定型文末表現の自動抽出

5B－3

加藤　直人　　森元　逞
（株）ＡＴＲ音声翻訳通信研究所

1　はじめに

音声対話の談話構造を解析する際には，ドメインへの依存性が少ないことから，発話意図に基づいて処理されることが多い．日本語の場合，発話意図は文末に現れやすく，文末表現から発話意図が同定できる．このような文末表現は定型的であるので，自動的に抽出することが可能である．

定型表現やＮグラムの自動抽出手法は様々提案されているが[新納 94][北 93][長尾 93][池原 95][浦谷 95][尾本 95][Lalit 89][田本 92]，本稿では文末表現のグループ分けという観点で定型文末表現を自動抽出する手法について述べる．実際には，文末をルートノードとする木構造で表わし，木全体のエントロピーを最大化する文字列を定型表現として抽出する．

2　定型文末表現

例１に示した文末の文字列から定型表現を抽出することを例に取り，定型表現について考えよう．

【例１】	文字列	出現頻度
［1］	ます	８０回
［2］	います	１２０回
［3］	します	９０回
［4］	てます	２０回
［5］	きます	３０回

（ここで，［1］は「べます」，「出ます」等，３文字では低頻度（ある頻度以下）である場合をまとめたもの．）

定型表現とは各文字列に共通する文字列であり，なるべく長いほうがよい．例えば，例１の中から１つ定型表現を抽出することを考える．「ます」はこの５種類すべての文末の文字列に共通し，「す」のみより長いので，「ます」が定型表現となる．これは見方を変えれば，５種類の文字列を，『文末から等しい部分を持つ文字列を持つものは同じグループにする』という基準で，１つのグループに分けていると考えることもできる．

次に３つの定型文末表現を抽出することを考える．すなわち，５種類の文末文字列を３つのグループに分ける．例えば，

a）	「ます」	頻度１９０回
	（「ます」，「します」，「てます」）	
	「います」	１２０回
	「きます」	３０回
b）	「ます」	１３０回
	（「ます」，「きます」，「てます」）	
	「います」	１２０回
	「します」	９０回
c）	「ます」	２２０回
	（「ます」，「います」，「てます」）	
	「します」	９０回
	「きます」	３０回

等と分けることが可能である．この中でｂ）が一番妥当であると考えられる．これは，ａ）やｃ）に比べてｂ）は，得られた定型表現の頻度がそれぞれ１３０，１２０，９０回と差が小さいからである．

そこでグループ分けの際には，得られた定型文末表現の頻度がなるべく平均的になるようにする．実際には相対頻度から計算されるグループ全体のエントロピーが最大になるグループ分けを選べばよい．

3　抽出アルゴリズム

定型表現を自動抽出するには，グループすべての組み合わせを作成して，エントロピーを最大とする文末表現のグループを求めればよい．しかし，グループの組み合わせの数は指数オーダーであるので，計算量が爆発を起こしてしまう．そこで，以下に示すアルゴリズムで準最適なグループ分けを求める．このアルゴリズムでは，はじめに入力となる文字列を出現頻度付きの木構造で表わし，前処理で低頻度の文字列はあらかじめ統合しておく．グループ分けはリーフノードとその親ノードとの統合によって行なう．統合するリーフノードは，統合後の木全体のエントロピーが最大となるリーフノードを選べばよいが，統合の度に木全体のエントロピー計算する必要はない．リーフノードjとその親ノード$par(j)$を統合したことによるエントロピーの減少分（エントロピー差分ΔE），

$$\Delta E = -p_j \log p_j - p_{par(j)} \log p_{par(j)} + (p_j + p_{par(j)}) \log (p_j + p_{par(j)})$$

Automatic extraction of fixed expressions lying on the parts at the end of sentences in Japanese.
Naoto Katoh and Tsuyoshi Morimoto
ATR Interpreting Telecommunications Research Laboratories
2-2 Hikaridai Seika-cho Soraku-gun Kyoto 619-02, Japan

を計算し，最小にするものを選択すればよい．

【アルゴリズム】

ステップ１：抽出対象となる文末表現すべてを，ダミーノードをトップノードとする木構造で表わす．各ノードには文字を対応づけ，そのノード以下の出現頻度，そのノード自身の出現頻度（真の出現頻度）をカウントする．例１の場合には図１i)のようになる．

ステップ２：深さの深いノードから順にノードたどり，ある頻度（f_0）以下のノードはその親ノードに統合する．例１でいえば，［１］の文字列がこのステップで得られる．

ステップ３：各リーフノードに対してエントロピー差分を計算した後小さい順にソートし，リーフノードのリスト（リーフノードリスト）を作成する．

ステップ４：リーフノードリストの中で値の一番小さいリーフノードを統合する．

ステップ５：統合したできたノード（統合ノード）がリーフノードであれば，エントロピー差分を計算してリーフノードリストにマージする．リーフノードでなければ，ソートリーフノードリストの中で，統合ノードを親ノードにもつノードに対してエントロピー差分を再計算して，リーフノードリストにマージする．

ステップ６：真の出現頻度が０でないノードが所望の数になるまでステップ４～５を繰り返す．

4　抽出実験

ＡＴＲ音声言語データベース[Morimoto 94]の中の３００対話（10,282文）を対象として文末表現の抽出実験を行なった．ただし，原理的には１文を入力とできるが，効率の面から考えて文末１５文字以下を入力とした．グループ数は１００とし，ステップ２のf_0は１０回とした．得られた文末表現のうち，出現頻度上位１５個を表１に示す．

表１　ＡＴＲ音声言語データベースからの抽出結果

頻度	文末表現	頻度	文末表現	頻度	文末表現
363	かしこまりました	216	分かりました	181	ですが
359	です	206	ですか	170	ます
241	はい	191	いますね	162	ありがとう
227	でしょうか	183	でございます	151	でございました
219	ですね	183	そうですか	150	ですけれども

5　おわりに

文末表現を頻度付き木構造で表わし，木のエントロピーを最大化することで，文末表現をグループ分けすることにより，定型文末表現を自動抽出する手法について述べた．今後は，他の定型表現抽出手法による結果との比較を行ないたい．また，本手法で抽出された文末表現を用いて，文末表現のバイグラムを作成し，談話構造解析等の精度向上を目指す．

参考文献

[池原 95]池原ほか：大規模コーパスからの連鎖型および離散型共起表現の自動抽出法，電子情報通信学会研究会報告，NLC-95-3，pp.17-24 (1995).

[北 93]北ほか：仕事量基準を用いたコーパスからの定型表現の自動抽出，情報処理学会論文誌，Vol.34，No.9，pp. 1937-1943 (1993).

[Morimoto 94] Morimoto, T.et al. : A Speech and Language Database for Speech Translation Research, Proc. of ICSLP-94, pp. 1791-1794 (1994).

[Lalit 89] Lalit Bahl et al. : A Tree-Based Statistical Language Model for Natural Language Speech Recognition, IEEE Trans. on A.S.S.P, Vol. 37, No.7, pp. 1001-1008 (1989).

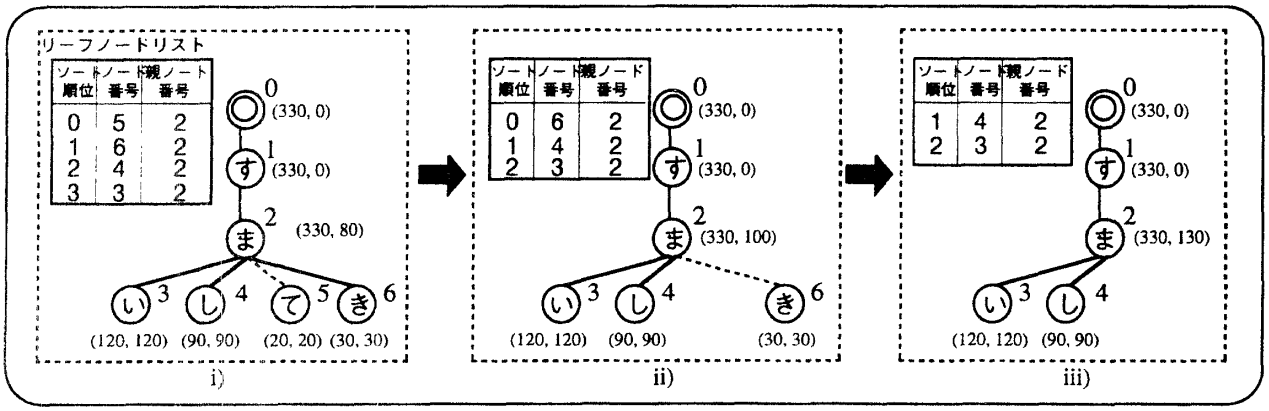
[長尾 93]長尾ほか：大規模日本語テキストのｎグラム統計の作り方と語句の自動抽出，情報処理学会研究会報告，NL-96-1，pp.1-8 (1993).

[新納 94]新納：文字列と後続文字との接続割合の変化を利用した定型的文末表現の自動抽出，情報処理学会報告，NL-104-6，pp.39-46 (1994).

[尾本 95]尾本ほか：木構造を用いたコロケーションの自動抽出，電子情報通信学会研究会報告，ET-94-148，pp.141-148 (1995).

[田本　92]田本ほか：木構造を用いた音韻連鎖統計モデル，電子情報通信学会研究会報告，SP-92-23，pp.77-84 (1992).

[浦谷 95]浦谷：ニュース原稿データベースからの表現パターンの抽出，情報処理学会第50回全国大会，1R-8(1995)



例１に対する抽出処理