自然言語処理 154−27
(2003. 3. 7)

Support Vector Machine を用いた談話構造解析

横山 憲司[†] 難波 英嗣[‡] 奥村 学[§]

概 要：人間は，テキストの文脈や談話構造を容易に理解することができるが，そのような構造をコンピューターが自動で解析することは非常に困難である．テキスト中の文間の関係を解析し，テキストの構造を明らかにすることは，一般に談話構造解析と言われている．本研究では，機械学習アルゴリズム Support Vector Machine(SVM) を用いた談話構造解析を提案する．

Discourse Structure Analysis using Support Vector Machines

Kenji YOKOYAMA[†] Hidetsugu NANBA[‡] Manabu OKUMURA[§]

Abstract: Although man can understand the context and the discourse structure of a text easily, it is very difficult for a computer to analyze the structure. Generally it is called discourse structure analysis to analyze the relation between sentences and to clarify structure of a text. In this paper, we propose a method for the discourse structure analysis using Support Vector Machines.

1 はじめに

一般に談話は，談話単位と呼ばれる談話の意味単位に分割され，その談話単位間には様々な関係が成立することが分かっている．このようなテキスト中の談話単位間の関係を解析し，テキストの構造を明らかにすることは，一般に談話構造解析と言われている．

談話構造解析ができると文脈や文間の係り受けを考慮した質の高い要約 [4] や機械翻訳によって生成された文の並べ替え [5]，文脈を考慮した照応解析 [2, 1] などが可能になる．

従来，談話構造解析の研究は，接続詞などの手がかり語を用いて係りやすさを決定するルールを人手で作成していた．しかし，談話構造解析で用いられる素性は膨大であり，一貫性，網羅性という点で問題が多い．また，解析規則のメンテナンスに多大な人的資源を投入する必要性もある．

これらの問題を解決するための方法として，機械学習を用いた談話構造解析が考えられる．機械学習を用いた先行研究には，野本ら [6] の決定木を用いた手法がある．しかし，決定木は素性の選択が難しく，有効な素性の組み合わせを学習できないという問題がある．本研究では機械学習アルゴリズム Support Vector Machine(SVM) を用いた談話構造解析を提案する．SVM は従来の学習モデルと比較して，高次元の入力ベクトルに対しても過学習しにくく，極めて高い汎化能力があるとされており，素性の次元に対して訓練データが少ない場合にも効率的に学習できる．談話構造解析のように有効な素性の組み合わせを求めることが難しい場合，SVM が現在最も有効な学習モデルである．

2 節では，談話構造解析の背景について説明し，従来研究について述べる．3 節では，本研究で用いる談話構造コーパスの仕様を述べる．4 節では，SVM を談話構造解析に適用する手法を述べた後，談話構造解析アルゴリズムについて説明する．5 節では，実装した談話構造解析システムを用いて実験し，評価を行う．最後に，6 節で本研究のまとめと今後の課題について述べる．

[†]東京工業大学大学院 総合理工学研究科
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
yoko@lr.pi.titech.ac.jp
[‡]広島市立大学 情報科学部
Graduate School of Information Sciences, Hiroshima City University
nanba@its.hiroshima-cu.ac.jp
[§]東京工業大学 精密工学研究所
Precision and Intelligence Laboratory, Tokyo Institute of Technology
oku@pi.titech.ac.jp

2 談話構造解析の背景

2.1 修辞構造理論

修辞構造理論 (Rhetorical Structure Theory, RST)[3] は，テキスト中の文の機能を特定し文間の依存関係を特定するモデルである．テキスト中の文は他の文に対し，説明，例示，帰結などの機能を持って存在する．修辞構造理論は，文と文に結束性があると認められた文章の分析から，文と文の関係を形式化して記述した理論である．そして，文と文の関係において，重要な方の文を**核**，重要でない方の文を**衛星**と呼ぶ．

2.2 従来研究

福本ら [10] は，新聞社説記事等の論説文が，ある事柄についての筆者の考えや意見などの筆者の主張を述べることを目的とした文章であるということを仮定した上で，筆者の主張に基づく日本語文章の構造化に注目し，文章の構造化を行っている．

住田ら [7] は，2 つの文章階層 (書式構造と修辞構造) から文章の構造を抽出している．書式構造とは文章の章や節の階層構造を表現し，修辞構造とは各章や節内の文や文のまとまりの間の論理的な関係を表現する．書式構造は章見出しや章番号，インデンテーションなどを手がかりを利用して，修辞構造は接続詞や照応表現，文末表現など言語的な手がかりとして解析している．

黒橋ら [9] は，技術論文を対象とし，さまざまな手がかり語に基づいた，依存構造パターン照合を主に利用している．接続詞を中心とした手がかり語，主題連鎖情報，類似度情報を利用して，手がかり語の結束関係のスコアの合計によって，構造を解析している．

先行研究では，対象とするテキストのジャンルが限られており，対象外のジャンルでは，パフォーマンスが悪い．また，解析規則を人手で作るため，規則が複雑になり，規則の修正・拡張が困難になっている．

3 コーパス

本研究で使用するSVMは，教師付き学習器であるため，訓練データとして談話構造を付与したコーパスが必要になる．しかし，これまでの談話構造解析の研究では，主に人手で作った規則を用いて解析する手法をとっていたため，談話構造解析用のコーパスは皆無に等しい．そこで，本研究では，談話構造を付与したコーパスを作成した．

3.1 コーパスに用いたデータ

本研究の談話構造コーパスは，テキスト自動要約タスクTSC(Text Summarization Challenge)[11]で使用されている新聞記事を元に作成したものである．合計180記事を使用した．

3.2 談話の基本単位

地の文に出現する句点“。”で区切られる文字列を談話の基本単位とする．つまり，会話文に出現する句点は談話の基本単位の区切りとして扱わない．以降，この談話の基本単位のことを簡単に文と呼ぶことにする．

3.3 談話構造の表現方法

談話構造は木構造であるという仮定の下で，談話構造を文と文の二項関係の集合で表現する．二項関係を用いることで，木のノード(文)とノードの結束関係を示すことができ，あらゆる木構造を表現することが可能になる．具体的には，修辞構造理論に基づいて，核と衛星を明示し，2文間の関係で談話構造を表現する．文と文の二項関係について，木の根に近い方に位置する文が核であり，葉に近い方が衛星である．

3.4 係り受け関係のタイプ

本研究では，談話構造だけではなく，係り受け関係のタイプも解析するため，二項関係に係り受け関係のタイプを付与する．本研究で使用する係り受け関係のタイプは，先行研究[10, 7, 3, 9, 8]で使用されている係り受け関係を比較統合し決定した．図1に，本研究で使用する係り受け関係8つを挙げ，その定義を示す．

因果	一方の文と，もう一方の文の間に因果関係が明確に存在する場合．実際の出来事の原因，筆者の主張の根拠，理由などが含まれる．
背景	一方の文で事実，出来事が述べられて，もう一方の文でそれらの背景となる事柄や前提が述べられている．歴史的背景などはこれに相当する．
呼応	一方の文の問いかけ文に対してもう一方の文で答えが示される．
並列	一方の文に追加する形で，もう一方の文でも同様な内容を述べている．同じ内容や種類の文の列挙，換言，要約がこれに相当する．
対比	一方の文に対して，もう一方の文が対比関係になっている．
転換	これまでの話題が一転して新たな話題に変わる．
補足	一方の文で述べられていた事柄についての補足や説明がもう一方の文で述べられている．一般的に，補足の文は文章から省いても，文章全体の意味は変わらない．
例示	一方で述べられた事柄の具体例がもう一方の文で提示される．

図1: 係り受け関係の定義

4 提案手法

4.1 解析のおおまかな流れ

構造木を構築する手続きは*shift-reduce*法と類似したものを使用する．談話構造解析は，談話構造木を構築する手続きのシーケンスを得ることに等しい．まず，コーパスから談話構造木を構築する手続きのシーケンスを抽出し，そのシーケンスをトレースしながら，構築する手続きと文間の関係タイプ，及びその位置での素性集合のペアを獲得する．この構築手続きと関係タイプ，及び素性集合の3つの組が一つの訓練事例となり，これが複数集まったものが訓練データとなる．このようにして獲得した訓練データからSVMを用いて，構築手続きの分類規則と，関係タイプの決定規則を学習する．訓練データから得た構築手続きの分類規則モデルと関係タイプの決定モデルを用いて，解析対象の文から抽出した素性集合に対する構築手続きと関係タイプを分類し，文章全体の談話構造木を構築していく．

4.2 談話構造解析アルゴリズム

談話構造解析アルゴリズムは，入力テキストに対し，テキストの始めの文から終わりの文にかけて順番に上昇型アルゴリズムを用いて解析木を構築する．本研究では，Marcu[4]の解析手法で使用している*shift-reduce*法を拡張して，2つのスタックを用いた解析アルゴリズムを考案した．構築手続きとして，解析位置を一つ進める**SHIFT**，交差した係り受けを受理できるようにするための手続き**PASS**と，実際に構造木を組み上げるための手続き**REDUCE-NS**，**REDUCE-SN**，**REDUCE-NN**の計5つを考える．以下に，5種類の構築手続き**SHIFT**，**PASS**，**REDUCE-NS**，**REDUCE-SN**，**REDUCE-NN**の詳細を述べ，図2を用いて，構築手続きを説明する．

- **SHIFT**: スタックを用いて解析位置を一つ前に進める手続きである．具体的な操作は，以下の通りである．入力リストからedt(elementary discourse tree；文番号のラベルが付いた四角い箱)を1つ取り出し，SHIFTスタックにpushする．図2のstep iに**SHIFT**を適用すると，step i+1の状態になる．同様にstep i+5に適用するとstep i+6になる．
- **PASS**: **SHIFT**と**REDUCE-{NS,SN,NN}**だけを用いて構築できる木は，図3のような係り受けが交差していない談話構造木に限られる．ところが，談話構造は係り受けが交差することがある．そこで，構築手続きに**PASS**を追加することにより，図4のような係り受けが交差した場合の解析も可能にした．具体的な操作は以下の通りである．SHIFTスタックのtopとtop-2以下にreduce可能な関係がある場合，top-1を取り出してPASSスタックにpushする．図2のstep i+2で，2と7がreduce可能になった場合，top-1の5をPASSスタックに移動する．step i+3でも同様に，top-1の3をPASSスタックに移動する．
- **REDUCE-{NS,SN,NN}**: この構築手続きは構造木を実際に組み立てる手続きである．具体的な操作は以下の通りである．SHIFTスタック

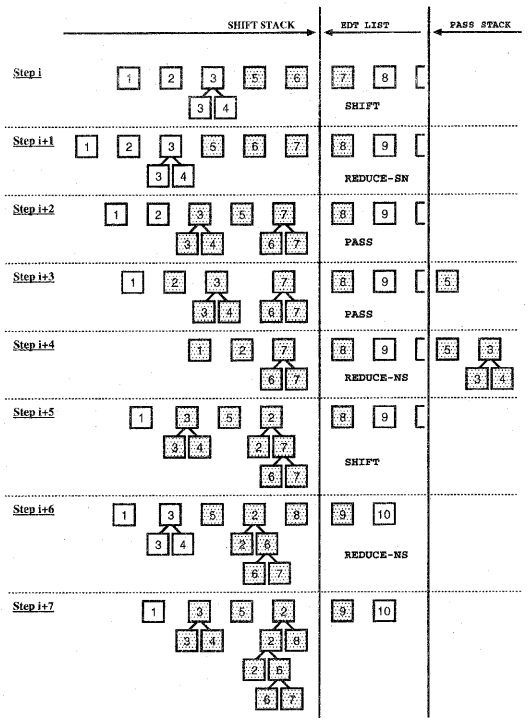

図 2: 談話構造解析における *shift-reduce* 手続きの例

から edt を 2 つ pop して得られる top と top-1 に対し，核を親ノードとして新しい木をつくる．次に，PASS スタックをすべて pop してスタックに push する．最後に，その新しく作られた木を SHIFT スタックに push する．核 (Nucleus) の文番号が先で衛星 (Satellite) の文番号が後の場合，**NS** であり，その逆の場合は **SN** である．また，多核関係で両方が核の場合 **NN** である．図 2 の step i+1 で **REDUCE-SN** が適用されると，6 と 7 が SHIFT スタックから pop され，6 が衛星で，7 が核となり，7 を親ノードとして新たな木 7 を作る．この場合，PASS スタックは空なので，そのまま，その新しい木 7 を SHIFT スタックに push して，step i+2 の状態になる．また，step i+4 で **REDUCE-NS** が適用されると，2 と 7 が SHIFT スタックから pop され，2 が核で，7 が衛星となり，2 を親ノードとして新たな木 2 を作る．この場合，PASS スタックには，3 と 5 が格納されているので，新たな木 2 を SHIFT スタックに格納する前に，3 と 5 を SHIFT スタックに移動しておく．その後で，新しい木 2 を SHIFT スタックに push して，step i+5 の状態になる．

次に，上に示した 5 種類の構築手続きを用いて，どのように解析木を構築するかを説明する．図 5 に解析アルゴリズムの疑似コードを示す．図 5 で，*edtList* は，入力リストを示している．これは，テキスト内における基本談話単位 (文) から生成した，一つのノードからなる基本談話木 (Elementary Discourse Tree) を格納したリストを示している．e_i は，第 i 番目の文から

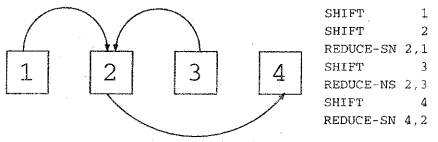

図 3: 交差した係り受け関係を持たない談話構造

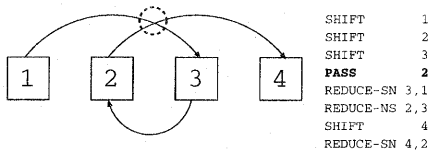

図 4: 交差した係り受け関係を持つ談話構造

生成した基本談話木である．談話構造木は，この基本談話木をボトムアップに積み上げていくことで得られる．また，*sftStack* は SHIFT スタックを，*pssStack* は PASS スタックを示している．関数 *get-features* は，入力リスト，SHIFT スタック，PASS スタックにおける注目木 (後述) から素性を抽出し，素性ベクトル feat を返す関数である．PROC_MODEL は，SVM を用いて構築手続きを学習して得たモデルを表している．一方，TYPE_MODEL は，SVM を用いて係り受け関係のタイプを学習して得たモデルを表している．関数 *classify-proc* は，PROC_MODEL と関数 *get-features* で抽出した素性ベクトル feat から，現在の解析位置における適切な構築手続き proc を返す．同様に，関数 *classify-type* は，TYPE_MODEL と関数 *get-features* で抽出した素性ベクトル feat から，先ほど得た構築手続き proc の係り受けのタイプ type を返す．関数 *reduce* は，構築手続き pro と係り受けタイプ type を引数にし，構築手続きを実行し新たな木を構築する．構築した木 newtree は，PASS スタックの要素をすべて SHIFT スタックに移した後で，SHIFT スタックに移す．

4.3 SVM を用いた解析木構築規則の学習

4.3.1 素性を抽出する範囲

素性は，木の集合から得られる素性と，木そのものから得られる素性に分けることができる．前者は，木の格納されたスタックや，木を構築するために適用した構築手続きの履歴などであり，後者は，木の示す文の文番号や，その文に使用されている文頭表現，文末表現などである．前者を**構築素性**と呼び，後者を**木素性**と呼ぶことにする．

木素性は，文字通り木から抽出した素性である．ここで，どの木から素性を抽出するかが問題となる．本研究では Marcu[4] を参考に，(1) 入力リストの先頭に格納されている木，(2)SHIFT スタックの上から 3 番目までに格納されている木，(3)PASS スタックに格納されているすべての木，から素性を抽出することにする．これら (1)，(2)，(3) の木を解析時に注目している木という意味で，**注目木**と呼ぶことにする．注目木は，図 2 の網掛けの部分である．

4.3.2 素性

本研究で使用する構築素性を以下に示す．

- SHIFT スタックのサイズ：`sstack_size`
  SHIFT スタックの中に格納されている談話構造

```
edtList := [e_1, e_2, ··· , e_i, ··· , e_n]
shiftStack := []
passStack := []
until |edtList| = 0∧|sftStack| = 1∧|pssStack| = 0
begin
  feat := get-features(edtList, sftStack, pssStack)
  proc := classify-proc(PROC_MODEL, feat)
  type := classify-type(TYPE_MODEL, feat)
  if proc = REDUCE-{NN,NS,SN}
    newtree := reduce(sftStack.pop,sftStack.pop,
proc, type)
    unitl |passStack| = 0
      sftStack.push(pssStack.pop)
    end
    sftStack.push(newtree)
  else if proc = PASS
    top := sftStack.pop
    pssStack.push(sftStack.pop)
    sftStack.push(top)
  else if proc = SHIFT
    sftStack.push(edtList.pop)
  end
end
```

図 5: 談話構造解析アルゴリズムの疑似コード

木の数である．

- PASS スタックのサイズ：pstack_size
  PASS スタックの中に格納されている談話構造木の数である．
- 解析場所：in_par
  解析している場所が，段落内であるか，段落外であるか．
- 構築手続きの履歴 (分割)：history-n
  過去 5 回分の履歴のシーケンスを 5 つに分割し，5 つの素性として扱う．n には，$1 \cdots 5$ が入る．
- 構築手続きの履歴 (結合)：history
  過去 3 回分の履歴のシーケンスを 1 つの素性として扱う．
- 文間距離：distance
  文間の距離を素性にする場合，どの文とどの文の距離を素性にするかが問題となる．本研究では，SHIFT スタックの top と top-1 の文の距離を素性にする．なぜなら，注目木から抽出した素性を用いて，構築手続きを分類し，実際に構築 (reduce) される 2 つの木は，SHIFT スタックの top と top-1 であるからである．このような理由から，SHIFT スタックの top と top-1 の文の距離を素性にするのが適切であると考えた．

本研究で使用する木素性を以下に示す．

- 文章内文番号：d_snum
  文章内における，文の位置である．
- 段落内文番号：p_snum
  段落内における，文の位置である．
- 文タイプ：type
  文タイプとは，文末文字列により決まる文のタイプのことである．本研究では，主張文，叙述文の 2 種類を用いる．本研究では解析対象の記事ジャンルを一つに固定しないため，記事ジャンルを表すような素性が必要になる．そこで，主張文が多ければ，筆者の意見を述べた文 (社説記事等) であり，少なければ，事実を述べた文章 (報道記事，解説記事等) であるという仮定の下で，この素性を使用する．この文タイプは，人手で作成した，文末文字列と文タイプの対応規則を用いて決定される．
- 文末表現：endexp
  文末表現とは，文末文字列により決まる文末の表現のことである．本研究では，過去，現在，断定，存在，推量，様態，問掛，判断，可能，理由，要望，叙述，義務，意見の 14 種類を採用した．この文末表現は，人手で作成した，文末文字列と文末表現の対応規則を用いて決定される．
- 文頭表現：conj
  接続詞をはじめとして，文と文の係り受け関係を明示する文字列は，文頭に現れることが多い．表 1 に，本研究で使用した 55 種類の文頭表現を示す．

けれど，こうしたことから，この，このうち，このため，これ，これに，これに対し，これまでは，さすがに，さて，さらに，しかし，しかしながら，しかも，したがって，すなわち，そこから，そこで，そして，その，そのため，その結果，そもそも，それ，それでも，それならば，ただ，ただし，たとえば，だから，だが，つまり，でも，というのは，ところが，ところで，とすれば，どうして，どのように，なかでも，なぜ，なぜなら，まず，また，もっとも，一方，一方で，逆に，結局，結論として，従って，当然，同様に，例えば

表 1: 文頭表現

- 文頭表現の有無：conjex
  接続詞をはじめとする文頭表現は，前の文と解析位置の文との係り受け関係のタイプを決定する上では，重要な素性になると考えられるが，文と文の結束関係を決定するために使用する素性としては，冗長であると考えられる．そこで，表 1 に挙げた文頭表現の有無を素性として採用する．
- 構築手続き：proc
  素性を抽出しようとしている木はすべて，基本談話木でない限り，構築手続き REDUCE-NS, REDUCE-NN, REDUCE-SN のいずれかによって構築されている．この情報を素性とする．

5 実験と考察

提案する手法に基づき，実際の談話構造コーパスを用いて，実験を行った．本節では，実験結果の報告と考察を行う．

5.1 実験環境

実験に用いたコーパスは，本研究で作成した談話構造コーパスである．談話構造コーパスは，新聞記事180記事に対し，4節で示した仕様に基づいて，人手で談話構造を付与して作成したものである．本実験で使用するコーパスは小さいので，解析における結果の妥当性・再現性を高めるために 10-fold cross validation を行った．

5.2 談話構造の解析精度

システムの出力した談話構造木と，コーパスの談話構造木の一致度を談話構造木の解析精度とする．木の一致度とは，コーパスの談話構造木を構成する基本談話木の二項関係からなる要素集合が，システムが出力した談話構造木のそれを内包する割合のことである．コーパスは基本談話木の二項関係の集合で談話構造木を表現しているので，システムの出力する談話構造木の表現もコーパスと同様，基本談話木の二項関係の集合としている．一致の基準として，二項関係を作る基本談話木の組み合わせと核，衛星の状態の両方が一致している場合と，二項関係を作る基本談話木の組み合わせだけが一致している場合の二種類を考える．前者の一致を Perm，後者の一致を Comb と呼ぶ．以上を具体例 (図 6) を用いて説明する．

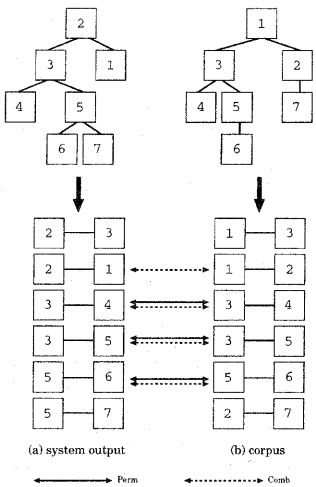

図 6: 談話構造木の一致の例

(a) がシステムの出力した談話構造木であり，(b) がコーパスの談話構造木，つまり正解データの談話構造木である．この談話構造木は，根に近い方が核で，葉に近い方が衛星となっている．(a) の談話構造木を基本談話木の二項関係 (前が核，後が衛星 ;「核-衛星」と表す) の集合で表すと (2-3, 2-1, 3-4, 3-5, 5-6, 5-7) となり，(b) の場合は，(1-3, 1-2, 3-4, 3-5, 5-6, 2-7) となる．Perm で一致した二項関係は，(3-4, 3-5, 5-6) の 3 つであり，Comb で一致した二項関係は (2-1, 3-4, 3-5, 5-6) の 4 つである．(a) のシステムが出力した談話構造木は，二項関係 6 つのうちで Perm は 2 つ一致しており，Comb は 4 つ一致している．したがって，Perm での一致度は $3/6 = 0.50$ であり，Comb での一致度は $4/6 = 0.66$ となる．

5.2.1 素性の組み合わせと解析精度の関係

カーネル関数には，RBF カーネルを使用し，ソフトマージンのパラメータ C は，すべての実験を通して 1000 に固定した．

まず，素性 14 種類すべてを学習に用いた S0(表 2) で実験をする．

ID	es	ss	ps	ip	hd	hu	ds	dn	pn	tp	ed	ct	ce	pr	Nm
S0	○	○	○	○	○	○	○	○	○	○	○	○	○	○	897

es：入力リストのサイズ，ss：SHIFT スタックのサイズ，ps：PASS スタックのサイズ，ip：解析場所，hd：構築手続きの履歴 (履歴を分割する場合)，hu：構築手続きの履歴 (履歴を分割しない場合)，ds：文間距離，dn：文章内文番号，pn：段落内文番号，tp：文のタイプ，ed：文末表現，ct：文頭表現，ce：文頭表現の有無，pr：構築手続き，Nm：素性数

表 2: 使用した素性 (S0)

段落内の文間の係り受けモデルと段落間の係り受けモデルは一般に異なっていると考えられる．そこで，段落内と段落外の両方で別々に係り受けモデルを作成した場合と，段落内と段落外を区別せずに係り受けモデルを作成した場合の両方について，解析精度を求めた．

また，本システムを評価するために，baseline を設定する．談話構造は，人間が文章を読む順番に依存することが多い．人間にとって文章を初めの文から終わりの文に向かって読むことが一般的な読み方だとすると，注目している文は，直前の文か直後の文との間になんらかの係り受け関係があると考えられる．そこで，注目している文が，常に次の文から係られているような談話構造を baseline とする．表 3 に，baseline の解析精度を示す．

	全体		段落内		段落外	
	Perm	Comb	Perm	Comb	Perm	Comb
B	39.06	58.39	52.64	77.82	13.28	17.81

表 3: baseline の解析精度

表 4 に，段落内と段落外を区別せずに作成した係り受けモデルを用いた場合の解析精度 (X) と，区別して係り受けモデルを作成した場合の解析精度 (Y) を示す．「段落内」は，段落内の文間の係り受けによって構築された談話構造木の解析精度を示し，「段落外」は，段落間の係り受けによって構築された談話構造木の解析精度を示している．また，「全体」は，テキスト全体に対して構築された談話構造木の解析精度を示している．

	全体		段落内		段落外	
	Perm	Comb	Perm	Comb	Perm	Comb
X	42.69	58.41	55.27	79.51	25.34	29.18
Y	43.30	59.14	55.35	80.02	25.85	29.44

表 4: S0 における解析精度

表 3，表 4 から，すべての項目の解析精度において，B < X < Y であることがわかる．この結果は，段落内と段落外での係り受けモデルは異なっており，文間の係り受け関係と，段落間の係り受け関係の性質が違うものであるということを意味している．以降の実験では，段落内と段落外で別々に作成した係り受けモデルを使用する．

次に，14 種類の素性をすべて使用したときの解析精度と，表 5 に示すように 1 つの素性を省いた残りの 13 の素性だけを使用した場合の解析精度とを比較するこ

とで，省いた素性が談話構造解析にとって有効かどうかを調査する．

ID	es	ss	ps	ip	hd	hu	ds	dn	pn	tp	ed	ct	ce	pr	Nm
S1	×	○	○	○	○	○	○	○	○	○	○	○	○	○	871
S2	○	×	○	○	○	○	○	○	○	○	○	○	○	○	884
S3	○	○	×	○	○	○	○	○	○	○	○	○	○	○	892
S4	○	○	○	×	○	○	○	○	○	○	○	○	○	○	895
S5	○	○	○	○	×	○	○	○	○	○	○	○	○	○	872
S6	○	○	○	○	○	×	○	○	○	○	○	○	○	○	798
S7	○	○	○	○	○	○	×	○	○	○	○	○	○	○	863
S8	○	○	○	○	○	○	○	×	○	○	○	○	○	○	608
S9	○	○	○	○	○	○	○	○	×	○	○	○	○	○	837
S10	○	○	○	○	○	○	○	○	○	×	○	○	○	○	881
S11	○	○	○	○	○	○	○	○	○	○	×	○	○	○	798
S12	○	○	○	○	○	○	○	○	○	○	○	×	○	○	704
S13	○	○	○	○	○	○	○	○	○	○	○	○	×	○	881
S14	○	○	○	○	○	○	○	○	○	○	○	○	○	×	877

表 5: 素性の組み合わせ (S1～S14)

表 5 に示す 14 種類の素性の組み合わせについて，解析精度を求めた．表 6 に，各素性の組み合わせとそれに対する解析精度を示す．

ID	全体		段落内		段落外	
	Perm	Comb	Perm	Comb	Perm	Comb
S1	**41.62**	57.75	**54.35**	79.43	**24.11**	27.76
S2	**42.93**	58.93	55.58	80.44	**24.53**	28.18
S3	43.28	59.19	**55.36**	80.07	25.73	29.37
S4	43.14	58.94	55.39	80.03	25.32	28.88
S5	43.37	59.22	55.66	80.18	25.68	29.56
S6	43.31	59.02	55.65	80.21	25.61	29.13
S7	**42.93**	58.89	**55.13**	80.11	25.65	29.14
S8	43.53	58.62	57.12	80.03	**23.84**	28.08
S9	43.28	58.71	56.22	80.23	24.90	28.64
S10	43.10	59.14	55.96	80.72	**24.59**	28.58
S11	44.00	59.34	57.57	80.51	24.85	29.42
S12	**41.34**	58.21	**53.48**	79.45	**24.12**	28.30
S13	44.30	59.47	56.92	80.42	26.26	29.74
S14	**42.51**	58.54	**54.92**	79.99	24.64	28.14

表 6: 素性の組み合わせと解析精度 (S1～S14)

各項目の Perm について，解析精度の低いものから 5 つをボールドで表示している．素性を削除して解析精度が下がる場合，その削除した素性は解析精度の向上に有効であり，逆に，削除して解析精度が上がる場合，その削除した素性は解析精度の向上に有効でないと考えられる．表 6 をもとに，素性を解析精度の向上に有効な順番にまとめたものが表 7 である．

順位	解析精度の向上に有効な素性		
	全体	段落内	段落外
1	文頭表現	文頭表現	文章内文番号
2	入力リストのサイズ	入力リストのサイズ	入力リストのサイズ
3	構築手続き	構築手続き	文頭表現
4	文間距離	文間距離	SHIFT スタックのサイズ
5	SHIFT スタックのサイズ	PASS スタックのサイズ	文タイプ
6	文タイプ	解析場所	構築手続き
7	解析場所	SHIFT スタックのサイズ	文末表現
8	段落内文番号	構築手続きの履歴 (統合)	段落内文番号
9	PASS スタックのサイズ	構築手続きの履歴 (分割)	解析場所
10	構築手続きの履歴 (統合)	文タイプ	構築手続きの履歴 (統合)
11	構築手続きの履歴 (分割)	段落内文番号	文間距離
12	文章内文番号	文頭表現の有無	構築手続きの履歴 (分割)
13	文末表現	文章内文番号	PASS スタックのサイズ
14	文頭表現の有無	文末表現	文頭表現の有無

表 7: 解析精度の向上に有効な素性

表 6，表 7 から以下のことがわかる．

- 文頭表現，入力リストのサイズ，構築手続き，SHIFT スタックのサイズは，段落内にも段落外にも有効である．
- 文間距離と PASS スタックのサイズは段落内に有効であり，段落外には有効でない．
- 文章内文番号と文タイプは段落外に有効であり，段落内には有効でない．
- 文末表現の有無は，段落内にも段落外にも有効でない．
- 文末表現は段落内では有効でない．
- 構築手続きの履歴に関しては分離，結合共に段落外には有効でない．

段落内の談話構造は文間の係り受け関係によって構築されるため，係り受け関係を明示する接続詞を含む文頭表現や，文と文の位置的な近さを示す文間距離などの文の結束性を直接的に反映するような素性が有効であると考えられる．逆に，文章内や段落内での文番号，文タイプなどは前後の文の結束性を明示する情報というよりは，むしろ，その文自体の役割や性質を表しているため，文間の係り受け関係のような直接的な結束性を必要とする解析には有効でない．一方，段落外の談話構造解析は，文章内文番号や，文タイプなど，段落内で有効でなかった素性が有効である．このことから，段落間の係り受け関係は文間の係り受け関係のように明示的な手がかり語や，位置的な近さだけでは決定できず，文タイプのような文自身の持つ役割や，テキスト内での位置によって決定される傾向があるとわかる．

次に，表 8 のような素性の組み合わせを作成した．これは，表 7 を参考に，解析精度に有効な素性上位 5 つを残して下位 5 つの素性を削除し (上位 5 つの素性は●，下位 5 つの素性は■で示している．)，それ以外の素性を一つずつ順番に削除していきながら素性の組み合わせを作成する．

つまり，段落内において解析精度の向上に有効でない上位 5 つ「文末表現 (ed)」,「文章内文番号 (dn)」,「文頭表現の有無 (ce)」,「段落内文番号 (pn)」,「文タイプ (tp)」を削除した組み合わせを S15，S15 から下位 6 番目の素性「構築手続きの履歴 (分割)(hd)」を削除した組み合わせを S16，S16 から下位 7 番目の素性を削除した組み合わせを S17 というように，上位 6 番目の素性になるまで繰り返して，S15～S19 を作成する．段落外の場合も段落内の場合と同様の手順で，S20～S24 を作成する．

ID	es	ss	ps	ip	hd	hu	ds	dn	pn	tp	ed	ct	ce	pr	Nm
S15	●	○	●	○	○	○	●	■	■	■	■	●	■	●	417
S16	●	○	●	○	×	○	●	■	■	■	■	●	■	●	392
S17	●	○	●	○	×	×	●	■	■	■	■	●	■	●	293
S18	●	×	●	○	×	×	●	■	■	■	■	●	■	●	280
S19	●	×	●	×	×	×	●	■	■	■	■	●	■	●	278
S20	●	●	■	○	■	■	■	●	○	●	○	●	■	○	718
S21	●	●	■	×	■	■	■	●	○	●	○	●	■	○	716
S22	●	●	■	×	■	■	■	●	×	●	○	●	■	○	656
S23	●	●	■	×	■	■	■	●	×	●	×	●	■	○	557
S24	●	●	■	×	■	■	■	●	×	●	×	●	■	×	537

表 8: 素性の組み合わせ (S15～S24)

表 9 に，各素性の組み合わせ (S15～S24) とそれに対する解析精度を示す．ここでは，さらにより細かな素性の組み合わせを学習するために，カーネル関数に，多項式カーネルを使用し，ソフトマージンのパラメータ C は，すべての実験を通して 1 に固定した．また，次元 p を 10 に設定した．

S15～S19 と baseline(表 3) を比較すると，これらは全体，段落内の Perm で共に約 13% 上回る結果となっている．段落外の Perm に至っては，baseline の約 20% 上回る結果となっている．また，14 種類の素性をすべて用いた S0 の解析精度 (表 4) と比べても，全体と段

ID	全体		段落内		段落外	
	Perm	Comb	Perm	Comb	Perm	Comb
S15	49.87	63.61	62.90	82.75	31.08	36.14
S16	50.60	64.29	63.04	83.00	32.48	37.46
S17	50.82	63.80	63.88	82.48	31.76	36.94
S18	52.35	64.59	65.40	82.65	33.39	38.58
S19	51.74	63.90	65.06	82.11	32.26	37.52
S20	49.23	63.32	61.82	82.94	30.68	34.97
S21	49.24	63.40	61.82	83.13	30.63	34.83
S22	49.62	63.48	62.33	83.07	30.90	35.19
S23	50.00	63.63	63.18	83.19	30.93	35.50
S24	50.16	64.04	63.17	83.72	31.02	35.72

表 9: 素性の組み合わせと解析精度 (S15～S24)

落内ともに Perm で約 10% 上回る結果となっている．しかしながら，S20～S24 は，段落外の解析精度の向上に有効な素性の組み合わせであるにも関わらず，低い解析精度となっている．これは，段落外の解析精度が段落内の解析精度に依存するため，段落内の解析精度が悪いとそれに応じて段落外の精度も悪くなるからである．

段落内の解析精度が最も良い S18 の係り受けモデルを段落内の解析に使用し，段落外の解析精度が最も良い S24 の係り受けモデルを段落外の解析に使用した場合の解析精度を求めたところ，全体の Perm が 53.82%，Comb が 65.19% となった．

5.3 関係タイプの分類精度

本研究では，談話構造のほかに，係り受け関係のタイプも決定する．本節では，関係タイプの分類精度を調査した．

5.3.1 実験結果

表 2，表 5 に示す素性の組み合わせ S0～S14 について，関係タイプの分類精度を求めた．カーネル関数には，RBF カーネルを使用し，ソフトマージンのパラメータ C は，すべての実験を通して 1000 に固定した．その結果を表 10 に示す．

ID	全体	段落内	段落外
S0	41.79	38.03	47.18
S1	42.32	**38.00**	47.47
S2	**40.81**	38.25	**44.21**
S3	41.87	38.15	47.38
S4	41.64	38.15	47.13
S5	**41.34**	38.21	**46.51**
S6	**40.64**	**37.32**	47.70
S7	41.91	38.29	46.98
S8	42.64	40.87	**45.34**
S9	41.59	38.86	46.99
S10	42.21	39.37	47.57
S11	**41.42**	**38.03**	47.00
S12	**39.50**	**34.59**	**46.57**
S13	41.77	38.45	46.94
S14	41.46	**38.14**	**46.91**

表 10: 素性の組み合わせと関係タイプの分類精度 (S0～S14)

各項目について，分類精度の低いものから 5 つをボールドで表示している．素性と分類精度の関係をまとめると，表 11 のようになる．

順位	関係タイプの分類精度の向上に有効な素性		
	全体	段落内	段落外
1	文頭表現	文頭表現	SHIFT スタックのサイズ
2	構築手続きの履歴 (統合)	構築手続きの履歴 (統合)	文章内文番号
3	SHIFT スタックのサイズ	入力リストのサイズ	構築手続きの履歴 (分割)
4	構築手続きの履歴 (分割)	文末表現	文頭表現
5	文末表現	構築手続き	構築手続き
6	構築手続き	解析場所	文頭表現の有無
7	段落内文番号	PASS スタックのサイズ	文間距離
8	解析場所	構築手続きの履歴 (分割)	段落内文番号
9	文頭表現の有無	SHIFT スタックのサイズ	文末表現
10	PASS スタックのサイズ	文間距離	解析場所
11	文間距離	文頭表現の有無	PASS スタックのサイズ
12	文タイプ	段落内文番号	入力リストのサイズ
13	入力リストのサイズ	文タイプ	文タイプ
14	文章内文番号	文章内文番号	構築手続きの履歴 (統合)

表 11: 関係タイプの分類精度の向上に有効な素性

表 7 から表 8 を作成したのと同じ方法で，表 11 から表 12 のような素性の組み合わせを作成した．

ID	es	ss	ps	ip	hd	hu	ds	dn	pn	tp	ed	ct	ce	pr	Nm
S25	■	●	■	○	●	●	■	■	○	■	●	●	○	○	527
S26	■	●	■	○	●	●	■	■	○	■	●	●	×	○	511
S27	■	●	■	×	●	●	■	■	○	■	●	●	×	○	509
S28	■	●	■	×	●	●	■	■	×	■	●	●	×	○	449
S29	■	●	■	×	●	●	■	■	×	■	●	●	×	×	429
S30	■	●	■	■	●	■	○	●	○	■	○	●	○	●	749
S31	■	●	■	■	●	■	○	●	○	■	×	●	○	●	650
S32	■	●	■	■	●	■	○	●	×	■	×	●	○	●	590
S33	■	●	■	■	●	■	×	●	×	■	×	●	○	●	556
S34	■	●	■	■	●	■	×	●	×	■	×	●	×	●	540

表 12: 素性の組み合わせ (S25～S34)

- 先行研究 [9] によると，係り受け関係タイプの判別には接続詞をはじめとする手がかり語が有効であると報告されている．本研究における手がかり語は，文頭表現や文末表現に相当する．表 11 を見ると，文頭表現が関係タイプの分類に最も有効であるという結果が出ている．また，文末表現も分類精度を良くしている．表 7 を見ると，段落内において，文末表現は最も解析精度の向上が期待できない素性であったにも関わらず，関係タイプの分類精度の向上に有効な素性となっている．
- 文間距離は段落内での解析精度を良くするが，段落内の関係タイプの分類精度を悪くする．
- 段落内では構築手続きの履歴 (結合) が有効である．これは，文の意味的なつながりに一定のパターンが存在することを示していると考えられる．
- 入力リストのサイズは，解析精度の向上に非常に有効な素性であったが，関係タイプの分類には有効でない．

表 13 に，各素性の組み合わせ (S15～S24) とそれに対する分類精度を示す．ここでは，さらにより細かな素性の組み合わせを調査するために，カーネル関数に，多項式カーネルを使用し，ソフトマージンのパラメータ C は，すべての実験を通して 1 に固定した．また，次元 p を 10 に設定した．

S25～S29 では，素性が多いときでも少ないときでも段落内の分類精度はあまり変わらない．段落外では，素性の数が多くても少なくても分類精度が悪くなる．S30～S35 を見ると，段落内では素性の数が少ない方が分類精度が良く，段落外では素性の数が多い方が分類精度が良い傾向がある．この結果から，段落内 (つまり，文と文の関係) のように人間にとって関係タイプがわかりやすい場合には，文頭表現などの有効な素性を使いさえすれば比較的高い分類精度が得られ，段落外 (つま

ID	全体	段落内	段落外
S25	45.75	44.42	46.99
S26	46.90	44.82	47.25
S27	47.17	44.79	47.35
S28	45.01	44.56	46.27
S29	44.11	44.18	46.31
S30	44.01	41.54	48.37
S31	43.69	40.56	46.87
S32	44.10	42.44	46.73
S33	43.49	42.31	45.89
S34	44.14	43.23	46.30

表 13: 素性の組み合わせと関係タイプの分類精度 (S15 ~S22)

り段落と段落の関係) のように人間にとっても関係タイプを決めるのが難しい場合は，多くの素性を用いて総合的に決定しなければ高い分類精度が期待できないということが分かる．

段落内での分類精度が最も良い S26 の関係タイプのモデルを段落内の分類に使用し，段落外での分類精度が最も良い S30 の関係タイプのモデルを段落外の分類に使用した場合の分類精度を求めたところ，全体で 48.03% の分類精度が得られた．

6 おわりに

本節では本論文を総括し，今後の検討課題を述べる．

6.1 本研究のまとめ

本研究の目的は，談話構造を機械学習を用いて解析し，文間の関係を明らかにすることであった．本研究で作成した談話構造解析のコーパスは小規模であり，どの素性が有効で，どの素性が有効でないかを決定することが難しいので，高次元の素性空間に対しても過学習しにくく極めて高い汎化能力を持つとされている SVM を使用した．

本手法により，段落内で 65.40% の解析精度が得られた．また，段落内と段落外で別々のモデルを使用することにより，全体で 53.8% の解析精度を達成した．一方，関係タイプの分類では，段落内で 44.82%，段落外で 48.37% が得られた．また，段落内と段落外で別々のモデルを使用することにより，全体で 48.03% の分類精度を達成した．

異なる人間が同じ文章を談話構造解析したとしても，その談話構造木間の一致率は 70% 程度にとどまり，決して高くはならない．談話構造解析がこのように非常に困難な解析であることを考えると，本研究で達成した精度は，談話構造解析の自動化の可能性を示すことができたと言えるだろう．

6.2 今後の課題

本研究で実装した談話構造解析システムは，解析精度の点でまだ実用レベルには達していない．そこで，解析精度を高めるための今後のアプローチとして，以下が考えられる．

- 人間が段落間の係り受け関係を判定するとき，段落で記述されている内容や意味が大きな手がかりとなる．しかし，意味そのものを抽出するというアプローチは現実的ではない．そこで，段落内で使用されている単語の頻度や，段落を構成する文間の語彙的結束性をもとに，段落の意味的な距離を素性にすることが考えられる．
- SVM では，どの素性が有効に働いていて，どの素性が有効に働いていないかという判断が付きにくい．そこで，最初に決定木で有効な素性の組み合わせを求めてから SVM を使用すればさらなる精度の向上が期待できるだろう．

参考文献

[1] Dan Cristea, Nancy Ide, Daniel Marcu, and Valentin Tablan. An empirical investigation of the relation between discourse structure and coreference. *Coling 2000*, pp. 208–214, July 2000.

[2] Barbara A. Fox. *Discourse structure and anaphora: written and conversational English.* Cambridge University Press, 1987.

[3] Mann, W.C., Thompson, and S.A. Rhetorical Structure Theory: Description and Construction of Text Structures. in Kempen, G. (ed.). *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pp. 85–96, 1987.

[4] Daniel Marcu. *The theory and practice of discourse parsing and summarization.* The MIT Press, 2000.

[5] Daniel Marcu, Lynn Carlson, and Maki Watanabe. The automatic translation of discourse structures. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2000.

[6] Tadashi Nomoto and Yuji Matsumoto. Discourse Parsing: A Decision Tree Approach. In *Proceedings of the 6th Workshop on Very Large Corpora*, pp. 216–224, Aug. 1998.

[7] 住田一男, 小野顕司, 知野哲朗, 三池誠司. 文書構造解析に基づく自動抄録生成と検索提示機能としての評価. 電子情報通信学会論文誌, Vol. 78, No. 3, pp. 511–519, 1995.

[8] 竹内和広, 松本裕治. 自動要約を視野にいれたテキスト構造解析実験. 情報処理学会研究報告, Vol. 99-NL-133, pp. 61–68, Sep. 1999.

[9] 黒橋禎夫, 長尾眞. 表層表現中の情報に基づく文章構造の自動抽出. 自然言語処理, Oct. 1994.

[10] 福本淳一. 筆者の主張に基づく日本語文章の構造化. 情報処理学会自然言語処理研究会研究報告, Vol. 78, No. 15, pp. 113–119, 1990.

[11] 難波英嗣, 奥村学. 第 2 回 NTCIR ワークショップ自動要約タスク (TSC) の結果および評価法の分析. 情報処理学会研究報告, Vol. NL-144, pp. 143–150, 2001.