音声言語情報処理 34－32
(2000. 12. 22)

単語およびクラスn-gram作成のためのツールキット

伊藤 彰則，好田 正紀

〒992-8510 山形大学工学部
米沢市城南 4-3-16
TEL: 0238-25-3369
E-mail: {aito,kohda}@ei5sun.yz.yamagata-u.ac.jp

あらまし 単語n-gramとクラスn-gramが作成可能なツールキットを作成した．このツールキットは，CMU-Cambridge SLM Toolkitとコマンドレベルで互換性があり，ARPA言語モデルを作成することができる．また，n-gram出現回数の混合による言語モデルや，線形結合による言語モデルの組み合わせをサポートしている．言語モデルの組み合わせはAPIレベルでサポートされているので，本ツールキットのライブラリを利用すれば，組み合わせ言語モデルを使ったシステムを手軽に作ることができる．本ツールを使った実験として，毎日新聞6年分を用いた言語モデルの評価実験を行った．単語モデルと品詞モデルのさまざまな組み合わせを比較した結果，単語trigramに品詞trigramを組み合わせることで，パープレキシティが若干改善できることがわかった．

キーワード 単語n-gram, 品詞n-gram, 言語モデルツールキット，パープレキシティ

Statistical Language Model Toolkit for Word and Class N-gram

Akinori Ito and Masaki Kohda

Faculty of Engineering, Yamagata University
4-3-16 Jonan, Yonezawa 992-8510
TEL: 0238-26-3369
E-mail: {aito,kohda}@ei5sun.yz.yamagata-u.ac.jp

Abstract This paper describes a statistical language model toolkit for word and class-based n-gram. This toolkit has command-level compatibility with CMU-Cambridge SLM Toolkit, and supports ARPA-style language model. Furthermore, the toolkit supports class n-gram and n-gram count mixture as well as combined language model using linear interpolation. As the language model combination is supported within the API level, the SLM library in this toolkit enables any tool to exploit the LM combination. To demonstrate the potential of the toolkit, several language models are created from six-year Mainichi Shimbun database. We evaluated verious combination of word n-gram and POS n-gram, and we found that the combination of word trigram and POS trigram reasonably improves the perplexity.

key words word n-gram, class n-gram, statistical language model toolkit, perplexity

1 はじめに

N-gram は現在最も広く用いられている言語モデルであり，連続音声認識はもちろん，その他の分野でも頻繁に用いられている．N-gram についてはさまざまな改良がこれまで提案されてきているが，多くのシステムで用いられているのは，最も基本的なバックオフ N-gram やヘルドアウト N-gram のみである．その他の「改良された」モデルは，それが従来法よりも優れていることを示す実験に用いられる他には，あまり使われることはない．その理由の一つとして，一般に広く利用することのできるコードが公開されていないことが挙げられよう．新たな手法の提案と同時に，その手法を用いたプログラムのソースコードが公開されれば，他の機関でその手法の検証を行うのは容易になるであろう．音声処理やデコーダについては，近年オープンソースとして公開されたものが多い[2, 1]．また，音響モデル作成については，HTK がフリーソフトウェアになった[3]ことから，今後さらにオープンソース化が進むと思われる．

CMU-Cambridge SLM Toolkit[4, 5](CMU-Cam tk) は，統計的言語モデル作成のためのフリーなツールキットとして広く利用されている．しかし，統計的言語モデルの研究を進めるにあたって，CMU-Cam tk をそのまま利用するには問題が出てきた．そのため，上記ツールキットとコマンドレベルで上位互換性を持つツールキットを作成した[1]．本稿では，その概要と，いくつかの実験結果について述べる．

2 CMU-Cambridge SLM toolkit とその問題点

CMU-Cambridge SLM toolkit は，バックオフ N-gram モデルが簡単に作成できるツールキット (複数のコマンド集) であり，ケンブリッジ大 (当時) の Philip Clarkson と，カーネギーメロン大の Ronald Rosenfeld によって書かれた．CMU toolkit の version 1 は Ronald Rosenfeld 一人によって書かれており，CMU-Cambridge SLM toolkit は version 2 にあたる．このツールキットの特徴は，

- 多くのコマンド群からなる
- 容易に N-gram が構築できる
- 高速な処理，圧縮ファイルのサポート
- 任意の N についての N-gram モデルのサポート
- 4 種類のバックオフ

という特徴を持つ．このツールキットを使うことで，以下のような作業を簡単に行なうことができる．

- 単語頻度リストの作成
- 語彙リストの作成
- バックオフ N-gram 言語モデルの作成
- バックオフ N-gram 言語モデルの評価

CMU-Cam tk は非常に簡単に使うことができた反面，次のような問題点がある．

- 作成できる言語モデルは単語ベースのバックオフ N-gram のみであり，バックオフは 4 種類のいずれかから選ぶだけである．
- 言語モデルファイルの種類 (binary 形式と ARPA 形式) によって言語モデルを利用するための API が異なっている．そのため，言語モデルの形式の違いがアプリケーション側まで影響してしまう．
- ツールキット内の言語モデルのデータ構造が特殊であり，このプログラムをベースに拡張・修正をすることが難しい．
- 利用条件が研究目的に限られている．

また，主要な作者の Philip Clarkson がケンブリッジ大を卒業してしまったため，今後ツールキットの積極的なサポートができない点も問題点として挙げられる．これらの点から，新しい基本ソフトウェアとしての言語モデル作成ツールキットが望まれていた．最近フリーになった言語モデルのツールキットとして SRILM [6]があり，こちらは CMU-Cam tk よりも多くの言語モデルをサポートしているため検討の余地があったが，今回は独自に実装を行うことにした．

3 開発方針

今回のツールキット開発方針として，次の点に考慮した．

- CMU-Cam tk とコマンドレベルで互換性を有すること．CMU-Cam tk とコマンドが微妙に異なるのは混乱の元なので，できるだけコマンドレベルでは CMU-Cam tk と互換にして，利用者が意識せずに移行できるようにする．
- 単語 N-gram だけでなく，クラス N-gram もサポートする．対話音声認識等では，クラス N-gram も重要になってくると思われる．そこで，各種のクラス N-gram をサポートする．
- 言語モデルの形式・種類に依存しない API．言語モデルの形式だけでなく，N-gram の種類をも API レベルで隠蔽する．これによって，言語モデルの種類をアプリケーション側で意識することなく，モデルのファイルを交換すれば自動的にそのモデルを利用するようになる．
- 単純なデータ構造．CMU-Cam tk では，省メモリのために複雑なデータ構造を用いている (例えば，n-gram の出現頻度の代わりに，n-gram の出現頻度テーブルの添字を保持するなど)．しかし，そのために拡張が難しくなっている面もある．現在のコンピュータのメモリ容量から考えて，省メモリのための工夫をするよりも，単純なデータ構造によるわかりやすいプログラムにした方が，研究用ツールとしては良いと考えられる．

[1] 本稿で述べるツールキットは，ftp://ei5nazha.yz.yamagata-u.ac.jp/speech/LM/からダウンロード可能になる予定である．

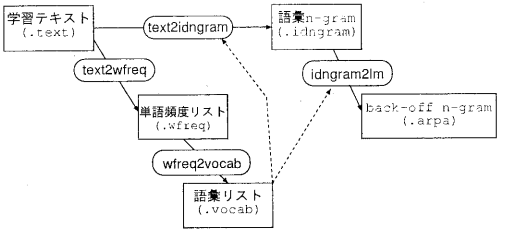

図 1: 単語モデル作成手順

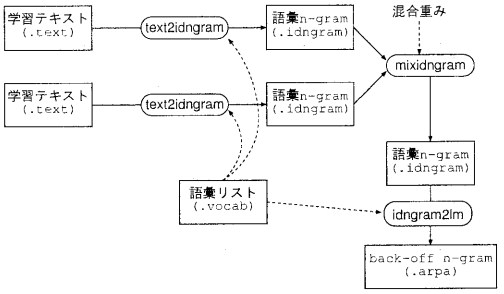

図 2: 混合モデル作成手順

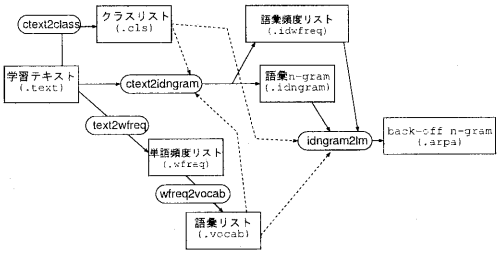

図 3: クラスモデル作成手順

ツールキットのコマンドはほとんど C 言語で実装されており，一部 (`text2class`, `text2idtext`) は ruby[7] スクリプトである．

4 言語モデルの作成

前述の通り，本ツールキットは CMU-Cam tk とコマンドレベルで互換性があるので，言語モデル作成方法は CMU-Cam tk とほとんど同じである．ただし，言語モデルの形式は ARPA 形式だけしかサポートしていない．

CMU-Cam tk にあるコマンドのうち，現時点で未実装なコマンドは，`text2wngram`, `ngram2mgram`, `wngram2idngram`, `idngram2stats`, `binlm2arpa` の 5 つである．逆に，オリジナルに追加されたコマンドとして，次のようなものがある．

- `mixidngram`: 複数の idngram を重み付きで加える．
- `ctext2class`: テキスト中から単語クラスを抽出して，その一覧を作成する．
- `text2idtext`: 単語列を単語番号列に変換する．
- `ctext2idngram`: テキストからクラス言語モデルのための ID n-gram を作成する．これについては後述する．

4.1 単語モデルの作成

基本的な言語モデル作成手順は CMU-Cam tk と同じである．これを図 1 に示す．単語モデル作成で追加された機能として，バックオフのないモデルを作るというものが挙げられる．この機能で作られたモデルは，単体で使用するのではなく，後述する「組みあわせ言語モデル」で利用される．

図 2 は，n-gram カウント混合による言語モデル[8]作成手順である．複数のテキストからそれぞれ ID n-gram を作成し，その出現回数に重みを付けて混合することで，混合モデルを作成することができる．ただし，出現回数に重みをかけることにより，バックオフ係数が影響を受けるので注意が必要である．後述するヘルドアウト法による平滑化を用いれば，この問題は生じない．

4.2 クラスモデルの作成

新しく追加された機能にクラス言語モデルがある．クラス言語モデルは図 3 のような手順で作成される．単語モデルと異なるのは，

- クラスリスト (.cls) を参照する．クラスリストはあらかじめ (人手等により) 用意しておくか，`ctext2class` コマンドによって生成しておく．
- .text ファイルにおいて，各単語は「単語表記+クラス名」という形をしている必要がある．単語表記とクラス名のデリミタである “+” として，別な文字を指定することもできる．また，クラス名は最後のデリミタから後とみなされるので，「単語表記+読み+クラス名」という表記でも問題ない．
- `ctext2idngram` コマンドを用い，text ファイルからクラスレベルの idngram と各単語の unigram 統計を算出する．
- 生成された各ファイルから言語モデルを生成するには，単語 N-gram と同じく `idngram2lm` コマンドを利用する．
- 生成された言語モデルファイルは，クラス文字列の ARPA 言語モデルに，各クラスから単語を生成する情報を追加したものになる．

クラスモデルでは，未知語と未知クラスの扱いについて次のような区別をすることができる．

- 未知クラスは存在しない．未知語はクラスごとに 1 個．(デフォルト)
- 指定されたクラスのほかに，「未知クラス」が 1 個存在する．未知語はクラスごとに 1 個．

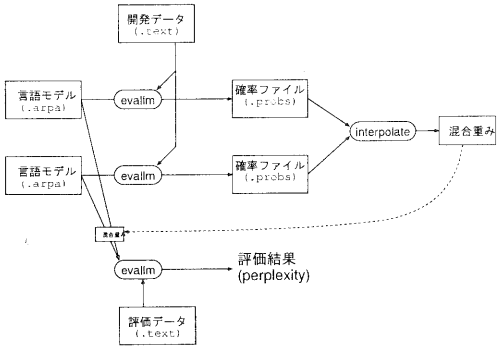

図 4: 言語モデルの組み合わせ手順

- 指定されたクラスのほかに,「未知クラス」が 1 個存在する．未知語は全体で 1 個．

4.3 組みあわせモデルの作成

上記のモデルをどれか一つだけ使うのではなく，複数を組みあわせて使うと，性能が向上することが知られている．例えば，単語モデルと品詞モデルを組みあわせる場合，線形補間，対数線形補間，最大エントロピー法などの方法が使われる．このうち，本ツールキットでは線形補間のみサポートしている．Trigram モデルを例に取ると，単語と品詞モデルを線形補間したモデルの確率 P' は

$$P'(w_i|w_{i-2}w_{i-1}) = \lambda P(w_i|w_{i-2}w_{i-1}) + (1-\lambda)P(w_i|c_i)P(c_i|c_{i-2}c_{i-1}) \quad (1)$$

と表される．ただし，c_k は単語 w_k のクラス，$0 \leq \lambda \leq 1$ である．この λ は，EM アルゴリズムによって求めることができる．

複数のモデルを組みあわせる手順を図 4 に示す．ここでの組みあわせは線形結合のみなので，結合重みを推定する手順が中心になる．まず，学習データと評価データのどちらからも独立な開発データを用意する．次に，`evallm` を使って，それぞれのモデルで開発データを評価したときの，単語ごとの確率ストリームを生成する．さらに，この確率ストリームを `interpolate` コマンドに入力することで，それぞれの確率ストリームに対する最適な重みが計算される．計算された重みを使って，再び `evallm` を使うことにより，組みあわせモデルを評価することができる．

5 言語モデルの評価

言語モデルの評価には `evallm` コマンドを使う．このツールキットの最大の特徴は,「組みあわせ言語モデル」が利用できることである．これは，

- 複数の言語モデルによる確率を重み付きで加える．
- ある長さの n-gram を，それより短い n-gram として利用する．

表 1: 学習，開発，評価セット

セット	文数	単語数
学習テキスト	995463	64528076
評価テキスト	80228	3578559
開発テキスト	80227	3562282

という 2 つの機能からなる．例えば，`a.arpa` と `b.arpa` による確率を 3:7 で組みあわせる場合 (つまり，$P(w|h) = 0.3P_a(w|h) + 0.7P_b(w|h)$)，

```
evallm -arpa 'a.arpa*0.3,b.arpa*0.7'
```

のように指定する．また，最尤推定による unigram・bigram・trigram を 1:4:5 で組みあわせる場合には，平滑化のない trigram モデル `ml.arpa` を作っておき，

```
evallm -arpa 'ml.arpa*0.5,ml.arpa;2*0.4,
                    ml.arpa;1*0.1'
```

のように指定できる．混合重みをヘルドアウトデータから推定すれば，ヘルドアウト法による平滑化が実現できる．

言語モデルの組みあわせは，ツールキットの関数内で全て処理されているので，このツールキットの関数を利用してプログラムを書けば，そのプログラムの中でも組みあわせ言語モデルが利用できるようになる．

6 実験

単語モデルとクラスモデル，バックオフモデルとヘルドアウトモデルの比較実験を行ってみた．

6.1 実験条件

学習・評価・開発セットとして毎日新聞データベースを用いた．形態素解析は，茶筌 2.02 を用いていったん解析を行った後，IPA の日本語ディクテーション基本ソフトウェア付属の 20k 語彙リストの品詞体系に合わせて品詞情報を修正したものを用いている．学習には 1991 ～ 1996 年分 (ただし 1994 年は 1 ～ 9 月分) を用いた．評価・開発セットには，1994 年の 10 ～ 12 月分のデータを用いた．このデータの中で，偶数番目の文を開発セット，奇数番目の文を評価セットとした．各セットの文数と単語数を表 1 に示す．語彙は 20000 である．

作成したモデルは，次の通りである．

- 単語 1 ～ 3gram (Witten-Bell バックオフ, ヘルドアウト法による平滑化, それらの組みあわせ)
- 品詞 1 ～ 5gram (Witten-Bell バックオフ, 茶筌の品詞体系を利用，品詞数 261)
- 単語 n-gram と品詞 n-gram の組みあわせ

ここで用いた品詞モデルは，未知語は品詞ごとに扱い，未知の品詞を 1 個使うモデルである．また，評価基準はパープレキシティである．

表 2: 言語モデルの性能 (1)

モデル	平滑化	cutoff	PP	
		2/3/4/5	開発	評価
W2	BO	3	112.6	112.4
W3	〃	3/3	77.9	77.7
W3NS	HO	3/3	98.8	98.7
P2	BO	0	257.3	256.5
P3	〃	0/0	222.8	222.0
P3	〃	0/1	223.0	222.6
P4	〃	0/0/0	191.2	190.7
P4	〃	0/1/1	207.1	206.8
P4	〃	0/1/2	207.0	206.7
P5	〃	0/0/0/0	188.0	187.5
P5	〃	0/1/1/1	185.3	184.9
P5	〃	0/1/2/3	185.9	184.4

6.2 単体での性能比較

単語モデル (Witten-Bell)，単語モデル (ヘルドアウト)，品詞モデルの性能を表 2 に示す.「モデル」の W2,W3 はそれぞれ単語 bigram と trigram を，P2,P3,P4,P5 はそれぞれ品詞 bigram ～ 5gram を表わす.「平滑化」の BO と HO はそれぞれバックオフモデル (Witten-Bell ディスカウティング) とヘルドアウト法による平滑化を表わす. バックオフモデルの場合，開発セットを必要としないので，表中の「開発」は単に開発セットに対して計算したパープレキシティを示す. ヘルドアウトモデルの場合，「開発」の項は開発セットに対して最適化した結果,「評価」はその重みを使って評価セットを評価した結果を示す.

単語モデルの場合，ヘルドアウトモデル（一種類の重み係数を使った場合）よりもバックオフモデルの方がパープレキシティが低いことがわかる. なお，ヘルドアウトモデルの混合重みは，unigram,bigram,trigram に対してそれぞれ 0.09:0.58:0.33 であった.

品詞モデルのパープレキシティは，単語モデルの 2 ～ 3 倍程度であった. 品詞モデルの場合，n-gram の長さを 5 まで伸ばしても，パープレキシティは下がりつづけた. 品詞を用いると，n を大きくしてもパープレキシティの低下が見られる点は他の報告[9]と一致する. また，品詞の場合，一つの品詞に対して相対的にデータ量が多いため，カットオフをしてもほとんどパープレキシティは変化しない. 5-gram の場合には，逆にカットオフによってパープレキシティが下がる現象が見られた. 品詞 n-gram のヒット率は非常に高く，カットオフなしの場合は 5-gram で 92.5%であった.

表 3: 言語モデルの性能 (2)

モデル	平滑化	cutoff	PP	
		2/3/4/5	開発	評価
W2+P3	BO	3+0/0	109.7	109.6
W2+P4	〃	3+0/0/0	105.1	105.0
W2+P5	〃	3+0/0/0/0	103.8	103.7
W3+P3	BO	3/3+0/0	76.2	75.3
W3+P4	〃	3/3+0/0/0	75.2	75.2
W3+P5	〃	3/3+0/0/0/0	74.4	74.3
W3+P5	〃	3/3+0/1/2/3	74.4	76.9

表 4: 言語モデルの性能 (3)

モデル	cutoff	PP	
	2/3/4/5	開発	評価
W3NS	3/3	98.8	98.7
W3NS+P5	3/3+0/0/0/0	89.6	89.5
W1+W2+W3	3/3	77.5	77.3
W1+W2+W3+P5	3/3+0/0/0/0	74.4	74.4

6.3 組み合わせモデルの性能

次に，これらのモデルを組み合わせたモデルを作成し，その性能を比較した. 組み合わせは線形結合で，結合重みは開発セットから学習した. まず，単語のバックオフ trigram と品詞モデルを組み合わせた場合の結果を表 3 に示す.

単語モデルと品詞モデルを組み合わせることにより，パープレキシティはやや下がる. 品詞モデル単独の場合には n-gram の長さを伸ばすことによってパープレキシティが低下していくが，単語モデルと組み合わせた場合はそれほど顕著な低下は見られなかった. 品詞モデルにカットオフを加えた場合とそうでない場合では，ほとんど差が見られなかった.

次に，バックオフなしの unigram ～ trigram を線形結合したもの（前述のヘルドアウトモデル）と，バックオフのある unigram ～ trigram を線形結合したもの，およびそれらと品詞モデルを組み合わせたものの比較を行った. 結果を表 4 に示す.

この結果から，バックオフ平滑化したモデルを線形結合しても，あまり性能は改善されないことがわかる. W1+W2+W3 の場合の結合重みは 0.003:0.103:0.894 であった. 平滑化なしのヘルドアウトモデルと比較して，trigram の重みが非常に大きいことがわかる. unigram はほとんど利用されておらず，bigram も 1 割程度しか貢献していなかった. ヘルドアウトモデルに品詞 5-gram モデルを併用すると性能が改善されるが，バックオフモデルには届かなかった. バックオフ単語 1 ～ 3gram と品詞 5gram モデルの併用は，バックオフ単語 trigram と品

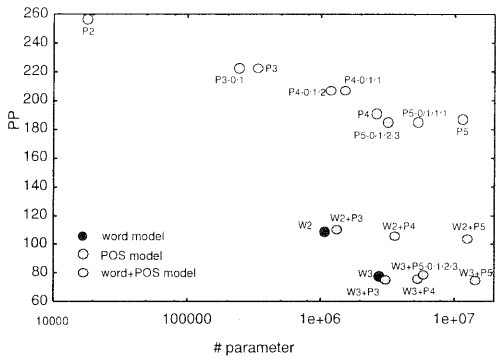

図 5: 各モデルのパラメータ数とパープレキシティ

詞 5gram の併用結果とほぼ同じになった.

6.4 パラメータ数とパープレキシティ

これまで作成した各モデルのパラメータ数とパープレキシティの関係を図 5 に示す. 品詞モデルは同じ長さの単語モデルのパラメータ数の 1/10 以下であるが, n を大きくするとパラメータ数は急激に増加し, 品詞 4 ～ 5gram のパラメータ数は単語 trigram と同等であった.

7 まとめと今後の予定

単語 n-gram とクラス n-gram が作成できるツールキットを作成した. 本ツールキットは CMU-Cambridge toolkit とコマンドレベルで互換性を持ち, さらにクラス言語モデルや混合言語モデル, 組み合わせ言語モデルを作成することができる. 評価実験として, 毎日新聞データベースから言語モデルを作成し, 評価した. その結果, 品詞 n-gram と単語 n-gram を組み合わせることにより, パープレキシティが低下することが確認された. 今回はパープレキシティのみの結果であったが, 品詞 n-gram を組み合わせたモデルを実際の音声認識に適用した場合の効果について検討していく予定である. ツールキットの改良・拡張としては, クラス n-gram のための単語クラスタリング[10]の実装, 各種のクラス・単語言語モデル[11][12]の実装などが挙げられる. また, 最大エントロピー法[13]や LLI[14]などの実装, SCFG 作成プログラム[15]のツールへの統合なども検討していきたいと考えている.

参考文献

[1] K. Sjander and J. Beskow: "WaveSurfer – an open source speech tool", Proc. ICSLP 2000, vol. IV, pp.464–467 (2000-10)

[2] 李, 河原, 堂下:「単語トレリスインデックスを用いた段階的探索による大語彙連続音声認識」, 信学論 (D-II), vol. J82-D-II, No. 1, pp. 1–9 (1999-1)

[3] http://htk.eng.cam.ac.uk/index.shtml

[4] P.R. Clarkson and R. Rosenfeld: "Statistical Language Modeling Using the CMU-Cambridge Toolkit",Proc. Eurospeech '97 (1997)

[5] http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html

[6] http://www-speech.sri.com/projects/srilm/

[7] http://www.ruby-lang.org/

[8] 伊藤, 好田:「N-gram 出現回数の混合による言語モデルの性能解析」, 信学論 (D-II), Vol.J83-D-II,No.11 (2000-11)

[9] T.R.Niesler, E.W.D.Whittaker and P.C.Woodland: "Comparison of part-of-speech and automatically derived catecory-based language models for speech recognition", Proc. ICASSP'98, vol. I, pp. 177–180 (1998)

[10] 森, 西村, 伊東:「クラスに基づく言語モデルのための単語クラスタリング」, 情処学論 vol. 38, No. 11, pp. 2200–2208 (1997-11)

[11] N. Kobayashi, T. Kobayashi: "Class-Combined Word N-gram for Robust Language Modeling", Proc. Eurospeech '99, pp. 1599–1602 (1999-9)

[12] H. Yamamoto, Y. Sagisaka: "Part-of-Speech N-gram and Word N-gram fused Language Model", Proc. Eurospeech '99, pp. 1803–1806(1999)

[13] R. Rosenfeld: "A maximum entropy approach to adaptive statistical language modelling", Computer Speech and Language, vol. 10, No. 3, pp. 187–228 (1996-7)

[14] D. Klakow: "Log-linear interpolation of language models", Proc. ICSLP'98, pp. 1695–1698(1998)

[15] 堀, 加藤, 伊藤, 好田:「音声認識のための確率文脈自由文法に基づく言語モデルの構築と評価」, 信学論 (D-II), Vol.J83-D-II,No.11 (2000-11)