ヒューマンインタフェース 97－26
音声言語情報処理 40－26
（2002．2．2）

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‡
† 東北大学大学院情報科学研究科
‡ 東北大学大学院工学研究科
〒980-8579 仙台市青葉区荒巻字青葉05
東北大学大学院工学研究科 電気・情報系 牧野研究室
TEL: (022)217-7085

あらまし 一般にn-gramによる言語モデルでは一つの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．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にサブタスクに対応したn-gram統計を自動的に獲得する．SSn-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，SSn-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11%パープレキシティを下げることができた．

キーワード 言語モデル, n-gram, HMM

Construction method of language model using stochastic switching n-gram based on HMM

Takeshi NAGANO†, Motoyuki SUZUKI‡, Shozo MAKINO‡
† Graduate School of Information Sciences, Tohoku University
‡ Graduate School of Engineering, Tohoku University
Aoba, Aramaki, Aoba-ku, Sendai-shi, 980-8579 Japan

Abstract In traditional speech recognition systems, a single kind of n-gram is used for n-gram language model in speech recognition system. If a task can divide into several small sub tasks, utilization of several kinds of n-gram can give 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 SSn-gram, each state corresponds to one kind of n-gram. When bigram is used as output probability, the perplexity with SSn-gram is reduced by 15% comparing to that with ordinary bigram. After the deleted interpolation method is applied, the perplexity is reduced by 11%.

Keywords language model, n-gram, HMM

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一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ことが知られている [1].

では，一つのタスクに対して単一の n-gram で十分か，というと必ずしもそうではない．例えばニュース番組タスクを考えた時，アナウンサー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．例に挙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（SSn-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．

一つのタスクをいくつかのサブタスクに分け複数の n-gram を求めるモデルが既にいくつか提案され [2, 3, 4]，一つの 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，本報告では SSn-gram とその学習アルゴリズムについて提案し，新聞記事をタスクとした評価実験について報告する．

2 SSn-gram モデル

2.1 SSn-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．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，その切り替えは HMM の状態遷移確率で表現される．よって，SSn-gram はサブタスクごとの n-gram を確率的に切り替える（Stochastic Switching）モデルであると言える．

以下に SSn-gram の学習アルゴリズムを述べる．

2.2 学習アルゴリズム

SSn-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 $O(= o_0 o_1 \cdots o_T)$ としたとき，出力確率 b_j を，

$$b_j(o_t) \rightarrow b_j(o_t|o_{t-(N-1)} \cdots o_{t-1})$$

のように拡張した．N は n-gram における N 個組の N 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．

$$\alpha_0(0) = 1, \alpha_0(i) = 0 \quad (1)$$

$$a_{0i} = \pi_i \quad (1 \leq i \leq n_s) \quad (2)$$

$$\alpha_t(j) = \sum_{i=0}^{n_s} \alpha_{t-1}(i) a_{ij} b_j(o_t|o_{t-(N-1)} \cdots o_{t-1}) \quad (1 \leq t \leq T), (1 \leq j \leq n_s) \quad (3)$$

$$P(O = o_0 o_1 \cdots o_T|\lambda) = \sum_{i=1}^{n_s} \alpha_T(i) \quad (4)$$

後ろ向き変数 $\beta_t(i)$　$\beta_t(i)$ は以下のように定義できる．

$$\beta_T(i) = 1 \quad (1 \leq i \leq n_s) \tag{5}$$

$$\beta_t(i) = \sum_{j=1}^{n_s} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}|o_{t-(N-2)} \cdots o_t) \cdot (t = T-1, T-2, \cdots, 2, 1, 0) \quad (0 \leq i \leq n_s) \tag{6}$$

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．

$$\overline{\pi_i} = \frac{\alpha_1(i)\beta_1(i)}{\alpha_0(0)\beta_0(0)} \tag{7}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T} \alpha_{t-1}(i) a_{ij} b_j(o_t|o_{t-(N-1)} \cdots o_{t-1}) \beta_t(j)}{\sum_{t=0}^{T-1} \alpha_t(i)\beta_t(i)} \tag{8}$$

$$\overline{b_j}(w_k|w_{l-(N-2)} \cdots w_l) = \frac{\sum_{i=0}^{n_s} \sum_{t=1}^{T} \alpha_{t-1}(i) a_{ij} \beta_t(j) b_j(w_k|w_{l-(N-2)} \cdots w_l)}{\sum_{i=0}^{n_s} \sum_{t=1}^{T} \sum_{k=1}^{n_v} \alpha_{t-1}(i) a_{ij} \beta_t(j) b_j(w_k|w_{l-(N-2)} \cdots w_l)} \tag{9}$$

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

ただし，n_v は語彙数，w_* は語彙 n_v に含まれる任意の 1 単語である．

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 のトポロジーは 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．従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．従って話題ごとに n-gram をとることで，それぞれの n-gram に偏りが表れるはずである．実際，話題ごとの n-gram をとり，それらを足し合わせて一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，それを初期値にした．

テキスト中の単語に着目したとき，話題によって付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} を用いた．

$$s_{ij} = \sum_{l}^{n} \left\{ H_i(l) \log \frac{H_j(l)}{H_i(l)} + H_j(l) \log \frac{H_i(l)}{H_j(l)} \right\} \tag{10}$$

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同士をマージする，(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．語彙 5,000 のセットについては，語彙 2,000 のセットで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を行って HMM の出力確率の初期値とした．また，比較のために bigram での実験も行った．

表 1: 実験条件

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	トポロジー:ergodic 出力確率:bigram

表 2: 実験セット

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

3.2 クラスタリングによって得られた初期値の結果

SSn-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 の時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 1 1 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悔やみの文であった．クラスタ 1 とクラスタ 2 でははっきりとした違いはわからなかったが，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.

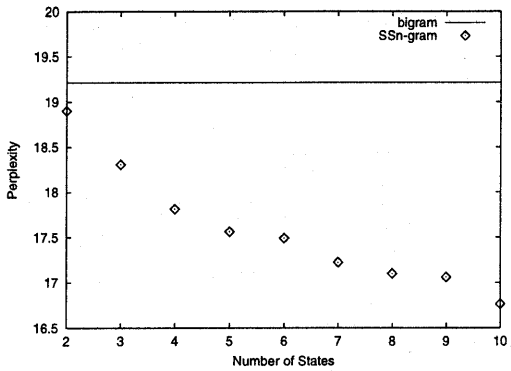

図 1: パープレキシティによる評価（カバー率 48%, small set）

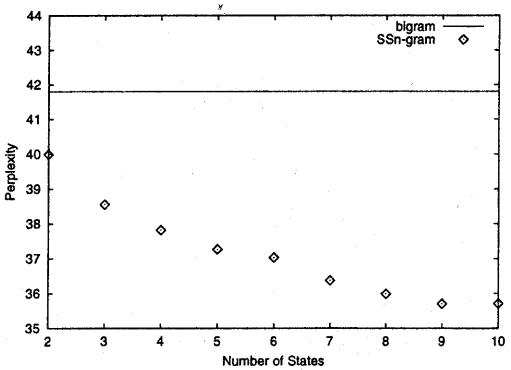

図 2: パープレキシティによる評価（カバー率 56%, large set）

3.3 perplexity による評価

SSn-gram のトポロジーを ergodic としたことで，SSn-gram と bigram の文カバー率は同じになった．テストセットに対する文カバー率は，small set で 48%, large set で 56%であった．評価はテストセットパープレキシティで行った．テストセットパープレキシティは，モデルで受理されたテキストを基に計算した.

図 1 に small set での結果を，図 2 に large set での結果を示す． 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 10 状態のモデルで約 15%，bigram と比べ SSn-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 SSn-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わかる．特に，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わかる．

4 SSn-gram への削除補間法の適用

SSn-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 SSn-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一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．例えば，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．

SSn-gram でもこれを応用してモデルの平滑化を行う．

4.1 SSn-gram の平滑化法

SSn-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 SSn-gram の全部の状態で共通な (n-1)-gram として平滑化に用いる．しかし，学習された SSn-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 SSn-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 SSn-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．

$$\overline{P_s}(w|w_{l-(N-2)}\cdots w_l) = \frac{\sum_{n=1}^{n_v} P_s(w|w_n w_{l-(N-3)}\cdots w_l)}{\sum_{m=1}^{n_v}\sum_{n=1}^{n_v} P_s(w_m|w_n w_{l-(N-3)}\cdots w_l)} \quad (11)$$

ただし，N は SSn-gram の各状態の出力確率の n-gram の N で，例えば bigram なら 2，trigram なら 3 のようになる．ここで得られた状態 s の (n-1)-gram 確率 $\overline{P_s}(w|w_{l-(n-2)}\cdots w_l), (n = 1,\cdots,N)$ 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．

$$\hat{P}_s(w|w_{l-(N-2)}\cdots w_l) = \lambda_s(N)P_s(w|w_{l-(N-2)}\cdots w_l) + \sum_{n=1}^{N-1}\lambda_s(n)\overline{P_s}(w|w_{l-(n-2)}\cdots w_l) \quad (12)$$

$$\sum_{n=1}^{N}\lambda_s(n) = 1 \quad (13)$$

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

ただし，$\hat{P}_s$ は平滑化後の状態 s の出力確率分布，$\lambda_s(n)$ は状態 s の n-gram 確率 $(n = 1,\cdots,N)$ 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 $v(t), (t = 1,\cdots,l)$ 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，

$$\hat{P}_{v(t)}(w|w_{l-(N-2)}\cdots w_l) = \lambda_{v(t)}(N)P_{v(t)}(w|w_{l-(N-2)}\cdots w_l) + \sum_{n=1}^{N-1}\lambda_{v(t)}(n)\hat{P}_{v(t)}(w|w_{l-(n-2)}\cdots w_l) \quad (14)$$

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

となる．

4.2 評価実験

評価は表 2 の small set，large set の両方で行った．3 節で構築したモデルに対し，平滑化を行った．また，比較のために削除補間法を用いて平滑化した bigram での実験も行った．また，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．それぞれの平滑化法を用いて平滑化した結果を示す．図 3 に small set での結果を，図 4 に large set での結果を示す．

図 3，図 4 中の「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 (12) で計算したものである．

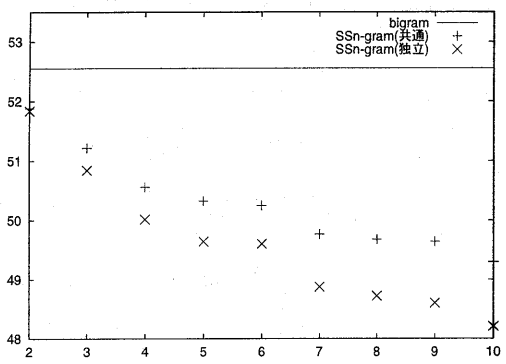

図 3: パープレキシティによる評価（small set, 補間後）

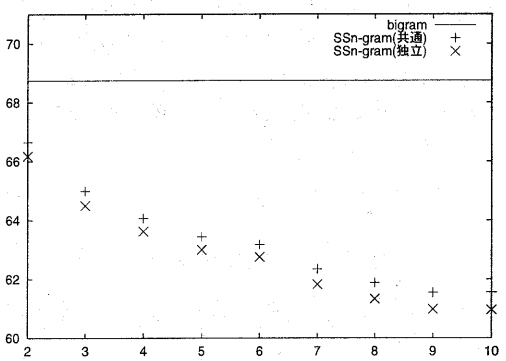

図 4: パープレキシティによる評価（large set, 補間後）

図3，図4から，SSn-gramは平滑化bigramと比べ平滑化後も低いパープレキシティを示した．small setでは図3からSSn-gram＋共通unigramでは約5%，SSn-gram＋独立unigramでは約6%，平滑化したbigramに比べてパープレキシティが低くなった．また，large setでは図4からSSn-gram＋共通unigramでは約10%，SSn-gram＋独立unigramでは約11%，平滑化したbigramに比べてパープレキシティが低くなった．また，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の方が低いパープレキシティが得られたと考えられる．

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small setで約13%，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，削除補間法で平滑化されたbigramと比べてsmall setで約6%，large setで約11%低いパープレキシティを得た．今後は，音声認識エンジンと組み合わせることで，SSn-gramが実際の音声認識でどの程度有効であるかを実験で確認する．

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] R.Iyer and M.Osterndorf: Modeling Long Distance Dependence in Languge: Topic Mixtures vs. Dynamic Cache Models, *ICSLP96*, Vol. I, pp. 236–239 (1996).

[3] 清水徹, 大野晃生, 黒岩眞吾, 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] F.Jelinek and R.Mercer: Interpolated estimation of Markov source parameters from sparse data, *Pattern Recognition in Practice*, pp. 381–397 (1980).