

電子商取引における商品ルール抽出支援手法

Supporting Business-Rule Extraction for E-Commerce

小澤 正幸¹ 岩井原 瑞穂²
上林 彌彦²

Masayuki KOZAWA Mizuho IWAHARA
Yahiko KAMBAYASHI

商品の価格や利用条件、割引条件などを記述する商品ルールは、電子商取引が普及している現在でも自然言語で記述されていることが多い。商品ルールを計算処理可能な論理的表現に変換することにより、条件の照合や高度な商品の検索などが可能になる。そのためには、自然言語による商品ルールから論理的表現を抽出する技術が必要となる。本論文では、航空券のルールを例に取り、表記のゆらぎを含む自然言語表現の商品ルールから、値の繰り返しまとめた正規表現を抽出し、最終的な論理式に変換するための支援手法について述べる。

Although e-commerce is becoming popular, product rules which describe various conditions such as prices, applicability, discount conditions are still expressed in natural languages. Providing logical rules for such product rules enable us more advanced product search functionalities and condition verification. To that end, developing technologies for extracting logical rules from natural language product rules is necessary. In this paper, we present a series of supporting methods for removing notational inconsistencies, discovering regular expressions from repetition of descriptions, and finally transforming into logical rules.

1. はじめに

電子商取引の普及に伴い、従来は人手で行われていた分野の商取引活動に対しても計算機による支援が求められるようになってきている。筆者らはその中でもビジネスルールの自動処理に着目し、ルール付商品、つまり商品ごとに多種多様な形式のルールが付され、その適用によって価格などの商品属性が変化するという特徴を持つ商品の電子商取引に対して、そのルール処理の支援を行う仕組みを研究してきた [4] [5] [7][8]。

実際に電子商取引市場で扱われている商品の中で、このような特徴を持つルール付商品の具体例としては、航空券やホテル、レンタカー、旅行パックやPCのBTO販売などが挙げられる。航空券ではTravelXMLやOpen Travel Alliance Specificationなどの記述言語が提案されているが、いずれも論理的なルールの記述は自然言語により行なわれ、人間による解釈を前提としている。

このような論理的表現は、将来的には商品データを作成する段階でデータ作成者によって与えられることが考えられる。しかし、現状のWebドキュメントのほとんどのルールデータが自然言語で記述されており、自然言語からこのような表現への変換作業が必要となる。このような作業の省力化は、たとえ

ばSemantic Web[1]においても重要な問題である。RuleML[2]やBRML[3]といったルール記述のためのXML言語は提案されているが、ウェブ文書からのビジネスルール抽出では文献[6]は書店の送料ルールを例にWebからのルール抽出を行っているが、人手で自然言語のルールにマークアップする作業を必要としている。

本論文では航空券販売を例に取りあげ、筆者らが開発を行っているビジネスルール処理のフレームワークである動的制約データベース用の表現 [7] に変換する作業を支援する手法を議論する。これは他の論理的表現への変換についても、共通して適用できうる部分を有する。

本論文の構成は、まず2.節でルール付商品とその例である航空券について述べ、3.節でルール付商品のモデル化を行う。そして、4.節で自然言語で記述されたルールの論理的表現への変換について述べ、5.節ではその変換支援手法のアルゴリズムと実行例を示す。7.節は結論である。

2. ルール付商品とその例

2.1 航空券販売

現在、国内でも旅行代理店・航空会社などが運営するサイトで電子商取引として航空券が販売されている。その取引形態は様々であり、Webサイトのみで商品選択と購入が可能なカタログ販売型とサイト上には簡潔な情報のみを置き、詳細については電子メールなどによって顧客と旅行代理店間で交渉を行う相対取引型が一般的である。

一般に格安航空券と呼ばれるものは、航空会社からまとめて仕入れた商品に、様々な条件を付けて小売りするものであり、航空会社のものより価格は安い様々な制約条件が付帯されているものが多い。この場合は様々な割引方式や複雑な価格構成が存在し、ルールには商品ごとのばらつきが見られる。ただし、商品ルールの自然言語記述には、表現の揺らぎはあるが、文の意味自体は曖昧性を持たないという特徴がある。

実際の航空券から抜き出したルールの例を表1に示す。

表 1: 航空券の商品ルールの例
Table 1: Examples of business rules of airline tickets

訪問可能都市	アトランタ, ポストン, パッファロー, ..., シアトル, サンフランシスコ, ソルトレイクシティ, バンクーバー, トロント
訪問可能都市数	上記指定都市内 3 都市周遊可能です。
有効期限	1 泊 3 日 ~ 1ヶ月以内 FIX (復路変更不可)
追加料金	週末料金: 現地発金・土・日は, ¥4,000UP
ストップオーバー	2 回まで無料。3 回目まで可: ¥10,000UP

2.2 商品ルールの論理的表現の応用

商品ルールを本稿で扱うような論理的表現に変換することにより、以下の応用が考えられる。

- 売り手の条件と買い手側の条件をそれぞれルールで表現し、それらの論理積の充足可能性や、含意の関係を調べることにより、互いの条件が満足されているかの検査を行なう。
- 商品にどのような付帯条件や制限事項があるかを検索し、表示する検索機能。
- 買い手の要求条件を、商品ルールに代入することにより、可能な選択枝を求め、それを買い手に選択メニューとして提示する。

¹学生会員 京都大学情報学研究所社会情報学専攻
²正会員 京都大学情報学研究所社会情報学専攻
iwaihara@i.kyoto-u.ac.jp

3. 商品ルールのモデル化

3.1 論理的表現の要求事項

- 論理的表現には曖昧性が無く、厳密な意味が与えられることが必要である。
- 計算処理が実用的時間になるように、表現力を適切なものとする。
- 論理的表現から元の自然言語の表現が直ちに求められるようにする。
- 実際には自然言語で記述されているルールが表す概念の全てが論理的表現に変換できるとは限らないため、部分的に自然言語表現を混在させることが必要である。自然言語による表現の部分は、利用者に表示して真偽値の判断を求めるようにする。

3.2 商品ルールの論理的表現のモデル

定義 1 1. 商品ルールは商品の各種特性を表す商品属性とその値を表す商品属性値からなる制約を表わす論理式である。

2. 1つの商品ルールは、いくつかのルール条項に分割することができ、各商品ルールはルール条項の論理積である。
3. 各商品属性には値域集合が割り当てられている。これはたとえば日付や金額、地名等である。
4. 商品ルールあるいはルール条項の自然言語による表現をそれぞれ商品ルール文およびルール条項文と呼ぶ。
5. 商品ルールおよびルール条項を表わす論理式のクラスとして、等号論理を用いる。これは、変数と定数、または変数同志の等号を AND, OR, NOT のブール結合子で結合した式であり、例えば $(x = "abc" \vee y \neq "def") \wedge (z = "ghi")$ のような論理式である。各変数はいずれかの商品属性に対応付けられ、定数は商品属性の値域集合に含まれるものとする。

上記の等号論理式の制約集合に対し、制約データベース [9] の考え方により代数演算を行なう質問言語として、動的制約代数 [4][5] があり、商品ルールの集成的操作および論理演算に用いることができる。

4. ルール条項文の変換処理

4.1 変換元の航空券データ

旅行代理店による航空券販売サイト (アルキカタ・ドット・コム <http://www.arukikata.com/>) から三ヶ月分 2000 件の航空券データを収集した。データには商品コードや訪問可能都市、出発地、出発日と基本料金の対応表、追加料金ルール、その他のコメントなど合わせて計 17 の項目分けがなされ、日本語で記述されている。

上記の日本語の記述は商品ルール文に対応することになる。商品ルール文をさらにルール条項文に分割してゆく必要があるが、意味的にまとまったものを分割するために、改行文字や箇条書きの区切りなどを手がかりとし、スクリプト処理により行った [8]。この分割では他を参照しなければ意味を把握できないルール条項文を作らないように、分割の基準を緩やかに取った。この作業により、収集した 2000 件の商品データに付随する商品ルールを分割したルール条項の数は 18482 件となった。さらに文字列の比較により重複したものを削除すると 3766 種類である。

4.2 類似ルールの例について

商品ルールの中には、他と同一か類似するルール条項文は多く、また日本語の表記の揺らぎがあるものや、論理的な構造は同じでも、具体的な値やその個数が異なるものがある。収集したルールのうち、週末料金に関するルール条項文の中から、このような表記の揺らぎや具体的な値、そして値の個数の違いを含む例を表 2 に示す。

表 2: 表記の揺らぎなどを含むルール条項文

Table 2: Rule clauses containing fractuating expressions

1	*週末料金：帰路，現地発が金・土曜日の場合は，¥3,000UP．
2	*週末料金：帰路，日本着が日曜日の場合 ¥5,000UP．
3	*週末料金：帰路，日本着が日曜日の場合は ¥3,000UP．
4	*週末料金：現地発 金曜・土曜 ¥5,000UP
5	*週末料金：現地発 土曜 ¥5,000UP
6	*週末料金：現地発，金・土曜日は ¥5,000UP．
7	*週末料金：現地発，金・土曜日は ¥3,000UP
8	*週末料金：現地発が金・土・日曜日の場合，¥4,000 円 UP
9	*週末料金：現地発金・土曜日は，¥3,000UP
10	*週末料金：現地発金・土曜日は，¥5,000UP
11	*週末料金：現地発金曜日は，¥5,000UP，土曜日は，¥8,000UP
12	*週末料金：現地発金・土曜日は ¥5,000UP
13	*週末料金：復路，現地発が金・土曜の場合は，¥5,000UP
14	*週末料金：復路，現地発が金・土曜日の場合 ¥3,000UP
15	*週末料金：復路，現地発が金・土曜日の場合 ¥3,000UP．
16	*週末料金：復路，現地発が金・土曜日の場合 ¥5,000UP．
17	*週末料金：復路，現地発が金曜日の場合 ¥5,000UP，土曜日の場合 ¥8,000UP．
18	*週末料金：復路，現地発が金曜日の場合は ¥5,000，土曜日の場合は ¥15,000UP．
19	*週末料金：復路，現地発が土曜日の場合は ¥3,000UP．
20	*週末料金：復路，現地発が土曜日の場合は ¥5,000UP．
21	*週末料金：復路，現地発が金・土曜日の場合，¥2,000UP．
22	*週末料金：復路，現地発が金・土曜日の場合，¥3,000UP．
23	*週末料金：復路，現地発が金・土曜日の場合は ¥5,000UP
24	*週末料金：復路，現地発が金・土曜日の場合は ¥5,000UP．
25	*週末料金：復路，現地発が金・土曜日の場合は ¥8,000UP．
26	*週末料金：復路，現地発が金・土曜日の場合は，¥5,000UP
27	*週末料金：復路，現地発が金曜・土曜の場合，¥5,000UP．
28	*週末料金：復路，現地発が日曜日の場合 ¥3,000UP．
29	*週末料金：復路，現地発金・土曜日は ¥5,000UP
30	*週末料金：復路，現地発金曜・土曜の場合は，¥5,000UP．
31	*週末料金：復路，現地発金曜の場合は ¥5,000UP・土曜の場合は ¥8,000UP．
32	*週末料金：復路，日本着が土・日曜日の場合 ¥3,000UP．
33	*週末料金：復路，日本着が土・日曜日の場合は ¥3,000UP．
34	*週末料金：復路，日本着が日曜場合，¥3,000UP．
35	*週末料金：復路，日本着が日曜日の場合 ¥2,000UP．
36	*週末料金：復路，日本着が日曜日の場合 ¥3,000UP．
37	*週末料金：復路，日本着が日曜日の場合，¥2,000UP．
38	*週末料金：復路，日本着が日曜日の場合，¥3,000UP．
39	*週末料金：復路，日本着が日曜日の場合は ¥3,000UP．
40	*週末料金：復路，日本着が日曜日の場合は ¥5,000UP．
41	*週末料金：復路現地発，金・土は ¥5,000UP．
42	*週末料金：復路現地発，金・土曜日は，¥3,000UP
43	*週末料金：復路現地発が土曜の場合，¥3,000UP
44	*週末料金：復路現地発が土曜の場合，¥5,000UP
45	*週末料金：復路現地発金曜・土曜の場合は，¥5,000UP．
46	週末料金：現地発金，土曜日は，¥3,000UP
47	週末料金：現地発金・土曜日は ¥3,000UP
48	週末料金：現地発金土日は，¥4,000UP
49	週末料金：現地発金土曜日は，¥5,000UP
50	週末料金：現地発金土曜日は，¥5,000UP
51	週末料金：現地発土，日曜日は ¥3,000UP
52	週末料金：現地発土曜日は，¥2,000UP
53	週末料金：現地発日曜日は ¥3,000UP
54	週末料金：現地発日曜日は ¥3,000UP
55	週末料金：現地発日曜日は，¥2,000UP
56	週末料金：現地発日曜日は，¥3,000UP
57	週末料金：復路現地発金・土曜日は，¥5,000UP
58	週末料金：復路現地発土曜日は，¥3,000UP
59	週末料金：現地発 金・土曜日は，¥3,000UP
60	週末料金：現地発：金・土曜：¥3,000UP

- 行頭の“*”や“ ”は箇条書きの先頭記号に由来する。統一されたフォーマットが存在するわけではないので、このようなルール条項文の記述者によると見られる不揃いが存在する。

- ルール 4,6,10,12,13,14,23,24,26,27,29,30,41,45,50,57 は表記の違いだけで、内容としては「復路の出発曜日が金曜日か土曜日であれば、5000 円の週末追加料金がかかる」という同一のものである。
- ルール 49,50 の違いは 2 文字 (“曜”の有無と金額の “4” と “5”) のみであり、文字列としての違いは同一の内容を持つルール 50 とルール 57 の違いよりも少ない。しかし解釈は異なる。
- 週末料金の対象となる曜日の値の個数は 1 曜日から 3 曜日まで様々である。
- ルール 11,17,18,31 では、曜日と追加料金額の対応関係が二つ示され、他の物と異なる。

5. 商品ルールの論理的表現の抽出支援手法

5.1 登録単語辞書

商品ルール文を解析し、論理的表現に変換するために、商品ルール文の記述に使用されている用語が登録されている登録単語辞書を用意する。登録単語のカテゴリーを表す値域集合の名前を単語型あるいは単に型と呼ぶ。たとえば「都市名」や「曜日」、「金額」、「否定語」が単語型である。登録単語辞書では、登録語から単語型を引けるようにする。また、商品ルール文は人手で書かれたものが多いため表記のゆらぎが存在する。例えば表 2 では、土曜日を表す語として、“土”、“土曜”、“土曜日”などがあり、このうちの 1 つを標準形として辞書に登録する。

5.2 論理的表現の抽出支援手法の流れ

1. 商品ルール文について、登録単語辞書を検索して標準形の登録語(トークン)に置換する。この過程をトークン抽出と呼ぶ。トークン抽出により、表記のゆらぎが吸収される。
2. 航空券においては、訪問可能都市や可能な曜日の列挙が見られるが、これらは同じ型のトークンの列として検出可能である。トークンの反復畳み込みとして、このようなトークンの列を検出する。これにより、トークン列の繰り返し回数や、カンマ等の句読点を無視した、同じ型のトークン列を認識する。
3. ルールのクラスタリングとして、トークンの反復畳み込みの結果が同一になったルール条項をまとめ込む。
4. 変換テンプレートの適用として、上記の結果のルール条項文にマッチする変換テンプレートを検索し、そのテンプレートに従ってルール条項文を論理的表現に変換する。

上記の手法は、完全に自動化できるものではなく、変換に誤りが含まれることがある。そのため、各段階で人手による変換結果の確認を行なう。最終的な判断を人間が行なうという意味で、本手法は抽出支援手法という位置づけになる。数千を越える商品ルールを全て人手で行なうのは困難であり、本抽出支援手法による半自動化の効果は大きいと考えられる。

5.3 抽出支援手法の実用例

5.3.1 ルールからのトークン抽出

トークン抽出において、ゆらぎのある表記は標準形であるトークンに変換される。表 2 のルール 3,4,17,41 に対しトークン抽出を行なった結果を、表 3 に示す。型名の先頭には#を、標準形には下線を付き、ルール文中に現れた表記については、二重引用符でかこった。これらについて型と標準形のみで見た場

合、ルール 4 と 41 そして前述の通りこれと同じ意味内容を持つルール(例えば 6,10,12,13,14,23,24,26,27,29,30,45,50,57)は、全て一致し、表記のゆらぎを吸収できたことが分かる。

表 3: トークン抽出を行なったルール条項文
Table 3: Rule clause sentences after token extraction

	(#型, 標準形, “ルール中の表記”)
3	(#週末料金, 週末料金, “週末料金”) (#復路日本着, 復路日本着, “帰路, 日本着”) (#曜日, 日曜日, “日曜日”) (#金額, ¥3000, “¥3,000”)
4	(#週末料金, 週末料金, “週末料金”) (#復路現地発, 復路現地発, “現地発”) (#曜日, 金曜日, “金曜”) (#曜日, 土曜日, “土曜”) (#金額, ¥5000, “¥5,000”)
17	(#週末料金, 週末料金, “週末料金”) (#復路現地発, 復路現地発, “復路, 現地発”) (#曜日, 金曜日, “金曜日”) (#金額, ¥5000, “¥5,000”) (#曜日, 土曜日, “土曜日”) (#金額, ¥8000, “¥8,000”)
41	(#週末料金, 週末料金, “週末料金”) (#復路現地発, 復路現地発, “復路現地発”) (#曜日, 金曜日, “金”) (#曜日, 土曜日, “土”) (#金額, ¥5000, “¥5,000”)

5.3.2 トークンの反復畳み込みとクラスタリング

トークンの反復畳み込みについて述べる。トークン列 $S = s_1s_2 \dots s_l$ について、各トークン s_i の型 t_i を調べ、型の部分列 $t_jt_{j+1} \dots t_{j+k}$ が 1 回以上の繰り返しになっている箇所が見つかれば、それを型の列の繰り返しを表わすトークン $(t_jt_{j+1} \dots t_{j+k})^+$ で置換する。この操作を S の先頭から順次行ない、適用できなくなるまで変換する。畳み込まれたトークン $(t_jt_{j+1} \dots t_{j+k})^+$ を縮退トークンと呼ぶ。

縮退トークン列の集合について、縮退トークン列の型が一致するものをまとめ込む操作をルールのクラスタリングと呼ぶ。そのクラスタに対応するルール条項文の集合は、表記の揺らぎや具体的な値の違い、そして反復回数の違いを無視した上で同じ構造を持つルール条項文をまとめた集合となる。

表 4: まとめ込まれたトークン列の型
Table 4: Types of grouped token sequences

	列型
a	“週末料金” ¹ , “復路現地発” ² (₁ (₂ 曜日 ³) ₂ 金額 ⁴) ₁ ⁺
b	“週末料金” ¹ , “復路日本着” ² (₁ (₂ 曜日 ³) ₂ 金額 ⁴) ₁ ⁺

表 3 の例に対して、反復畳み込みを行った結果を表 4 に示す。ルール 3 は b で表される列型に、ルール 4,17,41 は a で表される列型を持つ縮退トークン列に畳み込まれる。この他にも、表 2 で示した例は全てこれら a, b いずれかの列型を持つ縮退トークン列に畳み込まれる。

5.3.3 変換テンプレートの適用

得られた縮退トークン列の集合に対し、縮退トークン列ごとに論理的表現への変換を与える変換テンプレートを用意する。変換テンプレートでは、縮退トークンの型および定数に応じて、変数および定数を割り当て、さらに縮退トークンの括弧の入れ子構造に対応して、論理演算を割り当ててゆく情報を持たせる。

縮退トークン列 τ の各部分列に変換テンプレートの対応する変換を適用するために、 τ の開き括弧と閉じ括弧に同じ番号からなる識別子を与え、また τ に含まれる各トークン型にも先頭から番号の識別子を与える。表 4 には、型に番号付け

を行っており、トークン型の番号は上付で、括弧の番号は下付で示されている。

次に、括弧に対応する部分列に対する変換規則について述べる。以下の具体的な各関数 f の計算は文字列操作などであり、変換スクリプトとして定義される。

表 4 の列型 a,b に対する変換テンプレートの一部として、以下の変換関数を定義する。この定義において、“.” は文字列連接の演算子であり、 $std(t)$ は t に対応するトークンの標準形を与える。

$$f_{3a}(t) \stackrel{\text{def}}{=} 'v_a ='.std(t), f_{3b}(t) \stackrel{\text{def}}{=} 'v_b ='.std(t),$$

$$f_4(t) \stackrel{\text{def}}{=} 'p ='.std(t)$$

$f_{3a}(b)$ は型が曜日であるトークンを引数に取り、“ $v_a(b) = \text{土曜日}$ ” のような式を返す。 f_3 は型が金額であるトークンを引数に取り、“ $p = \text{¥5000}$ ” のような式を返す。 v_a, v_b, p はそれぞれ復路現地発の曜日、復路日本着の曜日、追加料金額を表す商品属性に対応する変数である。 f_{3a}, f_{3b} はそれぞれ列型 a,b へのテンプレートに用いられ、 f_4 は両方に共通して用いられる。

次に、変換テンプレートの定義を行う。列型 τ を持つ縮退トークン列集合 C_τ への変換テンプレート M_{C_τ} とは、以下の規則に従う、 τ の各構成要素への、変換関数 f 、論理演算子 op 、空文字 ϵ のいずれかの割り当てである。構成要素 a に A を割り当てることを $M_{C_\tau} : a \mapsto A$ と書く。変換テンプレートの形式的定義は省略するが、以下の適用例から容易に理解できるものとする。表 2 のルールに対して、表 5 のテンプレートを適用した結果を表 6 に示す。

表 5: 変換テンプレートの例

Table 5: Example of transformation template

a	b
$M_{C_\tau} : (.)_1 \mapsto ' \vee '$	$M_{C_\tau} : (.)_1 \mapsto ' \vee '$
$M_{C_\tau} : (.)_2 \mapsto ' \vee '$	$M_{C_\tau} : (.)_2 \mapsto ' \vee '$
$M_{C_\tau} : (.)_2 \# \text{金額}^4 \mapsto ' \wedge '$	$M_{C_\tau} : (.)_2 \# \text{金額}^4 \mapsto ' \wedge '$
$M_{C_\tau} : \# \text{曜日}^3 \mapsto f_{3a}$	$M_{C_\tau} : \# \text{曜日}^3 \mapsto f_{3b}$
$M_{C_\tau} : \# \text{金額}^4 \mapsto f_4$	$M_{C_\tau} : \# \text{金額}^4 \mapsto f_4$

表 6: 変換された論理式の例

Table 6: Example of transformed logic expressions

17, 9, 14, 15, 22, 42, 46, 48, 59, 60	$(v_a = \text{金曜日} \vee v_b = \text{土曜日}) \wedge (p = \text{¥3000})$
2, 40	$v_b = \text{日曜日} \wedge p = \text{¥5000}$
3, 34, 36, 38, 39	$v_b = \text{日曜日} \wedge p = \text{¥3000}$
4, 6, 10, 12, 13, 14, 16, 23, 24, 26, 27, 29, 30, 41, 45, 50, 57	$(v_a = \text{金曜日} \vee v_a = \text{土曜日}) \wedge (p = \text{¥5000})$
5, 20, 44	$v_a = \text{土曜日} \wedge p = \text{¥5000}$
8, 49	$(v_a = \text{金曜日} \vee v_a = \text{土曜日} \vee v_a = \text{日曜日}) \wedge (p = \text{¥5000})$
11, 17, 31	$v_a = \text{金曜日} \wedge p = \text{¥5000} \vee v_a = \text{土曜日} \wedge p = \text{¥8000}$
18	$v_a = \text{金曜日} \wedge p = \text{¥5000} \vee v_a = \text{土曜日} \wedge p = \text{¥15000}$
19, 43, 58	$v_a = \text{土曜日} \wedge p = \text{¥3000}$
21	$(v_a = \text{金曜日} \vee v_a = \text{土曜日}) \wedge (p = \text{¥2000})$
25	$(v_a = \text{金曜日} \vee v_a = \text{土曜日}) \wedge (p = \text{¥8000})$
28, 53, 54, 56	$v_a = \text{日曜日} \wedge p = \text{¥3000}$
32, 33	$(v_b = \text{土曜日} \vee v_b = \text{日曜日}) \wedge (p = \text{¥3000})$
35, 37, 55	$v_a = \text{日曜日} \wedge p = \text{¥2000}$
51	$(v_a = \text{土曜日} \vee v_a = \text{日曜日}) \wedge (p = \text{¥3000})$
52	$v_a = \text{土曜日} \wedge p = \text{¥2000}$

6. 実装と評価

5. 節で述べた手法に基づいて、辞書を元にルール条項文をトークン抽出し、反復積み込みとクラスタリングを行うプログラムを Perl で実装した。変換テンプレートは未実装である。

前述の航空券データより抽出した 3766 件のルール条項文について、クラスタリングの結果 322 種（括弧付けのみが異なる

物を含めると 495 種）のパターンに類別することが出来た。辞書データについては、単語の型として曜日、価格、日付、都市名など 11 種類を用意し、それぞれに属する語をルール文から抜き出し登録することで辞書を作成した。辞書が完全でない段階では、変換誤りが見られたが、辞書に用語を追加することによりトークン抽出およびクラスタリングの精度の向上が見られた。

次に本変換支援手法の課題について述べる。トークン抽出の段階で、文の区切り方の選択により異なるトークンが抽出されることがある。例えば“日本”の“日”を日曜日と認識することである。これについては、抽出されるのトークンの候補の中から最もスコアの高いものを選ぶことにより対処できると考えられる。実験で使用した商品ルール文の大部分は、変換結果として例題に見られる AND と OR で表記できる、比較的単純な等号論理の制約クラスであった。“～は不可”など否定を含む表現については今後の課題である。

変換した結果の誤りを人間が修正する形式を取っているが、登録単語辞書や変換テンプレートを修正過程で適宜入力して精度向上を図れるような対話型のシステムも今後の課題である。

7. まとめ

本論文では、Web 上に自然言語で記述されたルール付商品の商品ルールを論理的表現に変換するために、トークン抽出や変換テンプレートを用いた変換支援手法を提案した。今後、実装を進めてさらに手法の改良を行なう予定である。

[文献]

- [1] Berners-Lee, T.: The Semantic Web-LCS seminar, <http://www.w3.org/2002/Talks/09-lcs-sweb-tbl/>
- [2] Boley, H.: The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL transformations, *INAP2001*, Tokyo, Japan, in October 2001(2001).
- [3] Grosz, B. N., Labrou, Y. and Chan, H. Y.: A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML, *E-COMMERCE 99*, Denver, Colorado(1999).
- [4] Iwaihara, M.: Supporting Dynamic Constraints for Commerce Negotiations, *2nd Int. Workshop in Advanced Issues of E-commerce and Web-Information Systems (WECWIS)*, IEEE Press, pp.12-20, June (2000).
- [5] Iwaihara, M.: Matching and Deriving Dynamic Constraints for E-Commerce Negotiations, *Workshop on Technologies for E-Services, (informal proceedings)*, Cairo, Sep. (2000).
- [6] Kang, Juyoung and Lee, Jae Kyu: Extrraction of Structured Rules from Web Pages and Maintenance of Mutual Consistency: XRML Approach, *Proc. of RuleML 2003*, Springer-Verlag, LNCS2876, pp. 150-163, (2003).
- [7] Kozawa, M., Iwaihara, M. and Kambayashi, Y.: Constraint Search for Comparing Multiple-Incentive Merchandises, *Proc. of EC-Web 2002*, Springer, LNCS2455, pp.152-161, Sep. (2002)
- [8] 小澤正幸, 岩井原瑞穂, 上林彌彦: 商取引ルールの動的制約表現とデータベース化手法, 夏のデータベースワークショップ (DBWS2003) 予稿集, 情報処理学会研究報告 IPSJ SIG Technical Reports 2003-DBS-131 (II) pp.63, 2003/07
- [9] Kuper, G., Libkin L. and Paredaens, J. (Eds.): *Constraint Databases*, Springer-Verlog, (2000).

小澤正幸 Masayuki KOZAWA

2004 年 3 月京大情報学研究科社会情報学専攻修士課程修了。2004 年 4 月より凸版印刷株式会社勤務。

岩井原瑞穂 Mizuho IWAHARA

1993 年 3 月九大工学研究科情報学専攻博士後期課程修了。2001 年 4 月より京大情報学研究科社会情報学専攻助教授。

上林 彌彦 Yahiko KAMBAYASHI

1970 年京大工学研究科博士課程修了。京大情報学研究科教授および同研究科長、電子情報通信学会情報システムソサイエティ会長、日本データベース学会会長などを務める。2004 年 2 月 6 日逝去。