

概要

文章中での登場人物や物体がどの場所にあるのかを文章から解析する技術が必要とされている。この技術は文章の中の登場人物の場所の他、物が誰の手元 (もしくは場所) に存在しているのかも明確にする。そして、この技術は観光分析に必要とされている。先行研究 [1] では、ブログ記事から「ドクターイエロー」や「赤福」の場所を自動抽出する手法を提案した。しかし、文の中にある言葉の意味を辿るということを行っていなかった。そこで本研究は入力を一連の動作を表す文章とし、動作を表す用言の意味を辿ることで存在性情報を得ることを目的とする。

本研究の流れは以下の通りである。存在性をパターン型知識として収録した辞書を用いて、入力文を照合し、存在性情報を抽出する。この辞書からは、動詞の文字通りの解釈により、動詞ごとに事態の直前、最中、完了のタイミングで存在性が定まる。そして抽出した存在性情報を常識的な知識の解釈により、異なる存在性情報をより多く求め、信念の木というデータ構造 (正確にはラティス構造) に追加する。最後に作成した信念の木のルートからリーフへのパスを作成する。そして、どのパスが最も良いのかを解析するために、制定した条件に当てはまると得点が加算されるという方法で、全パスについて点数をつけて最も点数の高いパスを求める。

本研究では、「太郎が図書館で本を読む」、「太郎が本を借りる」、「太郎が家に帰る」という3文を用いて存在性情報を得るという実験を試みた。入力した3文について上記の流れで存在性情報を抽出した結果、まず、データ構造の信念の木は、深さ6、ノード数24、アーク数102となり最終リーフは4つとなった。このとき作成されたパスは1024個となった。次に、選択により得た最高得点は11点、最高得点に該当したパスは2つとなった。このパスの最終リーフは、人間が常識的に考える存在性と一致したので、存在性情報抽出は成功したといえる。

目次

第1章	はじめに	1
第2章	先行研究	2
2.1	場所と存在物の情報抽出	2
2.1.1	パターン辞書の作成と拡張	2
2.1.2	場所と存在物の抽出結果	3
2.2	存在物と場所の対応検出	4
2.2.1	コーパス作成	4
2.2.2	IOB タグの推定と抽出	5
2.2.3	物と場所の対応検出と評価	5
2.3	本研究の位置づけ	7
第3章	提案手法	8
3.1	存在性情報の知識表現	8
3.1.1	ベクトル表記	8
3.1.2	信念の木	8
3.2	手法	9
3.2.1	存在性情報付きパターン辞書	9
3.2.2	解釈	10
3.2.3	選択	12
第4章	実装	14
4.1	システム環境	14
4.2	入力文の解析	14
4.3	文の理解の流れ	15
4.3.1	文字通りの意味解析	15
4.3.2	常識的な解釈の追加	17

4.4	信念の木	20
4.5	選択	20
4.6	存在性解析結果	21
第 5 章	考察	24
5.1	提案手法の考察	24
5.1.1	解釈	24
5.1.2	選択	24
5.2	存在性情報の解析結果	24
5.2.1	信念の木	24
5.2.2	得られた存在性情報	24
第 6 章	おわりに	26

目 次

3.1	手法の流れ	9
3.2	パターン辞書の例	10
4.1	読むのパターン辞書	15
4.2	借りるのパターン辞書	16
4.3	帰るのパターン辞書	17

表 目 次

2.1	場所と存在物の抽出結果	3
2.2	注釈付けの例	5
2.3	抽出の結果	6
2.4	対応検出の評価 (リンク単位)	7
2.5	対応検出の評価 (名称単位)	7
4.1	深さとアーク	20
4.2	最終リーフ	21
4.3	最高得点該当パス 1	22
4.4	最高得点該当パス 2	23

第1章 はじめに

文章の中の登場人物が、今どの場所にいるのか、物が誰の手元(もしくは場所)に存在しているのかなどが明確になれば、その文章をより理解できるようになる。観光分析としても物や場所が明確になればより分析しやすくなる。先行研究ではブログ記事から文字列上での物と場所の単語の位置を手がかりに解析していた。

しかし、存在物と存在場所の対応を検出していても、動詞の意味を理解しなければ、解析できないことがある。例えば、「太郎が荷物を家からマンションに運ぶ」という文などは、太郎が荷物を運ぶまでと、運んだ後では、荷物が異なる所に存在するといったことが解析できなかった。このように言葉の意味を辿らなければわからないことは、解析できなかった。

そこで本研究では、入力を一連の動作を表す文章とし、動作を表す用言の意味を辿ることで存在性情報を得ることを目的とする。

第2章では先行研究について述べる。第3章では提案手法について述べる。第4章では本手法の実装について述べる。第5章では考察を行う。第6章ではまとめを行う。

第2章 先行研究

本章では、先行研究について述べる。まず、場所と存在物の情報抽出について述べる。次に、存在物と場所の対応検出について述べる。

2.1 場所と存在物の情報抽出

場所と存在物の情報を抽出するためのパターン対の辞書が開発された [1]。

2.1.1 パターン辞書の作成と拡張

パターン辞書の作成の手順を以下に示す。

手順1 用意したブログ文から、連続する単文（動詞述語に限る）の対を自動で抽出する。

手順2 抽出した単文対から、場所と存在物を含む単文対を、自動で選出する。

手順3 選出した単文対に対して、存在物、場所および重要表現の注釈を人手で付与し、不適切な単文対を取り除く。こうして、タグ付きの単文対を得る。

手順4 タグ付き単文対から、場所・存在物の分析用のパターン対を自動で作成する。

ただし、手順2では、係り受け解析ツール CaboCha による出力で、Location タグが付与されている名詞、または、CaboCha 意味属性に「地域」が付与されている名詞を場所とする。日本語語彙大系で一般名詞意味属性が具体物である名詞を、存在物とする。

3つのブログサイトから2008年から2009年までのうち約22ヶ月分のブログ記事を取得して、パターン辞書の作成を行った。

例文「週末に門司までドライブに行ってきました。フク井を食べてきました。」の2文から作成したパターン対を、例として以下に示す。

FP /N1 まで/' 行く'[MDLC2]。Location:N1

SP /N1 を/' 食べる'[MDLC2]。Object:N1

パターン対は連続する2文において、1文目がFPに適合して、2文目がSPに適合するならば、対が適合したとみなす。

作成した各パターンの動詞部分を、日本語語彙大系の用言意味属性が同一である単語と置き換えて、パターンを追加し、パターン辞書を拡張した。

2.1.2 場所と存在物の抽出結果

実験用の文対を、ブログサイトから47件抽出した。そのうち、理論上、場所と存在物のペアが得られる文対は26件であり、そのペア数は35組であった。逆に、場所と存在物が得られない文対は21件であり、便宜上のペアを21組作った。これより、正解の集合Aの大きさは56組となった。次の3つの手法について性能を比較する。

手法1 パターン辞書におけるパターン対が同時に2文対に適合する際に場所と存在物のペアを抽出

手法2 パターン対を分解し、FPとSPの全ての組み合わせで再構成したパターン辞書を用いて、場所と存在物のペアを抽出

手法3 パターン辞書を用いずに、3.1節・手順2で用いた場所と存在物の判定条件で、場所と存在物のペアを抽出

各手法の出力は場所と存在物のペアであり、同様に場所と存在物の得られない場合の出力はペアである。ここで、出力の集合をOとし、手法の性能を一致率 $2|A \cap O| / (|A| + |O|)$ で評価する。

結果を表2.1に示す。一致率が手法2で最大となった。わずかではあるが、手法2の一致率が手法3のものより高い値を示している。

表 2.1: 場所と存在物の抽出結果

手法	出力数	一致数	一致率
1	48	28	0.27
2	57	37	0.36
3	65	35	0.34

2.2 存在物と場所の対応検出

ブログ記事からパターン対を用いた場所と存在物の情報抽出が行われた [2]。ここで、存在物や存在場所の抽出は固有表現抽出の一種と考えられる。存在情報の抽出と固有表現抽出の差は、一般名詞による存在物や場所の表現を抽出しなければならないこと、および、存在物と存在場所の対応を検出しなければならないことである。そこで、SVMを用いて、文章から存在物と場所の抽出、および、それらの対応を検出した。

2.2.1 コーパス作成

手順1 ブログから「ドクターイエロー」に関する記事を抽出する。

手順2 記事内の文を CaboCha で構文解析し、単語、品詞、固有表現タグ、係り先の情報を得る。

手順3 存在物および場所の表現に IOB タグを人手で付ける。

手順4 存在物に ID を付与し、存在する場所に存在物 ID を「存在物リンク」として付与する。1つの場所に複数の存在物がある場合、複数の存在物 ID を付与する。存在物 ID は記事単位でユニークとする。

例文「名古屋駅で N700 系とドクターイエローを撮影しました」に注釈付けした例を表 2.1 に示す。

2013 年 2 月～4 月のブログからドクターイエローに関係する記事は 84 記事抽出された。文数は 1,507、単語数は 24,499 となり、存在物についてのタグは、B が 566、I が 983 で、場所についてのタグは B が 458、I が 421 になった。存在物リンクの付与された場所は 345ヶ所であった。存在物と場所のリンク数は 2,240 であった。対応する場所の無い存在物は 41 件であった。

表 2.2: 注釈付けの例

単語	存在タグ	存在物 ID	場所タグ	存在物リンク
名古屋	O		B	1;2
駅	O		I	
で	O		O	
N	B	1	O	
7	I		O	
0	I		O	
0	I		O	
系	I		O	
と	O		O	
ドクター	B	2	O	
イエロー	I		O	
を	O		O	
撮影	O		O	
し	O		O	
まし	O		O	
た	O		O	

2.2.2 IOB タグの推定と抽出

SVM を用いて文末から文頭の順に各単語の IOB タグを推定する。素性は、次の単語、品詞、固有表現タグ、係り先の情報、および、次の単語の推定 IOB タグとする。係り先の情報は現在の単語とその先の単語を組み合わせた単語列とする。

第 2 章 2 節 1 項のコーパスを用いて IOB タグの抽出を行う。表 2.1 に抽出性能を評価した結果を示す。ここで、適合率 $P = pp/(pp + pn)$ 、再現率 $R = pp/(pp + np)$ 、F 値 $= 2PR/(P + R)$ である。また、 pp は、「正解タグ B または I を、B または I と推定した数」、 pn は、「正解タグ O を、B または I と推定した数」、 np は、「正解タグ B または I を、O と推定した数」である。

2.2.3 物と場所の対応検出と評価

ブログ記事内にある存在物と場所の対応を SVM を用いて検出する。抽出した存在物の 1 つずつに注目し、その存在物ごとに、対応する場所を検出するタスクとする。

表 2.3: 抽出の結果

手法	p	R	F 値	pp	pn	np
B1 (存在物)	0.49	0.02	0.03	32	33	1,518
提案 (存在物)	0.84	0.06	0.11	94	17	1,456
B1 (場所)	0.84	0.60	0.70	530	96	348
提案 (場所)	0.83	0.60	0.70	534	103	344

1つの記事内全ての各場所を、注目する存在物とペアとする。各ペアが対応するべきか否かを、次の素性を用いて、SVMで判定する。

f1 存在物と場所の単語距離が全ペアのうち最短か否か。

f2 存在物/場所の表現（チャンク）の係り先の動詞の基本形のペア。

f3 存在物や場所の表現を含む文に出現する名詞および動詞の意味コード（日本語語彙大系の一般名詞意味属性および用言意味属性）のペア。

f4 場所の表現の直後の助詞。

f5 存在物と場所の間にある単語と、各ペアの末尾側の存在物/場所から文末側にある動詞または文末までの単語。

SVMの判定は、素性の組み合わせ方により次の4通りとする。

M1: f1 および f2 を用いる手法

M2: f1, f2, および, f3 を用いる手法

M3: f1, f2, および, f4 を用いる手法

M4: f1, f2, および, f5 を用いる手法

さらにSVMのスコアが正值かつ最大値のペアを推定結果とする方法Msgx, および, 正值のペアをすべて推定結果とする方法Mplxを設ける ($x = 1; 2; 3; 4$).

存在物と場所の得られるべきリンク数は2,240であり, この数についての評価結果を表3に示す. ドクターイエローの存在場所について, 得られるべき場所の文字列の異なり数は95であり, この数についての評価結果を表4に示す. ここで, 適合率 $P = (\text{一致数}) / (\text{推定数})$, 再現率 $R = (\text{一致数}) / (\text{得られるべき数})$ である. 前者の結果より, F値で

は Mpl2 や Mpl4 が優れるが、後者の結果によるとその限りではない。特に Mpl2 では「加島」という特定の表現が目立った。機械学習により特定の語が集められたためと考える。

表 2.4: 対応検出の評価 (リンク単位)

手法	P	R	F 値	一致数	推定数
B2	0.75	0.19	0.30	422	566
Msg1	0.72	0.18	0.29	407	566
Msg2	0.64	0.16	0.25	356	560
Msg3	0.72	0.18	0.29	409	566
Msg4	0.60	0.15	0.24	341	566
Mpl1	0.61	0.25	0.37	564	926
Mpl2	0.54	0.48	0.50	1,083	2,015
Mpl3	0.57	0.28	0.38	634	1,106
Mpl4	0.46	0.56	0.50	1,247	2,725

表 2.5: 対応検出の評価 (名称単位)

手法	P	R	F 値	一致数	推定数
B2	0.82	0.57	0.67	54	66
Mpl1	0.60	0.63	0.62	60	100
Mpl2	0.82	0.13	0.22	12	15
Mpl3	0.59	0.69	0.63	66	112
Mpl4	0.38	0.08	0.13	8	21

2.3 本研究の位置づけ

本研究の目的は、文章中の登場人物の存在場所、存在物の持ち主(または場所)を明確にさせるために、動作を表す用言の意味を辿ることで存在性情報を得ることが目的である。この目的に向けて、次の点を明らかにする。

- 存在性の知識表現
- 意味処理の構造

第3章 提案手法

本研究の提案手法の流れについて示す。存在性情報の知識表現について述べる。次に研究全体の流れを述べる。

3.1 存在性情報の知識表現

3.1.1 ベクトル表記

本研究の存在性情報は、ベクトルで表現する。まず存在性情報とは、存在する物とその場所の組(ペア)の集合である。物と場所のペアを(物, 場所)と表す。物と場所のペアに対して固有のID番号をつける。ID番号はベクトルの次元となる。物と場所のペアに対して存在性がはっきりと文から理解できるならば、そのペアの示す次元には、1と記載する。一方、存在性が不明の箇所は、nilと記載、存在性がはっきりと否定されていれば、0と記載する。ベクトルの次元はペアの数だけ増えていく。

例文「太郎が図書館から家に帰る」において、(太郎, 図書館), (太郎, 家)の2つのペアが生まれ、順に1次元目, 2次元目とする。ベクトルの次元は2次元となる。太郎がまだ図書館を出発していない場合、(太郎, 図書館)の次元は1となり、(太郎, 家)の次元はnilとなる。このときの存在性情報のベクトルは(1, nil)となる。

3.1.2 信念の木

入力文の存在性情報を順に辿るために、信念の木というデータ構造(ラティス構造)を用いる。信念の木は、縦方向(深さ)が入力文によって増えていき、上(ルート)から文脈の順を表す。横方向は入力文の解釈の曖昧さによって増えていく。そしてリーフに向けて信念の木のノードは、3.1.1節で示したベクトルを記憶する。アークは、ノードからノードを導く理由を示す。

例として、「荷物を運ぶ」という動作を行ったとする。荷物を運ぶ前の状態のノード1から荷物を運び終えた状態のノード2へエッジが伸びるとき、荷物を運び忘れたという

状態ノード3が発生したとしよう。このとき、ノード1からノード2へのエッジには運び終わったという理由が示され、ノード1からノード3へのエッジには運び忘れたという理由が示される。

3.2 手法

手法の流れを図3.1に記載する。本研究では入力を1つの文章とし、MorphAnalyzerを用いて文章中の文1つ1つを形態素解析する。形態素解析された文をパターン辞書を用いて、文からの存在性情報を抽出する。これだけでは、文字通りの情報なので、常識的な知識による解釈を行う。信念の木というデータ構造に解釈した存在性情報を追加する。最後に信念の木からリーフの選択を行い、存在性情報を決定する。

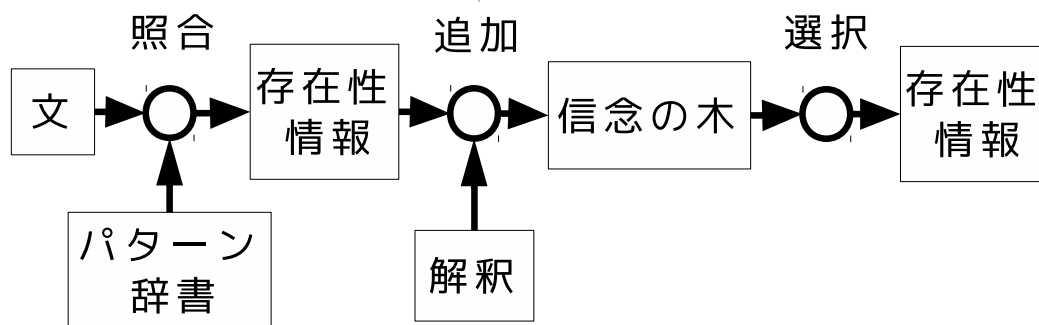


図 3.1: 手法の流れ

3.2.1 存在性情報付きパターン辞書

本研究のパターン辞書は基礎として日本語語彙大系を用いる。存在性情報を付与したこの辞書は本研究室で4, 5人による手作業で作成された。パターン数は14,819件である。1つのパターンをみると日本語パターンと意味属性制約の他に、変数Lの追加があり、タイミングのフラグ部分及び存在性言明がある。動詞は動作を表すのだから動作の直前、動作の最中、動作の完了がある。タイミングのフラグとは、直前、最中、完了の3つを表す。存在性言明とは物と場所のペアの存在性の説明である。パターン辞書内では

存在性が正しいと思われるタイミングでは、Tのフラグが立つ。動詞ごとに事態の直前、最中、完了のタイミングで存在性が定まる。

番号	パターン番号	見出し語	日本語文型パターン	英語文型パターン	直前	最中	完了	存在性言明
1	14312	読む	N1がN2を読む	N1 read N2	T	T	T	N1がLに存在する
2					T	T	T	N2がLに存在する
3	3201	借りる	N1がN2をN3から/より/ に借りる	N1 borrow N2 from N3	T		T	N1がLに存在する
4					T		T	N2がLに存在する
5					T		T	N3がLに存在する
6					T		T	N2がN3に存在する
7						T		N2がN1に存在する
8	2543	帰る	N1がN2から/よりN3に/ へ帰る	N1 return to N3 from N2	T	T	T	N1がLに存在する
9					T	T	T	N2がLに存在する
10					T	T	T	N3がLに存在する
11					T			N1がN2に存在する
12							T	N1がN3に存在する

*「Lで」が日本語文型パターンには暗黙で付いてくる。実行時にはパターンはコンバートされている(格要素の省略、順序の入れ替え、「は」と「が」の選択、修飾要素の挿入許可など)。日本語語彙大系の意味的制約はここでは表示していない。

図 3.2: パターン辞書の例

- Tのあるところは、そのタイミングで存在性言明がT(真)になることを表す。
- パターン辞書は断定できるものだけにTを立てているので空白は「変化なし」または、F(偽)と解析することがある..
- 全用言パターンは14,819件で、その89%に存在性の情報を付与している。付与した存在性情報は、変数Lに関するものが最も多く、直前、最中、完了の3つのタイミングを合わせて57,084回であった。変数L以外については、「N2がN1に存在する」が1,567回、「N1がN2に存在する」が1,545回等であった。

3.2.2 解釈

パターン辞書から得られた存在性情報に解釈を加える。本研究では以下の5つを用いる。

定理 1:継承 1つ前の文のノードから存在性情報を継承する.

定理 2:重複 ノードにおいて存在性情報が同じとき, ノードを重複する. このとき, エッジは引き継がれる

定理 3:非移動 動作主の位置が保たれていると仮定する

定理 4:移動 動作主の位置が変更されていると仮定する

定理 5:付帯 動作主が存在物を所持していると仮定する

この定理を解釈の定理とする.

3.2.2.1 定理 1:継承

1つ前の文のノードが $(1, \text{nil}, \text{nil})$ である場合, $(1, \text{nil}, \text{nil})$ を引き継ぎ, 現在の文の要素 $(1, \text{nil})$ を追加して, 現在の文のノードが $(1, \text{nil}, \text{nil}, 1, \text{nil})$ となる.

例文「太郎が本屋へ行く」「太郎が本を買う」と入力されるとき, 1文目の次元として (太郎, 本屋) の1次元が作成される. 太郎が本屋へ到着したとすると, 1文目のベクトルは (1) となる. この状態を継承して2文目を入力を行う. 2文目の次元は (本, 太郎) 作成される. ここで, 1文目の状態を継承すると, 2文目の次元は (太郎, 本屋), (本, 太郎) の2次元となる. このとき太郎は本を買う直前では (本, 太郎) は nil となる. そして2文目のベクトルは $(1, \text{nil})$ となる. また本を買い終わると $(1, 1)$ となる.

3.2.2.2 定理 2:重複

ベクトルが同じノードが2つ以上存在する場合, それらを全て重ねて1つだけのノードを出力する. 同時にアークも出力ノードに入るように変更する.

3.2.2.3 定理 3:非移動

動作主が「帰る」などの物理的移動の意味の動作を行わない場合, 動作主の存在している場所に変更されない.

3.2.2.4 定理 4:移動

動作主が「帰る」などの物理的移動の意味の動作を行う場合, 動作主の存在している場所を変更する.

3.2.2.5 定理 5:付帯

動作主が「借りる」などの所有的移動の意味の動作が行われた場合，物が動作主に存在する．

3.2.3 選択

信念の木のルートからリーフへは，複数のパスが作成される．どのパスが最も良いのかを解析するために，各パスに対して，加点方式の条件を6つ用いて，得点をつける．そして，最も点数の高いパスのリーフを求める．そのリーフが表す存在性情報を解析結果とする．

条件 1 1つ前の文のノードが継承されている

条件 2 動作主の移動，非移動が行われている

条件 3 動作主が存在物を付帯している

条件 4 完了を明示する動詞が出現しない場合での最中

条件 5 文章の最後は完了とする

条件 6 (物, 場所) において (a,b) かつ (c,a) ならば (c,b)

上記のいずれかに，該当すると1点加算される．

この条件を加点条件とする．

3.2.3.1 条件 1

1つ前の文のノードが(1,nil,nil)である場合，(1, nil, nil)を引き継ぎ，現在の文の要素(1, nil)を追加して，現在の文のノードが(1, nil, nil, 1, nil)となる．このように定理 1:継承が行われた場合に適用する．

3.2.3.2 条件 2

「帰る」などの物理的移動の意味の動作を行う場合，または，行われていない場合に適用する．

3.2.3.3 条件 3

「借りる」などの所有移転の意味の動作が行われた場合に適用する。

3.2.3.4 条件 4

「借りる」と「返す」など対となる動詞が発生しない限りその動作が継続するので、継続が続いている場合に適用する。

3.2.3.5 条件 5

文章の最後では、動作は全て完了となるので、最後の文を完了となっていれば適用される。

3.2.3.6 条件 6

ノードの要素(物, 場所)において, (太郎, 家)かつ(本, 太郎)の次元が1のとき, (本, 家)の次元に1と立つ場合に適用される。すなわち, 本の存在場所を推移律で解釈したことを評価する。

第4章 実装

3章の手法を用いて、入力文「太郎が図書館で本を読む」、「太郎が本を借りる」、「太郎が家に帰る」を順に入力する。

4.1 システム環境

本システムは次の環境下で実装する。OSにはVineLinux5.2を、プログラム言語にはRubyを用い、ツールはMorphAnalyzer(形態素解析)、Patlap(パターン処理器)、Graphviz(グラフ表現)を用いる。

4.2 入力文の解析

MorphAnalyzerを用いて形態素解析を行う。Patlapでパターン辞書を引く。適合するパターン及び付随する情報が得られる。

1 文目 「太郎が図書館で本を読む。」

[7, “読む”, “ny:23 身体動作 32 思考動作”, “lower:”, “before:太郎が図書館に存在する,本が図書館に存在する”, “progress:太郎が図書館に存在する,本が図書館に存在する”, “completion:太郎が図書館に存在する,本が図書館に存在する”]

[7, “読む”, “NY:23 身体動作 32 思考動作”, “attr:(動作)”, “eng:太郎 read 本”]

2 文目 「太郎が本を借りる。」

[5, “借りる”, “ny:19 所有的移動”, “lower:”, “before:太郎がφに存在する,本がφに存在する”, “progress:本が太郎に存在する”, “completion:太郎がφに存在する,本がφに存在する”]

[5, “借りる”, “NY:19 所有的移動”, “attr:(動作受身不可)”, “eng:太郎 borrow 本 from φ”]

3 文目 「太郎が家に帰る。」

[5, “帰る”, “ny:18 物理的移動”, “lower:”, “before:太郎がφに存在する, φがφに存在する, 家がφに存在する, 太郎がφに存在する”, “progress:太郎がφに存在する, φがφに存在する, 家がφに存在する”, “completion:太郎がφに存在する, φがφに存在する, 家がφに存在する, 太郎が家に存在する”]

[5, “帰る”, “NY:18 物理的移動”, “attr:(動作受身不可)”, “eng:太郎 return to 家 from φ”]

なお, 存在性言明にφが含まれる場合は, ペアを作れない。

4.3 文の理解の流れ

4.3.1 文字通りの意味解析

1 文目 「太郎が図書館で本を読む」

マッチしたパターン辞書のエントリーは図 4.1 である。前節 1 文目の結果に表示された通りである。

日本語文型/パターン	英語文型/パターン	直前	最中	完了	存在性言明
N1がN2を読む	N1 read N2	T	T	T	N1がLに存在する
		T	T	T	N2がLに存在する

* 「Lで」がパターン辞書に暗に有る

図 4.1: 読むのパターン辞書

直前のタイミングでは, 直前の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている (before スロットを参照)。このタイミングで存在性言明から読み取られるべき情報は, (太郎, 図書館), (本, 図書館) である。

最中と完了も同様である (progress スロット, completion スロットを参照)。

結論は, 「太郎は図書館にいる」, 「本は図書館にある」。

2 文目 「太郎が本を借りる」

マッチしたパターン辞書のエントリーは図 4.2 である。

日本語文型/パターン	英語文型/パターン	直前	最中	完了	存在性言明
N1が N2を N3から/より/に 借りる	N1 borrow N2 from N3	T		T	N1がLに存在する
		T		T	N2がLに存在する
		T		T	N3がLに存在する
		T		T	N2がN3に存在する
			T		N2がN1に存在する

* 「Lで」がパターン辞書に暗に有る

図 4.2: 借りるのパターン辞書

直前のタイミングでは、直前の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている。このタイミングでは、場所を表す名詞が入力されていないので、変数にマッチしない (ϕ を表示)。ゆえに存在性情報が得られない。

最中のタイミングでは、最中の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている。このタイミングでは、変数にマッチしているため、存在性情報は T であり存在性言明から得られるべき情報は、(本, 太郎) である。

完了のタイミングでは、完了の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている。このタイミングでは、場所を表す名詞が入力されていないので、変数にマッチしない。ゆえに存在性情報が得られない。

結論は、「太郎は本を持っている」。

3 文目 太郎が家に帰る

マッチしたパターン辞書のエントリーは図 4.3 である。

日本語文型/パターン	英語文型/パターン	直前	最中	完了	存在性言明
N1が N2から/より N3に/へ 帰る	N1 return to N3 from N2	T	T	T	N1がLに存在する
		T	T	T	N2がLに存在する
		T	T	T	N3がLに存在する
		T			N1がN2に存在する
				T	N1がN3に存在する

* 「Lで」がパターン辞書に暗に有る

図 4.3: 帰るのパターン辞書

直前のタイミングでは、直前の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている。このタイミングでは、場所を表す名詞が入力されていないので、変数にマッチしない。ゆえに存在性情報が得られない。

最中のタイミングでは、最中の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている。このタイミングでは、場所を表す名詞が入力されていないので、変数にマッチしない。ゆえに存在性情報が得られない。

完了のタイミングでは、完了の列を参照し T となる行の存在性言明を利用する。変数にマッチした単語が代入されている。このタイミングでは、変数にマッチしているので、存在性情報は T であり存在性言明から得られるべき情報は、(本, 太郎)である。

結論は、「太郎が家にいる」、「本が家にある」。

4.3.2 常識的な解釈の追加

1 文目では、(太郎, 図書館), (本, 図書館) という存在性情報が得られる。直前, 最中, 完了, とともに「太郎は図書館に存在する。本は図書館に存在する。」と解釈される。

2文目では(本, 太郎)という存在性情報が得られる。ここで, 解釈の定理1により1文目の存在性情報を継承したことにより, 存在性情報は(太郎, 図書館), (本, 図書館), (本, 太郎)となった。

- 「借りる」の直前の場合, 存在性情報の解釈は2パターン作成された
 - － 定理1により「太郎は図書館に存在する。本は図書館に存在する。太郎は本を持っていない。」と解釈される
 - － 解釈の定理が適用されなかった場合により, 「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っていない。」と解釈される
- 「借りる」の最中の場合, 存在性情報の解釈は4パターン作成された
 - － 定理3により, 「太郎が図書館に存在する。本は図書館に存在しない。太郎は本を持っている。」と解釈される
 - － 定理5により, 「太郎は図書館に存在しない。本は図書館に存在する。太郎は本を持っている。」と解釈される
 - － 定理3と定理5の組み合わせにより, 「太郎は図書館に存在する。本は図書館に存在する。太郎は本を持っている。」と解釈される
 - － 解釈の定理が適用されなかった場合により, 「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っている。」と解釈される
- 「借りる」の完了の場合では, 存在性情報の解釈は4パターン作成された
 - － 定理3により, 「太郎が図書館に存在する。本は図書館に存在しない。太郎は本を持っていない。」と解釈される
 - － 定理5により, 「太郎は図書館に存在しない。本は図書館に存在する。太郎は本を持っていない。」と解釈される
 - － 定理3と定理5の組み合わせにより, 「太郎は図書館に存在する。本は図書館に存在する。太郎は本を持っていない。」と解釈される
 - － 解釈の定理が適用されなかった場合により, 「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っていない。」と解釈される

3文目では(太郎, 家), (本, 家)という存在性情報が得られる。ここで, 解釈の定理1により2文目の存在性情報を継承したことにより, 存在性情報は(太郎, 図書館), (本, 図書館), (本, 太郎), (太郎, 家), (本, 家)となった。

- 「帰る」の直前の場合, 存在性情報の解釈は8パターン作成された
 - － 定理1により「太郎は図書館に存在する。本は図書館に存在する。太郎は本を持っている。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 定理1により「太郎は図書館に存在する。本は図書館に存在しない。太郎は本を持っていない。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 定理1により「太郎は図書館に存在する。本は図書館に存在する。太郎は本を持ってない。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 定理1により「太郎は図書館に存在する。本は図書館に存在しない。太郎は本を持っている。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 定理1により「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っている。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 定理1により「太郎は図書館に存在しない。本は図書館に存在する。太郎は本を持っていない。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 定理1により「太郎は図書館に存在しない。本は図書館に存在する。太郎は本を持っている。太郎は家に存在しない。本は家に存在しない。」と解釈される
 - － 解釈の定理が適用されなかった場合により, 「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っていない。太郎は家に存在しない。本は家に存在しない。」と解釈される
- 「帰る」の最中の場合, 存在性情報の解釈は2パターン作成された
 - － 定理4のみと定理5のみと定理4と定理5の組み合わせにより, 「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っている。太郎は家に存在しない。本は家に存在しない。」と解釈される

- 解釈の定理が適用されなかった場合により、「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っていない。太郎は家に存在しない。本は家に存在しない。」と解釈される
- 「帰る」の完了の場合では、存在性情報の解釈は2パターン作成された
 - 定理4のみと定理5のみと定理4と定理5の組み合わせにより、「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っている。太郎は家に存在する。本は家に存在する。」と解釈される
 - 解釈の定理が適用されなかった場合により、「太郎は図書館に存在しない。本は図書館に存在しない。太郎は本を持っていない。太郎は家に存在する。本は家に存在する。」と解釈される

4.4 信念の木

Graphviz を用いて信念の木を表現する。3.2.2 節の解釈の定理5つを用いる。

1文目入力時では、深さ2、ノード数3、アーク数4、リーフ数が1つだけとなった。2文目入力時では、深さ4、ノード数13、アーク数22、リーフ数が8つとなった。3文目入力時は深さ6、ノード数24、アーク数102、最終リーフ数が4つの構造となった。

各深さごとのアークは表4.1である。

表 4.1: 深さとアーク

深さ	アーク
0 → 1	2
1 → 2	2
2 → 3	2
3 → 4	16
4 → 5	16
5 → 6	64

4.5 選択

作成された信念の木に縦型探索を行い、ルートからリーフへのパスを作成する。この作成されたパスは1,024個作成され、パス1つ1つに加点条件を当てはめる。そして、

当てはまると加点される。今回の入力文だと以下のいずれかに該当すると加点する。

- ルートから1文目から2文目, 2文目から3文目に存在性情報の継承が行われていれば加点
- 2文目の「借りる」という動作を行って, そのまま, 図書館に留まっているならば加点
- 2文目の「借りる」という動作を行って, 本を所持しているならば加点
- 2文目の「借りる」という動作を行って, 最中の状態であるのならば加点
- 2文目の「借りる」という動作を行って, 「太郎が図書館に存在する. 本が図書館に存在する. 太郎は本を持っている.」という存在性情報が得られるなら加点
- 3文目の「帰る」という動作を行って, 図書館から移動しているならば加点
- 3文目の「帰る」という動作を行って, 本を所持しているならば加点
- 3文目の「借りる」という動作を行って, 「太郎が家に存在する. 本が家に存在する. 太郎は本を持っている.」という存在性情報が得られるなら加点
- 3文目の「借りる」という動作を行って, 完了の状態であるのならば加点

この加点された最高得点のパスを選択する。

4.6 存在性解析結果

最終リーフより4つの解釈が可能。表4.2は各最終リーフをノードを表す。

表 4.2: 最終リーフ

ノード番号	状態				
	(太郎, 図書館)	(本, 図書館)	(本, 太郎)	(太郎, 家)	(本, 家)
40	nil	nil	1	nil	nil
41	nil	nil	1	1	1
46	nil	nil	nil	nil	nil
47	nil	nil	nil	1	1

この表はそれぞれ以下の意味を表現したものである。

ノード 40 「太郎は図書館にも家にもいない。太郎は本を持っている。」

ノード 41 「太郎は家にいる。本は太郎の所にあり，家にある。」

ノード 46 「太郎は図書館にも家にもいない。太郎は本を持っていない。」

ノード 47 「太郎は家にいる。本は家にある。太郎は本を持っていない。」

この4つの中で人間の常識に最もふさわしいと考えられるリーフは41である。

作成されたパスは，1,024個であり，3.2.3節の加点条件による最高得点が11点であった。最高得点に該当したパスは2個であった。

表4.3と表4.4に最高得点該当パスを記載する。この表はノード，エッジという順で表記する。ノード番号とノード内のベクトルとエッジのラベルを記載する。

表 4.3: 最高得点該当パス 1

ノード	ノード番号	ベクトル
エッジ	ラベル(タイミング)	ラベル(解釈)
ノード	1	
エッジ	文1:直前	(継承あり)
ノード	2	(1,1)
エッジ	文1:完了	
ノード	4	(1,1)
エッジ	文2:直前	(継承あり)
ノード	7	(1,1,nil)
エッジ	文2:最中	非移動付帯
ノード	8	(1,1,1)
エッジ	文3:直前	(継承あり)
ノード	25	(1,1,1,nil,nil)
エッジ	文3:完了	移動付帯
ノード	41	(nil,nil,1,1,1)

表 4.4: 最高得点該当パス 2

ノード	ノード番号	ベクトル
エッジ	ラベル (タイミング)	ラベル (解釈)
ノード	1	
エッジ	文 1:直前	(継承あり)
ノード	2	(1,1)
エッジ	文 1:最中	
ノード	4	(1,1)
エッジ	文 2:直前	(継承あり)
ノード	7	(1,1,nil)
エッジ	文 2:最中	非移動付帯
ノード	8	(1,1,1)
エッジ	文 3:直前	(継承あり)
ノード	25	(1,1,1,nil,nil)
エッジ	文 3:完了	移動付帯
ノード	41	(nil,nil,1,1,1)

最高得点の2個のパスのリーフは41だけとなった。正解であるリーフ41に絞り込むことに成功した。

第5章 考察

5.1 提案手法の考察

5.1.1 解釈

解釈の定理に，他の文を持ち込んだとしても，上手くいくとは限らない．解釈の定理というのは，動詞の意味に合わせてマッチするものである．今回は5つ用意したが，今後さらに増やしていく必要がある．

5.1.2 選択

1,024個のパスというのは，今回用意した3文から生成されたものである．もっと文を増やしたり，違う文を用意すれば，パスはもっと増えるだろう．そのパスの中から，理想的な存在性情報を選択するためには，さらに条件を増やしていく必要がある．

5.2 存在性情報の解析結果

5.2.1 信念の木

ノード内のベクトルが同じであるとき重複すると設計した．それでも，信念の木はかなりの複雑さである．エッジなどは図からでは，辿ることが非常に困難である．3文でこの様子なので，文が増えた際に，信念の木をより見やすくする設計をしなければならない．

5.2.2 得られた存在性情報

1文目の「読む」の動作のときは，「太郎は図書館にいる．本は太郎の所にあり，図書館にある．」が得られる．2文目の「借りる」の動作のときは，「本は太郎の所にある．」という結果を得る．これらの過程を経て，3文目「帰る」という動作のときは，「太郎は家

にいる。本は太郎の所にあり，家にある。」という結果が得られる。用意した3文を動詞の意味を考えて，常識的に解析した結果，このような結果を得ることができた。

しかし，3文しか解析していないので，もっと多様な文を試し，解析していく必要がある。

第6章 おわりに

本研究の目的は、文章中の登場人物の存在場所、存在物の持ち主(または場所)を明確にさせるために、動作を表す用言の意味を辿ることで存在性情報抽出することが目的である。存在性情報の知識表現では、物の存在性情報をベクトルで表現し、信念の木で多様な解釈を表現した。意味処理の構造では、文字通りの情報をパターン辞書で表現し、解釈の定理を用いて存在性情報を追加し、選択により常識的な情報を抽出した。

本研究では、パターン辞書を用いて、3文と照合させ、文字通りの存在性情報を抽出した。その得られた存在性情報を、解釈の定理により文外の常識で補足した後、信念の木というデータ構造に追加した。信念の木から加点条件を用いて理想的なパスを選出し、常識的な存在性情報を得ることができた。

解釈では定理を5つ用意し、これにより、信念の木は、深さ6、ノード数24、アーク数102となり、最終リーフは4つの構造となった。選択では条件を6つ用意し、これにより、選択前では1,024個あったパスは、選択後2個まで絞り込むことができた。この2個のパスは、最終リーフが2つともノード番号41だった。ノード番号41というのは、もっとも人間の常識にふさわしいと考えられる正解データなので、存在性情報の抽出は成功したといえる。

ただし、本研究では3文しか試していない。今後の課題は、より多くの多様な文を用い、解釈の定理と加点条件を増やして、動作を確認することである。

参考文献

- [1] 北尾祐樹, 徳久雅人, 村田真樹, 村上仁一: “2 文からの場所と存在物の解析”, 鳥取大学工学部知能情報工学科卒業論文, 2013.
- [2] 菊池春香, 徳久雅人, 村田真樹, 村上仁一: “文章から存在性情報の抽出”, 情報処理学会第 76 回全国大会講演論文集, pp.93-94, 2014.
- [3] 池原悟, 宮崎正弘, 白井諭, 横尾昭男, 中岩浩巳, 小倉健太郎, 大山芳史, 林良彦: “日本語語彙大系”, 岩波書店, 1997.

謝辞

徳久雅人講師には，終始に渡り研究の進め方や論文の書き方など，細部に渡るご指導を頂きました．心から御礼申し上げます．

また，本研究を進めるに当たり，種々の御助言を頂きました村田真樹教授，および，村上仁一准教授に心から御礼申し上げます．

その他様々な場面で御助力を頂いた計算機工学 C 講座の学生の皆様に感謝の意を表します．