1ZA-1

経験的性質を考慮した高速な係り受け解析

生田 哲也 † 鈴木 寿 ††
† 中央大学大学院理工学研究科 †† 中央大学理工学部

1 序論

本稿では，動的計画法に基づく統計的係り受け解析の高速化を目的として，経験的性質を考慮した効率的なツリー状探索法を提案する．

従来から，文中の任意の 2 文節間の係りやすさを統計的手法により数値化した行列を作成し，動的計画法により文全体を最適にする係り受け関係を求めるモデルに基づく方法が提案されている [1]．このモデルを用いる場合，動的計画法による解析時に発生する計算複雑性を克服する必要があるが，これらの方法は日本語の係り受け関係の制約をうまく利用することにより克服している．

本稿では，計算複雑性を克服するための新たな方法を提案する．提案手法は，係り受け関係の制約を利用して構築したツリーを，経験的性質を利用して効率よく探索するものである．

なお，提案手法に基づく係り受け解析器により，50 文節程度までの現実的な文節数を持つ入力文に対して高速に解を得られる．

2 係り受け尤度の学習法

従来法に習い，文節列 $\{b_i\}_{i=1}^{n}$ を B，係り受けパターン列 $\{d_i\}_{i=1}^{n-1}$ を D と定義する．ただし，d_i は，文節 b_i の係り先文節番号を表す．D は以下の制約を満たす．

- 文末以外の全ての文節は，その文節の後方側に必ず 1 つの係り先を持つ（後方参照性）．
- 係り受けは交差しない（非交差性）．

提案手法では，文節 b_i の文節 b_j への係りやすさを係り受け尤度 L_{ij} により数値化する．L_{ij} を大規模コーパスから学習する．ある係り受けにおける，係り元文節の品詞を S_{dep}，係り先文節の品詞を S_{dest}，文節間距離を d とする．係り受け解析済み日本語文の集合を大規模コーパスとして用いた場合，大規模コーパスは 3 項組 $(S_{\mathrm{dep}}, S_{\mathrm{dest}}, d)$ の集合と考えられる．大規模コーパスの中で，係り元文節の品詞が S_i である 3 項組の個数を $C(S_i)$ とする．また，係り元文節の品詞が S_i かつ係り先文節の品詞が S_j である 3 項組の個数を $C(S_i \cap S_j)$ とする．さらに，係り元文節の品詞が S_i かつ係り先文節の品詞が S_j かつ文節間距離が d である 3 項組の個数を $C(S_i \cap S_j \cap d)$ とする．L_{ij} を式 (1) のように定義する．

$$L_{ij} = \frac{C(S_i \cap S_j)}{C(S_i)} \cdot \frac{C(S_i \cap S_j \cap d)}{C(S_i \cap S_j)} \tag{1}$$

なお，L_{ij} は，b_i の品詞，b_j の品詞，d をパラメータとする 3 次元ハッシュとしてあらかじめ用意できる．

3 提案手法

後方参照性，非交差性を満たす D を係り受け列挙木を用いて系統的に列挙し，その中で式 (2) に示す係り受け尤度の積 $M_1(D)$ が大きいものを解とする．その際に，経験的性質を考慮した枝刈りをおこなうことにより，計算複雑性を効率的に克服する．

$$M_k(D) = \prod_{i=k}^{n-1} L_{ij} \tag{2}$$

アルゴリズムを以下に示す．

1. $j = n, P = \phi, F_i = 0$ $(i = 0, \cdots, n-1)$ とする．
2. 2 つの制約を満たす係り受け $(b_{j-1}, b_{d_{j-1}})$ を全て列挙する．それぞれを P と結合し，空のリスト L に追加する．係り受け尤度が大きい要素 x から順に $P := x$ として処理 3 を実行する．
3. $M_{j-1}(P) > F_{n-j+1}$ ならば $F_{n-j+1} := M_{j-1}(P)$ とする．
4. $t := j+\epsilon$ とする．ただし ϵ は定数である．$t > n-1$ ならば $t := n-1$ とする．
   (a) $j = 2$ かつ $M_1(P) > F_{n-1}$ ならば P を解リストに追加して終了する．
   (b) $j \neq 2$ ならば以下を実行する．
      i. $M_{j-1} \leq F_t$ ならば終了する．
      ii. $M_{j-1} > F_t$ ならば，L から係り受け尤度が最大の要素 x を取り出し，$P := x, j := j-1$ として処理 2 を実行する．

以下でアルゴリズムの説明をおこなう．

第 j 文節から第 $n-1$ 文節までの各文節を係り元文節とする係り受けパターン列 $\{d_i\}_{j}^{n-1}$ を親ノードとして考える．また，親ノードに制約を満たす全ての係り受け $(b_{j-1}, b_{d_{j-1}})$ を結合した係り受けパターン列 $\{d_i\}_{j-1}^{n-1}$ を子ノードとして考える．親ノードに対して子ノードを列挙する操作を再帰的におこなうことによ

Fast Dependency Analysis Considering Empirical Properties

† Tetsuya IKUTA(ikuta@suzuki-lab.ise.chuo-u.ac.jp)
†† Hisashi SUZUKI(suzuki@ise.chuo-u.ac.jp)
Graduate School of Science and Engineering, Chuo University (†)
Faculty of Science and Engineering Chuo University (††)

り，D を系統的に列挙する．処理 2 において，子ノードの列挙をおこなう．

$0 \leq L_{ij} \leq 1$ であるので，子ノードの係り受け尤度の積は親ノードのそれ以下であることが保証される．あるノード N の係り受け尤度の積が葉ノード，すなわち解の 1 つのそれよりも小さいことがわかった場合，N から派生するノードはより良い解になりえないので，探索を打ち切ることができる．

この枝刈り法には，確実に解になりえない枝のみを刈ることができるという利点がある一方で，葉ノードの係り受け尤度の積が小さくなりすぎてしまい，解でないノードに対しても，それが解でないと判断するまで探索を続行し続け，枝刈りを実行するレベルを必要以上に深くしてしまうという短点がある．枝刈りを実行できるレベルは，多くの枝において文節数に比例して深くなる．

解になる見込みのないノードに対して，できるだけ浅いレベルで解にならないという判断を下し，探索を打ち切ることにより，探索ノード数を効率良く削減できる．そこで，経験的な性質を用いて解にならないという判断を早く下す工夫をおこなう．レベル i のノードの係り受け尤度の積 $M_{n-i}(P)$ の最大値を F_i として保存する．F_i の更新は処理 3 でおこなう．経験的に，レベル i のノードの係り受け尤度の積が，レベル i よりも若干深いレベル $i+\epsilon$ の係り受け尤度の積の最大値 $F_{i+\epsilon}$ よりも小さければ，そのノードから派生するノードは解にならないと判断できる．この性質を利用して，$F_{i+\epsilon}$ を用いてレベル i のノードから派生する枝を刈る．この枝刈りは処理 4 でおこなう．

ϵ を大きくすると，レベル i のより多くのノードにおいて，その係り受け尤度と $F_{i+\epsilon}$ の差が大きくなるため，枝刈りを実行できる枝が少なくなる．その一方で，レベル $i+\epsilon$ とレベル $n-1$ の間で係り受け尤度の積が逆転し，レベル i では解にならないと判断したものの実際には解になるという可能性をより少なくできると考えられるが，経験的に，この逆転現象により得られる解は，上位の解にならないことがわかっており，ϵ を大きくする必要はない．

4 計算機実験による性能評価

経験的性質を考慮した枝刈りの有効性を実験的に示す．実験は次のようにおこなう．2004 年 12 月 20 日の毎日新聞社会面から無作為に 200 文を抜き出し，それぞれに形態素解析，文節分割を施す．その結果に提案手法を適用する．また，比較のため，枝刈りをおこなわない全数探索法を適用する．実験条件として $\epsilon = 1$ とする．また，形態素解析器に JUMAN Version 3.0 を用いる．係り受け尤度は京大コーパスから学習する．

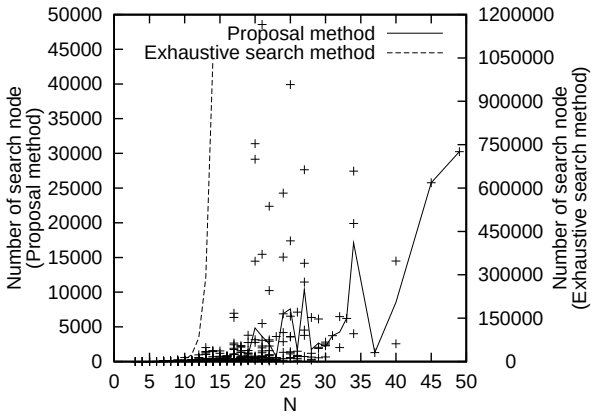

図 1: 文節数と探索ノード数の関係

新聞記事は多くの文が 10 文節から 20 文節程度であり，30 文節を超える長い文は稀である．本実験もその影響を受けている．

提案手法を適用した場合の，各入力文の文節数 N に対する探索ノード数を図 1 にプロットし，その平均値を実線で示す．また，全数探索法を適用した場合の，入力文の文節数 N に対する探索ノード数を図 1 に破線で示す．全数探索法は，入力文の内容によらず，入力文の文節数によってのみ探索ノード数が決まるので，各文節数について 1 つだけ実行する．

全数探索法においては 14 文節以上の入力で探索ノード数が発散する一方，提案手法においては 48 文節の入力まで，探索ノード数が良好な範囲に収まることが確認できる．なお，最も時間がかかった入力文に対しても，Intel Celeron 500[MHz] プロセッサを搭載するコンピュータを用いて 310[ms] で処理を完了できる．

入力文 $S_i = \{b_j\}_{j=1}^{n}$ の正しい係り受けパターン列を $\{P(S_i)_j\}_{j=1}^{n-1}$ とする．また，文 S_i の解析結果の係り受けパターン列を $\{Q(S_i)_j\}_{j=1}^{n-1}$ とする．S_i の文節数を $N(S_i)$ とする．文節正解率 R を式 (3) に定義する．ただし，$|X|$ は X が真ならば 1，偽ならば 0 であることを表す．

$$R = \frac{\sum_{i=1}^{200}}{200} \cdot \frac{\sum_{k=1}^{N(S_i)-1} |P(S_i)_k = Q(S_i)_k|}{N(S_i) - 1} \qquad (3)$$

入力文 $S_1, \cdots, S_{200}$ に対して R は約 71% である．

5 結論

係り受け解析の高速化を目的として，効率的なツリー状探索法を提案した．また，係りやすさの学習法を工夫した．現実的な入力に対して，高速に解を得られることを実験的に確認した．

参考文献

[1] 関根論，内元清貴，井左原均“ 文末から解析する統計的係り受け解析アルゴリズム，”**自然言語処理**，Vol.6，No.3，pp.59-73，1999.