クラスタリングとDPマッチングによるネットワーク文法の自動生成*

丸山　誠**
森元　逞***
高橋伸弥***

Automatic Generation of Network Grammar Using DP matching and Clustering

Makoto Maruyama, Tsuyoshi Morimoto and Shin-ya Takahashi

In general, language models for speech recognition are constructed from large text corpora with statistical approach. It is however required very large effort to collect such large copora with high quality. On the other hand, network grammars are also used for not large vocaburary tasks. It however takes a very large time to manually construct these network grammars. To cope with this issue, this paper proposes a method for automatically generating a network grammar using DP matching algorithm and clustcring technique. In addition we utilize a thesaurus to increase the number of accptable sentence patterns. To show the effectiveness of this method, we conduct speech recognition experiments. From the experimental result, compared with the result using a stastical language model (bi-gram model), the network grammar generated by the proposed method can improve the speech recognition performance extensively.

Key Words: Network Grammar, Automatic Generation, Speech Recognition, Language Model

1. はじめに

音声認識システムでは高性能な言語モデルが必要である．言語モデルとは単語の規則を定義したモデルである．近年，大規模なディクテーションシステムが開発されているが，そこでは一般に統計的言語モデル[(1)]が用いられている．この言語モデルは，学習テキストに出現する単語の頻度にもとづいて，その統計量をモデル化したものである．しかし，これらの統計的言語モデルにおいて精度の良い統計量を推定するには，多量の電子化されたデータが必要となる．例えば，日本語ディクテーション基本ソフトウエア（Julius）に用いられている言語モデルの学習には，12年間分（1991年～2002年）の新聞記事が用いられている[(2)]．さらに，これらの学習テキストは基本的には書き言葉であるため，話し言葉には十分に対応できない．しかし，多量の電子化された話し言葉の学習テキストは入手が困難である．また，少量の話し言葉の学習テキストを用いて統計的言語モデルを作成すると，性能の良くない言語モデルになってしまう恐れがある．

一方，言語モデルの一つに中規模の語彙数を対象とするネットワーク文法[(3)]がある．これは，対象とする文パターンをネットワークで表現したモデルであり，正規文法と同等の記述能力を持つ．このネットワーク文法は一般には人手で作成されるため，データ量が増大するにつれて作成が困難となる[(4)]．また，人手による記述ミスにより，予期しない文パターンを受け付けてしまう恐れも

* 平成18年１月20日受付
** 電子情報工業専攻
*** 電子情報工学科

ある．

そこで，本論文ではこれらの問題を解決するために，クラスタリングと DP マッチングを用いてネットワーク文法を自動的に作成する手法を提案する．

2. 基本的な考え

本手法の処理の流れを図１に示す．まず，学習テキストを類似した文にクラスタリングする．次にそれぞれのクラスタごとに DP マッチングを行ってネットワーク文法を作成する．最後に，それらのネットワーク文法に共通の開始（ST）ノードと終了（ED）ノードを繋げて一つのネットワーク文法に統合する．

さらに上記の処理に加え，受理可能な文の数を増やすために単語の意味素性を導入する．例えば「ここは空港ですよ」という文において，「空港」という単語は「公共機関の場所」という意味素性を持つが，これと同じ意味素性を持つ「駅」や「港」と置き換え可能とすることにより，元の１文に対し３文を生成させることができるようになる（図２）．

なお後述するように，システムの内部では意味素性の名前を一定の数字列で表している．以下ではこの数字列を「クラス名」と呼ぶことにする．この意味素性を導入するためには，あらかじめ学習テキストに出現する単語がどのクラスに属するかを定義した情報（クラスファイル）が必要となる．そこで，既存のシソーラスを参照してあらかじめこのクラスファイルを作成しておく．

意味素性を導入してネットワーク文法を作成する処理の流れを図３に示す．まず，クラスファイルを参照して学習テキストに出現するすべての名詞をクラス名に置き換える．次に，そのクラス名に置き換わった学習テキストを用いてネットワーク文法を作成する．最後に作成したネットワーク文法のクラス名をそのクラスに属する名詞に展開する．

3. 言語モデルの自動生成

3.1 DP マッチング

与えられたネットワーク文法と新たに追加したい１文

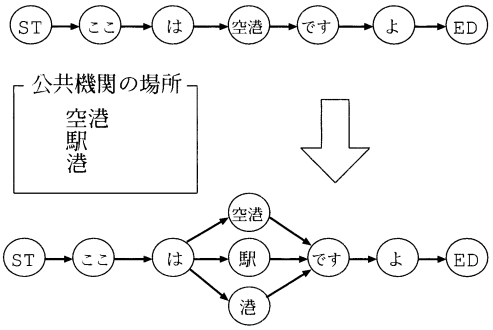


図2 意味素性の導入

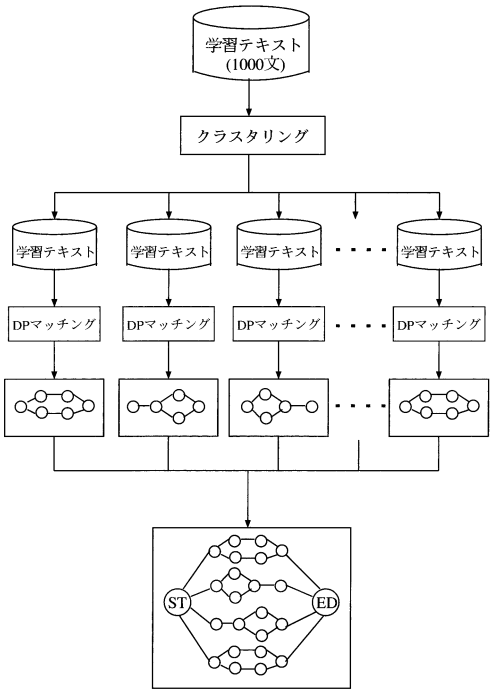


図1 処理の流れ

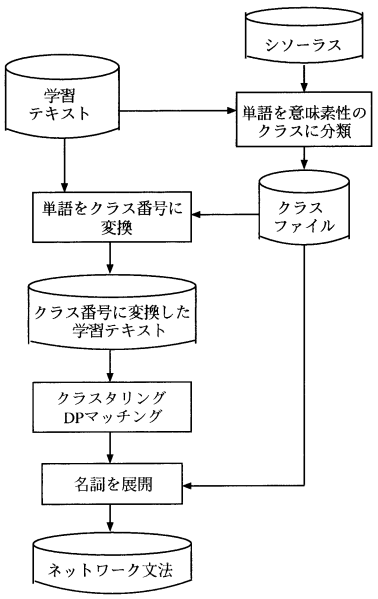


図3 意味素性を導入した処理の流れ

とのDPマッチングを行って単語の関係（挿入，削除，置換，一致）を求め，この結果に基づいてネットワーク文法に新しいノードを追加してゆく．また，この処理を繰り返し行うことによって，ネットワーク文法を順次拡大してゆく．例えば，図４(a)のようなネットワーク文法に「ここは空港らしいですよ」という１文（図４(b)）をDPマッチングさせると，「らしい」が挿入の関係となり，また，「か」と「よ」が置換の関係になるため，図４(c)のようなネットワーク文法となる．

次にDPマッチングの具体的な処理の流れを説明する．まず，図５のように与えられたネットワーク文法(注1)と追加したい１文を２次元配列にそれぞれ格納する．ただし，y軸についてはその接続関係を設定する．

次にすべての格子点についてローカル距離を

$$d(x,y)=\begin{cases}0\cdots x\text{と}y\text{の単語が一致}\\1\cdots \text{それ以外}\end{cases}$$

から求め，STからEDまでの最短距離$g(x,y)$を次式から算出する．

$$g(x,y)=d(x,y)+min\begin{bmatrix}g(x-1,y)\\g(x,\Delta y)\\g(x-1,\Delta y)\end{bmatrix}$$

$\Delta x:y$に遷移可能なすべてのポイント

最後に，求めたSTからEDまでの最短距離にしたがって，現在のポイントと一つ前のポイントのノードの関係（一致，置換，挿入，削除）を順に調べて，新しいノードをネットワークにマージしてゆく（図６）．

3.2 クラスタリング

ネットワーク文法を作成する場合，学習テキストに出現した順に文をDPマッチングしてゆくと，ネットワークの合流が多く発生するために，性能が良くないネットワーク文法を生成する恐れがある．例えば，図４(c)のようなネットワーク文法の場合，「です」のノードでネットワークが合流しているので，「ここはどこですよ」のような非文を受理してしまう．そこで，このようなネットワークの合流を防ぐために，クラスタリングを用いて学習テキストをあらかじめ類似した文に分類する．そして，それぞれのクラスタごとにネットワーク文法を作成する．

まず文間の距離の求め方について述べる．原理的には対象となる２文において，一致する単語の数を調べて距離を求めれば良い．しかし，例えば“This is a pen”と“Is this a pen”のような２つの文では出現する単語は全く同じであるため，これらの文を区別することができない(注2)．そこでこの問題を回避するため，それぞれの文の連続する２単語をワードセットとし，一致するワードセット数により距離を求めることとする．すなわち，文A,Bの距離D_{AB}を計算するために，まずワードセットを要素とする集合A',B'を求め（図７），次にA'とB'から計算した$n(A'\cap B')$と$n(A'\cup B')$から

$$D_{AB}=1-\frac{n(A'\cap B')}{n(A'\cup B')}$$

を求める．ここで$n(\ \ )$は集合の要素数を示す．最後

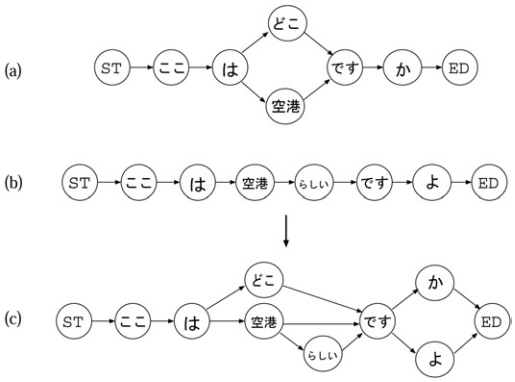


図4 ノードの追加

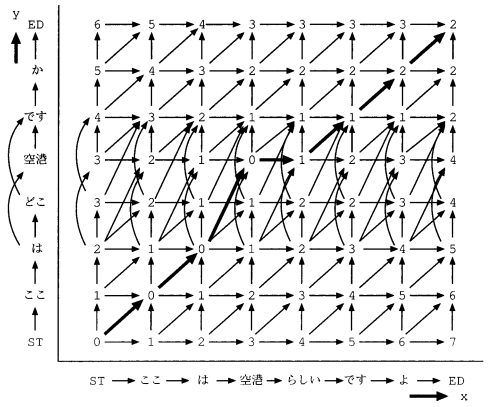


図5 DPマッチング

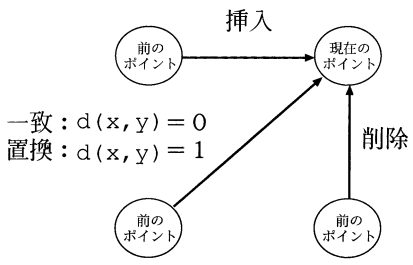


図6 ノードの関係

に，全ての文間の距離が求まったら，最大距離アルゴリズム（付録参照）を用いてクラスタリングを行う．なお，本アルゴリズムでは，クラスタ数をあらかじめ指定することができる．

3.3 シソーラスを用いたクラスの分類

前述したように学習テキスト中の対象となる名詞についてシソーラスより意味素性を求め，クラスに分類する．なお，シソーラスとしては国立国語研究所で作成された分類語彙表[5]を用いることとした．なお本シソーラスにおいては，図８のように意味素性のレベルを分類番号の桁数により表している．そこで，対象ドメインへの適合性を考慮し，分類番号の左から５桁までの番号を並べたものをクラス名とした．また，学習テキスト中に出現した名詞のみを分類の対象とした．本システムで用いたクラスファイルの詳細を表１に示す．

4. 評価実験

提案した手法で作成したネットワーク文法について実験と評価を行った．実験で用いた学習テキストは一般的な旅行会話文(注3)1000文（異り語彙数1193，文当たりの平均単語数8.87）をあらかじめ茶筌(注4)により形態素に区切ったものである．表２に本実験に用いた学習テキストの部を示す．

なお，クラスタ数はそれぞれ30，100の２種のケースで行った．また，性能を比較するためベースモデルとしてバイグラム言語モデル(注5)を用いて同様の実験を行った．

4.1 言語モデルの生成

生成したそれぞれの言語モデルのノード数とリンク数を表３に示す．また，表１には参考のために DP マッチングのみ（クラスタ数１）で作成したネットワーク文法についても示している．なお，括弧の中の数字はクラスタ数である．

表３より，作成したネットワーク文法はバイグラム言語モデルに比べて，ノード数とリンク数がともに増加していることが分かる．これはバイグラム言語モデルでは１単語が１ノードに対応しているのに対し，ネットワーク文法では１単語でも複数のノードに対応するからである．また，クラスタ数を増加させると　ノード数，リンク数がともに増加している．これはクラスタごとに独立したネットワーク文法が作成され，同じ単語でもクラスタごとに別のノードとして生成されるからである．さらに，意味素性を導入した場合，それぞれのクラス内には分類された複数の単語がノードとして存在し，ネットワーク文法において，クラス番号を名詞に展開するたびにそれらを毎回参照するので，ノード数とリンク数が大幅に増加している．

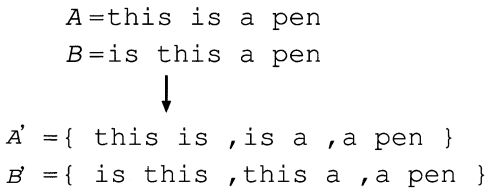


図7 ワードセット集合の例

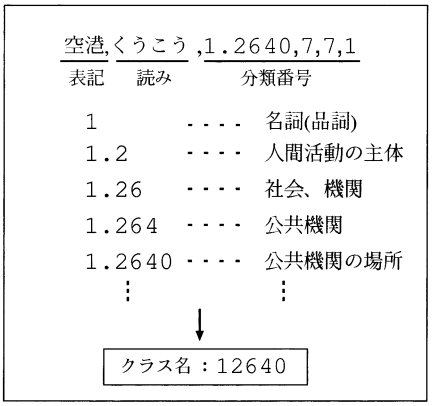


図8 分類語彙表の分類番号

表1 クラスファイルの詳細

対象の異り単語数	760
総クラス数	282
クラス当たりの単語数	2.69

表2 学習テキストの例

明日の晩空いている部屋はありますか
ここでタバコをすってもいいですか
乗り換え切符をもらえますか
バスの路線図をお願いします
パスポートが見当たりません
シカゴまで長距離電話をかけたいのです

表3 言語モデルの詳細

	ノード数	リンク数
バイグラム言語モデル	1195	3282
クラスタ（1）	3299	4822
クラスタ（30）	3582	5047
クラスタ（100）	4006	5445
意味素性（30）	17363	29941
意味素性（100）	18576	31820

次にこれらの言語モデルからHSGen(注6)によりランダムに500文を生成して平均分岐数と平均単語数を調べた（表４）．本手法によるネットワーク文法ではクラスごとにネットワークを作成しているため，平均分岐数が半分ほどに低下している．また，意味素性を導入した場合，出現する名詞がクラスとして抽象化され，単語の入れ換えが可能になったため，平均分岐数が若干増加している．

4.2 音声認識実験

上記で作成した言語モデルを用いて音声認識実験を行った．条件は表５のとおりである．

音声データは学習テキストからランダムに100文を抜き出し，発話者３人によりそれぞれ異る文を読み上げたものを用いた．なお，限られたデータを用いてオープンなデータに対する認識率を求めるために，Cross Validation（交差検定）(注7)法で実験を行った．結果を表６に示す．

表６から，本手法によるネットワーク文法はバイグラム言語モデルに比べて音声認識率が向上していることが分かる．単語認識率の向上は2.5ポイント程度であるが，文認識率では約15ポイントから20ポイントと大幅に向上している．

4.3 言語モデルの受理能力の評価

上で述べたように，本手法で作成したネットワーク文法ではバイグラム言語モデルよりも高い音声認識率を達成することができた．しかし，一方では平均分岐数が少ないために，受理可能な文の数が減少する恐れがある．そこで，本手法で作成したネットワーク文法について受理能力を評価した．

4.3.1 受理能力の計算

受理能力の求め方について述べる．まず，平均分岐数を求める際に生成した500文について，日本語として正しい文（正文）か日本語として誤った文（非文）かを目視でチェックし，正文の比率を求める．次に，このデータをもとに，言語モデルの受理可能な正文の数を

$$\text{受理可能な正文の数} = \text{平均分岐数}^{\text{平均単語数}} \times \text{正文の比率}$$

で算出し，学習テキスト1000文に対する受理能力を

$$\text{受理能力} = \text{受理可能な正文の数} / \text{学習テキスト}$$

で求める．例えば，DPマッチングのみでネットワーク文法を作成した場合，生成させた500文のうち正文の比率は54.0%，平均分岐数は2.89，平均単語数は9.23であるので上記の式より

$$\text{受理可能な正文の数} = 2.89^{9.23} \times 0.54 = 9693$$

つまり，9693文を受理することが可能である．すなわち，学習テキストに対して約9.7倍のテキストを受理することができることになる．

表4 各言語モデルでの平均分岐数

	平均分岐数	平均単語数
バイグラム言語モデル	6.50	8.27
クラスタ（1）	2.89	9.23
クラスタ（30）	2.52	9.22
クラスタ（100）	2.43	9.12
意味素性（30）	3.60	9.24
意味素性（100）	3.27	8.93

表5 音声認識の実験条件

音声データ	100文（サンプリング周波数16kH）
音声認識エンジン	HVite(7)
音響モデル	HMM（４混合性別非依存トライフォン）

表6 音声認識実験の結果

	音声認識	
	文認識率	単語認識率
バイグラム言語モデル	62.0%	92.4%
クラスタ（1）	75.0%	94.0%
クラスタ（30）	82.0%	94.4%
クラスタ（100）	83.0%	95.0%
意味素性（30）	76.0%	93.5%
意味素性（100）	79.0%	94.3%

表7 受理能力

	正文の比率	受理能力（倍）
クラスタ(1)	54.0%（272文）	9.7
クラスタ(30)	53.0%（265文）	2.7
クラスタ(100)	61.8%（341文）	2.0
意味素性(30)	35.2%（176文）	48.6
意味素性(100)	36.2%（182文）	14.2

4.3.2 受理能力の評価

ネットワーク文法について上記の式より受理能力を求めた結果を表７に示す．

この結果によると，クラスタリングのみの場合，意味素性を導入した場合，いずれにおいてもクラスタ数が増加するにつれて受理能力は減少している．また，音声認識実験の結果ではクラスタ数が増加するにつれて音声認識率は向上している（表６）から，クラスタ数の増減については音声認識率と受理能力はトレードオフの関係にあることが分かる．また，単純にクラスタリングを行っただけのケースでは，受理能力は元の学習データの２倍～３倍程度にしかならないが，意味素性を導入することにより10倍～40倍程度に向上している．このとき音声認

識率もいくらか低下するが，それでもバイグラム言語モデルに比べれば文認識率は10ポイント以上も高くなっている（表６）．なお，意味素性を導入した時に音声認識率が低下するのは，正文の比率が低下したためであろうと思われる．これは，現在のシステムではシソーラスから意味素性のクラスに分類する際，シソーラスの先頭から順に単語を検索して最初にマッチしたエントリを参照しているため，不適切なクラスに分類されてしまうケースがあったのが原因であると考えられる．

5. まとめ

本論文ではクラスタリングとDPマッチングを用いてネットワーク文法を自動的に作成する手法を提案した．この手法で作成したネットワーク文法はバイグラム言語モデルに比べ，音声認識率を大幅に向上させることができた．また，さらに意味素性を導入することによりある程度の受理能力を確保することができた．

意味素性を導入する際，シソーラスを参照して自動的にクラスに分類したが，正文の比率は低い結果となってしまった．これは，シソーラスでの単語の検索方法に問題があるためであり，この検索方法を改良することにより，さらに性能の良いネットワーク文法を作成することができると考えられる．今後はこの点についてさらに検討していく予定である．

注

(1) 与えられたネットワーク文法がない場合，先頭の１文よりネットワーク文法を作成する．
(2) ここでは，大文字／小文字や“？”は考えないものとする．
(3) 市販の旅行会話のテキスト数冊より抽出．
(4) 奈良先端科学技術大学院で開発された形態素解析システム．
(5) CMU-Cambridge SLM Toolkit[6]により作成．
(6) HTK ツールキットにおける評価テキスト生成プログラム[6]．
(7) 標本サイズｎのデータセットに対して，n-1個を使ってモデルを構成し，残しておいた１個のデータに対してモデルを適用して性能を評価することをｎ回くり返す方法．

参 考 文 献

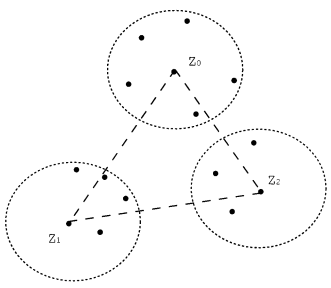
(1) 北研二：確率的言語モデル，東京大学出版会（1999）
(2) 河原達也，武田一哉，山本幹雄，伊藤克亘，鹿野清宏，李晃伸，山田篤：連続音声認識コンソーシアムの活動報告及び最終版ソフトウエアの概要，情報処理学会研究報告，SLP-49-57（2003）
(3) 鹿野清宏，伊藤克亘，河原達也，武田一哉，山本幹雄：音声認識システム，オーム社（2001）
(4) T. Morimoto, S. Orimoto and S. Takahashi: Automatic Extraction of Language Models for Speech Translation from a Travel Conversation Bilingual Corpus, Proc. of PACLING ’05（2005）
(5) 国立国語研究所：「分類語彙表」形成による語彙分類表（増補版）（1996）
(6) P.R. Clarkson and R. Rosenfeld: Statistical Language Modeling Using the CMU-Cambridge Toolkit, Proc. of ESCA Eurospeech（1997）
(7) S. Young *et. al.*: The HTK Book Ver 3.0,（1999）［http://htk.eng.cam.ac.uk］

付録　最大距離アルゴリズム

最大距離アルゴリズムは，付図１に示すように，クラスタ中心からの距離が大きなサンプル点は別のクラスタとみなす処理を繰り返すことで，クラスタリングを行う処理である．以下に，その処理手順を示す．

まず，Ｎ個のサンプル集合 $X=x_1,x_2,\cdots,x_n$ においてそれぞれのサンプル間の距離を計算する．次に，以下の手順により各クラスタ中心 $z_0,z_1,\cdots,z_n$ とサンプル x_n の属するクラスタ c_j を求める．

１．サンプル集合 $X=x_1,x_2,\cdots,x_n$ から任意に１点を選び，その点をクラスタ中心 z_0 とする．
２．サンプル集合 X のそれぞれの点と z_0 との距離Ｄを計算し，その中で最も距離の大きいサンプルをクラスタ中心 z_1 とする．
３．クラスタ中心間の距離の最大値を D_{max} とおくが，ここでは初期値として $D_{max}=\mathrm{D}[z_0][z_1]$ とする．
４．集合 X の各サンプル x_n について z_0,z_1（すべてのクラスタ中心）との距離をそれぞれ計算し，一番近いクラスタ中心を選び，そのサンプルとクラスタ中心間の距離を D_{x_n} とする．



付図1 最大距離アルゴリズム

5．D_{x_n} の X における最大値 D_X をとり $D_X \geq \theta \cdot D_{max}$ $\left(\theta \geq \frac{1}{2}\right)$ だったらその x_n を新たなクラスタ中心 z_2 とする．

6．すべてのクラスタ中心間の距離を求め，その中で最大のクラスタ中心間の距離を新たな D_{max} とする．

7．4～6を同様に繰り返し，5の処理において $D_x \leq \theta \cdot D_{max}$ になったらクラスタ中心はそこまでに得られたものとし，それらのクラスタ中心をサンプル x_n の属すクラスタ c_j とする．