音声言語情報処理 9－16
(1995. 12. 15)

文字列パターンのN-gramによる文節モデルの検討

伊藤 彰則　好田 正紀

山形大学 工学部
〒992 山形県米沢市城南4丁目3-16
0238-26-3369
aito@ei5sun.yz.yamagata-u.ac.jp　kohda@ei5sun.yz.yamagata-u.ac.jp

あらまし 日本語文/対話音声認識において，N-gramに代表される統計的言語モデルを用いようとした場合，その単位が問題となる．英語の場合には単語を単位としたN-gramを用いるのが一般的であるが，日本語の場合には単語に分かち書きされないため，事前に形態素解析が必要となる．しかし，対話などの場合は従来のシステムによる形態素解析が難しい．そこで本稿では，形態素解析を行なわずに性能の良いN-gramを作るための手法，誤りを含んだ形態素解析によるN-gram，さらに文字列パターンのクラス化によるN-gramについて検討を行なった．その結果，パターンクラスによる方法で人手による形態素解析を越える結果を得ることができた．

キーワード 言語モデル，N-gram，単語間距離

Language Modelling by String Pattern N-gram

Akinori Ito and Masaki Kohda

Faculty of Engineering, Yamagata University
4-3-16 Jonan, Yonezawa-shi, Yamagata 992 JAPAN
0238-26-3369
aito@ei5sun.yz.yamagata-u.ac.jp　kohda@ei5sun.yz.yamagata-u.ac.jp

Abstract Markov model based language models (N-gram) are popular among sentence/dialog speech recognition. On applying these models to Japanese speech recognition, one has to dicide what to be a unit of N-gram. As Japanese sentence is not divided into words, the morphemic analysis is required before word-by-word processing. But it is difficult to get the precise analysis automatically for spontaneous speech transcription. In this paper, we propose several language models which enable fully automatic construction of the model. We examined three types of models: N-gram by string pattern, N-gram by automatic morphemic analysis and string pattern class N-gram. These models were compared by perplexity. From the experimental results, the string pattern class N-gram got better performance than morpheme N-gram.

key words Language Model, N-gram, word similarity

1 はじめに

現在我々は文節構造をベースとした対話音声認識を目指して研究を進めている．その一環として，これまで人間同士の対話から文節構造モデルを構築するという研究を行ってきた[1]．しかし，単純な有限オートマトンによる文法では認識に用いる際の制約能力が不十分であることから，N-gramなどの確率モデルの利用を検討している．

N-gramによる確率モデルを日本語に適用する際に問題になるのは，何をモデルの単位とするかである．英語の場合，学習用のテキストは単語ごとにわかち書きされているため，単語を単位としたN-gramを構成するのが最も簡単かつ自然である．一方，日本語は単語ごとにわかち書きされないので，単語を単位とした処理を行うためには，事前に形態素解析が必要である．しかし，従来の形態素解析システムを用いて会話文のような文章を解析するのは難しく[5]，正確な解析は望めない．筆者らの構築した文節モデルによって形態素解析をすることも可能であるが，解析結果に曖昧性があるため，解析を自動化することは難しい．確率モデルを構築するには大量の学習用テキストが必要であるため，学習テキスト整備には多大な労力が必要となる．

形態素解析の問題を回避するための一つの方法は，音素・音節・文字などの単位を用いた確率モデルを使うことである[2, 3, 4]．文献[4]では，学習テキストに出現する漢字・仮名とその読みを用いてN-gramを構成している．この方法は，電子化されたテキストをほぼそのまま学習テキストとして利用できるという点で優れた方法であるが，文字が単位であるため，元のテキストが持つ制約を十分に利用しているとは言えない．もちろん，文字単位であっても，統計をとる連鎖の長さを長くすることによって制約を強めることは可能である．しかし，その場合には，計算量や記憶容量の点で困難が生じる．

そこで本稿では，次のような手法によって自動的にN-gramを生成する方法について検討する．

1. 「単語単位」と「文字単位」の中間的な単位として，学習テキストから自動的に抽出できる単位を用いて確率モデルを構成する．この方法は，学習テキストから何らかの基準で文字列を抽出し，これを単位として学習テキストを分割し，その単位のN-gramを構成するものである．
2. 学習テキストの文字列を何らかの基準でクラスに分け，そのクラスと文字のN-gramを構成する．
3. 学習テキストに対して文節数最小基準による形態素解析を行ない，その結果からN-gramを生成する．

今回は，文節単位のデータを用いてN-gramを構成し，マルコフモデルによる文節モデルを構築している．原理的には，文節モデルだけでなく，文のモデルも構築可能である．しかし，今回利用したデータが均一なタスクでないこともあり，文単位のモデル構築は行わなかった．

2 文字列パターンによるN-gram

2.1 文字列パターンの抽出によるN-gram構成

この方法は，学習テキストから何らかの基準で文字列を抽出し，これを単位として学習テキストを分割し，その単位のN-gramを構成するものである．例えば，「いらっしゃいませんので」という文字列には，「い」，「いら」，「いらっ」，…，「ら」，「らっ」，「らっし」，…などの部分列が含まれている．これらの部分列に何らかの基準で順位をつけ，順位の高い方から適当な数だけ選び出す．選び出された部分列を単語だと考え，これらのパターンに基づいて元のテキストを解析する．例えば，基準として部分列の出現頻度を用い，頻度順に2000個を抽出して最長一致法で解析した場合，「いらっしゃいませんので」という文節は「いら/っしゃ/いません/ので」と分割される．元のテキストのうち，この方法で解析されなかった部分は，文字単位に分割する．あとは，この系列から部分列のN-gramを構成する．実験では，計算量の問題から，部分列の長さを最高6に制限している．部分列の順位付けの方法として，「頻度による方法」と「パターン間距離による方法」の2種類について検討した．

頻度によるパターン抽出法とは，上記の部分列の出現頻度を数え，頻度の高いパターンから選んでいく方法である．頻度を計算するにあたっては，互いに重なったパターンもそれぞれ独立に1個と数えている．例えば「それぞれ」という文字列については，「そ」「ぞ」「それ」「れぞ」「ぞれ」「それぞ」「れぞれ」「それぞれ」が各1回出現，「れ」が2回出現と数えている．高頻度パターン1000個の場合，例えば次のようなパターンが抽出されている．

長さ	パターン例
1	い で す か と の う ん て ま は ...
2	です ます ん で っ て すか そ う い う ...
3	んです ですか ですね りま す う です ...
4	そうです んですけ ですけど あります ...
5	そうですね んですけど でしょうか ...

パターン間距離による方法とは，各部分列の間の距離を適当な方法で計測し，距離の近いペアを一定個数抽出する方法である．抽出されたペアに含まれている部分列を解析に用いる．今回は，パターン間距離として divergence に基づく方法[8]を用いた．2種類の部分列を p_1, p_2 とし，その部分列が出現したコンテキストが c である確率を $P(c|p_1), P(c|p_2)$ とするとき，p_1 と p_2 との距離を

$$d(p_1, p_2) = \sum_c d(p_1, p_2; c)$$

$$d(p_1, p_2; c) = \begin{cases} (P(c|p_1) - P(c|p_2)) \log \dfrac{P(c|p_1)}{P(c|p_2)} \\ \quad \text{if} \quad P(c|p_1) > 0 \text{ and} \\ \qquad P(c|p_2) > 0 \\ \alpha \quad \text{otherwise} \end{cases}$$

とするものである．α は適当なペナルティである．この尺度では，部分列 p_1, p_2 の両方が出現しないコンテキスト c における距離 $d(p_1, p_2; c)$ が本来の定義と違っている (本来は 0 になるが，ここの定義では α)．これは，「あるコンテキストで p_1, p_2 がともに生起しない」ということを否定的にとらえたものである．この尺度では，出現頻度の高いパターンほど，他のパターンとの距離が近くなりやすい．今回の実験では，コンテキストとしてパターン前後の 1 文字を用いた．たとえば，「お願いしたいんですが」という文節における「願い」のコンテキストは「お / し」になる．

これらの方法で抽出された文字列パターンを使って学習テキスト / 評価テキストを分割するわけだが，このとき左最長一致法と文節 (形態素) 数最小法[6]の 2 種類の方法を比較した．

2.2 Multigram による文の分割

次に，Multigram[7]の手法によって文を分割する手法を検討した．この手法は英文を単語列のパターン系列として確率を与える言語モデルである．N-gram とは異なり，1 つ以上の単語からなる単語列パターンが情報源から生成されるというモデルになっている．これを図 1に示す．例えば abc という入力系列に対して，単語列 ab がパターンとして出現する確率を $P([ab])$ とするとき

$$P(abc) = \max \begin{cases} P([a])P([b])P([c]) \\ P([a])P([bc]) \\ P([ab])P([c]) \\ P([abc]) \end{cases}$$

のようにして全体の出現確率を求める．各単語列パターンの出現確率は EM アルゴリズムによって求めることができる．今回はこの手法を応用し，入力を最適な文字列に区切った．このときの Multigram 確率から直接 perplexity を求めることもできるが，今回はこの手法によって分割したテキストからあらためて N-gram を求め，これを用いた．

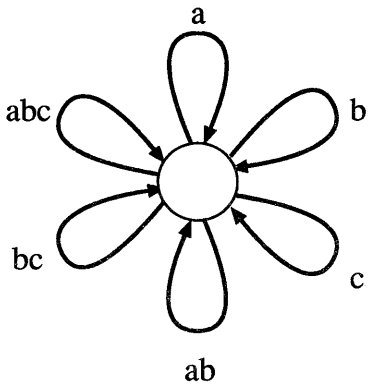

図 1: Multigram モデル

2.3 形態素解析による N-gram

前述の通り，対話データの完全な形態素解析は難しい．しかし，形態素数最小基準で解析を行ない，何らかの結果を出すことは可能である．そこで，このようにして得られた解析結果から N-gram を作り，他の方法と比較した．

2.4 パターンクラスによる N-gram

文字列パターンをクラスにまとめ，パターンクラスと文字の N-gram を構成する方法を試した．まず，前述の単語間距離と基本的に同じ方法 (距離の定義が若干違う) を用いてパターン間の距離を求め，距離の近いパターンを一定個数抽出する．次に，それらのパターンのうち，コンテキストを共有しているものを 1 つのクラスにまとめる．この操作により，文字列パターンを「コンテキストについて互いに素なクラス」に分類することができる．この処理において，パターン間距離は次の式を用いている．2種類の部分列を p_1, p_2 とし，その部分列が出現したコンテキストが c である確率を $P(c|p_1), P(c|p_2)$ とするとき，p_1 と p_2 との距離を

$$d(p_1, p_2) = \sum_c d(p_1, p_2; c)$$

$$d(p_1,p_2;c)=\begin{cases}(P(c|p_1)-P(c|p_2))\log\dfrac{P(c|p_1)}{P(c|p_2)} & \text{if } P(c|p_1)>0 \text{ and } P(c|p_2)>0\\ 0 & \text{if } P(c|p_1)=0 \text{ and } P(c|p_2)=0\\ \alpha & \text{otherwise}\end{cases}$$

とする．この場合，どちらの単語も出現しないコンテキストについての距離が0になっており，本来のdivergenceの定義に近くなっている．この距離を使うと，偶然同じコンテキストで1回だけ出現した2つのパターンの距離が0になってしまう．そこで，出現回数が2回以下のパターンは距離計算から除外している．

3 平滑化とPerplexity

評価テキスト中には，学習テキストに出現しなかったN-gramの組みあわせ(未知語)が存在する．これに対処するため，平滑化を行なう．今回は，unigram, bigram, trigramのそれぞれに，改良されたback-off平滑化を組みあわせたもの[10]を用いている．この方法は次のようなものである．

単語をw，単語の出現したコンテキストをxとし，コンテキストxでwが出現する回数を$C_{w|x}$，コンテキストxで出現する単語の数をn_x，コンテキストxで出現する単語の種類をr_xとしたとき，

$$\hat{P}(w|x)=\begin{cases}\dfrac{C_{w|x}}{n_x+r_x} & \text{if } C_{w|x}>0\\ \dfrac{r_x}{n_x+r_x}\hat{P}(w|x') & \text{if } C_{w|x}=0 \text{ and } n_x>0\\ \hat{P}(w|x') & \text{if } n_x=0\end{cases}$$

ここでx'はxより1つ低次のコンテキストである．今回の実験では，最終的にzerogram($=\frac{1}{r_0}$)までback-offを行なっている．

各モデルの評価として，文字単位の補正perplexity[11]を用いている．通常のperplexityは，評価テキストを$w_1\ldots w_n$とすると

$$PP=P(w_1\ldots w_n)^{-\frac{1}{n}}$$

で与えられる．しかし，この評価基準では，評価テキスト中の未知語の扱いによって値が大きく変化し，異なる言語モデル間の評価基準として適当ではないことが指摘されている[11]．ここで用いている補正perplexity(APP)は，

$$APP=\left(P(w_1\ldots w_n)m^{-n_u}\right)^{-\frac{1}{n}}$$

で表される．ただし，n_uは評価テキスト中に出現した未知パターンの文字数，mは評価テキスト中に出現した未知パターンの文字種数である．

4 学習データ・評価データ

学習に用いたのは，日本音響学会連続音声データベース[9]の模擬対話書きおこしテキスト(CRL, ETL, KIT, NTU, OSA, TOH, TSU) 44対話，3606発話，19031文節，52588形態素，87520文字．この学習データに出現する形態素の語彙数は3922，文字種は1378である．perplexityの評価用に用いたのは，同データベースの5対話(HOS) 649発話，2446文節，6698形態素，11278文字．形態素の語彙数は945，文字種は611である．学習用データと評価用データのタスクは一致していない．形態素単位の実験に用いたデータは，各テキストを筆者らの構築した文節モデルで形態素解析し，目視で一意化したものを用いている．

5 各手法でのperplexityの算出

文字列パターンの抽出によるN-gramの場合，まず文字単位の学習テキストの中から，それぞれの方法で文字パターン集合を抽出し，(擬似)形態素解析を行なう．高頻度パターンの場合，上位250 ～ 8000個のパターンを抽出した．また，単語間距離の場合，距離の近い1000 ～ 8000ペアを抽出した．これらのパターンを用いて学習テキストを分割し，そのパターン系列からN-gramを生成する．また，同じ文字パターン集合を使って評価テキストを擬似形態素解析し，これをN-gramで評価してperplexityを算出した．

Multigramによる分割の場合，まずEMアルゴリズムによって学習テキストの部分パターンの出現確率を求める．次に，その出現確率が最大になるようにテキストを分割し，これをN-gramの学習データとする．次に，同じ出現確率によって評価データを分割し，これをN-gramで評価した．

形態素解析によるN-gramの場合，まず学習テキストをすべてカバーする辞書を使って学習テキストの形態素解析を行ない，形態素数最小基準による解析結果を得る．次に，以下の3種類の評価テキストについてN-gramによる評価を行なった．

1. 目視によって形態素解析した評価テキスト．
2. 学習テキストのみをカバーする辞書を使って自動形態素解析した評価テキスト．
3. 学習テキストと評価テキストをカバーする辞書を使って自動形態素解析した評価テキスト．

表1 実験結果

N-gramの単位			パターン数		補正 perplexity			n_u	m
			抽出	出現	unigram	bigram	trigram		
形態素単位			–	3922	104.412	56.367	81.932	2100	417
文字単位			–	1378	164.710	51.814	43.307	93	51
高頻度パターン	最長一致	250	250	1524	96.659	39.823	43.714	93	51
		500	500	1690	82.742	37.572	45.095	93	51
		1000	1000	2007	70.062	37.598	48.197	100	58
		2000	2000	2607	62.954	39.856	53.896	159	74
		4000	4000	3665	55.358	42.977	61.083	287	105
		8000	8000	5279	60.151	53.534	77.492	669	148
	最小	250	250	1524	95.838	39.744	43.498	93	51
		500	500	1690	82.737	37.901	45.146	96	54
		1000	1000	2007	70.885	37.493	47.863	101	59
		2000	2000	2607	62.861	39.816	53.117	145	79
		4000	4000	2665	55.382	42.395	58.698	250	104
		8000	8000	5279	57.944	50.963	70.707	539	149
単語間距離	最長一致	1000	210	1522	103.861	41.942	44.899	93	51
		2000	377	1668	90.892	40.506	46.925	95	53
		4000	621	1862	80.420	39.338	48.353	95	53
		8000	987	2145	75.199	39.893	51.156	115	67
	最小	1000	210	1522	103.397	43.167	46.586	93	51
		2000	377	1668	90.196	42.427	49.509	97	51
		4000	621	1892	77.198	41.494	51.387	102	53
		8000	987	2145	70.835	43.471	56.369	117	63
multigram/N-gram			–	16231	136.777	138.914	193.933	3333	534
形態素解析	目視		–	3925	105.211	57.327	84.296	2100	417
	辞書＝学習		–	3925	101.094	56.432	83.073	2115	416
	辞書＝学習＋評価		–	3925	138.037	83.206	108.948	3397	437
パターンクラス	100000		1838	3128	–	44.934	35.025	100	55
	200000		2679	3907	–	50.130	27.540	101	56

2の場合には，未知語を含む文節は解析不能であるため，文節全体が1つの形態素として扱われている．

パターンクラスによるN-gramの場合，まず学習テキストから近距離ペアを100000～200000個抽出し，パターンクラスを構成した．100000ペアのデータからは520クラス，200000ペアのデータからは422クラスが構成された．次に，学習テキストのどの部分がパターンクラスに属するパターンであるかを決定している．これには，単純な左最長一致法を用いた．この結果，次の例のようにクラスと文字の並んだデータができる．

元の文字列	処理後
どちら様でしょうか	C2310 ら 様 で し ょ う か
いらっしゃいませんのでは	C2440 ま C0729 で C2470

このクラスと文字の並びからN-gramを構成し，また各クラスCから文字列パターンwが生成される確率$P(C \to w|C)$を求めておく．次に，評価テキストを同じように解析し，N-gramの確率と，クラスから文字列パターンが生成される確率とを掛けて全体の生成確率を得る．例えば，「わかりまして」という文字列は，「わかりまし」の部分がクラスC2443に解析される．この場合のbigramの生成確率は，

$$P(\text{C2443}|\$)P(\text{て}|\text{C2443})P(\$|\text{て}) \times P(\text{C2443} \to \text{わかりまし}|\text{C2443})$$

と計算される($は文の先頭と末尾を表す記号である)．このようにして求めた確率からperplexityを求めた．

6 評価実験結果

以上の方法によって求めたperplexityを表1に示す．表中の「パターン数」は，抽出されたパターン数と，学習テキスト中に出現したパターン数(解析できない部分を文字単位に分割したものを含む)を示している．また，n_uは評価テキスト中に出現した未知語の文字数，mは評価テキスト中に出現した未知語の文字種数である．全体にtrigramによるperplexityが大きいが，これは学習テキストの量の不足によると思われる．全体で最もperplexityが小さくなったのは，200000ペアのパターンクラスのtrigramによる27.54で，文字単位，形態素単位よりも低い結果になった．抽出パターンによる方法では，高頻度パターン1000を使って形態素数最小で解析したbigramが最も良い結果で，37.49であった．Multigramによる方法は，全体としてあまり良い性能が得られなかった．形態素解析による方法は，予想よりも良く，目視による形態素単位のN-gramに近いperplexityを得ることができた．辞書の違いによって性能に差が出ていることから，大規模な辞書による解析によって良い性能が得られる可能性がある．

7 まとめ

日本語文/対話音声認識をターゲットとして，自動的にN-gramを構築するための手法の比較検討を行なった．今回は，「パターンの抽出による方法」「Multigramによる方法」「形態素解析による方法」「パターンクラスによる方法」について比較検討を行った．比較実験の結果，パターンクラスによる方法で，人手による形態素解析を越える性能が得られることがわかった．

この方法の現在の問題点としては，文字によるテキストをそのまま用いているため，読みの情報がないという点が挙げられる．これについては，読みの情報をいれた言語モデル[4]を検討する必要があるであろう．また，Ergodic HMM[12]などの言語モデルとの比較も行っていきたい．

参考文献

[1] 伊藤，牧野：「音声認識のための文節構造モデルとその制約について」，情処学会研究報告 95-SLP-6-7, pp.43-50 (1995-5)

[2] T. Kawabata *et. al.*:"Japanese Phonetic Typewriter Using HMM Phone Recognition and Stochastic Phone-Sequence Modeling", IEICE Trans. Information and Systems, Vol. E74, No. 7, pp.1783-1787 (1991)

[3] 荒木，村上，池原：「2重音節マルコフモデルによる日本語の文節音節認識候補の曖昧さの解消効果」，情処学論, vol. 30, No. 4, pp.467-477 (1989)

[4] 山田，松永，川端，鹿野：「音声認識における仮名・漢字文字連鎖確率に基づく統計的言語モデルの利用」，信学論(A) Vol. J77-A, No. 2, pp.198-205 (1994)

[5] 日本音響学会：「日本音響学会研究用連続音声データベース」 vol.6, 付録A (1992)

[6] 田中穂積：「自然言語解析の基礎」，産業図書 (1989)

[7] Sabine Deligne and Frederic Bimbot: "Language Modelling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams", Proc. ICASSP-95, vol.I, pp. 169-172

[8] Michael K. McCandless and James R. Glass: "Empirical Acquisition of Language Models for Speech Recognition", Proc. ICSLP94, pp.835-838 (1994)

[9] 小林，板橋，速水，竹沢：「日本音響学会研究用連続音声データベース」，音響学会論文誌 vol.48, No.12, pp.888-893 (1992)

[10] Paul Placeway, Richard Schwartz, Pascale Fung and Long Nguyen: "The Estimation of Powerful Language Models from Small and Large Corpora", Proc. ICASSP-93, vol.II, pp.33-36 (1993)

[11] Joerg Ueberla:"Analysing a simple language model — some general conclusions for language models for speech recognition", Computer Speech and Language, Vol. 8, No. 2, pp.153-176 (1994-4)

[12] T. Kuhn, *et. al.*: "Ergodic Hidden Markov Models and Polygrams for Language Modeling", Proc. ICASSP94, Vol. 1, pp. 357-360 (1994)