並列制約最適化問題解決としてのテキスト解析

吉見毅彦 †* 吉田宏和 † 池口大輔 ‡ 田村直之 ‡

† 神戸大学自然科学研究科
‡ 神戸大学工学部
神戸市灘区六甲台町 1-1
{yoshi,sarion,ikeguchi,tamura}@timpani.seg.kobe-u.ac.jp

Abstract

This paper proposes a new approach to text analysis, in which incremental analysis is realised by viewing text analysis as concurrent constraint optimisation problem solving. Multiple agents work concurrently and cooperatively to solve the constraint optimisation problem. Links representing dependency and correferentiality relations between two words are considered as the agents. Every time a word is inserted or deleted, the agents holding the prioritised constraints relevant to the words are either generated or removed. We have applied this model to an analysis of dependency and correferentiality in Japanese texts.

1 はじめに

従来の自然言語処理システムにおける入力テキスト解析系の多くは，形態素解析系，構文解析系，照応解析系などのモジュールを直列につないだ構成をしており，各モジュールでの処理は，それ以前のモジュールでの入力全体に対する処理が完了した時点で初めて起動されるバッチ式である．このような直列・バッチ式のモジュール構成は，入力が完了していない時点でも意味的にまとまりのある部分に対する解釈を出力する必要がある同時通訳システムや，人間とのやりとりを自然にかつ円滑に行なう必要がある対話システムなどの高度な自然言語処理システムには適してない．

このようなシステムに適用可能なテキスト解析系は，漸進性と即応性を備えていていなければならない．漸進性は，テキストの変化を検知した後できるだけ速やかに，その変化に関連する部分の解釈だけを更新できる (関連しない部分の無駄な更新を避ける) 能力である．テキストの変化として，テキストの任意の位置への新たな語句の追加と任意の既存語句の削除に対応できる漸進性が広義の漸進性である．広義の漸進性を持つ解析系としては，チャート構文解析法に基づく手法がこれまでに提案されている [Wir89]．これに対して，狭義の漸進性を持つ解析系 [Mel85, Had87, AIOT93, Mil95] は，テキストの変化として，新たな語句のテキスト右端への追加のみを許しており，通常，語が一語追加されるたびに解釈を更新する．

即応性は，適切な解釈を許容できる時間内に見つけ出すことができる能力である．テキスト解釈の曖昧性解消は，可能な解釈の中から種々の選好に基づいて最適な解釈を選び出すことによって実現される [YJN+97]

*兼 シャープ (株) ソフト事業推進センター

が，一般に，音声言語処理などのように時間的制約が厳しい状況の下では，真の最適解を常に見つけ出すことは容易ではない．このような状況下では，真の最適解を見つけ出すために過度に時間を費やすことよりも，即応的に準最適解を見つけ出すことの方が重要である [Men94]．従って，本稿では，テキスト解析における真の最適解を求める手法ではなく，許容される時間内で準最適解を求める手法を開発することに重点を置く．

広義の漸進性と即応性を備えたテキスト解析系を実現するために，テキストの解釈を，二つの文節を結ぶ文法的なリンクの集合で表し，各リンクを分散制約最適化問題におけるエージェントとみなす．これによって，テキスト解析は，各エージェントが他のエージェントとの通信によってエージェント群における自己の役割を決定する過程とみなされる．なお，テキスト解析を制約最適化問題解決と捉えるという点で本手法と類似した手法として，係り受け関係を優先度付き制約マトリックスで表現し制約伝播によって曖昧性の解消を行なう手法がこれまでに提案されている [Mar90, Nag90]．

テキストに新たな文節が追加されるたびに，あるいは，既存の文節が削除されるたびに，それらを結ぶリンクすなわちエージェントが動的に生成されたり消滅したりすると考えると，このようなエージェントの動的な生成と消滅によって，解釈を漸進的に更新することができるようになる．テキストの変化が検知されたとき，動作する必要があるのは，その変化によって影響を受けるエージェントだけであり，それ以外のエージェントは動作する必要はないので，テキストの局所的な変化がテキスト全体の再解析につながることはない．

解析系が即応的に振舞えるようにするために，あるエージェントが通信できるエージェントの範囲を，存在するエージェント群の一部に限定し，エージェントが局所的に振舞えるようにする．エージェントを局所的に動作させることで，解析系全体の処理効率向上が期待される．

2 支配従属リンクと照応リンクによる解釈の表現

本稿では，テキストの解釈を二種類の二項関係の集合で表す．一つは，従属要素と支配要素の間の支配従属関係であり，もう一つは，先行要素と照応要素 (anaphoric) の間の前方照応関係である．ある文において，二つの文節 w_i と w_j の間で次の条件が満たされるとき，両者の間で支配従属関係が成り立ち， w_i が w_j に従属する，または w_j が w_i を支配するという． w_i が w_j に従属するするとき，それらを w_i から w_j へ向かうリンク (以降，支配従属リンクと呼ぶ) で結ぶ．

1. w_i が w_j の前方に現れる．
2. w_i の従属属性と w_j の支配属性が相容れる．
3. w_i と w_j を結ぶ支配従属リンクと交差する他の支配従属リンクが存在しない．

適格な日本語文では，それを構成するすべての文節 (ただし，文末の文節を除く) は，上記の条件を満たす文節のうち，唯一つの文節に従属する．

本稿では，頑健な解析を実現するために，条件 2と 3を絶対的な条件とはみなさず，優先的な条件とみなす．条件 2と 3を絶対的な条件とみなすと，これらを満たさない解釈は棄却されるが，優先的な条件とみなすと，これらを満たさない解釈は，適格な解釈と比較したときの相対的な良さ (優先度) が下がるだけで，棄却されずに保持される．このため，不適格な文に対しても何らかの解釈を出力することができる．

二つのテキスト構成要素の間で照応が成り立つかどうかの判断は，陳述圧縮パラダイム [JYN$^+$95] に基づいて行なう．陳述圧縮は，ある表現で言及した事象にテキストの後方で再び言及するとき，以前の表現を圧縮した言語形式が用いられる現象である．陳述縮約に関する規範は，完全形 (full form)[1]がゼロ形に縮約される過程を 11 段階の陳述縮約度に分類し，指す言語表現の陳述縮約度と指される言語表現の陳述縮約度の間で成り立つ条件を記述したものである．二つの文節 w_i と w_j の間で次の条件が満たされるとき， w_j が w_i を指すといい，それらを w_j から w_i へ向かうリンク (以降，照応リンクと呼ぶ) で結ぶ．

1. w_i は w_j の前方に現れる．

[1]書き手が記述しようとしている事柄についての知識を読み手が全く持っていないと書き手が判断したときに用いる言語表現．

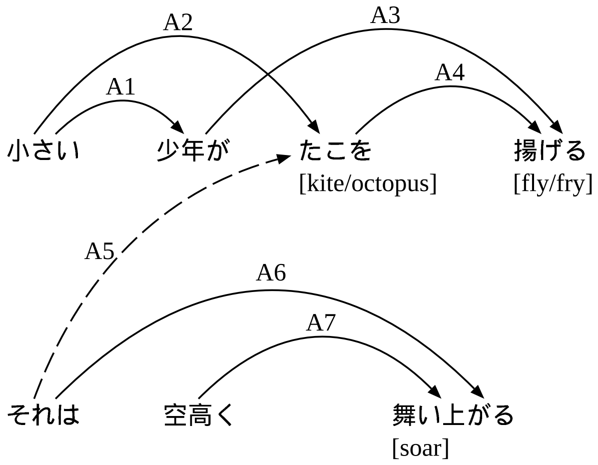

図 1: サンプルテキストの支配従属リンクと照応リンク

2. w_i の意味と w_j の意味が矛盾しない．

3. w_j は w_i を圧縮した言語表現である．すなわち，w_j の陳述圧縮度は w_i の陳述圧縮度よりも低い．

テキストがこれらの条件に従っているならば，通常，テキストの読み手は縮約形 (depredicated form) [2]から完全形を復元することができるので，そのテキストは適格であるとみなせる．

3 分散制約最適化問題

制約最適化問題は，一般に，1) N 個の変数 $v_1, v_2, \cdots, v_N$，2) 各変数 v_i が値をとる有限領域 $D_1, D_2, \cdots, D_N$，3) 変数への値の割り当てに関する制約，4) 割り当てに対する評価関数で構成される．制約最適化問題を解くことは，制約をすべて満たす変数への値割り当てのうち，評価関数が最適値を返すような割り当てを見つけることである．分散制約最適化問題は，変数，制約，評価関数が各エージェントに分散された制約最適化問題である．各エージェントは，自分が知っている制約をすべて満たし，評価関数から返される値が最適値となるように，自分が知っている各変数に値を割り当てる．

4 エージェントの生成

本稿で提案するテキスト解析モデルでは，支配従属リンクと照応リンクを分散制約最適化問題におけるエージェントとみなす．エージェントは，エージェント生成系によって，形態素解析結果から生成されるものとする．次のような簡単なテキストを考える．

> 少年がたこを揚げていた．
> それは空高く舞い上がっていた．

このテキストに対しては，図 1に示すエージェントが生成される． あるリンクに対応するエージェント A は，一般に，1) A の始点側の文節の語義 (意味素性)，2) A の終点側の文節の語義，3) A と始点を共有するエー

[2] 読み手に既知であると書き手が判断した情報を明示しない言語表現．

エージェント	優先度付き制約リスト
A_1	$[5 : A_2 = \mathrm{inactive},$
	$0 : A_2 = \mathrm{active}]$
A_2	$[5 : A_1 = \mathrm{inactive} \wedge A_3 = \mathrm{inactive},$
	$0 : A_1 = \mathrm{active} \vee A_3 = \mathrm{active}]$
A_3	$[5 : A_2 = \mathrm{inactive},$
	$-99 : A_2 = \mathrm{active}]$
A_4	$[5 : \mathrm{tako.SC} = \mathrm{kite} \wedge \mathrm{ageru.SC} = \mathrm{fly},$
	$5 : \mathrm{tako.SC} = \mathrm{octopus} \wedge \mathrm{ageru.SC} = \mathrm{fry},$
	$-5 : \mathrm{tako.SC} = \mathrm{octopus} \wedge \mathrm{ageru.SC} = \mathrm{fly},$
	$-5 : \mathrm{tako.SC} = \mathrm{kite} \wedge \mathrm{ageru.SC} = \mathrm{fry}]$
A_5	$[5 : \mathrm{tako.SC} = \mathrm{kite} \wedge \mathrm{sore.SC} = \mathrm{kite},$
	$5 : \mathrm{tako.SC} = \mathrm{octopus} \wedge \mathrm{sore.SC} = \mathrm{octopus},$
	$0 : \mathrm{tako.SC} = \mathrm{octopus} \wedge \mathrm{sore.SC} = \mathrm{kite},$
	$0 : \mathrm{tako.SC} = \mathrm{kite} \wedge \mathrm{sore.SC} = \mathrm{octopus}]$
A_6	$[5 : \mathrm{sore.SC} = \mathrm{kite} \wedge \mathrm{maiagaru.SC} = \mathrm{soar},$
	$-5 : \mathrm{sore.SC} = \mathrm{octopus} \wedge \mathrm{maiagaru.SC} = \mathrm{soar}]$
A_7	$[5 : \mathrm{true}]$

表 1: サンプルテキストにおける優先度付き制約リスト

ジェント名，4) A と終点を共有するエージェント名，5) A と交差する可能性があるエージェント名を変数とする優先度付き制約を持つ．サンプルテキストにおける各エージェントが持つ優先度付き制約を表 1に示す．例えば，A_4 の優先度付き制約において，変数 “tako.SC” は，「たこ」の意味素性を意味する．「たこ」には “kite” と “octopus” の二通りの語義 (意味素性) があり，「揚げる」には “fly” と “fry” の二通りの語義があるものとすると，「たこを」と「揚げていた」を結ぶ支配従属リンクに対応するエージェント A_4 の優先度付き制約リストは，“kite” と “fly” の間，“octopus” と “fry” の間ではそれぞれ選択制限[3]が満たされるが，“octopus” と “fly” の間，“kite” と “fry” の間ではそれぞれの選択制限が満たされないことを表している．エージェントは，選択制限を破る解釈を，選択制限を満たす解釈の評価値よりも評価値が低い解釈として保持する．これによって，比喩などに対して頑健な解析が実現できる．

「小さい」の従属属性は「少年」の支配属性とも「たこ」の支配属性とも相容れるが，2節で述べた「文節は他の唯一つの文節に従属する」という支配従属の制約から，ある一つの解釈では，エージェント A_1 と A_2 のいずれか一方しか活性状態になれない．従って，エージェント A_2 が活性状態かどうかを，エージェント A_1 の優先度付き制約として記述しておかなければならない．エージェント A_1 は，その優先度付き制約リストの第一要素が真であるとき，活性状態になれる．

エージェント A_2 は，また，エージェント A_3 とも競合するので，エージェント A_2 の優先度付き制約には，エージェント A_1 の状態だけでなく，エージェント A_3 の状態も記述される．たとえエージェント A_2 が不活性状態となっても，適格な解釈は生成できるが，エージェント A_3 が不活性状態となると生成できない．従って，エージェント A_3 が不活性状態になるのをできるだけ避けるために，エージェント A_3 の優先度付き制約リストの第二要素の評価値は，他の値に比べてかなり低く設定される．

代名詞「それ」は名詞「たこ」を指せる，すなわち，「それ」と「たこ」が 2節で述べた照応制約を満たす可能性があるので，破線で示したエージェント A_5 が生成される．

[3] 文節間の意味的な整合性に関する条件．

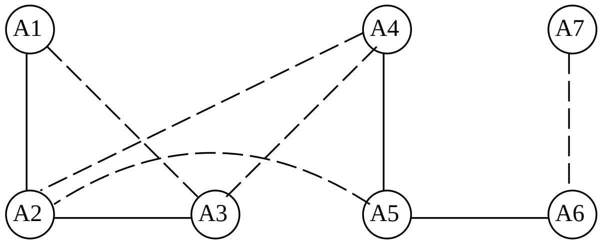

図 2: サンプルテキストにおけるエージェント通信ネットワーク

5 エージェント間の競合解消

解析過程で，変数への値の割り当てに関して，エージェント間で競合が生じることがある．このため，すべてのエージェントが最適な割り当てを獲得できるとは限らない．各エージェントは，できるだけ良い割り当てが獲得できるように振舞う．すなわち，後述するプロトコルに従って他のエージェントと通信を行ない，最終的に合意に達する．本稿のモデルでは，あるエージェント A が通信できるエージェントを，次の二種類に制限している．

1. ある文節をエージェント A と共有するエージェント
2. エージェント A と交差する可能性があるエージェント

サンプルテキストにおけるエージェント間の通信ネットワークを図 2に示す． エージェント間の制約は，実線で結ばれたエージェント間にのみ存在する．破線で結ばれたエージェントは互いに通信することができるが，それらの間には制約がないので，通信する必要はない．例えば，エージェント A_1， A_2， A_3 は，通信線を通じて互いに通信できるが， A_1 と A_3 が通信する必要はない．これに対して，全体で最適な解釈を発見するためには，エージェント A_4 と A_6 の間で通信を行なう必要があるが，本稿のモデルでは， A_4 と A_6 は文節を共有しないし交差もしないので，通信することはできない．このように，あるエージェントが通信できる範囲を，存在するエージェント群の一部に限定すると，全体での最適解釈が発見できなくなることがある．しかし，このような制限を設けることによって，エージェントが局所的に処理を進めることが可能になり，準最適解釈を適切な時間内で発見することが可能になる．

実線で結ばれていないエージェントは，互いに独立に動作することができる．図 2では， A_1， A_2， A_3 から成るグループ， A_4 と A_5 から成るグループ， A_7 から成るグループの三つのグループが並列に動作できる．

エージェント A は，次の通信プロトコルに従って動作する．

1. 現在の値割り当てが最適ならば，何もしない．
2. もしそうでなければ，現在の値割り当てをより良いものに変更したい旨を，通信線で直接結ばれている各エージェントに伝える．
3. エージェント A の意図を知ったエージェント B は， A が行なう値割り当ての変更によって B の評価値がどれだけ増減するかを A に返答する．ここで，エージェント B は，エージェント A が知らない変数への値割り当ての変更は行なわないという条件の下で，評価値の増減の見積りを行なう．
4. エージェント A は，現在の値割り当てを変更した場合の評価値の増加分に，通信線で直接結ばれている各エージェントの評価値の増減分を加え，その和が正ならば値割り当てを実際に変更し，正でなければ変更を断念する．

このプロトコルでは，評価値の見積りを行なう際に，エージェント A にとって未知の変数への値割り当ての変更は行なわないという条件を設けることによって，エージェントの通信の範囲を抑えている．もし，このような条件を設けなければ，値割り当ての一貫性を保つために，エージェント B は第三のエージェントと通信しなければならなくなる．

サンプルテキストの解析過程のある時点で，エージェント A_4 が表 1の優先度付き制約リストの第一要素で示される状態にあり，エージェント A_5 と A_6 がそれぞれの優先度付き制約リストの第二要素で示される状態にあるとする．通信は，最適な値割り当てを得ていないエージェント A_5 と A_6 によって開始される．今，エージェント A_5 から動作しはじめるとものする．エージェント A_5 は，変数 “sore.SC” の値を “octopus” から “kite” に変更すれば，最適な値割り当てを獲得することができる．エージェント A_4 はこの変更による影響を受けないが，エージェント A_6 は，この変更によって，最適な値割り当てを得ることができる．従って，この場合，三つのエージェント間に値割り当ての競合は生じず，全体での最適値 15 点が得られる．

次に，エージェントの通信範囲を制限しているために，処理が局所最適に陥る例を示す．エージェント A_4，A_5，A_6 が，それぞれの優先度付き制約リストの第二要素で示される状態にあると仮定する．エージェント A_6 は，変数 “sore.SC” の値を “octopus” から “kite” に変更すれば，最適な状態に移り，評価値が 10 点上がる．この変更が実行されれば，エージェント A_5 の評価値は 5 点下がるが，全体としては，エージェント A_6 の増分 10 点を加えると，5 点あがるので，この変更は実行される．その後，エージェント A_5 は，最適の状態に戻るための通信を開始する．最適の状態に戻るためには，エージェント A_5 は，変数 “tako.SC” の値を “kite” に変更しなければならない．この変更が実行されれば，エージェント A_5 の評価値が 5 点上がるが，エージェント A_4 は，第四要素で示される状態になり評価値が 10 点下がるので，全体としては 5 点下がる．このため，エージェント A_5 は，変数 “tako.SC” の値を “kite” に変更することはできない．従って，どのエージェントもこの状況を改善することができず，処理は局所最適に陥る．変数 “tako.SC” の値を “kite” に変更すると，エージェント A_4 が第四要素で示される状態にまで落ちる原因は，プロトコルの第 3ステップで，A_4 の評価値の増減の見積りが，エージェント A_5 にとって未知の変数 “ageru.SC” への値割り当ての不変であるという条件の下で行なわれていることにある．

6 実装

現在，提案したモデルの実装を行なっている．制約最適化問題を解くエージェントの実装は JAVA を用いて行なった．テキストの形態素解析結果から制約最適化問題を生成するモジュールの実装が残されている．形態素解析には形態素解析システム「茶筌」[MKY+97] と EDR 辞書 [EDR96] を利用している．

7 今後の課題

実験を通じて，どのような言語現象を含むテキストを処理した場合に，真の最適解からどの程度離れた局所最適に陥るかを調べ，必要ならば，局所最適からの脱出策を検討する．

8 おわりに

漸進的で即応的なテキスト解析システムを実現するための基本的な考え方として，二つの文節を結ぶ文法的リンクをエージェントとみなし，テキスト解析を，各エージェントが他のエージェントとの通信によってエージェント群における自己の役割を決定する過程と捉えた．漸進性は，エージェントの動的な生成と消滅によって，また即応性は，エージェントの通信範囲の制限によって実現される．

参考文献

[AIOT93] 秋葉友良, 伊藤克亘, 奥村学, 田中穂積. 増進的曖昧性解消モデルに基づいた日本語解析. コンピュータソフトウェア, Vol. 10, No. 1, pp. 29–40, 1993.

[EDR96] 日本電子化辞書研究所. EDR 電子化辞書 1.5 版, 1996.

[Had87] N. J. Haddock. Incremental Interpretation and Combinatory Categorial Grammar. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI87)*, pp. 661–663, 1987.

[JYN+95] J. Jelinek, T. Yoshimi, O. Nishida, N. Tamura, and H. Murakami. Text-Wide MT Grammar. In *Proceedings of the 3rd Natural Language Processing Pacific Rim Symposium (NLPRS95)*, pp. 449–454, 1995.

[Mar90] H. Maruyama. Structural Disambiguation with Constraint Propagation. In *Proceeding of the 28th Annual Meeting of the Association for Computational Linguistics (ACL90)*, pp. 31–38, 1990.

[Mel85] C. S. Mellish. *Computer Interpretation of Natural Language Description*. Ellis Horwood, 1985.

[Men94] W. Menzel. Parsing of Spoken Language under Time Constraints. Published as http://xxx.lanl.gov/abs/cmp-lg/9409008, 1994.

[Mil95] D. Milward. Incremental Interpretation of Categorial Grammar. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL95)*, pp. 119–126, 1995. Also Published as http://xxx.lanl.gov/abs/cmp-lg/9503015.

[MKY+97] 松本裕治, 北内啓, 山下達雄, 今一修, 今村友明. 日本語形態素解析システム「茶筌」使用説明書. 奈良先端科学技術大学院大学テクニカルレポート NAIST-IS-TR97007, 1997.

[Nag90] K. Nagao. Dependency Analyzer: A Knowledge-Based Approach to Structural Disambiguation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING90)*, pp. 282–287, 1990.

[Wir89] M. Wiren. Interactive Incremental Chart Parsing. In *Proceeding of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL89)*, pp. 241–248, 1989.

[YJN+97] 吉見毅彦, J. Jelinek, 西田収, 田村直之, 村上温夫. Text-Wide Grammar に基づくテキスト解析. 自然言語処理, Vol. 4, No. 1, pp. 3–21, 1997.

[YKH+97] T. Yoshimi, T. Kisaku, R. Hokabira, H. Yoshida, and N. Tamura. Text Analysis as Concurrent Constraint Optimization Problem Solving. In *Proceedings of the 4th Natural Language Processing Pacific Rim Symposium (NLPRS)*, pp. 395–400, 1997.