

HMMを用いた複数 n -gram モデルによる言語モデルの構築

長野 雄[†] 鈴木 基之^{††} 牧野 正三^{††}

一般に n -gram による言語モデルでは 1 つの n -gram 統計を学習する。タスクがいくつかのサブタスクに分けられる場合、複数の n -gram 統計を用いる方が単一の n -gram 統計を用いるよりも性能を上げることができると考えられる。そこで本論文では HMM を用いた複数 n -gram モデルによる言語モデル SS (Stochastic Switching) n -gram を提案する。SS n -gram は、HMM の出力確率を n -gram 確率分布にしたモデルで、学習を行うことで各状態にサブタスクに対応した n -gram 統計を自動的に獲得する。SS n -gram の出力確率を bigram とした場合、新聞記事タスクにおいて bigram と比べ約 15% パープレキシティを下げることができた。また、SS n -gram に削除補間法を適用することで、平滑化した bigram と比べ平滑化後も約 8% パープレキシティを下げることができた。

Construction Method of Language Model Using Stochastic Switching n -gram

TAKESHI NAGANO,[†] MOTOYUKI SUZUKI^{††} and SHOZO MAKINO^{††}

In traditional speech recognition systems, a single kind of n -gram is used for n -gram language model. If a task can divide into more small sub tasks, utilization of several kinds of n -gram gives better performance. In this paper, we propose a so-called SS (Stochastic Switching) n -gram which consists of several kinds of n -gram stochastically obtained using a discrete-type HMM. In SS n -gram, each state has one kind of n -gram as output probability. When bigram is used as output probability, the perplexity with SS n -gram is reduced by 15% comparing to that with ordinary bigram. After the deleted interpolation method is applied, the perplexity is still reduced by 8%.

1. はじめに

連続音声認識のための言語モデルとしてよく用いられるモデルに n -gram がある。通常 n -gram による言語モデルでは 1 つのタスクに対し単一の n -gram を学習して使用する。タスクが異なれば語彙や文体が異なるため、それぞれのタスクのテキストを用いて n -gram を学習した方が性能が上がるということが知られている¹⁾。

では、1 つのタスクに対して単一の n -gram で十分か、ということ必ずしもそうではない。たとえばニュース番組タスクを考えたとき、アナウンサの朗読部とインタビューのような対話部では文体などが異なるため、それぞれに対応した n -gram を用いた方が性能が良いことが予想される。例にあげたニュース番組以外にもタスクをより小さなタスク(ここではサブタスクと呼

ぶ)に分けることができるタスクが存在すると考えられる。

そもそもここでいうタスクとは認識対象のことを指し、これは人間が考えた意味的な「かたまり」であるため、単一の n -gram の予測性能を最大にする「かたまり」と対応するとは限らない。そこで、タスクをサブタスクに分割し、 n -gram の予測性能を最大にする「かたまり」とサブタスクを一致させることができれば、それぞれに対応した n -gram を用いることで、単一の n -gram を用いるよりもより良い性能が得られると考えられる。そこで本論文では、複数の n -gram を確率的に切り替えて用いるモデル(SS n -gram: Stochastic Switching n -gram)を提案する。このモデルでは尤度最大基準で学習テキスト中のサブタスクを自動的に抽出し、対応した n -gram を学習することができる。

1 つのタスクをいくつかのサブタスクに分け複数の n -gram を求めるモデルがすでにいくつか提案され^{2)~4)}、1 つの n -gram を使う場合に比べ、良い性能を示すことが報告されている。しかし、これらのモデルでは求めた複数の n -gram を足し合わせて 1 つのモ

[†] 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

^{††} 東北大学大学院工学研究科
Graduate School of Engineering, Tohoku University

デルにしてしまう。もし、複数の n -gram を切り替えて使うことができれば、さらに良い性能が得られるはずである。

以下、本論文では、2章で SS n -gram とその学習アルゴリズムについて提案し、3章で新聞記事をタスクとした評価実験について、4章では SS n -gram の平滑化の方法について述べる。

2. SS n -gram モデル

2.1 SS n -gram の概要

タスクをより小さな複数のサブタスクへ分割し、それぞれのタスクごとに求めた n -gram を適切に切り替えて用いることができれば、性能の高いモデルになると考えられる。SS n -gram は HMM の出力確率を n -gram 確率分布にしたモデルで、学習を行うことで各状態にサブタスクに対応した n -gram を自動的に獲得する。また、その切替えは HMM の状態遷移確率で表現される。よって、SS n -gram はサブタスクごとの n -gram を確率的に切り替える (Stochastic Switching) モデルであるといえる。

以下に SS n -gram の学習アルゴリズムを述べる。

2.2 学習アルゴリズム

SS n -gram は HMM で表現されているため、学習アルゴリズムとして通常の HMM の学習に用いられる Baum-Welch アルゴリズムを応用し、出力確率分布が n -gram 確率となるよう再推定式を定義した。

再推定式を定義する前に、出力確率、前向き変数、後ろ向き変数を定義する。

出力確率

観測系列を $O(= o_0o_1 \cdots o_T)$ としたとき、 n -gram 確率 P を、

$$P(O, t, n) = \begin{cases} P(o_t) & n = 1 \text{ のとき} \\ P(o_t | o_{t-(n-1)} \cdots o_{t-1}) & n \geq 2 \text{ のとき} \end{cases} \quad (1)$$

と定義する。ただし、 n は n -gram における n 個組の n で、たとえば unigram なら $n = 1$ 、bigram なら $n = 2$ 、trigram なら $n = 3$ のようになる。状態 j の出力確率 b_j を、

$$b_j(O, t, n) = P_j(O, t, n) \quad (2)$$

のように拡張した。 $P_j(O, t, n)$ は状態 j の n -gram 確率である。

前向き変数 $\alpha_t(i)$

$\alpha_t(i)$ は以下のように定義できる。ただし、 a_{ij} は状態 i から状態 j への遷移確率、 n_s は HMM の状態

数である。

$$a_{0i} = \pi_i \quad (3)$$

• $t = 0$ のとき

$$\alpha_0(i) = \begin{cases} 1 & i = 0 \text{ のとき} \\ 0 & i = 1 \cdots n_s \text{ のとき} \end{cases} \quad (4)$$

• $t = 1$ のとき

$$\alpha_1(i) = \begin{cases} 0 & i = 0 \text{ のとき} \\ \alpha_0(0)a_{0i}b_i(O, 1, n) & i = 1 \cdots n_s \text{ のとき} \end{cases} \quad (5)$$

• $t \geq 2$ のとき

$$\alpha_t(i) = \begin{cases} 0 & i = 0 \text{ のとき} \\ \sum_{j=1}^{n_s} \alpha_{t-1}(j)a_{ji}b_i(O, t, n) & i = 1 \cdots n_s \text{ のとき} \end{cases} \quad (6)$$

モデルパラメータのセット $\lambda = (\pi, \mathbf{a}, \mathbf{b})$ が与えられたとき、系列 $W = w_0w_1 \cdots w_l$ に対する確率は、

$$P(W = w_0w_1 \cdots w_l | \lambda) = \sum_{i \in \text{終状態集合}} \alpha_l(i) \quad (7)$$

として計算される。

後ろ向き変数 $\beta_t(i)$

$\beta_t(i)$ は以下のように定義できる。

• $t = T$ のとき

$$\beta_T(i) = \begin{cases} 1 & \text{状態 } i \text{ が終状態のとき} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

• $t \geq 1$ のとき

$$\beta_t(i) = \begin{cases} 0 & i = 0 \text{ のとき} \\ \sum_{j=1}^{n_s} a_{ij}b_j(O, t, n)\beta_{t+1}(j) & i = 1 \cdots n_s \text{ のとき} \end{cases} \quad (9)$$

• $t = 0$ のとき

$$\beta_0(i) = \begin{cases} 1 & i = 0 \text{ のとき} \\ 0 & i = 1 \cdots n_s \text{ のとき} \end{cases} \quad (10)$$

パラメータの再推定

上で定義された出力確率 b 、前向き変数 α 、後ろ向き変数 β を使うとパラメータ (初期確率 π 、状態遷移確率 a 、出力確率 b) の再推定式は以下のように定義できる。

$$\bar{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\alpha_0(0)\beta_0(0)} \quad (11)$$

$$\bar{a}_{ij} = \frac{\sum_{t=0}^{T-1} \alpha_t(i) a_{ij} b_j(O, t, n) \beta_{t+1}(j)}{\sum_{t=0}^{T-1} \alpha_t(i) \beta_t(i)} \quad (12)$$

$$\begin{aligned} \bar{b}_j(W, n, n) \\ = \frac{\sum_{i=0}^{n_s} \sum_{t=0}^{T-1} \alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(W, n, n)}{\sum_{i=0}^{n_s} \sum_{t=0}^{T-1} \alpha_t(i) a_{ij} \beta_{t+1}(j) \sum_{q \in V} b_j(W', n, n)} \end{aligned} \quad (13)$$

$$(0 \leq i \leq n_s, 1 \leq j \leq n_s)$$

$$(S = w_1 \cdots w_{n-1}, W = Sw_n, W' = Sq)$$

$$\{w_i | w_i \in V, i = 1 \cdots n\}$$

(V は語彙に含まれるすべての単語の集合)

本論文では、学習アルゴリズムに意図的な制約はいれず、単語ごとに n -gram 統計を切り替えられるモデルとした。しかし、意図的な制約が必要であれば学習アルゴリズムに制約を容易にいれることができる。たとえば、文ごとに n -gram 統計が切り替わる、というような制約が必要であれば、文の区切りでだけ状態遷移を許すような制約を学習アルゴリズムに組み込めばよい。

また、学習された SSn -gram では、通常の n -gram のような直前の n 単語での制約だけでなく、学習によって得られたサブタスクによる制約が反映されることが期待される。

2.3 HMM の初期値の決定

Baum-Welch アルゴリズムでは設定した初期値によって学習後のモデルの性能が大きく変わってしまうため、初期値の決定が重要となる。予備実験として、HMM のトポロジは ergodic、出力確率は bigram、初期値として初期確率 π と状態遷移確率 a をそれぞれ等確率とし、出力確率分布を乱数で与えて実験を行ったところ、発生させた乱数によって学習されたモデルの性能がばらついてしまい、良い性能のモデルは得られなかった。したがって、モデルの学習がうまく行えるように HMM の初期値に意図的な偏りを与える必要がある。

複数のサブタスクに分けることができるタスクでは、話題ごとに文体や語彙がある程度決まってくると考えられる。したがって、話題ごとに n -gram をとることで、それぞれの n -gram に偏りが現れるはずである。実際、話題ごとの n -gram をとり、それらを足し合わせて 1 つのモデルを作ることで性能が上がったという報告がなされている²⁾。そこで SSn -gram の出力確率の初期値を決めるために、学習テキストを話題ごとにクラスタリングし、そのクラスタごとに n -gram 確率分布を求め、それを初期値にする。

テキスト中の単語に着目したとき、話題によって付

表 1 実験条件

Table 1 Experimental condition.

使用コーパス	毎日新聞
形態素解析	RWCP データベース ⁵⁾ に収録されたもの
使用語彙	1991 年 ~ 1994 年の出現頻度が上位のものから選択
HMM	トポロジ : ergodic 出力確率 : bigram

表 2 実験セット

Table 2 Experimental set.

セット名	語彙	学習	評価
small	2,000	1 年分 (1993) 約 25,000 文	1 年分 (1994) 約 30,000 文
large	5,000	3 年分 (1991 ~ 1993) 約 280,000 文	1 年分 (1994) 約 70,000 文

属語の使われ方は変わらず、自立語の使われ方が変わると考えられる。そのため、単語の頻度分布をとると、話題に関係なく使われる単語 (付属語) の頻度にはあまり変わりがなく、そのテキストの話題に固有の単語の頻度が他の話題のテキストに比べて高くなるはずである。そこで、本論文ではテキストを話題で分けるために、クラスタリングの尺度としてクラスタに含まれるテキストの単語の頻度分布の類似度 s_{ij} を用いる。

$$s_{ij} = \sum_l^n \left\{ H_j(l) \log \frac{H_j(l)}{H_i(l)} + H_i(l) \log \frac{H_i(l)}{H_j(l)} \right\} \quad (14)$$

ここで、 H_i, H_j はそれぞれクラスタ i, j の単語の頻度分布である。

学習テキストのクラスタリングは (1) 1 文 1 クラスタとする (2) 類似度が最も近いクラスタどうしをマージする (3) 所望のクラスタ数 (=HMM の状態数) になるまで (2) を繰り返す、として行った。

初期確率や状態遷移確率の初期値は等確率とした。

3. 評価実験

3.1 実験条件

実験条件を表 1 に示す。

学習テキストは、表 2 のように語彙と使用するテキストの量によって小規模なセット (small set) と大規模なセット (large set) に分けた。出力確率の初期値を与えるための学習テキストのクラスタリングには、small set を使用した。large set については、計算機のメモリの制限のためにクラスタリングを行うことができなかったため、small set をクラスタリングして得られた初期値を用い、残りの 3,000 語についてはフロアリング (フロアリング値 : $1.0e-05$) を行って HMM

の出力確率の初期値とした．small set のデータで初期値にある程度の偏りが与えられていると考えられるため，フロアリングによる影響はほとんどないと考えられる．また，比較のために bigram での実験も行った．なお，ここでの実験では bigram, SS n -gram とともにカットオフ，平滑化などの操作はいっさい行っていない．

3.2 クラスタリングによって得られた初期値の結果
SS n -gram の初期値を与える際に行ったクラスタリングの結果について，クラスタ数が 3 のときの結果の一部を以下に示す．

クラスタ 1

日本の政治改革はできそうにない．

それは日本の選挙制度に問題があると思う．

政党も，今までの政党ではもう対応できない．

⋮

クラスタ 2

資金運用によるもので，増加は 2 カ月連続．

2 年連続は円高不況から 11 年ぶり．

現在 3 期目．

⋮

クラスタ 3

喪主は長男博氏．

喪主は長男文夫氏．

喪主は妻良子さん．

⋮

クラスタ 3 に含まれる文はすべておくやみの文であった．クラスタ 1 とクラスタ 2 でははっきりとした違いは分らなかったが，クラスタ 2 では数字が含まれている文が多かった．

このように，話題に固有の単語によってある程度のクラス分けができており，意図した結果が得られている．

3.3 perplexity による評価

平滑化などの操作を行っていないため，構築されたモデルで確率が 0 になる文章が評価セット中に存在する．SS n -gram のトポロジを ergodic としたことで，SS n -gram と bigram で確率が 0 になる文章はまったく同じになった．確率が 0 になる文章が評価セット中にどのくらいの割合で存在するのかを表す指標として文カパー率を式 (15) で定義する．

$$\text{文カパー率} = \frac{\text{確率が 0 にならない文章数}}{\text{評価文章数}} \times 100[\%] \quad (15)$$

テストセットに対する文カパー率は，small set で 48%，large set で 56%であった．また，テストセット

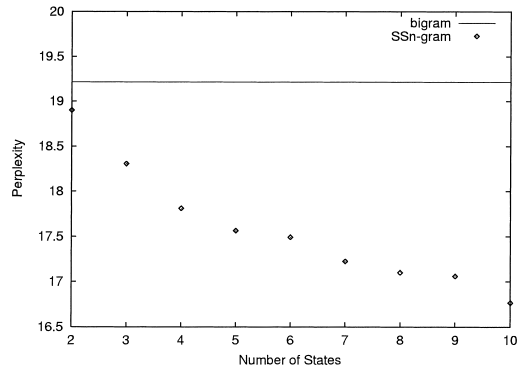


図 1 パープレキシティによる評価 (small set)

Fig. 1 Test set perplexity (small set).

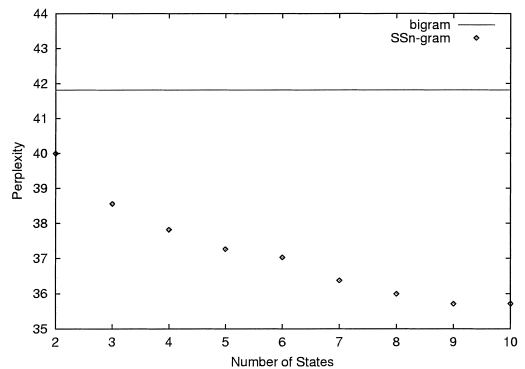


図 2 パープレキシティによる評価 (large set)

Fig. 2 Test set perplexity (large set).

パープレキシティは確率が 0 になる文章では計算できないため，評価セットで確率が 0 にならなかった文章で計算した．評価セットのすべての文章でパープレキシティを計算することはできなかったが，small set で約 15,000 文，large set で約 40,000 文程度でパープレキシティを計算できているのである程度の精度が得られていると考えられる．

図 1 に small set での結果を，図 2 に large set での結果を示す．図 1，図 2 から small set, large set とともに状態数が増えることでパープレキシティが減少していく傾向があることが分かる．small set では 10 状態のモデルで，large set では 9 状態のモデルでパープレキシティが一番低くなった．small set, large set とともに状態数が増えるに従い，パープレキシティの変化量が小さくなっている．このため，これ以上モデルの状態数を増やしても性能にはあまり影響がないように思われる．図 1 より small set では 10 状態のモデルで約 13%，図 2 より large set では 9 状態のモデルで約 15%，bigram と比べ SS n -gram の方がパープレキシティが低い．学習セットの規模や状態数によらず

SS n -gram の方が bigram に比べて低いパープレキシティが得られているのが分かる。

4. SS n -gram への削除補間法の適用

SS n -gram は n -gram と同様に学習テキストの量が十分でない場合、テストセットに対する文カバー率が十分でないため、なんらかの形でモデルの平滑化を行う必要がある。そこで SS n -gram のための平滑化法を提案する。

n -gram で行われる平滑化の方法の 1 つとして制約の弱いモデルとの重み付きの和がよく用いられている。たとえば、bigram でモデルの構築を行った場合、より制約の弱い unigram 確率との重み付きの和でモデルの平滑化を行う。

SS n -gram でもこれを応用してモデルの平滑化を行う。

4.1 SS n -gram の平滑化法

SS n -gram は出力確率として n -gram モデルを持つので、制約の弱いモデルとして $(n-1)$ -gram モデルを考える。平滑化に使用する $(n-1)$ -gram モデルの計算法として、すべての学習テキストから $(n-1)$ -gram モデルを学習する方法が考えられる。このようにして計算された $(n-1)$ -gram を SS n -gram のすべての状態で共通な $(n-1)$ -gram として平滑化に用いる。しかし、学習された SS n -gram は各状態の出力確率が異なっているため、共通な $(n-1)$ -gram で平滑化してしまうと各状態の出力確率の異なりが平滑化されてしまい、平滑化後の性能が悪くなると思われる。そのため SS n -gram の各状態で独立な $(n-1)$ -gram モデルを計算した方がよい性能が得られると考えられる。そこで、本論文では SS n -gram の各状態の n -gram 確率から状態ごとに独立な $(n-1)$ -gram 確率 \overline{P}_s を計算する。

$$\overline{P}_s(W, n, n-1) = \frac{\sum_{p \in V} P_s(W, n, n)}{\sum_{p \in V} \sum_{q \in V} P_s(W', n, n)} \quad (16)$$

$$(S = w_2 \cdots w_{n-1}, W = pSw_n, W' = pSq)$$

$$\{w_i | w_i \in V, i = 2 \cdots n\}$$

(V は語彙に含まれるすべての単語の集合)

ここで得られた状態 s の $(n-1)$ -gram 確率 $\overline{P}_s(W, n, n-1)$ を用いて、各状態の出力確率は以下のように平滑化される。

$$\hat{P}_s(W, n, n) = \lambda_s(n) P_s(W, n, n) + \sum_{m=1}^{n-1} \lambda_s(m) \overline{P}_s(W, n, m) \quad (17)$$

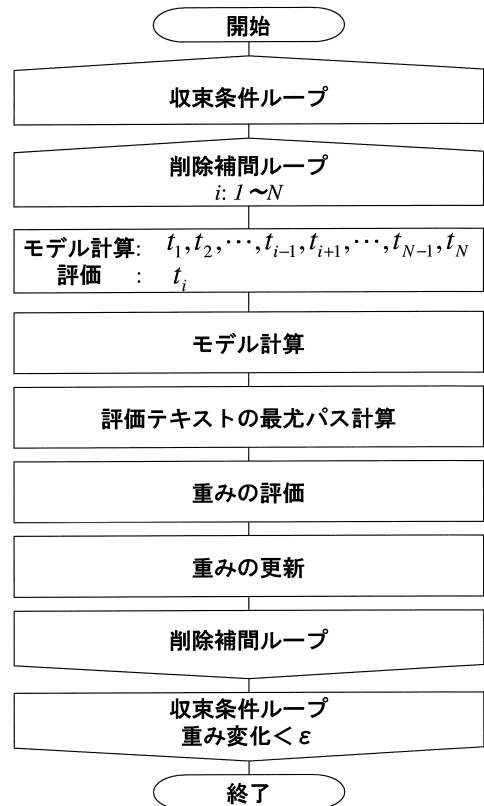


図 3 SS n -gram の削除補間に基づく平滑化の流れ

Fig. 3 Flowchart of interpolation algorithm for SS n -gram based on deleted interpolation.

$$\sum_{m=1}^n \lambda_s(m) = 1 \quad (18)$$

$$(s = 1, \dots, n_s)$$

ただし、 \hat{P}_s は平滑化後の状態 s の出力確率分布、 $\lambda_s(m)$ は状態 s の n -gram 確率 ($m = 1, \dots, n$) の重みである。

この重み $\lambda_s(n)$ の推定は削除補間法⁶⁾を用いて行った。削除補間法を用いる際の重み λ の評価は、評価サンプルの Viterbi をとり、その最尤パス $v(t)$, ($t = 1, \dots, l$) 上で行った。ただし、 $v(t)$ は最尤パス上での時刻 t にいる状態、 l は評価サンプルの長さである。したがって削除補間法を適用した際の評価式は、

$$\hat{P}_{v(t)}(W, n, n) = \lambda_{v(t)}(n) P_{v(t)}(W, n, n) + \sum_{m=1}^{n-1} \lambda_{v(t)}(m) \overline{P}_{v(t)}(W, n, m) \quad (19)$$

$$(t = 1, \dots, l)$$

となる。

平滑化法のフローチャートは図 3 のようになる。

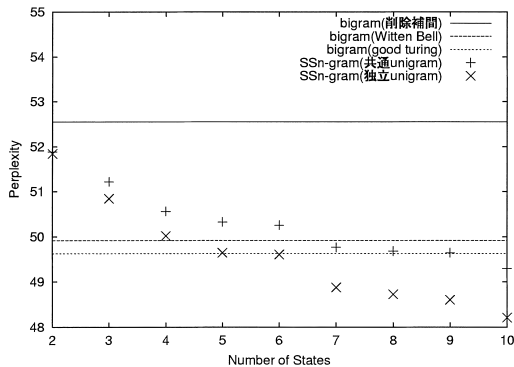


図4 パープレキシティによる評価 (small set)

Fig. 4 Test set perplexity (small set).

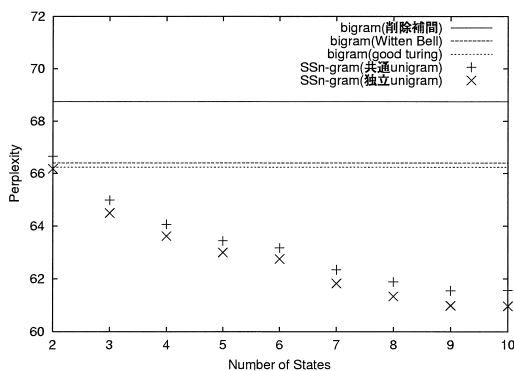


図5 パープレキシティによる評価 (large set)

Fig. 5 Test set perplexity (large set).

4.2 評価実験

実験に使用したデータは表2と同じである。3章で構築したモデルに対し、削除補間法に基づく平滑化を行った。SSn-gramで各状態ごとに独立な unigram モデルを計算する方法の有効性を調べるために、すべての学習テキストから計算した unigram を各状態で共通に用いる方法についても実験を行った。

また、比較のために、削除補間法で平滑化された bigram と、一般によく用いられている good turing⁷⁾, Witten Bell⁸⁾ の discounting による back-off bigram による評価も行った。good turing, Witten Bell の discounting による back-off bigram については palmkit (palmkit, version 1.0.17⁹⁾) を使用してモデルの構築および評価を行った。

small set での結果を図4に、large set での結果を図5に示す。

図4, 図5中で「SSn-gram(共通 unigram)」はすべての状態で共通の unigram モデルを用いたもの、「SSn-gram(独立 unigram)」は unigram 確率を各状態ごとに式(17)で計算したものである。また、それ

ぞれの bigram の平滑化法は図中のカッコ内に示したとおりである。平滑化を行ったことですべてのモデルで文カバール率は 100%となった。

図4, 図5から平滑化後は small set, large set ともに 10 状態のモデルでパープレキシティが一番低くなっていることが分かる。SSn-gram(共通 unigram), SSn-gram(独立 unigram), それぞれの結果の比較から、各状態で unigram 確率を計算することで unigram 確率に偏りが反映されたため、SSn-gram(独立 unigram)の方が set によらず低いパープレキシティが得られたと考えられる。SSn-gram は同様に削除補間法を適用した bigram(削除補間)と比べ低いパープレキシティが得られた。10 状態の SSn-gram(独立 unigram) は bigram(削除補間)よりも small set では約 6%, large set では約 13%低いパープレキシティが得られた。また、平滑化された bigram の中で最もパープレキシティの低かった bigram(good turing) と比べても SSn-gram は低いパープレキシティを示した。10 状態の SSn-gram(独立 unigram) は bigram(good turing) よりも small set では約 3%, large set では約 8%低いパープレキシティが得られた。

5. おわりに

HMM を用いた複数 n -gram モデルによる言語モデル SSn-gram を提案した。SSn-gram は HMM の出力確率を n -gram 確率分布にしたモデルで、学習を行うことで各状態のサブタスクに対応した n -gram を自動的に獲得するモデルである。HMM の初期値の設定をうまく行うことで、新聞記事をタスクとした実験で bigram に比べて large set で約 15%, 低いパープレキシティを得ることができた。また、SSn-gram に削除補間法を適用し、各状態で unigram 確率を計算することで、平滑化後も低いパープレキシティが得られた。SSn-gram は同様に削除補間法で平滑化された bigram と比べ large set による実験で約 13%低いパープレキシティが得られた。また、平滑化された bigram の中で最も低いパープレキシティを示した good turing discounting による back-off bigram と比べ large set による実験では約 8%低いパープレキシティが得られた。今後は、trigram での実験や学習の条件などの検討を行う予定である。なお、本論文では提案手法の評価に「CD-毎日新聞 91~94 年版」を用いたことを付記する。

参 考 文 献

- 1) 伊藤彰則, 好田正紀: N-gram 出現回数の混合によるタスク適応の性能解析, 信学論, Vol.J83-D-II, No.11, pp.2418-2427 (2000).
- 2) Iyer, R. and Osterndorf, M.: Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models, *ICSLP96*, Vol.I, pp.236-239 (1996).
- 3) 清水 徹, 大野晃生, 黒岩真吾, 樋口宜男: 文クラスタ混合分布 N-gram の検討, 信学技報, Vol.SP98, No.101, pp.41-48 (1998).
- 4) 阿部芳春, 伍井啓恭, 丸田裕三, 中島邦男: 混合言語モデル作成のためのコーパスクラスタの分割の検討, 音講論, Vol.I, No.3-P-17, pp.197-198 (2001).
- 5) データベースワークショップテキストグループ: テキストデータベース報告書, 技術研究組合新情報処理開発機構 (1995).
- 6) Jelinek, F. and Mercer, R.: Interpolated estimation of Markov source parameters from sparse data, *Pattern Recognition in Practice*, pp.381-397 (1980).
- 7) Katz, S.M.: Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer, *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.ASSP-35, No.3, p.400 (1987).
- 8) Witten, I. and Bell, T.: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression, *IEEE Trans. Information Theory*, Vol.37, No.4, pp.1085-1094 (1991).
- 9) 伊藤彰則, 好田正紀: 単語およびクラス n -gram 作成のためのツールキット, 信学技法, Vol.SP2000, No.106, pp.67-72 (2000).
- 10) 松岡, 大附, 森, 古井, 白井: 新聞記事データベースを用いた大語い連続音声認識, 信学論, Vol.J79-D-II, No.12, pp.2125-2131 (1996).
- 11) 長野 雄, 鈴木基之, 牧野正三: 出力確率を bigram とする HMM による言語モデルの検討, 音講論, Vol.I, No.2-1-18, pp.83-84 (2001).
- 12) 中川聖一: 確率モデルによる音声認識, 電子情報通信学会 (1988).
- 13) Rabiner, L., Juang, B.-H., 古井貞熙 (監訳):

音声認識の基礎(上,下), NTT アドバンステックテクノロジー株式会社 (1995).

14) 鹿野清宏, 中村 哲, 伊勢史郎: 音声・音情報のデジタル信号処理, 昭晃堂 (1997).

15) 田中穂積: 自然言語処理基礎と応用, 電子情報通信学会 (1999).

(平成 13 年 11 月 21 日受付)

(平成 14 年 4 月 16 日採録)



長野 雄

平成 9 年東北大学工学部電子工学科卒業。平成 12 年同大学大学院情報科学研究科情報基礎科学専攻博士前期課程修了。現在, 同大学院博士後期課程在学中。音声認識の研究に従事。日本音響学会会員。



鈴木 基之

平成 5 年東北大学工学部情報工学科卒業。平成 7 年同大学大学院工学研究科電気・通信工学専攻博士前期課程修了。同年博士後期課程進学。平成 8 年同課程退学。同年同大学大型計算機センター助手。平成 13 年同大学情報シナジーセンター助手。同年同大学大学院工学研究科助手。音声認識の研究に従事。電子情報通信学会, 日本音響学会各会員。



牧野 正三 (正会員)

昭和 44 年東北大学工学部電子工学科卒業。昭和 49 年同大学大学院博士課程修了。同年同大学電気通信研究所助手。昭和 56 年同大学応用情報学研究センター助手。同助教を経て, 現在, 同大学大学院工学研究科教授。工学博士。昭和 59~61 年アメリカ合衆国 STL 客員研究員。言語情報を利用した音声認識の研究, 音声データベース, 音声信号処理, 画像情報処理, 文字認識の研究に従事。電子情報通信学会, 日本音響学会, IEEE 各会員。