日本語意味解析システムSAGEの高速化・高精度化と精度評価

田淵 和幸*　　原田 実**

青山学院大学　理工学研究科　経営工学専攻*　　青山学院大学　理工学部　情報テクノロジー学科**

1. はじめに

原田研究室ではこれまで，EDR電子化辞書[1]に記載された情報を元に，日本語文を意味解析し格フレーム群に自動変換するシステムSAGE98[2] (Semantic frame Automatic GEnerator)とSAGE99[3]を開発してきた．SAGE99は一応正しく動作するが，解析時間や解析の正解率において，実利用するには十分なレベルに達していない．本研究の目的は，解析速度と解析精度の両面で実用可能レベルの意味解析システムSAGE2000の開発を行うことであり，具体的には以下の3つを行う．

1. SAGEの高速化: Jiri-Haradaアルゴリズムにより解析速度を向上する．

2. 解析精度の自動評価: 解析済みコーパスを用いてSAGEの解析精度を自動的に評価する．

3. SAGEの高精度化: 合成語や複文などSAGE99では解析できなかった問題を解決し、解析精度を向上する．

2. 探索的アルゴリズムによるSAGE99の問題

SAGEで係り受け関係にある文節間の意味解析を行う前段階として形態素解析と係り受け解析を行う．本研究では，形態素解析システムには『茶筌』を，係り受け解析システムには『茶掛』を利用した．これらのシステムは奈良先端科学技術大学院大学の松本研究室で開発されたツールである[4]．またpreSAGEでは，茶掛の出力ファイルを，prologで扱いやすい形のリスト形式(tree述語形式)に変換する．このtree述語を受けてSAGEは意味解析を行う．SAGE99は(SAGE99とSAGE2000で処理の流れに大きな差はない)，図1に示すようにEDRを元に意味解析を行うシステムであり，SAGE本体，CIP，Corpus，Arrangeという4つのコンポーネントからなる．処理の流れを「人が降りる」という例に添って説明する．まずSAGE本体がtree述語形式ファイルを読み込み，そこから係り受け関係にある2文節を取り出す．例では「人が」と「降りる」．このとき，「降りる」にあたる語を係り元語，「人が」にあたる語を係り先語と呼ぶ．文法的な修飾関係における係り受けとは逆になる．これら2文節をCIPとCorpusに引き渡す．CIPでは，図2にも示すように渡された2文節の中心語(「人」と「降りる」)の語意とそれらの間にどのような格関係が考えられるのかをEDR辞書で検索し，それぞれの語意と格の組み合わせ(これを語意-格組と呼ぶ)の尤もらしさを語意-格確率として求める．Corpusでは，助詞と係り元中心語(「～が」と「降りる」)から，その助詞と単語が共に出現した場合の2文節間における各格の出現確率を格確率として求める．さらに，係り元中心語と係り先中心語(「降りる」と「人」)から，この2つの語が共に出現した場合の，2語の語意の組の出現確率を語意確率として求める．Arrangeでは個々の語意-格組毎に語意-格確率と格確率と語意確率の和を語意-格総合評価値として算出し，図1に示すようにCase述語としてSAGE本体に引き渡す．

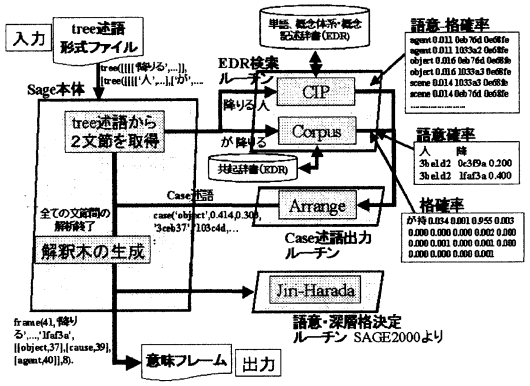

図1 SAGE2000の基本処理の流れ

これらの作業を係り受け関係にある全ての2文節に対して行い，図2に示すように文の係り受け木の各枝が表す語意-格総合評価値を割り当てる．これを解釈木と呼び，これらの解釈木ごとに全ての枝に対する語意-格総合評価値の和を求める．これを確信度と呼ぶ．この確信度が最も大きくなるような解釈木を統計的に尤もらしい木として採択

Speed-up and accuracy improvement of Japanese semantic analysis system SAGE

Kazuyuki Tabuchi*　Minoru Harada**

*Graduate School of Industrial and System Engineering, Department of Science and Engineering, Aoyama Gakuin University.

**Department of Integrated Information Technology, Faculty of Science and Engineering, Aoyama Gakuin University.

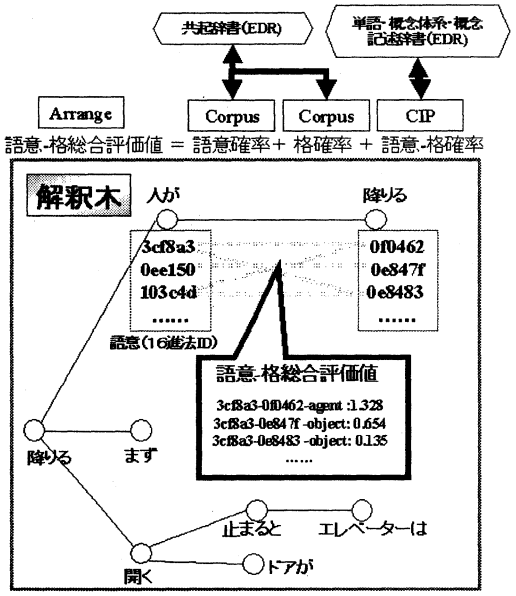

図 2 解釈木と EDR 辞書からの確率

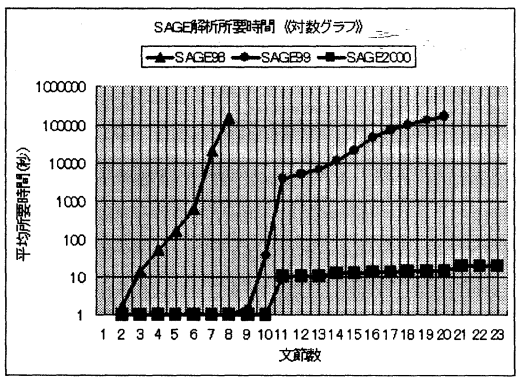

図 3 解釈木構築における平均所要時間(対数グラフ)

する. 係り受け木における 1 つの枝には多くの語意-格組候補があり, 解釈木の数も膨大なものとなる. 図2に示す「エレベータは止まるとドアが開き, まず降りる人が降りる」という例に対しては, 約630 億通りもの解釈木が存在する. この解析には図3の SAGE98 に示すように指数オーダの時間がかかる. この問題点に対し水野 [6] は精度を落とさないことを基本方針に, 「最大評価優先法」と「分枝限定法」という2つの手法による高速化を提案・実装した. その結果として図3の SAGE99 に示すように約 10 文節までの解析を数秒で行うことに成功した. しかし 10 文節を越えた文に対しては実用的な時間内では解析できず, 線形オーダのアルゴリズムが求められている.

3. SAGE の高速化

3.1. Jiri アルゴリズム

Jiri らは, 英文の構文木中の各節の語意を決定する高速アルゴリズムを提案している [7]. 彼らは, 係り受け関係における各節の中心語(head)とその修飾語(modifier)の間の構文的な意味関係を分類し, それ毎に両語の語意の確率をコーパスから統計的に求め, これを関係行列(relational matrix)として算出している. 各語の語意の決定は, まず語毎にその様々な語意の確率をベクトルとしたもの(sense score vector)の初期値を統計的に求め, 次に構文木の葉から始めて, それらが修飾している head との関係行列から head の sense score vector を更新する. 同時に各 modifier の sense score vector を head の語意毎に並べて意味得点行列(sense score matrix)とする. この過程を head が構文木の根になるまで行い, 根の sense score vector が確定すると, その中の最大確率を持つ語意を根の語意とする. ここまでを Bottom-up 集約という. これが決定すると今度は Top-down 決定を行う. ここでは根から始めて, modifier の語意を順に決定していく. この際, 既に決まっている head の語意を固定して, その中で sense score matrix 要素が最大値になる modifier の語意を決定する。このプロセスは探索を含まず決定的に行われるので高速に実行できる.

3.2. Jiri-Harada アルゴリズム

我々の目的は各語の語意を決定するのみならず, 語間の深層格も決定するということにある. 従って, 語意の決定も単純に語毎の語意確率というよりは, 他の語との深層格の関係においての語意の確率を重要視している. Jiri らの方法も確かに語意を表す sense vector の更新を他の語の関係を表す relational matrix を用いて行っているが, 我々は係り受け関係にある2語の語意とその間の深層格の3つ組毎にその出現確率を用いる方が, これらの最適値を決定するにはより適切であると考えている. そこで我々は以下のように Jiri らのアルゴリズムを拡張した. ここでは主に語意-格組の出現確率の算出方法を変更し, Bottom-up 集約と Top-down 決定という全体的なアルゴリズムは同様とした. 図4に添って以下にそのアルゴリズムを示す.

【Jiri-Harada アルゴリズム】

Step1(Bottom-Up 集約): まず各ノード m_i に対して，sense score vector $M_i(u)$の初期値を，m_iの語意uとそれが修飾している語 h の語意 j の組み合わせに対する語意確率のうち，語意uを含むものの和とする．さらに，sense score matrix $Q_i(k,j,u)$に，m_iの語意uとhの語意jとその間の格kに対する語意-格総合評価値を割り当てる．

次に最下層より1つ上以上の各ノード(例，h)において，そのsense score vector $M_h(j)$を，その直下のノード群 $\{m_i\}$ のsense score matrixを用いて式①のように更新する．

$$M_h(j) = \frac{L_j}{L} M_h(j) \quad \cdots\cdots ①$$

$$L_j = \sum_k \max_u (Q_i(k,j,u)) \quad \cdots\cdots ②$$

$$L = \sum_j L_j \quad \cdots\cdots ③$$

ここで②の$\max_u(Q_i(k,j,u))$は，hの語意jを一定にしてm_iの語意uを変化させた時の最大値，Σ_kは上記の最大値を格kを変化させた時の和である．また③のΣ_jはhの語意jを変化させたときの和である．

Step2(Top-Down 決定): まず，最上位のノードrの語意をそのsense score vector M_rの要素の最大値を与えるインデックス1とする．次に，この最上位のノードhから始めて，その修飾語m_iの語意とhとの間の格を，m_iのsense score matrix $Q_i(k,j,u)$を用いて，hの語意jを固定してm_iの語意uとhとの間の格kを変化させたとき，$Q_i(k,j,u)$の最大値を与える語意uと格kとする．

Jiri アルゴリズムとの差は，本アルゴリズムでは relation matrix を必要としないこと，またsense score matrixを2次元ではなく格kの次元を加えた3次元行列として，その値を語意-格総合評価値で直接的に与えていることである．これは，SAGE では図4に示すように2語の語意と語間の格の全ての組み合わせに対する出現確率が，先に述べたように，EDR から求まるからである．

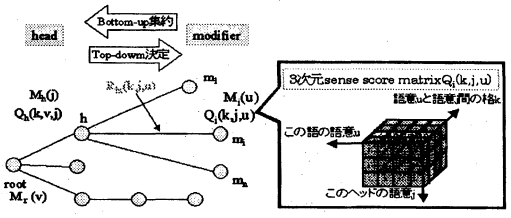

図 4 Jiri-Harada アルゴリズム

4. SAGE2000 の処理の流れ

SAGE2000 の処理の流れは図 1に示すようにSAGE99 の解釈木構築を Jiri-Harada というコンポーネントに変更したものになっている．文内の全ての係り受けにおける辞書検索が終了すると SAGE 本体は Arrange からの Case 述語を Jiri-Harada に渡す．Jiri-Harada はこの Case 述語から上述のアルゴリズムにより各語の語意と語間の格を決定する．この結果を基に SAGE 本体が語毎に意味フレームとして出力する．

5. 解析精度の自動評価

SAGE の解析精度を自動的に評価するシステムを構築し，実際に 100 文に対して評価を行った．我々は評価対象文として EDR 電子化辞書のコーパス辞書に記述されている例文をランダムに選ぶことにした．このコーパス辞書は新聞や雑誌などから抽出した文と，それを専門家が意味解析した結果データを保持している．この意味解析済みデータには，①構成要素情報，②形態素情報，③構文情報，④意味情報がある．このうち図 5に示すように，④意味情報は形式は異なるが，SAGE が出力する意味フレームと同等の情報を保持している．

本評価システムは，図6のように2つのコンポーネントから構成されている．形式変換 corpus Yxx Japanese は解析済みコーパスデータを SAGE の出力データの表現形式に変換する． EvalSAGE は両者の照合を行う．具体的には，フレーム毎にその語意と，他のフレームと関係があるならば，その相手先のフレームが同じかどうか，その間の格が同じかどうかを調べる．結果は評価値をただ出力するだけではなく，図 7のように Excel 表形式でフレーム毎にこの 3 つの検査項目を出力する．この結果，後の誤りの解析において誤りの分類がしやすくなる．例では「石」と「なり」の語意が異なっていることや，コーパスでは「救わ」と「姫」の間に object 格がないことがわかる．なお前者ではコーパス辞書が正しいが，後者ではむしろ SAGE の解析結果のほうが正しい．

6. SAGE の高精度化

前章のシステムで，SAGE99 の誤りを分析した結果，①合成語の中心語以外の語意が決定されていないこと，②用言間の関係における複文の解析が誤っていることがわかった．前者では、共起辞書を用い，中心語とそれ以外の構成

語の2語をキーワードとして出現確率の最も高いものを採用することにした. 後者では、2文節の中心語が用言である場合には, 接続詞/接続助詞と係り受け関係にある他の用言の語意や助動詞の活用形などを考慮して決定するようにした.

6.1. 評価

コーパス辞書の例文100文において, SAGE99とSAGE2000が生成した意味フレームを, 精度自動評価システムを用いて評価した結果を図8に示す. 本論文の高精度化により, 語意正解率が43.96%から81.09%へ, 格正解率が54.39から60.70%へ, 格の宛先正解率が71.93%から73.33%へ向上をしている. なお, ここではコーパス辞書が正しいとした場合の正解率であるが, 先にも指摘したようにコーパス辞書が誤っていることもあり, 実際の正解率はもう少し高くなると思われる. これによってSAGE2000は実利用を開始できる精度に至ったといえる.

7. おわりに

本研究により, SAGEは解析速度と解析精度ともに実利用可能なレベルに近づいたといえる. 今後は, 速度面では辞書検索の速度の向上, 精度面では更なる誤り分析による改良を行う必要がある.

Corpusの正解データ

```
1[ [main 16;結婚;0e51a0]
   [agent 14;ペルセウス;"ギリシア神話の登場人物"]
   [agent 9;姫;104f79]
   [sequence 11;救;3ceb79]
   [and [ [main 6;な;3cea3]
          [object 1;鎖;0edf59]
          [manner 3;たちまち;109a8b]
          [goal 4;石;0ee74f]]]]
```

Sageの出力データ

```
frame(1,'ペルセウス','未知語','JSA','ペルセウス','名詞-サ変接続','none','ffffff',[],1).
frame(2,'姫','ヒメ','JN1','姫','名詞-一般','none','104f79',[],1).
frame(3,'救わ','スクフ','JVE','救う','動詞-自立','未然形','3ceb79',[[object,2]],1).
frame(4,'れ','レ','none','れる','動詞-接尾','連用形','000000',[],1).
frame(5,'救われ','none','JPR','救われ','none','none','000000',[[consist,3],[consist,4]],1).
                    ・・・・・・・・・・
frame(11,'する','スル','JVE','する','動詞-自立','基本形','000000',[],1).
frame(12,'結婚する','none','JPR','結婚する','none','none','000000',[[consist,10],[consist,11]],1).
```

図5 解析済みコーパスデータとSAGEの出力データの表現形式

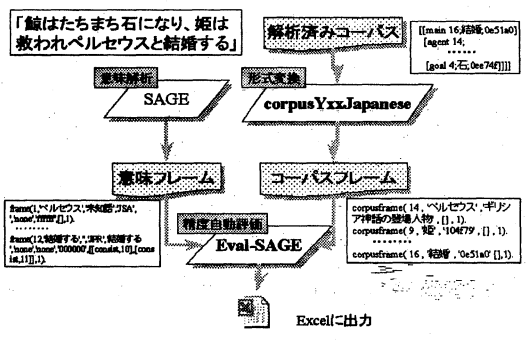

図6 精度自動評価システムの流れ

8. 謝辞

本研究を進めるにあたり, 日本語形態素解析システム『茶筌』, 係り受け解析システム『茶掛』を提供してくださった奈良先端科学技術大学院大学の松本裕治教授に深く感謝いたします.

Corpus frame番号	Sageframe番号	語	Corpus語意	Sage語意	語意の正誤(%)	Corpus格	Corpus格あて先	Sage格	Sage格あて先	格の正誤(%)	あて先の正誤(%)	文番号
					71.429					75	83.33	
14	1	ペルセウス	ギリシア神話の登場人物	ffffff	*	[	]	[	]	1	1	1
9	2	姫	104f79	104f79	1	[	]	[	]	1	1	1
11	3	救わ	3ceb79	3ceb79	1	[	]	[object	姫]	0	0	1
1	9	鎖	0edf59	0edf59	1	[	]	[	]	1	1	1
3	7	たちまち	109a8b	109a8b	1	[	]	[	]	1	1	1
4	6	石	0ee74f	0e434c	0	[	]	[	]	1	1	1
6	8	な	3cea3	101d86	0	[manner	たちまち]	[manner	たちまち]	1	1	1
6	8	な	3cea3	101d86	*	[goal	石]	[goal	石]	1	1	1
6	8	な	3cea3	101d86	*	[object	鎖]	[	]	0	0	1
16	10	結婚	0e51a0	0e51a0	1	[and	な]	[sequence	な]	0	1	1
16	10	結婚	0e51a0	0e51a0	*	[sequence	救わ]	[sequence	救わ]	1	1	1
16	10	結婚	0e51a0	0e51a0	*	[agent	ペルセウス]	[agent	ペルセウス]	1	1	1

図7 解析済みコーパスデータとSAGEの出力データのframe毎の比較表

	SAGE99 正解率(%)	SAGE2000 正解率(%)
語意	43.96	81.09
格	54.39	60.70
格の宛先	71.93	73.33

図8 SAGE99とSAGE2000の精度評価

9. 参考文献

[1] (株)日本語電子化辞書研究所: EDR電子化辞書仕様説明書, (株)日本語電子化辞書研究所 (1995).

[2] 尾見孝一郎, 原田実, 岩田隆志, 水野高宏: 日本語文章からの意味フレーム自動生成システムSAGE(Semantic frame Automatic GEnerator)の開発研究, 人工知能学会第13回全国大会論文集, pp. 213-216 (1999).

[3] 水野高宏, 原田実: 日本語意味解析システムSAGEの高速化と精度向上, 人工知能学会第14回全国大会論文集, pp. 149-152 (2000).

[4] 松本裕治, 北内啓, 山下達雄, 平野善隆, 今一修, 今村友明: 日本語形態素解析システム『茶筌』version 2.0 使用説明書, 奈良先端科学技術大学院大学松本研究室 (1999).

[5] 原田実, 水野 高宏: "EDRを用いた日本語意味解析システムSAGE", 人工知能学会論文誌, Vol.16, No.1, pp.85-93 (2001.1).

[6] Jiri Stetina, Makoto Nagao: General Word Sense Disambiguation Method Based on a Full Sentential Context, Journal of Natural Language Processing, Vol.5, No.2, pp.47–74 (1998).