博士学位論文

知識に依存しない言語モデルに関する研究

東北大学大学院工学研究科電気・通信工学専攻

森　大毅

目 次

第 1 章　序論 **1**
1.1 研究の背景 . . . 1
1.1.1 パターン認識と自然言語 . . . 1
1.1.2 言語モデル . . . 3
1.2 研究の目的 . . . 12
1.3 本論文の構成 . . . 13

第 2 章　文字の連鎖確率を用いた頑健な言語モデル **14**
2.1 はじめに . . . 14
2.2 文字 n-gram モデルの定式化 . . . 15
2.3 trigram モデルと少数データ問題 . . . 17
2.4 trigram モデルのスムージング . . . 22
2.4.1 フロアリング . . . 23
2.4.2 削除補間法 . . . 23
2.5 スムージング法の比較及び検討 . . . 29
2.5.1 パープレキシティ . . . 29
2.6 まとめ . . . 33

第 3 章　文字と単語の連鎖確率の融合に基づく言語モデル **36**
3.1 はじめに . . . 36
3.2 関連する研究 . . . 38
3.3 言語モデルの定式化 . . . 39
3.3.1 文字 n-gram モデル . . . 39
3.3.2 単語 n-gram モデル . . . 40
3.3.3 文字・単語の複合 n-gram モデル . . . 41
3.4 モデルの学習法 . . . 42

3.4.1 単語 n-gram モデルの学習 42
3.4.2 複合 n-gram モデルの学習 44
3.5 評価実験 46
3.6 まとめ 50

第 4 章 単語知識を必要としない高精度な言語モデル 52
4.1 はじめに 52
4.2 superword モデルの定式化 53
4.3 superword モデルの学習法 55
4.3.1 superword 集合の獲得 55
4.3.2 確率分布の Forward-Backward 学習 56
4.4 長さ制限の導入 61
4.5 複合モデル 62
4.6 評価実験 63
4.7 まとめ 66

第 5 章 文字認識システムへの統合と評価 68
5.1 はじめに 68
5.2 文字認識の情報理論的解釈 69
5.3 尤度関数の推定法 71
5.4 文字 trigram モデルを用いた認識システムの実現 74
5.4.1 認識アルゴリズム 74
5.4.2 認識実験 75
5.5 低品質文書の認識のための単語・文字列認識システムの実現 78
5.5.1 単語・文字列認識のための探索アルゴリズム 78
5.5.2 認識実験 81
5.6 まとめ 85

第 6 章 結論 86
6.1 本論文の成果 86
6.2 おわりに 87

謝辞 88

参考文献 90

研究業績 **97**

索引 **99**

図 目 次

1.1 単独では読めない文字 3
1.2 前後の文脈によって初めて読むことができる 3
1.3 n-gram の単位の分類 11

2.1 文の生成モデル。破線はヌル遷移 (出力のない状態遷移) を表す . . . 16
2.2 trigram の出現回数 19
2.3 訓練テキストの量による異なり trigram 数の増加傾向 20
2.4 n-gram の被覆率 21
2.5 trigram モデルを bigram, unigram, zerogram によって内挿した隠れマルコフモデル 25
2.6 削除補間法の処理の概要 28
2.7 訓練テキストの量と削除補間法による重み係数の関係 30
2.8 「 社説 」モデルのパープレキシティ 31
2.9 タスク間の関係と言語モデル 34
2.10 「 社説 」モデルと「 天声人語 」モデルのパープレキシティによる比較 35

3.1 複合モデルの状態遷移図。ここでは単語 trigram、文字 trigram の例 43
3.2 未知文章の解析シミュレーションによるパラメータ学習法 45
3.3 単語モデルによる評価テキストの解析結果 47
3.4 新聞社説タスクに対するパープレキシティ 48
3.5 コラム記事タスクに対するパープレキシティ 49
3.6 ディスク使用量 50

4.1 multigram の状態遷移図 55
4.2 再現性のある文字列を求める再帰的手続き 57
4.3 2 回以上出現する n-gram の種類 58
4.4 「 東北大学 」という superword の各文字に対応した副状態の系列 . 59

4.5 「東北大学」というテキストの解析。矩形は superword を、実線は可能なパスを表す。 “$” は文の終端 60
4.6 「東北大学」の部分文字列を superword の集合とする時の長さ制限モデル 61
4.7 superword unigram モデルにおける長さ制限の効果 64
4.8 各モデルの性能評価 65

5.1 情報理論と文字認識の対応関係。[14] を参考にした 69
5.2 Viterbi アルゴリズムによる最尤系列の導出 76
5.3 訓練テキスト量、スムージング手法と誤認識率の関係 77
5.4 セグメント同期型多重ビームサーチ 82
5.5 各モデルによる低品質文書の認識性能比較 84

表 目 次

2.1 実験に用いる日本語テキストデータ (単位 字) 18

3.1 評価テキスト 46

4.1 訓練テキスト・評価テキストの量 63

5.1 個別文字認識系の実験条件 (1) 77
5.2 個別文字認識系の実験条件 (2) 83

第1章　序論

1.1　研究の背景

1.1.1　パターン認識と自然言語

パターンを認識し、理解する能力は、人間のみならず高等動物全てにおいて基本的に備わっている能力である。

人類は、高度な社会生活によって存続する種である。社会生活の根本は、他の個体とのコミュニケーションにある。コミュニケーションを欠いては、人間としての生活を営むことはできない。

コミュニケーションは、伝え手の動作によって生じる物理的な変化を、受け手が感覚器官を介して知覚する、そのやり取りのことである。したがって、言葉を発したり文字を書いたりする能力と同様に、言葉を聞いたり文字を読んだりする能力は、人間にとって本質的なものである。

高度情報化社会という言葉で現代を形容するようになって久しいが、それがもっとも顕著に現れているのが、コンピュータの普及であろう。計算をする機械という本来の使命をはるかに超越し、事務処理や生産支援、オートメーション、情報通信など、およそコンピュータと無縁な社会というものが考えられなくなっている。社会のすみずみにまでコンピュータが浸透している現状を考えれば、社会を構成する一要素として、コンピュータに人間とのコミュニケーションの能力を期待するのはごく自然であろう。

しかし、コンピュータが近い将来その期待に真に応えるだろうと考える人は、現在ではほとんどいない。コミュニケーションの最も基本的な条件である、言葉を聞いたり文字を読んだりするパターン認識の能力においてさえ、人間と同等にまで高め得るという命題に対して多くは悲観的である。人工知能の花形分野として、コンピュータの黎明期から多くの研究者がパターン認識問題の解決を目指して取り組んできた。しかし、それによって一層はっきりしたことは、人間が苦もなく行っている一見単純な情報処理は、実際には巨大な世界知識に裏打ちされた極め

て多角的なものだということであった。そして、今日に至るまでパターン認識が多くの未解決問題を抱えている根本的な原因は、コンピュータ上で実現可能な知識の質および量が、人間の持つ世界知識に全く及ぶべくもないという点にある。

読む、聞くという基本的な情報処理を裏打ちする知識の中で最も重要なものの一つは、言語の能力である。すなわち、我々が持つ高いパターン認識能力は、高度な自然言語処理能力に負うところが大きいのである。

このことは、音声言語においてより顕著である。音声によるコミュニケーションは、音声波形という連続的な物理量を介して行われる。これは、文字が持つ離散的性質や論理的性質と対照的である。発声内容が文章の朗読である場合のように論理的なものであっても、発声を実際に分析してみると、しばしば行われるべき発音が省略されていたり、非常に曖昧であったりする。このため、連続発声から切り出された音節を単独で認識することは人間にとっても難しい。にも関わらず音声による意志伝達が可能なのは、語彙 (lexicology)、文法 (syntax) の知識に加え、状況による判断 (pragmatics) が可能とする強力な言語理解能力が備わっているからに他ならない。

一方、文字による情報は、音声言語に比べコンピュータにとって扱いやすい性質を備えている。このため、文字認識の研究は数十年の間に大きな成果を挙げた。現在では日本語の印刷文字の認識に関してはすでに解決済みであるとさえ言われる [1]。また従来困難とされてきた手書き漢字の認識についても、電子技術総合研究所の手書き文字データベース ETL9B 上での認識実験において、99%を超える結果が近年報告されるようになった [2, 3]。しかし、これが文字認識問題の克服を意味するものでないことは言うまでもない。誰しも他人の悪筆に悩まされた経験を持つものであるが、曖昧な文字は無意識にその前後関係から類推して読んでいる。また、認識が比較的容易であると考えられている印刷文字についても、同形文字や類似文字の曖昧性、さらには未知フォントの問題などから、100%の精度での認識は望めない。市販の OCR ソフトウェアが、謳い文句の通りの性能をなかなか発揮しないという意見はよく聞かれるが、それらの認識誤りは、言語的な「常識」があればあり得ないものがほとんどである。さらに、ファクシミリや複写機によって文字の印字品質が劣化している場合には、一般に単独の文字だけでは情報が不十分である。そのような例を図 1.1、図 1.2 に示す。 図 1.1 の文字はつぶれにより特徴を失っている。約 25 人を対象にした簡単な調査では、この文字単独で正しく判読できた者はなかった。しかし、図 1.2 のように同じ文字でも前後の文脈が与えられると、全体を明らかに読むことができる。我々が文を読む時には必ず

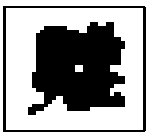

図 1.1: 単独では読めない文字

図 1.2: 前後の文脈によって初めて読むことができる

内容を理解しながら読んでいるはずであるから、網膜に映る像にこれと同様な曖昧性があったとしても、無意識下での自然言語処理により問題なく判読できているのだと考えられる。言語的な処理なしに人間は文字を正しく読むことができている、と考えるのは誤っている。

1.1.2　言語モデル

このように、コンピュータによるパターン認識の能力を人間が持つそれに近づけるためには、コンピュータに自然言語理解の能力を与えることが必要である。しかし、言語の理解こそは人間のもっとも人間らしいと言える知的情報処理であり、上で指摘したような、コンピュータ上に実現できる規模をはるかに超越した世界知識そのものである。現在我々にとって可能なのは、有限の記述長を持つアルゴリズムによって、人間の言語活動を近似することまでである。

自然言語の生成過程に対する近似を定式化したものを、言語モデルという。パターン認識における言語モデルとは、パターン認識の曖昧性を削減するために導入された、認識結果の言語的適格性を与える枠組として定義されることが多い。本論文においても、この立場を踏襲する。

言語モデルは近似の仕方を規定するものであるから、アプリケーションによって最適な言語モデルは各々異なる。パターン認識における言語モデルの目的は、曖昧な認識結果の中から正しいものを導くことであるから、一般論として、より正しい結果を導くことのできる言語モデルが望ましい、ということは言える。それ

以外の言語モデルの評価基準としては、次のようなものが考えられる。

- 記述の容易さ。文法の場合には、文法記述の負担の程度に相当する。
- 柔軟性。新しい語彙・言語現象に対応できるか。その際に要する人的負担の量はどうか。
- 頑健性。実世界での言語活動に対処できるか。実際に正しい文であるにも関わらず言語モデルによって排除されるものがどの程度存在するか。
- 可搬性。多くのタスクに適用できるか。多言語に適用できるか。
- 処理量。モデルの構築や認識システムでの計算量は少ないのが望ましい。
- 言語資源 (コーパス) を必要とするか。必要ならば、どの程度あればよいか。

以下では、過去にパターン認識において用いられてきた言語モデルを分類・整理し、上記のような観点からその得失を論じる。

知識に基づく言語モデル

人間の言語活動を定式化し記述するために古くから採られてきた方法に、言語学的アプローチがある。これは、語彙論・形態論・統語論・意味論・語用論などの言語学的分析法に基づいて規則を記述するものである。これらは知識として人間が与えるものであるから、知識に基づく言語モデルと呼ぶこともできる。

このような立場を採る際に最も基礎的な知識となるのは、一般に単語辞書という名前で呼ばれることの多い、語彙の知識である。その他の知識については、互いに密接な依存関係にあるため、ひとくくりにして文法と呼ぶのがふさわしい。どのような文法を設定するかという問題が、どのような言語モデルを用いるかという問題と同等である。以下では、パターン認識システムにおいて用いられている文法について述べる。

形態素解析、単語照合

形態素とは語を構成する接辞などの要素であり、英語における形態素解析とは、接辞、屈折形および品詞の同定の処理である。これに加え、連続音声認識や日本語の書き言葉においては単語境界がわからないから、辞書中の見出し語とパターンを照合し、単語境界を同定することが形態素解析処理の中心となる。

一般に、形態素解析は曖昧である。曖昧性を減らすために、通常は単語の照合に加え、単語相互間の接続可能性が考慮される。これは、後述する有限オートマトンによる文法記述の特殊例とみなすことができる。

パターン認識において形態素解析に基づく言語モデルを用いる利点を以下に示す。

- 処理が単純で、高速な実行が可能。
- 辞書および文法の記述がコンパクト。これは、後述するより複雑な文法に比べ人手で作成すべき知識の量が少なく、メンテナンスが比較的容易であることをも意味する。
- 既存の文法知識 (学校文法など) が、多くのタスクに対してただちに適用可能。既存の形態素解析システムを利用することも可能である。例外の一つは自然発話の認識タスクであり、間投詞の挿入、助詞の省略などに対する対策を施す必要がある。

その反面、次のような欠点もある。

- 詳細な文法に比べ、制約が弱い。すなわち、明らかな非文を含む多くの可能性を許してしまう。特に、連続音声認識のように曖昧性が大きいパターン認識においては、可能性をしぼりこむ力が弱すぎるので実用的ではない。
- 辞書と文法の記述を人手で与えなければならない。他の文法を用いる場合でも、全ての処理に先立って形態素解析は必要であるから、このことは知識に基づいた言語モデルに共通した欠点と言える。

有限オートマトン　有限オートマトンが受理する言語のクラスは Chomsky 階層の中でもっとも小さく、自然言語の近似モデルとしては能力が弱い。したがって、受理できる正文の割合を大きくしようとすると非文も多く受理してしまうという制約の弱いものになってしまう。

それにも関わらず、自然言語処理において有限オートマトンが重要なのは、その処理の容易性にある。処理量の観点からは、系列長に対して線形オーダーで解析が可能である (後述する文脈自由文法では 2 乗から 3 乗)。また、後続単語が容易に予測できるという利点もあり、連続音声認識などの実時間処理に向く。

有限オートマトンの設計は一般に難しく、専門家の手によらねばならない。このため、まず人手で記述した文脈自由文法を用意し、これを有限オートマトンに近似変換する方法も採られている [4]。

前述したように、表現能力の点で劣るモデルであるから、文法全体の有限オートマトンによる記述は限定されたタスク向きである。また、文節などの文要素を表現するのには向いているから、複雑な文法を用いた処理の前処理として使うこともできる。

構文解析、文脈自由文法　文脈自由文法は自然言語の良い近似を与える枠組として知られている。英語の場合には動詞の性質に応じて大きく 5 つの文型があるが、これらが文脈自由文法における書き換え規則と良く対応することからも、自然言語との整合の良さがうかがえよう。また、形式言語理論の教えるところによれば、$A \stackrel{*}{\Rightarrow} \alpha_1 A \alpha_2$ なる非終端記号 A を除去できない文法は正規ではないが、これはまさに日本語における複文の埋め込み構造そのものである。

このように、文脈自由文法は人手で文法を記述するのに向いており、限定されたタスクにおいては適切な文法を与えることによって望ましくない認識結果を排除することが可能である。

しかし、タスクを限定しないパターン認識のために文脈自由文法を適切に記述するのはかなり難しく、これまでのところ実用に足る汎用の日本語構文解析システムは存在しないと言ってもよい。これは、日本語の構文が比較的自由なものであることにも関係している。すなわち、主語を必要とせず、文が名詞を中心に組み立てられ、係り受けを基礎とした修飾関係が語順にゆるやかな制限を与えるのみで、はっきりとした構造が定まっていない。また、音声言語においては語順はさらに自由で、省略も頻繁であり、そもそも文法的とは言いがたい。

このようなことから、もし大きなタスク向けの、あるいはタスクに依存しない文法を文脈自由文法で記述するとしても、日本語の場合には構文的な制約による曖昧性のしぼりこみ効果はあまり期待できない。したがって、日本語の構文的制約の記述だけを積極的に行うのではなく、むしろ後述する意味解析のインプリメンテーションのための枠組として利用するのが適当だと考えられる。

意味解析　意味解析は、言語理解の手段としてだけでなく、構文解析の曖昧性、さらにはパターン認識の曖昧性を削減するのにも役立つ。例えば、朝日新聞「社説」中にある次の文

こんどの 結果が もたらした 意味は 大きい。　(1.1)

を誤って

∗こんどの 結果か もたらした 意味は 大きい。　(1.2)

と認識することは、構文的知識を用いるだけでは防ぐことができない。しかし、動詞「もたらす」が主格として動作主を要求することに着目すれば、後者には「もたらす」の動作主がないから非文と判定することができる。

このような意味解析を行うための枠組の代表的なものに、Fillmore の格文法がある。格文法では、意味の取り扱いを動詞を中心として整理する。すなわち、各々の動詞がいかなる格を要求し、そこにどのような素性を持つ語が入るかという情報 (格フレームと呼ばれる) を記述しておく。また、語彙項目にはその語がいかなる素性を持つかを記述しておく。

他に、格関係を含む種々の依存関係を、構造を持った非終端記号 (素性構造と呼ばれる) を媒介として解析する単一化文法という枠組がある。単一化文法は句構造文法 (実質的には文脈自由文法) に素性構造を導入することで拡張したものと考えることができ、各種の構文的・意味的制約を素性構造として語彙項目に持たせることで、文法規則数の爆発を防いでいる。

しかし、これらを含むいずれの枠組においても、文法の記述、辞書の整備には膨大な人的コストがかかる。また、新しい語彙の登録を自動で行うことは不可能で、文法の構造を理解した専門家の助けが必要である。

また、これらの意味解析は単一化と呼ばれる操作に基づいて処理されることが多いが、この処理の計算量は一般に非常に大きい。

コーパスに基づく言語モデル

以上述べてきたように、規則に基づく言語モデルには一長一短があるが、総じて言えることとして、文法や辞書の記述に労力がかかる、という共通の欠点がある。特に、曖昧性削減能力を高めるために複雑なモデルへ移行しようとすると、必要とする知識には量的のみならず質的な高度さが要求されるようになり、いよいよ職人芸とならざるを得ない。

知識に基づくモデルには、柔軟性の点からも多くの問題がある。形態素解析の場合には、新しい語彙に対応するためには人手による辞書のメンテナンスが必要である。また、「ら抜き言葉」に代表されるような言語現象の変化への対応は難しい。意味解析を行う場合には語彙そのものが多くの情報を持つことになるが、そ

の場合の辞書のメンテナンスがさらに難しいことはすでに述べた。

可搬性についてはどうだろうか。連続音声認識にはほとんどの場合タスクが設定される。しかも、ある一定レベルの認識性能を達成するために、タスク記述はむしろタスクに特化した形で行われることが多い。これを、1970 年代に始まる ARPA(Advanced Research Projects Agency) 主導による連続音声理解プロジェクトの結果を受け、「タスクを制限しなければ高い認識精度は望めない」という観念が一般化してしまった結果と解釈する向きもある [5]。しかし、ARPA プロジェクト発動から 20 余年を経て、今だに個々の事例に個別の対処を続けていく態度が人間の知的情報処理能力へ近づく道であろうはずもない。むしろ、タスクに依存しない言語モデルの枠組を見出すのが本筋であろう。だが、大きなタスク、あるいはタスクに依存しない文法を記述し、十分な性能を得ることが非常に難しいことは、先に述べた通りである。

このような欠点を持つ知識に基づく言語モデルに代わる、あるいは補完するものとして、近年コーパスに基づく言語モデルが注目されてきている。コーパスとは広義には文書・音声データベースや電子化辞書、シソーラスを含む言語資源を指すが、本論文ではテキストコーパスのことをいう。つまりコーパスに基づく言語モデルとは、電子化された自然言語テキストを事例集とし、事例に基づいてパラメータを獲得し、あるいは自己組織化する自然言語の生成モデルをいう。

テキストコーパスは、形態的には次の 3 種に分類される [6]。

- 生コーパス
- タグ付きコーパス
- 分析済みコーパス

生コーパスはオリジナルのテキストそのものを電子化したものであり、明らかにもっともありふれたコーパスである。生コーパスの場合、一文一文の質よりは、むしろコーパス全体としての等質性や収集方針の方が問題となる。タグ付きコーパスは単語に品詞などの標識を付与したもので、日本語の場合には単語境界も付与される。分析済みコーパスは構文木などの文法的分析を施したもので、作成のコストは当然最も大きい。タグ付きコーパスや分析済みコーパスの作成および利用に際しては、何らかの意味で言語学的立場の統一が保たれなければならない。しかし、実際には同一のコーパスにおいてもしばしば不統一が見られる [7]。また一

般に、異なる言語学的立場で作成されたコーパスが混在した形で利用することは難しい。

コーパスに基づく言語モデルが一般化した背景事情としては、主なものを 2 つ挙げることができよう。1 つめは、音声認識における確率・統計的手法の大きな成功である。特に、隠れマルコフモデル (HMM: Hidden Markov Model) の登場は、それまでの動的時間伸縮 (DTW: Dynamic Time Warping) によるテンプレートマッチングを基礎とする音声認識の流れを大きく転換し、確率モデルの時代を切り開いた。コーパスに基づく言語モデルの多くは確率的モデルであるが、これらには HMM に代表される音声認識研究の成果に触発されたという側面もあろう。音声認識システムは大きく音響処理部と言語処理部から成るが、現在では音響処理部に確率モデルが使われるのが普通であるから、確率的言語モデルとの相性が良いことも指摘しておきたい。2 つめは、機械翻訳に代表される人工知能研究が、いわゆる知識獲得のボトルネックからゆきづまりを見せ、これを打破するものとして、事例ベースのアプローチが注目されるようになったことである。事例に基づく自然言語処理の代表的な例としては、分析済みコーパスを用いた語彙的知識の獲得に関する研究や、対訳コーパスを用いた構造的言語知識の獲得に関する研究が挙げられる [8]。パターン認識技術がそうしたアプローチから受けた刺激はむろん無視しがたいものがあるが、実際にはむしろ学際的なコーパスベース技術への期待の高まりに応え、大規模コーパスの開発・整備・流通が促進されたことによる影響の方がより大きいと考える。

以下では、パターン認識に関連の深い言語モデルについて述べる。

HMM　隠れマルコフモデル (HMM) は、出力シンボルの系列によって一意に状態遷移系列が決まらないという意味において非決定的な確率付き有限オートマトンである。HMM は従来の DP マッチングによる DTW を一般化したモデルとみなすことができ、また多量のサンプルを用いることでパラメータの学習を効率良く行える手法が知られていることから、連続音声認識手法の主流となっているものである。

音響モデルとしての HMM では、時間的非可逆性を表現するために一方向的な状態遷移のみが許されるのが普通であるが、HMM を言語モデルとして使う場合には全状態間の遷移を許すことが多い [9]。また、構造を持つ HMM の一種である HMNet[10] を日本語の文節モデルに適用する試みもある [11]。

また、前述したように単語相互間の接続可否に基づく形態素解析は有限オート

マトンの特殊例であるが、品詞から単語への生成コストや単語の接続コストを導入したものは、HMM の特殊例になる。形態素解析システムの設計にあたって、これらのコストは経験的あるいは実験的に設定されるが、そのようなパラメータ調整を HMM の学習によって行うことで改良する研究がある [12]。

SCFG　文脈自由文法の書き換え規則に確率を付与して拡張したものが、確率文脈自由文法 (SCFG: Stochastic Context Free Grammar) である。すでに示したように、有限オートマトンと文脈自由文法の間には、本質的な能力の差がある。同様に、パラメータを適切に与えることができれば、HMM よりも SCFG の方が記述効率が高く、自然言語のモデルとして向いていることが考えられる。

SCFG は、分析済みコーパスが与えられていれば、原理的には書き換え規則の適用回数から確率を推定することができる。しかし実際には、非終端記号数の爆発をどう抑えるかという問題があり、簡単ではない。

SCFG の学習方法としては他に、Inside-Outside アルゴリズム [13], CYK 法に基づくアルゴリズム [14] などが知られている。Inside-Outside アルゴリズムは文法が未知の場合にも推定が行えるが、その学習に要する計算量がサンプル系列の長さ及び非終端記号数の 3 乗に比例して増加することから、タスクを制限してパラメータ数を小さく抑えなければ応用が難しい。CYK 法を用いる場合は文法を人手で与えねばならないので、知識に基づく言語モデルと同様な問題が残る。

n-gram　n-gram とは本来 n 個の組を意味する語であるが、自然言語処理においては記号の生成確率がマルコフ過程 (正確には $(n-1)$ 重マルコフ過程) に従うとするモデルのことを n-gram という。

n-gram の歴史は古い。n-gram が自然言語の性質を良く反映することは Shannon[15] によっても言及されている。しかし、n-gram が単に良く知られている単純なモデルであるというだけでなく、音声認識の曖昧性を減少させる極めて有効な道具となり得ることを初めて示したのは、IBM ワトソン研究所のグループによるディクテーションシステム Tangora[16] である。以来、特に連続音声認識を中心とした研究成果により、n-gram の有効性がさらに明らかになった [17, 18, 19, 20, 21, 22]。大規模コーパスが一般化した今日においては、n-gram の重要性はなお一層増すばかりのように思われる。

n-gram モデルの特長は、次の通りである。

- 柔軟である。文法が仮定されている言語モデルは、その文法から逸脱した系

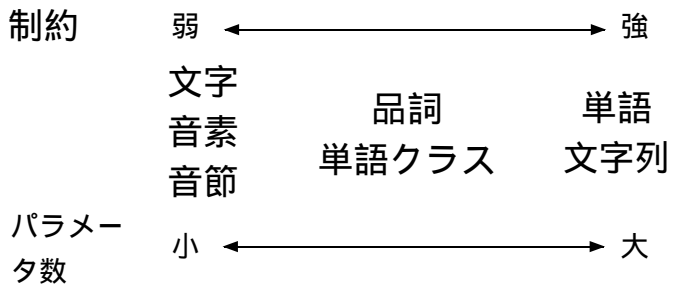

図 1.3: n-gram の単位の分類

列を受理しない。n-gram モデルは文法を仮定せず、用例にある表現は例え非文法的であっても許容する。これはさまざまな表現を含んだ実際の文章を扱う応用においては有利である場合がある。

- 学習が簡単である。n-gram モデルの学習は、単に n 個組の出現頻度を数え上げるだけで実現できる。
- 処理が高速である。

反面、次のような欠点もある。

- n を大きくとらないと、探索空間が大きすぎて制約とならない。しかしモデルのパラメータ数、言い換えれば異なる n-gram の数は、終端記号の数を N とすると最悪で N^n 個であり、実際には n は高々3 か 4 程度にしかできない。この場合の n-gram モデルは、日本語のように係り受けを基礎とした文構造を持つ言語が対象の場合に大域的な統語構造を記述できない。また、十分に探索空間を制限するに足る n-gram の持つパラメータを正確に推定できるほど大量の訓練テキストは用意できない (少数データ問題)。訓練テキストの量が小さいと、未知データに対する頑健性が失われる。

n-gram によるモデル定式化にあたっては、その単位を何にするかが問題である。n-gram の分類を図 1.3 に示す。最も簡単なのは単語の n-gram で、英語などではこれがよく用いられる。しかし、日本語、中国語などは分かち書きをしない。したがって、学習のためにはタグ付きコーパスを利用するか、さもなければ生コーパスに何らかの前処理が必要である。

単語の n-gram は非常に種類が多くなるから、品詞の n-gram を代わりに使うことも多い [23, 24, 25]。これによりパラメータ数が大幅に削減され、少数データ問

題が回避できる。しかし、この方法では学習にタグ付きコーパスが必要となる (なくても不可能とは言えないが、効果については疑わしい)。また、品詞の定義はコーパスに依存するため、複数のコーパスを混在させるのは難しい。さらに、品詞という範疇が言語モデルとして最適かどうかはわからない。このため、品詞に相当する単語のクラスを相互情報量に基づくクラスタリングで自動的に見出す方法も提案されている [26]。

また、文字を単位とした n-gram も文字認識 [28]、音声認識 [19, 21] などで用いられている。このモデルは、生のコーパスからただちに求めることができる点で優れている。しかし、文字連鎖を単純に長くすることによってはモデルの精度を上げるのは困難と考えられる。音声認識では音素・音節を n-gram の単位とすることも行われている [27, 18, 19] が、n を大きくできないために制約が局所的である点では文字 n-gram と類似したものと言える。

このように、既存の n-gram にはそれぞれ一長一短があり、曖昧性解消能力・頑健性などを総合的に高めた新しい言語モデルの開発が望まれている。

1.2　研究の目的

本研究が究極的に目指すものは、コンピュータ上に人間と同等の自然言語理解能力を与えることにより、文字認識や音声認識の精度を高め、人間とコンピュータの円滑なコミュニケーションを実現することである。

しかし、本論文では特に、従来の自然言語処理・自然言語理解の研究に見られる知識への依存による問題点、すなわち知識獲得のボトルネックを重要視し、知識への依存の度合いが少ない、あるいは全く依存しない方法という枠の中でパターン認識の曖昧性を解消する強い能力を持つ言語モデルを見出すことを目的とする。また、特定タスクに特化したモデルではなく、可搬性のあるモデルを得たい。このため、新聞社説などの比較的大きなタスクを研究の対象とする。

確率的言語モデルでは、訓練テキストの量と曖昧性解消能力との関係が問題となる。本論文では、言語モデルの能力を、訓練テキスト量とパープレキシティと呼ばれる基準との関係において論ずる。

また本論文では、提案する言語モデルのパターン認識における有効性を評価するために、印刷文書を対象にした文字認識システムを構成し、認識実験を行う。

1.3　本論文の構成

本論文の構成は以下の通りである。

第 2 章では文字の n-gram に基づく日本語のモデル化について定式化を行う。また、少数データ問題に対処するために 2 種類のスムージング法について比較検討を行う。また、構成した言語モデルの曖昧性削減能力をパープレキシティにより評価する実験を行う。さらに、認識対象の言語的性質の相違と言語モデルの効果の関係についても考察する。

第 3 章では、単語 n-gram の欠点である、サンプルが少ない場合に単語解析精度が低下する問題について論じ、この問題を解決する新しい方法を提案する。すなわち、文字と単語の n-gram を融合することにより、解析精度の低下による悪影響を低減する言語モデルについて述べる。また、未知文章の解析シミュレーションという方法を用いたモデル学習法について述べる。さらに、従来のモデルとの比較実験により、提案するモデルの有効性について述べる。

第 4 章では、言語知識を全く必要としない、高精度な新しい言語モデルを提案する。まず superword と呼ばれる再現性に基づく文字列集合の概念を与え、その学習アルゴリズムについて述べる。次に、superword に基づくモデルを定式化し、学習法を与える。また、複数のモデルを融合して汎化能力を改善したモデルを提案する。さらに、これらのモデルと前章までで提案したモデルの性能を実験により比較検討する。最後に、モデルの性能を落とさず計算量を削減するための検討を行う。

第 5 章では、提案した言語モデルを用いた文字認識システムを構築し、総合的な評価を行う。まず、第 2 章で提案する文字 trigram を用いた認識システムについて述べ、認識実験結果とパープレキシティによる評価との対応関係について論じる。続いて、第 3 章で提案した文字と単語を融合したモデルを用いた文字認識システムの構成法を示し、実験により提案したモデルの有効性を確認する。

第 6 章は結論である。

第2章　文字の連鎖確率を用いた頑健な言語モデル

2.1　はじめに

本章では、知識に依存しない言語モデルの最も基礎的な形態である、文字の n-gram モデルについて述べる。このモデルは、あらゆる言語モデルの中でも最も単純なものの一つであると言ってよい。既に図 1.3 に示したように、文字・音素・音節の n-gram は品詞や単語の n-gram に比べパラメータ数が少なく、学習が容易であると言える。

音素・音節の統計情報を生コーパスから得るには形態素解析が必要であるが、これらの要素を単位とする言語モデルでは、品詞や単語に基づく言語モデルと違い、得られる統計情報が形態素解析に必要な言語的知識にほとんど依存せず、ほぼ同じ結果となる。このため、音声認識においては認識の単位となる音素・音節を n-gram の単位とすることによって、音響処理部と言語処理部の整合が取れたシステムを実現することができる [27, 18, 19]。

この議論を延長すると、文字認識の場合には文字 n-gram を用いるのがふさわしく思える。しかし、これは全ての言語に対して必ずしも自明でないことに注意しておきたい。まずヨーロッパ系言語における文字、すなわちアルファベットについてであるが、アルファベットを単位とする n-gram はほとんど用いられていない。この原因についてはいくつか可能性があるが、その一つとしてアルファベットが表音文字であることが挙げられよう。つまり、日本語の n-gram と異なり、アルファベットの局所的な連鎖には意味の上でのつながりがない。仮に意味の上でのつながりまで表現しようとするならば、連鎖の単位は少なくとも形態素まで拡張せねばならない。また、アルファベットの数は一般に非常に少なく、十分な制約を得るためには n を大きくせねばならないが、これを形態素レベルまで拡張するとパラメータ数が大きくなり過ぎ、必要なコーパスの量と得られる制約との間のバランスを欠く。また、朝鮮語などの膠着語においては、形態素の区切りが文字の区

切りにならないことがある。そのような言語を文字を単位とした n-gram でモデル化すると、語の派生に関する一般性を獲得することができず、効率が悪い。このため、朝鮮語 OCR の後処理システムにおいては形態素解析と音節 n-gram を組み合わせた方法も試みられている [32]。

これに比べ、日本語の場合には活用する機能語はそのほとんどが字種の少ない音節文字から成り、字種が多い漢字は主として活用しない実質語の表記に用いられる。これは文字を単位とする n-gram を利用する上では極めて望ましい性質である。また、実質語の多くは 1 文字から 3 文字で構成されている。これは、文字 n-gram を使う場合に n をあまり大きくすることなく効率良くモデル化できる可能性を示唆している。

本章では、まず日本語文字 n-gram モデルの定式化を行った後、少数データ問題を実験により検証する。これにより n-gram モデルのスムージングが必要であることを明らかにし、連続音声認識の分野で知られている 2 種類のスムージング法について詳細を述べる。またこの手法を用いて実際に文章のサンプルから日本語文字 n-gram モデルを学習し、情報論的な立場から得られた言語モデルの曖昧性解消能力について検討を行う。

2.2　文字 n-gram モデルの定式化

ここでは日本語文字系列のマルコフ性を仮定する。マルコフ性とは、全ての時点での文字の発生確率が、過去にどのような系列が発生したか (文脈) だけで決まることである。文字系列 $C_1C_2\cdots C_{i-1}$ の次に文字 C_i が生起する条件付き確率を $P(C_i|C_1C_2\cdots C_{i-1})$ と表記する。文脈の数は無限にあり得るから、学習が可能であるためには何らかの近似を導入しなければならない。一つの方法は、この情報源をマルコフ連鎖で近似することである。すなわち、

$$P(C_i|C_1C_2\cdots C_{i-1}) \approx P(C_i|C_{i-(n-1)}\cdots C_{i-2}C_{i-1}) \tag{2.1}$$

この場合の文字 C_i の生起確率は、直前の $(n-1)$ 文字 $C_{i-(n-1)}\cdots C_{i-2}C_{i-1}$ だけで決まることになる。この意味で、$C_{i-(n-1)}\cdots C_{i-2}C_{i-1}$ を状態と呼ぶ。式 (2.1) はいわゆる $(n-1)$ 重マルコフモデルであるが、これは長さ n の系列 $C_{i-(n-1)}\cdots C_{i-1}C_i$ が出現する頻度を表すものであることから、n-gram (n 字組) モデルとも呼ばれる。本論文では後者の呼び方を採用する。また、式 (2.1) の右辺を n-gram 確率と呼ぶ。

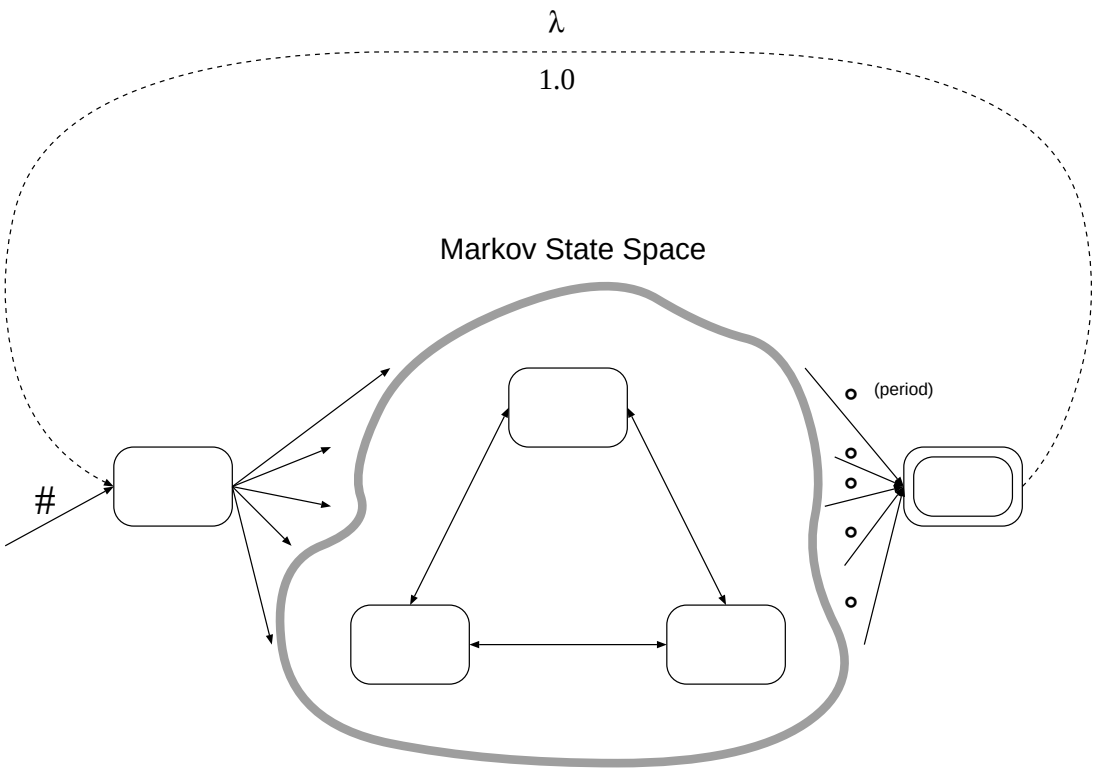

図 2.1: 文の生成モデル。破線はヌル遷移 (出力のない状態遷移) を表す

ここでの n-gram モデルは文字の連鎖を規定するものだが、文の連鎖も考えられる。しかし、文字の n-gram モデルでは文の連鎖を適切に表現することはできないと考える。よって、言語のモデル化にあたっては、各文の生起確率は独立であると仮定する。すなわち、

$$P(C_i|C_{i-(n-1)}\cdots 。 C_{i-j}\cdots C_{i-1}) = P(C_i|。 C_{i-j}\cdots C_{i-1}) \quad (2.2)$$

$$= P(C_i|\# C_{i-j}\cdots C_{i-1}) \quad (2.3)$$

ただし、ここでは「＃」は文書の最初を表す特殊な文字である。このことを状態遷移図で表すと、図 2.1 のようになる。この図では文は初期状態から何らかの状態遷移および文字の出力を経、最終的には句点「。」を出力した後に終状態に至る。終状態と初期状態を 1 状態に縮退させれば、ヌル遷移のないモデルとなり、出力系列と状態遷移系列が 1 対 1 に対応する。このようなモデルはユニフィラーであるといわれる。

2.3　trigram モデルと少数データ問題

n をいくつに選べば良いかは重要である。序論で言及したが、n の選択にあたって何が問題となるかをもう一度整理しておきたい。

- n が小さいと、探索空間が広くなりすぎてモデルの曖昧性解消効果はなくなってしまう。
- n が大きいと、訓練サンプルがパラメータ数に比べて少なすぎるので学習が難しくなってしまう。

本研究と同様に日本語文字 n-gram モデルを採用しているものに、bigram モデル $(n = 2)$ をオンライン手書き文字認識に応用した研究がある [28]。ここでは、マルコフモデルとしては最も簡単な bigram モデルを採用し、さらに複数の文字カテゴリを代表するマクロ文字を導入してカテゴリ数を減らすことによってパラメータ数を小さく抑えている。またそれに伴って探索空間が拡大するのを防ぐため、文字に品詞属性を与えて文法的な制約を加味している。この手法によって新聞記事の認識後処理実験を行ったところ、候補内に正解があるうちの約半数の誤りが救済できたと報告されている。しかし、この手法は訓練テキストを形態素解析する必要があり、マクロ文字の品詞属性が形態素解析の品詞分類を踏襲する点で、知識に依存したモデルであると言える。また、このモデルはハードウェア的な制約が大きいパームトップ、ラップトップ型のキーボードレスコンピュータ上での実装を意識したものであり、簡便ではあっても強力とは言いにくい。

音節文字 (かな) については、字種が少ない割には連続して多用されることが多く、bigram では十分な制約とならないことが知られている [29]。かな trigram を形態素解析の仮文節境界の推定に利用することの有効性が報告されている [30] が、最近では 4-gram(3 重マルコフモデル) を用いた方が精度が良いとも言われている [31]。本論文で扱う日本語文中にはかなの連続する部分が非常に多いことを考えると、$n \geq 3$ が必要である。また、字種が約 3000 と多く、これ以上状態数を増やすのは得策ではない。したがって、本研究では trigram モデル、つまり $n = 3$ のモデルを採用し、効果を確かめることとする。日本語文字の trigram モデルの有効性は、音節列からの漢字かな系列の生成において確かめられている [19][21]。

trigram モデルを採用したことによって、少数データ問題が深刻となることが考えられる。このことを確認するため、日本語テキストデータをコーパスとして用い、trigram モデル学習の傾向を調べた。本論文で用いたテキストデータを表 2.1

表 2.1: 実験に用いる日本語テキストデータ (単位 字)

	訓練サンプル	テストサンプル
「社説」(朝日新聞)	580 万　(A)	80098　(B)
「天声人語」(朝日新聞)	170 万　(C)	23315　(D)
「科研費概要」		34219　(E)

に示す。表中で、「科研費概要」とは、学術情報センター科研費補助金研究成果概要データベースの中から「情報」をキーワードに選んだ抄録である。このテキストデータは、後の章での認識実験にも用いる。簡単のため、テキストの種類は表にある通り (A) から (E) の記号で呼ぶことにする。以後、断わらない限り、モデルの訓練に用いるデータは (A) の「社説」である。

最初に、訓練テキスト中の trigram の出現回数分布を調べた。その結果を図 2.2 に示す。この図からわかるように、半数以上の trigram は再現性のないものであり、また 10 回以下の trigram だけで全体の 9 割以上を占めてしまう。出現回数が十分大きく統計的に信頼できる n-gram については、その出現確率を最尤推定によって求めて良いが、Katz[33] は実験によってこの信頼するに足るとみなせる回数の閾値を 6 回とした。このことからも、ほとんどの trigram は統計的な信頼性が低く、確率の推定の仕方を工夫せねばならないと言える。なお荒木ら [27] は、音節の trigram について出現回数の少なかったものの確率を 0 とすることを試みているが、これは訓練データの多少に関わらず逆効果で、確率を 0 としないときに比べ性能が低下することを報告している。また、Federico et al.[39] がスペイン語、英語の 3 種類のコーパスにおいて、出現回数が 1 回の bigram を削除して back-off スムージングをしたところ、他のモデルに比べて成績が悪かったことが報告されている。これらの結果を踏まえ、本研究では出現頻度の低い trigram をも利用することとする。

次に、訓練テキスト中のユニークな trigram の種類を調べた。その結果を図 2.3 に示す。この図から、全訓練データを用いても trigram 数は収束せず、多少の訓練データの追加では対応できそうにないことがわかる。この状態では、訓練データには含まれなかった trigram が未知の文書に現れることが予想される。

そこで、訓練テキストが持つ n-gram の未知データに対する被覆率を調べる。被覆率は次のように定義する。訓練テキスト $\mathcal{T}$ 中に n-gram $\boldsymbol{C}_k^{k+(n-1)}$ が存在すると

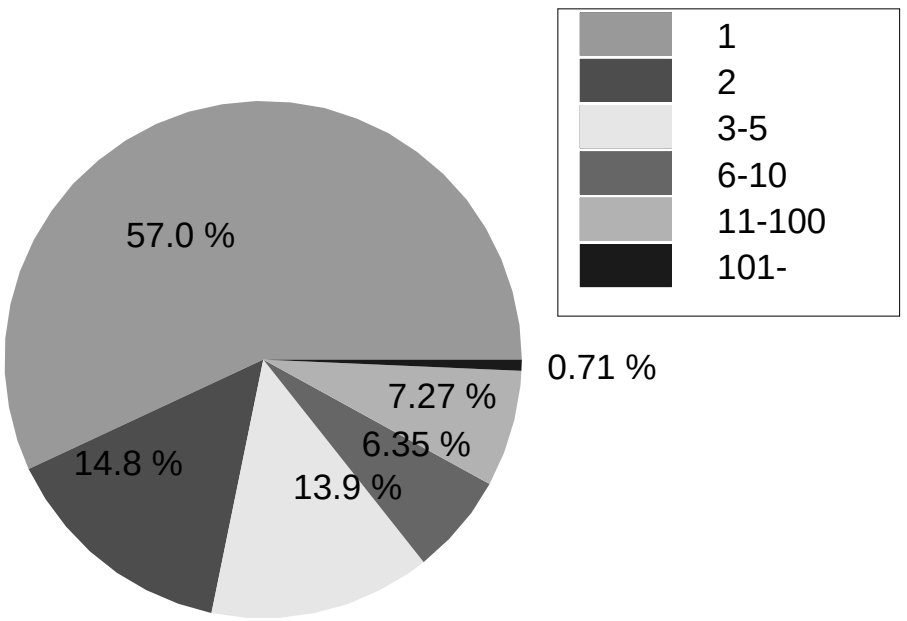

図 2.2: trigram の出現回数

き 1, そうでないとき 0 を取るような関数 $\delta(\boldsymbol{C}_1^n, \mathcal{T})$ を用い、評価用の未知テキスト $C_1C_2\cdots C_l$ に対して

$$(\text{被覆率}) \stackrel{\text{def}}{=} \frac{1}{l}\sum_{i=1}^{l-(n-1)} \delta(\boldsymbol{C}_i^{i+(n-1)}, \mathcal{T}) \tag{2.4}$$

この値は、評価用テキストに出現する n-gram を全て予想できる場合に 1 を取る。訓練テキストを (A)、評価用の未知テキストを (B) としたところ、被覆率は図 2.4 のようになった。trigram は 90%程度しかカバーしていない。したがって、訓練テキスト中に存在する trigram だけを認めるようなモデルでは、認識対象の概ね 10 文字に 1 文字の割合で「あり得ない」と判断する文字が現れることになり、このような文字を含む系列の生起確率を 0 と推定してしまう。これを放置すれば、系列が長くなるにしたがい、その生起確率の期待値は急激に減少する。これは、言語モデルとしての性能の急激な劣化を意味する。同様の問題点は bigram の場合にも当てはまる。unigram、つまり文字そのものですら、被覆率 100%とはならない。

以上のような傾向は典型的な少数データ問題と言える。したがって、trigram モデルを用いる場合には訓練データが少ないことを考慮し影響を小さくするような処置を施さなければならない。これについては次節で詳しく述べる。

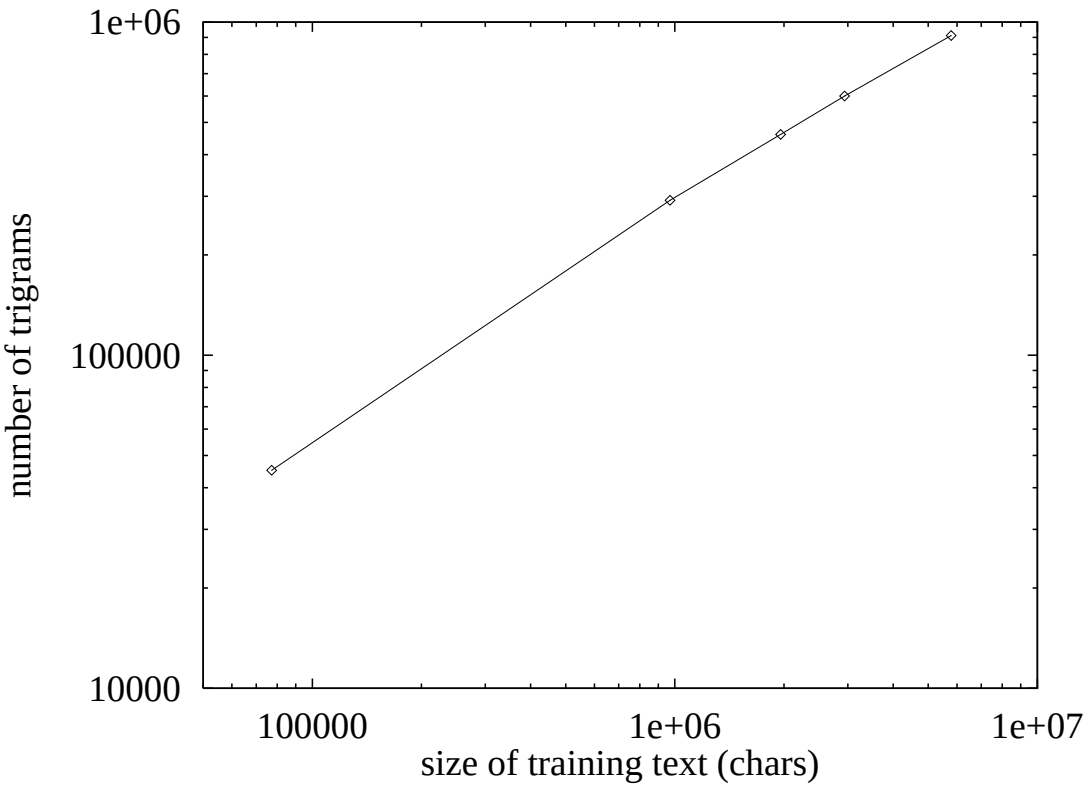

図 2.3: 訓練テキストの量による異なり trigram 数の増加傾向

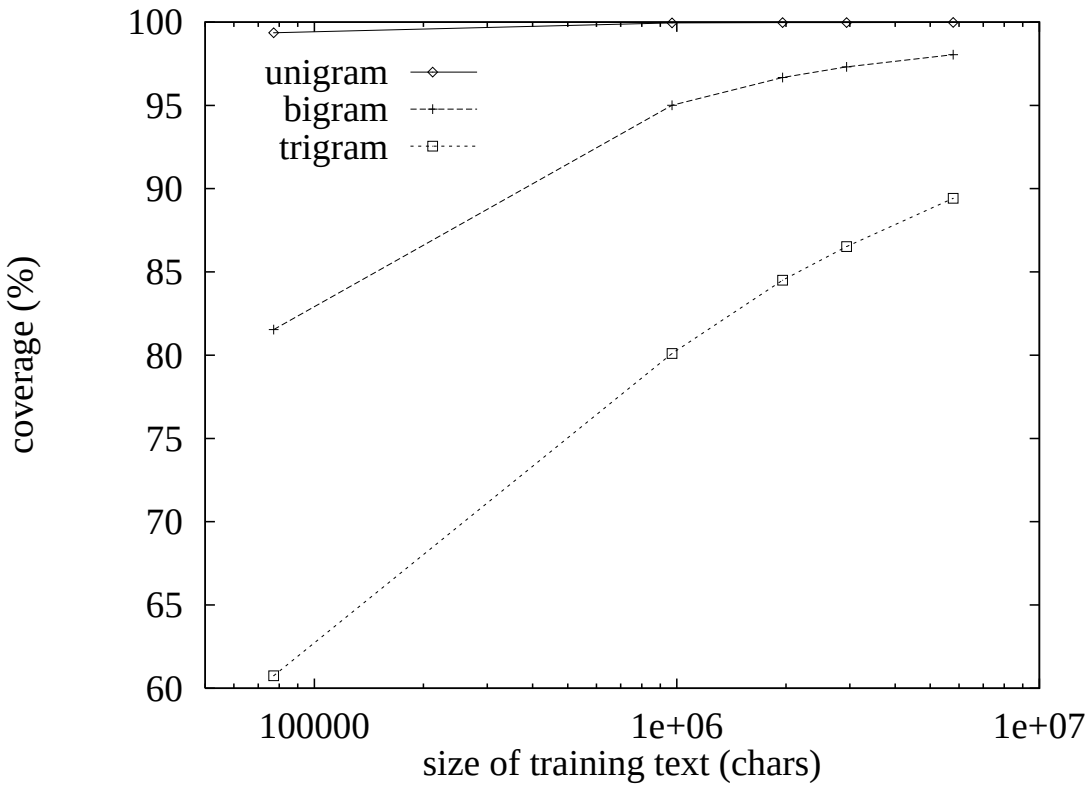

図 2.4: n-gram の被覆率

2.4　trigram モデルのスムージング

$\mathcal{A}$ を情報源記号の集合とし、$\mathcal{S}$ を $\mathcal{A}^{n-1}$ 上で定義された状態の集合とすると、一般に n-gram モデルは写像 $\mathcal{S}\times\mathcal{A}\to\mathbf{R}$ で定義される。値域は確率であり、$[0,1]$ である。この値は、情報源からの通報 (本論文では自然言語のサンプル) から、n-gram の出現頻度を求めることによって推定できる。長さ l の通報中に n-gram $C_1C_2\cdots C_n\in\mathcal{A}^n$ が現れた回数を $N(C_1C_2\cdots C_n)$ とすると、

$$\overline{P}(C_n|C_1C_2\cdots C_{n-1}) = \frac{N(C_1C_2\cdots C_n)}{N(C_1C_2\cdots C_{n-1})} \qquad (n\geq 2) \tag{2.5}$$

$$\overline{P}(C_1) = \frac{N(C_1)}{\sum_{C_1\in\mathcal{A}} N(C_1)} \qquad (n=1) \tag{2.6}$$

これは n-gram 確率の最尤推定式になっており [14]、通報の長さ $l\to\infty$ とすると真値 $P(C_n|C_1C_2\cdots C_{n-1})$ に一致する。したがって、通報の長さをいくらでも大きくできるのならば、最尤推定法が最も適当なモデルを与える。(なお、式 (2.2), (2.3) が示すように句点や文章の最初を含む trigram, bigram については特別に考える必要があるが、最尤推定式は自明であるからここでは省略する。)

しかし、実際にはモデルのパラメータを推定するための訓練テキストを必要なだけ多く取ることはできない。特に本論文で扱う日本語文字 trigram モデルの場合にはこの問題が深刻であることは 2.3 節で述べた。図 2.4 に示したように、未知テキストに対しては多くの trigram について $N(C_1C_2C_3)=0$ となり、したがって $\overline{P}(C_3|C_1C_2)=0$ である。

このように、一般に n-gram モデルは少数データ問題が深刻であり、このために様々な対応策が取られてきた [34]。これらの手法が行っていることは、サンプルが少ないために不連続的になっている状態空間 $\boldsymbol{S}$(またはパラメータ空間 $\mathcal{S}\times\mathcal{A}$) を何らかの意味で平滑化することであり、スムージングと呼ばれる。本論文では 2.3 節で述べたような問題が無視できないものと考え、次に示すように 2 種類のスムージング手法について検討する。

2.4.1　フロアリング

第 1 の手法は非常に簡単なもので、スムージングされた trigram 確率は次式で推定される。

$$\hat{P}(C_i|C_{i-2}C_{i-1}) \stackrel{\text{def}}{=} \begin{cases} \varepsilon & \text{if } \overline{P}(C_i|C_{i-2}C_{i-1}) = 0 \\ \overline{P}(C_i|C_{i-2}C_{i-1}) & \text{otherwise.} \end{cases} \tag{2.7}$$

ここに、$0 < \varepsilon \ll 1$. これは、いかなる trigram 確率も正の値をとるべきことを主張しているものであり、この手法を用いることによって全ての系列の生起確率を正の値とすることができる。被覆率は 100%となる。確率の定義から、trigram 確率に関して次式が満たされることが要求される。

$$\sum_{C_i} \hat{P}(C_i|C_{i-2}C_{i-1}) = 1. \tag{2.8}$$

今、$\varepsilon > 0$ としたため、$\sum_{C_i} \hat{P}(C_i|C_{i-2}C_{i-1}) > 1$ となってしまい式 (2.8) は満たされない。このために、$\overline{P}(C_i|C_{i-2}C_{i-1}) > 0$ なる trigram に関する確率を小さくなる方へ調整して式 (2.8) を満たすようにすることも行われる [14]。しかし、ε を十分小さくすれば式 (2.8) は近似的に成り立つので、本論文では式 (2.7) を用いる。また $\varepsilon = 1.0 \times 10^{-5}$ とした。

このように ε を正の値設定することはフロアリング、または底上げと呼ばれる。フロアリングは訓練データの量が不十分な場合に良く用いられる手法であり、種々のバリエーションがある [14]。全ての n-gram の存在を許すフロアリングは、n-gram が存在し得るか否かだけが制約となる binary n-gram[35, 36] の考え方とある意味では対立するものであり、直感的にも実際とそぐわない印象がある。したがって、この手法は被覆率が低い場合でのみ用いられる便宜上のものである。被覆率が十分大きければフロアリングは不要となるが、サンプルが非常に少ない場合の HMM による音素認識において、フロアリングが逆効果となることも報告されており [37]、どのような場合に有効であるかは一概には言えない。

2.4.2　削除補間法

手法に差はあれ、スムージングは結局訓練サンプル中に一度も出現しなかった n-gram に何らかの正なる値を割り当てることに他ならない。この値をある定数としたのが前出のフロアリングであった。この値を、訓練サンプル中に出現した既

知の n-gram を用いて推定することも考えられる。最も現実的な方法の一つは、低次の n-gram から情報を得ることであろう。図 2.4 からもわかるように、n が小さければ被覆率は向上し、より信頼できる統計量が得られる。ただし、その分情報量は少なくなるから推定精度は落ちる。したがって、低次の n-gram を併用する場合はその手法が重要である。発見的な手法ではかえって精度を落とすことにもなりかねない [27]。

このように、複数の確率モデルを併用したモデルを最大尤度基準に基づいて最適化する手法として、削除補間法 [38] と呼ばれるものが連続音声認識の分野において知られている。最大尤度基準とは、できるだけ未知テキストを生成しやすい (生起確率が高い) ようにモデルをつくることであり (最も生成しやすいモデルは自然言語ソースである)、具体的には後述するパープレキシティを近似的に最小化することである。削除補間法の特徴は、モデル学習用のサンプルのセットと未知サンプルのセットを別々に用意するのではなく、学習用サンプルの一部を「削除」して未知サンプルとみなすことを学習サンプルの全部分について繰り返すことによって、見かけ上大量の未知サンプルを使っているのとほぼ同じになることである。このような仮想的な未知サンプルによる評価は、一般に擬似的なオープン実験のためにしばしば用いられる技術であり、クロスバリデーションという名で呼ばれることがある。

ここでは、n-gram モデルを次式によってスムージングし、未知パラメータである各項の重み係数を訓練テキストによって最適化することを考える。

$$\begin{aligned} & \hat{P}(C_i|C_{i-2}C_{i-1}) \\ \stackrel{\text{def}}{=} \; & \lambda_3\overline{P}(C_i|C_{i-2}C_{i-1}) + \lambda_2\overline{P}(C_i|C_{i-1}) + \lambda_1\overline{P}(C_i) + \lambda_0\overline{P_0} \end{aligned} \tag{2.9}$$

ただし $\lambda_3 + \lambda_2 + \lambda_1 + \lambda_0 = 1$. また $\overline{P_0}$ は zerogram 確率と呼ばれる定数 P_0 の推定値であり、P_0 は次式で定義される。

$$P_0 = \frac{1}{\sum_{C\in\mathcal{A}} 1} \tag{2.10}$$

ところが、日本語の文字言語の場合この式の分母を正確に知る方法はない。本論文では、これを訓練テキスト中に出現した字種数とした上で $\overline{P_0}$ を定義する。$\overline{P_0} > 0$ とすることによって、被覆率は 100%となる。式 (2.9) は明らかに確率の定義である式 (2.8) を満たす。

式 (2.9) は、重み係数 λ_3 の確率で trigram 確率に、λ_2 で bigram 確率に (以下同じ) それぞれしたがって文字系列を生成するようなマルコフ情報源を表している。

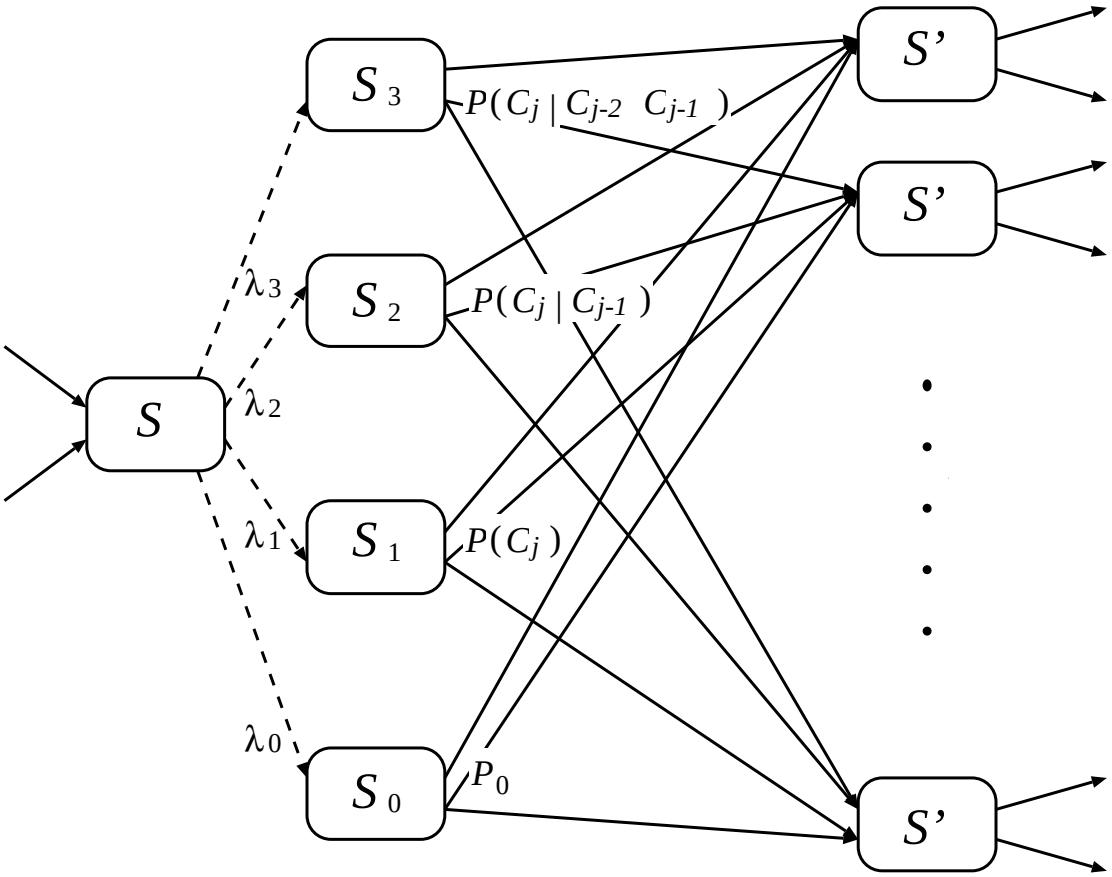

図 2.5: trigram モデルを bigram, unigram, zerogram によって内挿した隠れマルコフモデル

これを図で表すと、図 2.5 のようになる。$\lambda_3 = 1$ であれば、式 (2.1) の通常の 2 重マルコフ過程と同じである。ここでは trigram モデルの状態 S_3 の他に、遷移確率が直前の 1 文字だけによって決まる bigram 状態 S_2, 及び文脈によらない unigram 状態 S_1, zerogram 状態 S_0 を導入して、状態空間を補間している。破線で示すアークは出力のないヌル遷移である。通常のマルコフモデルと違い、このようにヌル遷移を含むマルコフ情報源からの通報は、状態遷移と 1 対 1 に対応しないことから、隠れマルコフモデル (HMM) と呼ばれる。このようなモデルのパラメータ推定を行う手法として、以下で述べる Baum-Welch のアルゴリズム Baum-Welch アルゴリズムが知られている。Baum-Welch アルゴリズムは、HMM のパラメータ推定のために必要な状態遷移確率の計算を前向き (Forward) 確率および後向き (Backward) 確率を用いることで高速に求めるアルゴリズム、および EM アルゴリズムと呼ばれる山登り法によるパラメータの準最適値計算法から成り、Forward-Backward ア

ルゴリズムとも呼ばれる。Baum は一般のパラメータ推定問題について次の定理を証明した [14]。

$$L(\tilde{\boldsymbol{\theta}};\boldsymbol{y}) \geq L(\boldsymbol{\theta};\boldsymbol{y}) \tag{2.11}$$

等号成立の必要十分条件は $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$ である。ここで、$\tilde{\boldsymbol{\theta}}$ は確率変数の実現値 (観測値) の組 $\boldsymbol{y}$ と初期パラメータの組 $\boldsymbol{\theta}$ とを用いて再推定されたパラメータの組である。また L は尤度関数、すなわち最尤推定における目的関数である。この定理は、ある初期パラメータを元にパラメータ自身を最尤推定し、推定されたパラメータを元にまたパラメータの推定を行うという繰り返し計算により、尤度関数が局所的最大値に収束することを意味している。この繰り返しプロセスは EM アルゴリズムと呼ばれ、音声認識における HMM の推定のみならず、画像工学など多方面で広く用いられている。

式 (2.9) の HMM の推定においては $\boldsymbol{\theta} = \{\lambda_3, \lambda_2, \lambda_1, \lambda_0\}$ であり、これらの係数を含む式 (2.9) を用いて $\boldsymbol{\theta}$ を再推定する。今なすべきことは観測系列を最も良く生成するようにモデルのパラメータを決定することであるから、尤度関数は十分長い観測系列 $\boldsymbol{C} = C_1C_2\cdots C_k$ をモデルが生成する確率

$$\hat{P}(\boldsymbol{C}) = \prod_{i=1}^{k} \hat{P}(C_i|C_{i-2}C_{i-1}) \tag{2.12}$$

である。観測系列に trigram $C_{j-2}C_{j-1}C_j$ が現れたとき、直前に $S_0, \ldots, S_3$ のそれぞれの状態を通過してきた確率は、Bayes の定理を用いると

$$P(S_i|S, C_j) = \frac{P(S_i|S)P(C_j|S, S_i)}{\sum_{m=0}^{3} P(S_m|S)P(C_j|S, S_m)} \qquad (i = 0, 1, 2, 3) \tag{2.13}$$

となり、$\lambda_i = P(S_i|S)$ と式 (2.5), (2.6), (2.9) を用いると、次のように推定できる。

$$\hat{P}(S_3|S, C_j) = \frac{\lambda_3\overline{P}(C_j|C_{j-2}C_{j-1})}{\hat{P}(C_j|C_{j-2}C_{j-1})} \tag{2.14}$$

$$\hat{P}(S_2|S, C_j) = \frac{\lambda_2\overline{P}(C_j|C_{j-1})}{\hat{P}(C_j|C_{j-2}C_{j-1})} \tag{2.15}$$

$$\hat{P}(S_1|S, C_j) = \frac{\lambda_1\overline{P}(C_j)}{\hat{P}(C_j|C_{j-2}C_{j-1})} \tag{2.16}$$

$$\hat{P}(S_0|S, C_j) = \frac{\lambda_0\overline{P_0}}{\hat{P}(C_j|C_{j-2}C_{j-1})} \tag{2.17}$$

式 (2.13) が λ_i そのものであったことに注意し、$\hat{P}(S_i|S, C_j)$ を trigram $C_{j-2}C_{j-1}C_j$ の関数として $\hat{\lambda}_i(C_{j-2}C_{j-1}C_j)$ と書き直すと、系列 $\boldsymbol{C}$ 全体を用いて最尤推定量は次のように得られる。

$$\hat{\lambda}_i = \frac{1}{k}\sum_{j=1}^{k} \hat{\lambda}_i(C_{j-2}C_{j-1}C_j) \qquad (i = 0, 1, 2, 3) \tag{2.18}$$

これらを用いて EM アルゴリズムを実行することにより、最適な重み係数が得られる。なお、式 (2.9) の形式の補間においては、局所的最大値が存在しないため、補間係数の初期値に関わらずこのアルゴリズムによって最適解が得られることが知られている。

しかし、注意すべきことは、この状態空間の補間は訓練サンプルを十分に取ることができないために導入したものであるから、$\boldsymbol{C}$ は既知テキストでなく未知テキストでなければならない点である。仮に $\boldsymbol{C}$ を n-gram 確率を推定するために用いた訓練テキストと同じものとすると、このモデルは通常の最尤推定式 (2.5) と同じになってしまう。なぜなら、最尤推定によるパラメータは既知テキストの生起確率を最大にするからである。真に未知のテキストは予め知ることができないから、既知テキストの一部を補間のために確保しておき、それを含まないテキストから n-gram 確率を推定することが考えられる。これは held-out 補間法 [38] と呼ばれ、確保しておいたテキストを held-out データと呼ぶ。held-out 補間法には、次の 2 つの問題点がある。

- n-gram 確率の推定のためのテキストと補間係数の推定のためのテキストのバランスは非常に重要だが、これを決める良い基準がない。
- もともと少数である訓練テキストが、補間のためにさらに削られてしまう。

そこで、訓練テキストの一部を順番に削除し、その部分を未知テキストとみなして補間係数を推定するという手法が考えだされた。これが削除補間法である。本論文では、削除する単位を文とし、句点「。」によって訓練テキストを分割した。訓練テキスト $\mathcal{T}$ は文章の集合であるが、これを一列につなげて文の系列 $\boldsymbol{S}_1\boldsymbol{S}_2\cdots\boldsymbol{S}_l$ とする。m 番目の文に関する式 (2.18) の補間係数の再推定値を $\hat{\lambda}_{im}$ と表記する。この値を計算するときに用いる n-gram 確率は $\mathcal{T}$ から $\boldsymbol{S}_m$ を削除したものによって求める。最終的に、次の式によりパラメータの更新を行えば良い。

$$\hat{\lambda}_{i\mathrm{D.I.}} = \frac{1}{l}\sum_{m=1}^{l} \hat{\lambda}_{im} \qquad (i = 0, 1, 2, 3) \tag{2.19}$$

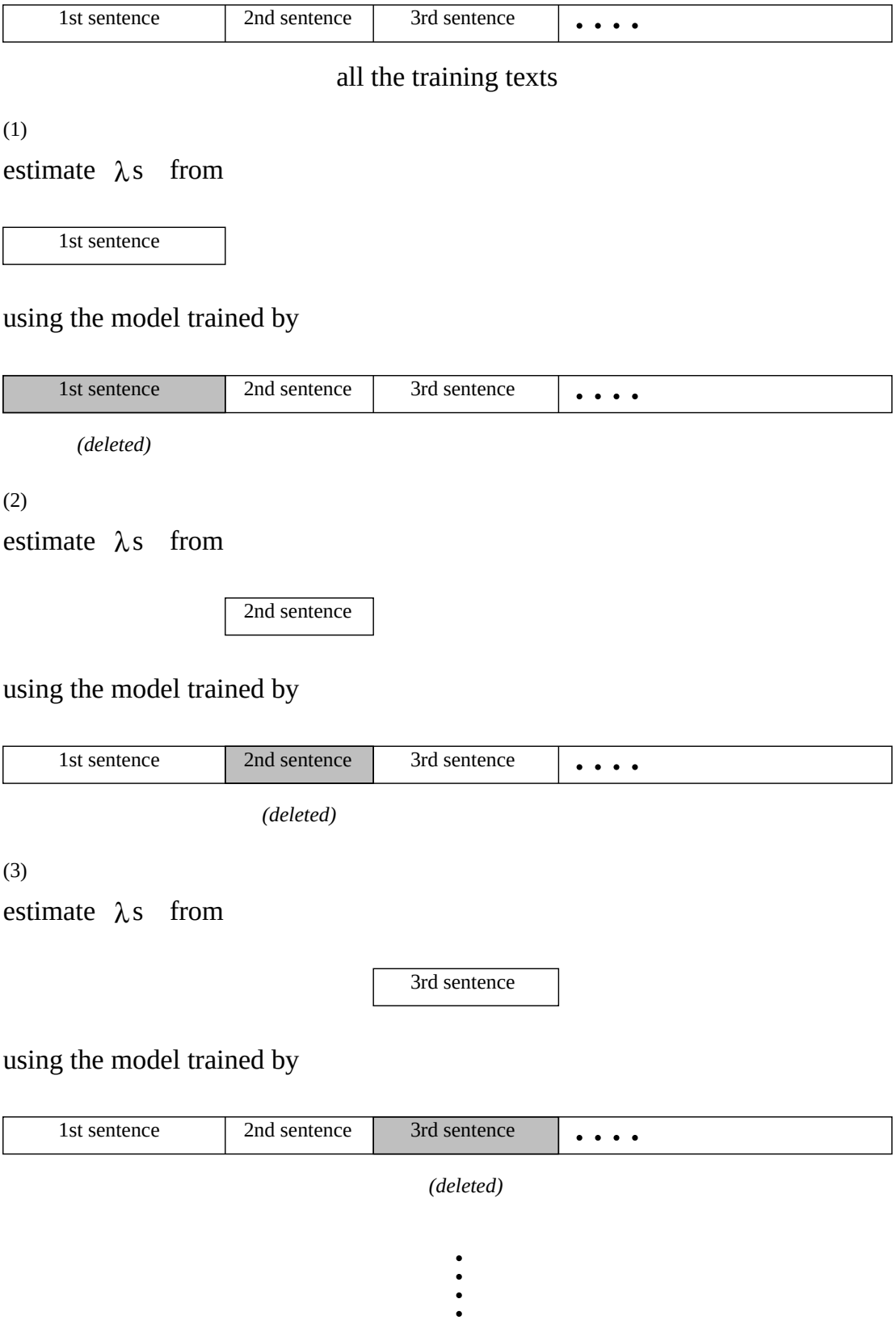

図 2.6: 削除補間法の処理の概要

以上の操作の概念図を図 2.6 に示す。なお、ここで説明したスムージング手法は、正確には「trigram 確率を bigram 確率などとの線形形式で表し削除補間法によって重みを推定する手法」とでも呼ぶべきものであるが、簡単のために以下では単に「削除補間法」と呼ぶ。

実際に、訓練テキストとして (A) の社説を用いて削除補間法を適用し、重み係数を算出した。その結果を図 2.7 に示す。サンプルが少なくなり、統計的な信頼性が低くなるのと共に、trigram 確率に代わって bigram 確率の重みが増加しているのがわかる。

2.5　スムージング法の比較及び検討

この節では、前節で説明した 2 種類のスムージング手法を適用した日本語文字 trigram モデルの評価を行う。前述した通り、n-gram モデルで最も重要なことは少数データ問題である。したがって、モデルの評価は未知テキストに対して行われなければならない。また、対象とする認識タスクの設定の仕方によってその評価は左右される。

2.5.1　パープレキシティ

確率的言語モデルの曖昧性解消能力の目安として良く用いられる尺度に、パープレキシティと呼ばれるものがある。これは情報理論的な意味での平均分岐数であり [14]、定性的には、ある言語モデルが真の言語の確率分布からどれだけ離れているかの指標である。言語 L の (真の) エントロピーを $H(L)$ と書く。ある言語モデルが言語 L と同じ確率分布を持つなら、その言語モデルのパープレキシティは $2^{H(L)}$ であり、これがパープレキシティの下限である [34]。

L がエルゴード性のマルコフ過程であれば、L の十分長い系列 $\boldsymbol{C} = C_1 C_2 \cdots C_k$ から、1 文字あたりのパープレキシティ PP は次式で推定できる。

$$PP \simeq \hat{P}(\boldsymbol{C})^{-1/k} \tag{2.20}$$

本論文で扱うスムージングされた文字 trigram モデルであれば、$\hat{P}(\boldsymbol{C})$ は式 (2.7), (2.9), (2.12) で定義される。これらの式からもわかるように、パープレキシティはある言語モデルを使って推定された L の 1 記号あたりの平均生起確率の逆数である。L の冗長度が大きい (事前に予測しやすい) ほど、また言語モデルの精度が高

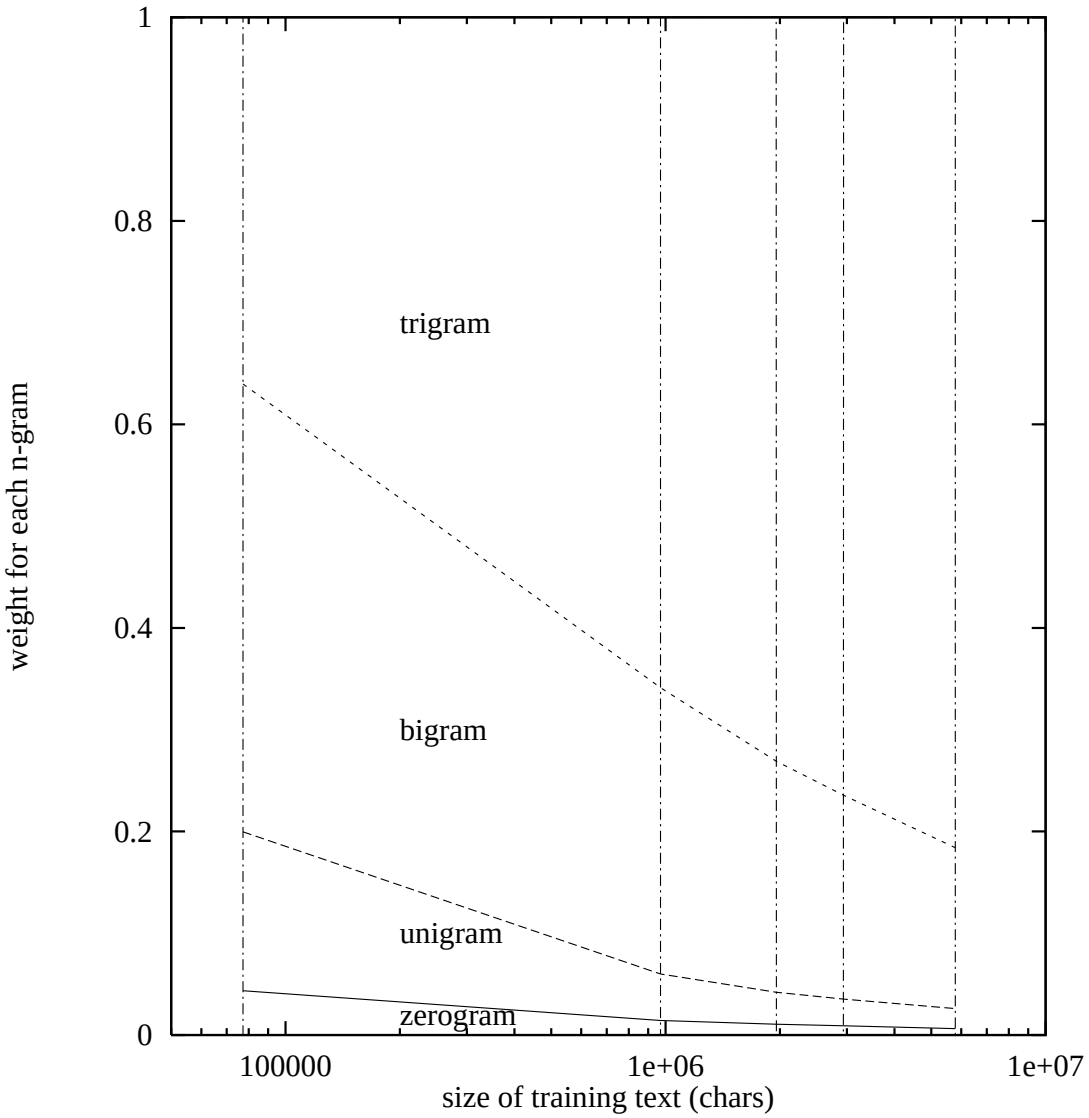

図 2.7: 訓練テキストの量と削除補間法による重み係数の関係

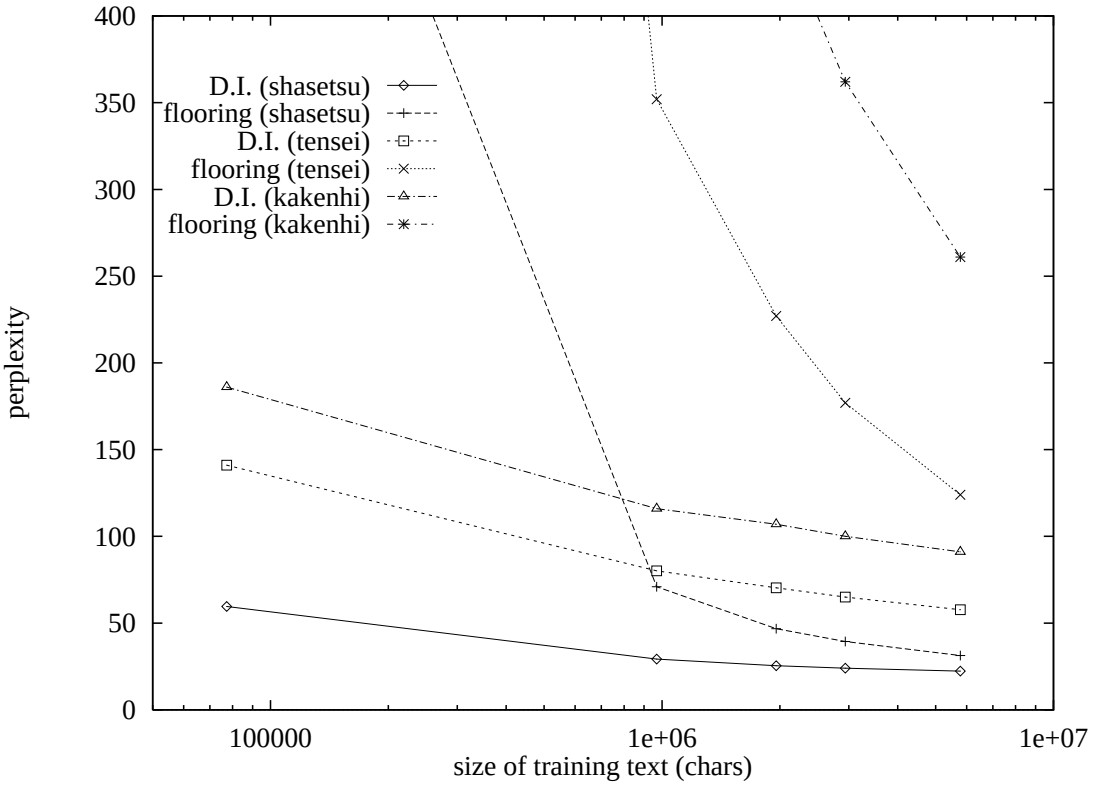

図 2.8: 「社説」モデルのパープレキシティ

い (確率分布が似ている) ほどパープレキシティは小さくなる。連続音声認識の分野ではパープレキシティが認識率と強い相関があることが知られ、言語モデル設計の有力な指標となっている。

定義から、パープレキシティは L の領域に依存する量である。本研究では n-gram モデルによってできるだけタスク依存性の少ない頑強な言語モデルを得ることを目的としているが、モデルを構成するための訓練データは L の特定領域からの標本にならざるを得ない。したがって、言語モデルの評価にあたっては、モデルを構成するためのサンプルと同一のタスク、および異なるタスクの両方を考慮すべきである。

訓練データとして「社説」(A) を用いた日本語文字 trigram モデルに対し、前章で導入したフロアリングと削除補間法の 2 種類のスムージングを適用してそれぞれ言語モデルを構成した。これらの言語モデルのパープレキシティを、「社説」(B),「天声人語」(D),「科研費概要」(E) をそれぞれ評価テキストとして求めた。図 2.8 にその結果を示す。横軸は、訓練テキストの量である。図中の凡例で、D.I. とあ

るのは式 (2.9) の削除補間法によるモデル、flooring とあるのは式 (2.7) のフロアリングによるモデルである。スムージングを行わない場合には、前述したように系列全体の生起確率が 0 になるため、パープレキシティは定義できない。(生起確率を 0 に近づければ、パープレキシティは無限大に近づく。)

図 2.8 について考察する。まず言語モデルが利用できない場合を考える。言語モデルを利用しないのなら、系列の事前確率分布の最尤推定は全シンボルの生起確率を等しくすることで与えられる。これは zerogram とも呼ばれる。この場合のパープレキシティはシンボル数に等しく、本研究で扱う日本語文字の場合には約 3000 である (ただし実際に出現が確認されたものに限る)。言語モデルのパープレキシティの評価は、シンボル数に比べどの程度小さいかが一つの目安となる。

評価テキストが「社説」(B) の場合を見てみると、全訓練テキストを用い、さらに削除補間法を適用したモデルではパープレキシティは 22.3 となった。これは、仮に全ての文字の認識が同程度に難しいとすることができるならば、この言語モデルを用いることによって、カテゴリ数約 3000 が等価的に約 22 にまで減少することを意味している [14]。このことから、削除補間法を適用した trigram モデルは強い曖昧性解消能力を持つことが裏付けられる。また、フロアリングを適用したモデルではパープレキシティは 31.3 であり、この訓練テキストの量では削除補間法によるスムージングの方が効果的であると言える。さらに、訓練テキストが少なくなると、2 種類のスムージング手法による差が非常に大きくなる。このことから、削除補間法は特に訓練サンプルが少ない状況において適切な n-gram モデルを構成する能力があることがわかる。

次に、認識タスクの性質が言語モデルの訓練に用いたサンプルと異なる場合について検討する。評価に用いたテキストは、「天声人語」(D) と「科研費概要」(E) である。「天声人語」は随筆であり、比較的自由な文体で書かれ、しかも取り上げる話題が多彩である。「科研費概要」は専門用語を豊富に含む。全訓練テキストを用いた場合について見てみると、削除補間法を用いたモデルのパープレキシティは「天声人語」で 57.7,「科研費概要」で 91.1 であった。このことから、訓練サンプルと異なる領域に属するテキストに対しても、ある程度の曖昧性解消の効果が期待できそうである。しかし、「天声人語」に対するパープレキシティは「社説」に対するそれの 2 倍以上であり、「科研費概要」に対してはさらに大きくなる。これは、訓練サンプルの確率分布が評価サンプルの確率分布から外れていることを示している。本研究が扱う印刷文書中には、このように訓練サンプル中には現れなかった多彩な表現が出現することも考えられ、確率分布が異なる対象の認識精

度の低下も予想される。

対象が「天声人語」「科研費概要」である場合のスムージングの効果を調べてみると、ここでは削除補間法の優位性が明らかとなる。すなわち、図 2.8 が示すように、削除補間法によるモデルとフロアリングによるモデルのパープレキシティの差は、訓練テキストが少ない時、及び認識タスクがより困難である時に顕著となる。タスクと言語モデルの関係の概念図を図 2.9 に示す。

図 2.8 では、訓練サンプルと認識タスクの確率分布の違いによるタスクの困難さが評価された。図 2.10 は、タスク自身の困難さを評価するために、「天声人語」(C) を訓練テキストとして用い、構成した言語モデルのパープレキシティを、「天声人語」(D) を用いて評価した結果である。この図から、「天声人語」は「社説」に比べ本質的に困難なタスクであるということが言える。

2.6　まとめ

本章では、文字認識のための日本語の確率モデルとして文字の n-gram モデルを用いること、また n-gram モデルにスムージングを導入することを提案し、実験により、削除補間法によってスムージングされたモデルが曖昧性の削減に有効であることを明らかにした。

削除補間法は実験で示した通り、訓練テキストが少ない時の言語モデルの性能の低下を抑える有効な手段である。しかし、自動学習が可能であることを割り引いてもこの言語モデルが最良のものであるとは言えない。式 (2.9) の形の確率モデルの欠点として、一度でも出現した trigram についてはほぼ無条件に信頼されてしまうことが挙げられる。Katz[33] のように、統計的信頼性を加味することによって、より適切なスムージングが行える可能性がある。

本章では困難なタスクに対する曖昧性解消性能の低下を指摘したが、この問題に関する対策である n-gram モデルの適応化手法とその認識精度への影響については本論文で触れるゆとりがない。筆者の過去の研究として、文献 [40] を挙げておく。) 知識に依存しない言語モデルは、適応化に関しては有利な点を多く持っている。特に、事前の人手による分析が許されない適応データからの教師なし適応のためには、知識に依存しない言語モデルが必須である。文字の n-gram は、知識に依存しない言語モデルの基礎として、適応においても重要な役割を持つことになるだろう。

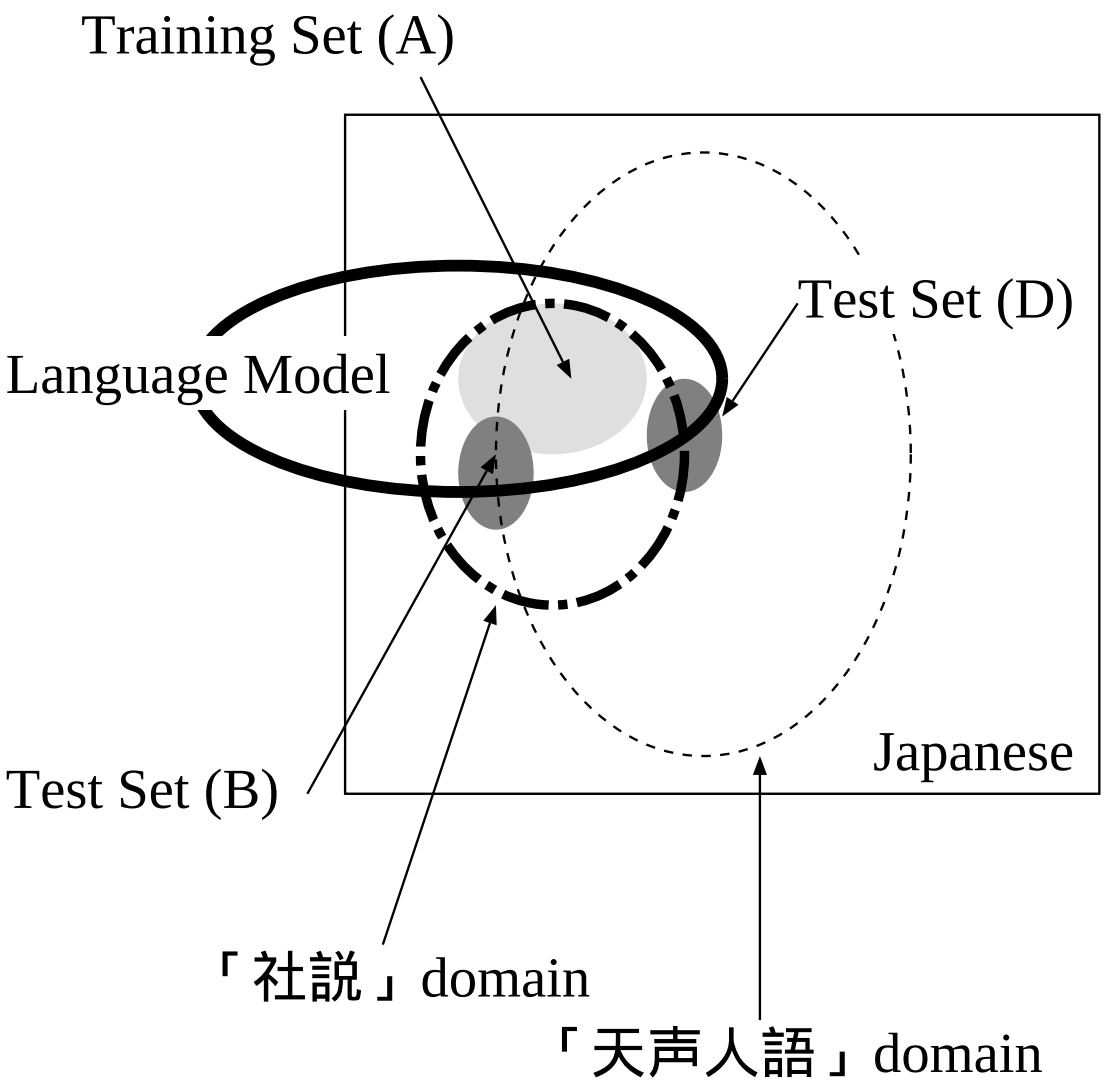

図 2.9: タスク間の関係と言語モデル

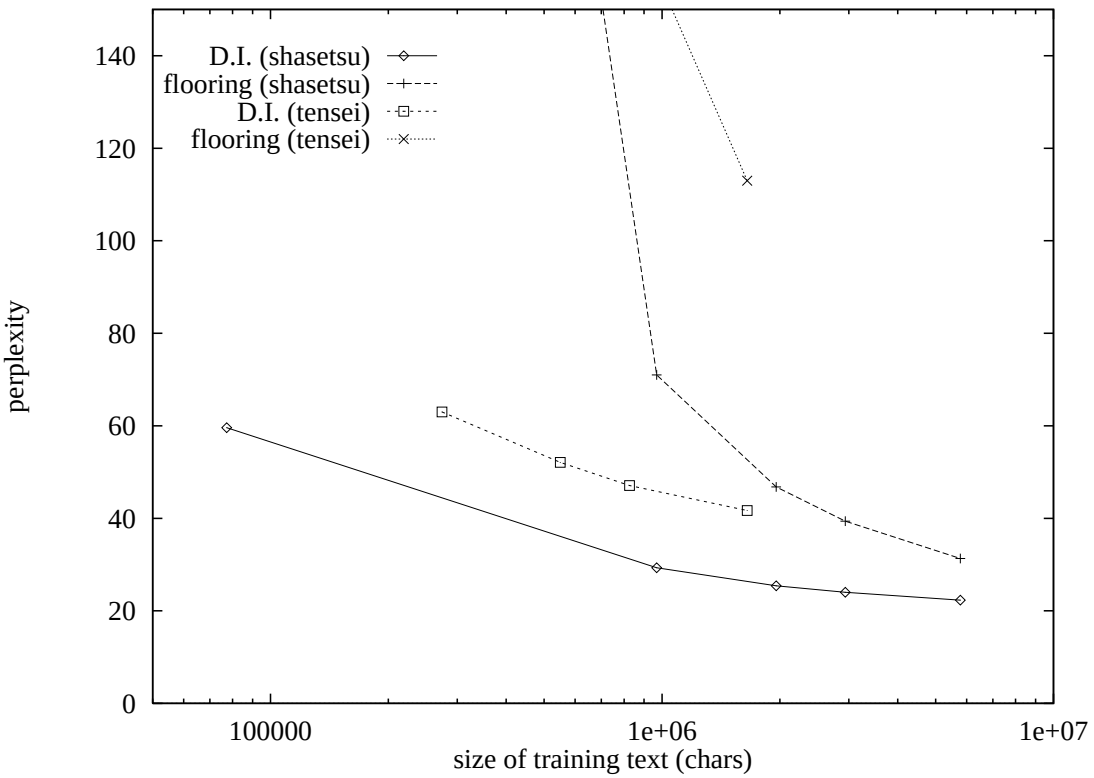

図 2.10: 「社説」モデルと「天声人語」モデルのパープレキシティによる比較

第3章　文字と単語の連鎖確率の融合に基づく言語モデル

3.1　はじめに

序論でも述べたように、文字認識・音声認識などにおける曖昧性の削減のために、近年コーパスに基づく言語モデルがさかんに研究されている。テキストコーパスの形態的分類は序論で述べたが、その中でもタグ付きコーパスは人手による形態素解析を施されたものであり、言語モデルを構築する上で種々の望ましい性質を備えている。また、分析済みコーパスではさらに構文・意味的情報をも利用することができるため、人間の高度な情報処理により近い、複雑な言語モデルの獲得に役立つと考えられる。

しかし、EDR コーパス [7] のようなタグ付き、あるいは解析済みの日本語文コーパスはまだ広い分野で使えるように整備されてはいない。また、日本語の処理のための形態素や文法として利用できる体系には標準と呼べるものが存在しないため、異なった文法体系に基づいたコーパスを同時に用いることが難しい。

これに比べ、タグのないべた書きのテキストは、文章を計算機上で処理する場合の基本的な形式であるために、さまざまな分野のものがコーパスとして利用できることが期待される。このような理由から、近年ではべた書きテキストをコーパスとして利用する自然言語処理技術が注目されている [41, 12]。

コーパスに基づく言語モデルの中でも、n-gram モデルは数え上げを基本とする単純な構成法によって獲得できるため、広く用いられている。英語などのヨーロッパ系言語では、単語を単位とした n-gram が良く用いられる [16]。これらの言語においては、単語間の区切りが明確に与えられるため、数え上げのみで容易に単語 n-gram を得ることが可能である。

一方、日本語においては分かち書きの習慣はなく、単語の概念も曖昧である。このため、単語 n-gram を構成するためには初めから単語に分割されているコーパスを用いるか、あるいはべた書きのコーパスを何らかの方法で予め分割する必要が

ある。

テキストの分割に自動形態素解析システムを用いると、ヨーロッパ系言語と同様な方法によって単語 n-gram を構成することができるようになる。しかし、実際の文字認識や音声認識が対象とする文章は一般に未知であり、コーパスから単純に構成した単語 n-gram モデルでは尤もらしいと判断できない文が出現する。尤もらしさの判断を困難にする要因として、いわゆる未知語の存在が挙げられる。未知語とはモデルで想定していない単語で、次のように分類できる。

- 形態素解析システムが持つ語彙に含まれていない語
- 学習用コーパスに含まれていない語

未知文章に形態素境界が与えられていない場合には、単語 n-gram モデルを用いた未知文章の生起確率の計算は、形態素区切りの可能性の中から確からしいものを推定しながら行う必要があるが、未知語が存在する場合には、その語および周囲の語の解析精度が低下し、確率計算に悪影響を与える。単語を単位とした日本語のモデル評価においては、未知語の扱いが難しいため、評価テキストから未知語を含む文を排除する場合もある [22]。しかし、現実のアプリケーションにおいて未知語の存在は本質的であることが多く、そのような場合にはモデル評価も未知語が存在しないことを仮定しない方が適当である。

未知語の発生の原因の 1 つは、モデル学習用のコーパスが少ないことである。しかし、コーパスの量を増やすことには限界がある。少数サンプルの問題を緩和するため、通常はモデルにスムージングを施す。第 2 章でも述べたように、n-gram モデルのスムージング法としては、確率の重み付き線形和による方法と、back-off による方法 [33] が良く用いられている。しかし、単語 n-gram モデルの未知語の問題に対しては、いずれのスムージング法においても、未知語の周囲の語の誤解析までは考慮されていない。

日本語文の確率モデルとして、他に第 2 章で取り上げた、文字を単位とする n-gram モデルがある。このモデルはべた書きテキストコーパスから形態素解析をすることなく構築することができる点で優れている。しかし、一般に n-gram モデルにおいては n の指数オーダーのパラメータを推定せねばならない。このため、もっと強い制約を持つモデルを必要とする場合であっても、単純に n を大きくするだけでは十分な効果が得られない。

本章では、べた書きテキストから構成することのできる、文字認識・音声認識用の強力な言語モデルの開発を目的とし、これを実現する日本語文字列の確率モデ

ルとして、単語 n-gram と文字 n-gram を融合した新しいモデルを提案する。このモデルは、少数サンプルが原因で起こる未知文章の単語解析精度の低下を文字 n-gram が補うように構成されている。このため、単語 n-gram が有効に働く部分以外の所でも文字 n-gram に近い性能が維持される。後で述べるように、この効果は単語 n-gram のスムージングだけでは得ることができないものである。更に本章では、文字・単語を融合した n-gram モデルの学習法を示すと共に、本手法によって高い曖昧性削減能力を持つモデルが得られることを、パープレキシティを基準とした性能評価実験により示す。

3.2　関連する研究

単語 n-gram の推定誤差を軽減するための過去の研究として、いくつか関連するものを挙げる。連続音声認識においては、間投詞などの不要部分をスキップするために、単語 n-gram と不要部分を表す音素 n-gram のいずれかを選択する言語モデルによって得られた尤度を最大にするパスから不要部分を同定する方法が提案されている [42]。また、辞書との単語照合に基づく文字認識においては、未知語が周囲の語の解析に悪影響を与えるのを防ぐため、未知語と思われる範囲を同定するためのモデルが提案されている。高尾ら [43] はヒューリスティックな未知語テンプレートを用いている。Nagata[44] は、未知語を特別なカテゴリと考えた品詞 n-gram を用いて、未知語の確率を文字 n-gram モデルによって与えている。本論文で提案するモデルがこれらの手法と最も異なる点は、不要部分や未知語の位置の同定を行わないことである。これらの手法では未知語などの位置を同定しなければならないが、これを高い精度で行うのは難しい [45]。また、これらの手法では未知語などに認定されなかった部分に関する文字 n-gram モデルのスムージング効果は全くない。これは、タグのないコーパスを自動形態素解析して訓練テキストとして用いる場合などで、単語 n-gram の推定精度そのものが悪い場合には不利である。

これに対し、本手法では訓練テキストに出現した単語のみを用いて未知テキストの分割を行う。このため、未知語部分を同定するという困難な問題は回避される。また、すべての単語の発生確率が単語 n-gram と同時に文字 n-gram によって与えられる。このため、単語 n-gram の推定における統計的な困難が推定の容易な文字 n-gram により解消され、頑健で強力な言語モデルを得ることができる。本手

法は形態素解析に依存しないため、次章で述べるような形態素解析以外の方法に基づく n-gram にも容易に適用できることも利点の一つである。更に、タスクに応じて単語 n-gram と文字 n-gram のバランスを自動で調整することができる点でも有利である。

3.3　言語モデルの定式化

ここでいう言語モデルとは、文字の系列 $\boldsymbol{C} = C_1 C_2 \cdots C_k$ に対してその生起確率 $P(\boldsymbol{C})$ の推定値 $\hat{P}(\boldsymbol{C})$ を与えるものである。

3.3.1　文字 n-gram モデル

第 2 章では文字の n-gram モデルについて詳細に述べたが、定式化の都合上、ここで再定義を行う。文字 n-gram 確率とは、直前 $n-1$ 文字のコンテキストに関する、文字の条件付き生起確率 $P_{\mathrm{c}}(C_i|C_{i-(n-1)}\cdots C_{i-2}C_{i-1})$ である。文字 n-gram モデルは、$P(\boldsymbol{C})$ を文字 n-gram 確率の積で推定したものである。すなわち、

$$P(\boldsymbol{C}) \simeq \prod_{i=1}^{k} P_{\mathrm{c}}(C_i|C_{i-(n-1)}\cdots C_{i-2}C_{i-1}). \tag{3.1}$$

文字 n-gram 確率の過学習の影響を低減するためのスムージングの方法の一つとして、より低次の n-gram 確率を併用し、全体の確率を複数のコンテキスト下での確率の重み付き線形和で表す方法がある。

$$\begin{aligned} & \hat{P}_{\mathrm{c}}(C_i|C_{i-(n-1)}\cdots C_{i-2}C_{i-1}) \\ \equiv\ & \sum_{j=0}^{n} \lambda_{\mathrm{c}\,j} \overline{P}_{\mathrm{c}\,j}(C_i|C_{i-(n-1)}\cdots C_{i-2}C_{i-1}) \end{aligned} \tag{3.2}$$

但し $\sum_{j=0}^{n} \lambda_{\mathrm{c}\,j} = 1$、また

$$\begin{aligned} & \overline{P}_{\mathrm{c}\,j}(C_i|C_{i-(n-1)}\cdots C_{i-2}C_{i-1}) \\ =\ & \begin{cases} \overline{P}_{\mathrm{c}}(C_i|C_{i-(j-1)}\cdots C_{i-2}C_{i-1}) & \text{if } j > 1 \\ \overline{P}_{\mathrm{c}}(C_i) & \text{if } j = 1 \\ \overline{P}_{\mathrm{c}0} \quad (\text{const.}) & \text{if } j = 0 \end{cases} \end{aligned} \tag{3.3}$$

は文字 j-gram 確率の最尤推定値である。第 2 章で述べたように、削除補間法 [38] を適用することにより、このような複数の n-gram モデルを併用する場合の重み

係数を、未知テキストの生成確率が近似的に最大となるように決定することができる。

3.3.2　単語 n-gram モデル

ここでいう単語とは、任意の長さの文字列である。各単語は品詞や活用形などの属性を持たない。単語 n-gram モデルは単語集合をパラメータの一部として持つ。与えられた文字列 $\boldsymbol{C}$ は、単語の列 $w_1 \cdots w_l$ に分割できる。この分割は一般には一意ではない。分割のひとつ $w_1 \cdots w_l$ を $\boldsymbol{C}$ のひとつの解析結果と呼び、$w_1 \cdots w_l \in \boldsymbol{C}$ と書く。分割が一意でないことは、単語境界が曖昧であることを意味している。

単語 n-gram モデルは、$P(\boldsymbol{C})$ を $\boldsymbol{C}$ の解析結果ごとの単語生起確率の積の総和で推定するものであり、次式で与えられる。

$$P(\boldsymbol{C}) \simeq \sum_{w_1 \cdots w_l \in \boldsymbol{C}} P(w_1 w_2 \cdots w_l) \tag{3.4}$$

$$\simeq \sum_{w_1 \cdots w_l \in \boldsymbol{C}} \prod_{i=1}^{l} P_{\mathrm{w}}(w_i | w_{i-(n-1)} \cdots w_{i-2} w_{i-1}) \tag{3.5}$$

但し、$P_{\mathrm{w}}(w_i | w_{i-(n-1)} \cdots w_{i-1})$ は単語 n-gram 確率であり、直前に $w_{i-(n-1)} \cdots w_{i-1}$ が生起したと仮定した時の単語 w_i の条件付き生起確率である。観測系列が分かち書きされていない場合には単語列は一意に決まらないため、単語 n-gram モデルは隠れマルコフモデルの一種であると言える。

単語 n-gram 確率は、訓練テキストを自動形態素解析などによって一意な単語列に分割したものを用いることにより、n 単語の組の出現頻度の計数によって最尤推定できる。この時得られた単語をすべて集めたものを獲得単語集合と呼ぶ。獲得単語集合には、形態素解析の段階で未定義語として扱われた文字列も含まれる。また、訓練テキストに関わりなく、長さ 1 の文字列 (1 文字語と呼ぶ) すべてを獲得単語集合に加えたものを、単語 n-gram モデルの単語集合と定める。この定義から、未知テキストは単語集合に含まれる単語の列に必ず分割できる。この意味で、定義した単語 n-gram モデルは、文字列による n-gram [41] と同様に、いわゆる未知語を陽に扱わないものとなっている。1 文字語の付加により、すべての $\boldsymbol{C}$ に対して少なくとも 1 通りの解析結果が存在することが保証され、スムージングによってすべての文が正の生起確率を持つことになる。例えば、獲得単語集合が { スト } で

ある場合の trigram モデルでは、「テキスト」という文に対する解析結果の集合は

$$\{\,\text{テ/キ/ス/ト}, \text{テ/キ/スト}\,\}$$

(/は単語境界を表す) であるから、

$$\begin{aligned} P(\text{テキスト}) &= P_{\mathrm{w}}(\text{テ}\,|\#)P_{\mathrm{w}}(\text{キ}\,|\#/\text{テ})\cdot \\ &\qquad \{P_{\mathrm{w}}(\text{ス}\,|\,\text{テ/キ})P_{\mathrm{w}}(\text{ト}\,|\,\text{キ/ス}) + P_{\mathrm{w}}(\text{スト}\,|\,\text{テ/キ})\} \end{aligned}$$

と計算される。

線形補間によってスムージングされた単語 n-gram 確率の式を定める。これは、次のようになる。

$$\begin{aligned} & \hat{P}_{\mathrm{w}}(w_i|w_{i-(n-1)}\cdots w_{i-2}w_{i-1}) \\ \equiv\ & \sum_{j=0}^{n} \lambda_{\mathrm{w}\,j}\overline{P}_{\mathrm{w}\,j}(w_i|w_{i-(n-1)}\cdots w_{i-2}w_{i-1}) \end{aligned} \tag{3.6}$$

但し $\sum_{j=0}^{n}\lambda_{\mathrm{w}\,j}=1$,

$$\begin{aligned} & \overline{P}_{\mathrm{w}\,j}(w_i|w_{i-(n-1)}\cdots w_{i-2}w_{i-1}) \\ =\ & \begin{cases} \overline{P}_{\mathrm{w}}(w_i|w_{i-(j-1)}\cdots w_{i-2}w_{i-1}) & \text{if } j>1, \\ \overline{P}_{\mathrm{w}}(w_i) & \text{if } j=1, \\ \overline{P}_{\mathrm{w}0}\quad (\text{const.}) & \text{if } j=0. \end{cases} \end{aligned} \tag{3.7}$$

3.3.3　文字・単語の複合 n-gram モデル

単語 n-gram モデルについても、削除補間法を用いることによって少数サンプルの影響を低減することはできる。しかし、単語の場合にはこれでも不十分であると考えられる。その理由は次のとおりである。

- 未知文章中にいわゆる未知語が存在することを避け得ない
- 単語の種類は文字に比べてはるかに多く、少数サンプルの影響が深刻

単語 n-gram モデルを少数サンプル下でも頑健なものに改善するために、文字 n-gram モデルとの融合を考える。削除補間法は、一般に複数の異なる言語モデルの併用のために適用可能であると主張されている [38]。本論文で提案する複合 n-gram モデルもこの考えに基づいている。

複合 n-gram モデルは、$P(\boldsymbol{C})$ を $\boldsymbol{C}$ の解析結果ごとの複合 n-gram 確率の積の総和で推定するものであり、次式で与えられる。

$$P(\boldsymbol{C}) \simeq \sum_{w_1 \cdots w_l \in \boldsymbol{C}} \prod_{i=1}^{l} \hat{P}_n^m(w_i|w_1 \cdots w_{i-2}w_{i-1}) \tag{3.8}$$

但し $\hat{P}_n^m(w_i|w_1 \cdots w_{i-2}w_{i-1})$ は文字 m-gram、単語 n-gram の複合確率であり、次式で定義される。

$$\hat{P}_n^m(w_i|w_1 \cdots w_{i-2}w_{i-1}) \equiv$$
$$\lambda_{\mathrm{w}}\hat{P}_{\mathrm{w}}(w_i|w_{i-(n-1)} \cdots w_{i-2}w_{i-1}) + \lambda_{\mathrm{c}}\hat{P}_{\mathrm{c}\,m}(w_i|w_1 \cdots w_{i-2}w_{i-1}) \tag{3.9}$$

但し $\lambda_{\mathrm{w}} + \lambda_{\mathrm{c}} = 1$。また、式 (3.9) の右辺第 2 項は単語 w_i が生起する確率を補間された文字 m-gram 確率の積によって求めたものである。すなわち、単語 w_i の先頭から j 文字目を $C_{i,j}$、単語列 $w_1 \cdots w_{i-1}$ の最後尾からの文字を $C_{i,0}, C_{i,-1}, C_{i,-2}, \cdots$ と表記すると、

$$\begin{aligned} &\hat{P}_{\mathrm{c}\,m}(w_i|w_1 \cdots w_{i-1}) \\ = &\left(\prod_{j=1}^{L(w_i)} \hat{P}_{\mathrm{c}}(C_{i,j}|C_{i,j-(m-1)} \cdots C_{i,j-1}) \right) \cdot d(L(w_i)). \end{aligned} \tag{3.10}$$

但し、$L(w_i)$ は単語 w_i の長さを表す。また、$d(\tau)$ は文字モデルが生成する単語の長さに関する分布関数である。ここでは次式の指数分布を仮定する。指数分布の妥当性については 3.5 節で述べる。

$$d(\tau) = {P_{\mathrm{d}}}^{\tau-1}(1 - P_{\mathrm{d}}) \qquad (0 \leq P_{\mathrm{d}} < 1) \tag{3.11}$$

複合 n-gram モデルの状態遷移の模式図を図 3.1 に示す。図中、点線の矢印はヌル遷移を示す。戻るヌル遷移の矢印は、図の上半分の単語モデル部で出力する単語の長さに相当する回数だけ下半分の文字モデル部で同じ文字列を出力することを意味している。

削除補間法によって求めるべきパラメータは、式 (3.2) の $\lambda_{\mathrm{c}\,j}$、式 (3.6) の $\lambda_{\mathrm{w}\,j}$ の他、文字モデルへの重み係数である λ_{c}、および式 (3.11) を支配する P_{d} となる。

3.4　モデルの学習法

3.4.1　単語 n-gram モデルの学習

単語 n-gram 確率の学習は、訓練テキストを予め汎用形態素解析ツール JUMAN[46] によって一意な単語列に分割して行った。JUMAN の語彙数は約 12 万である。JU-

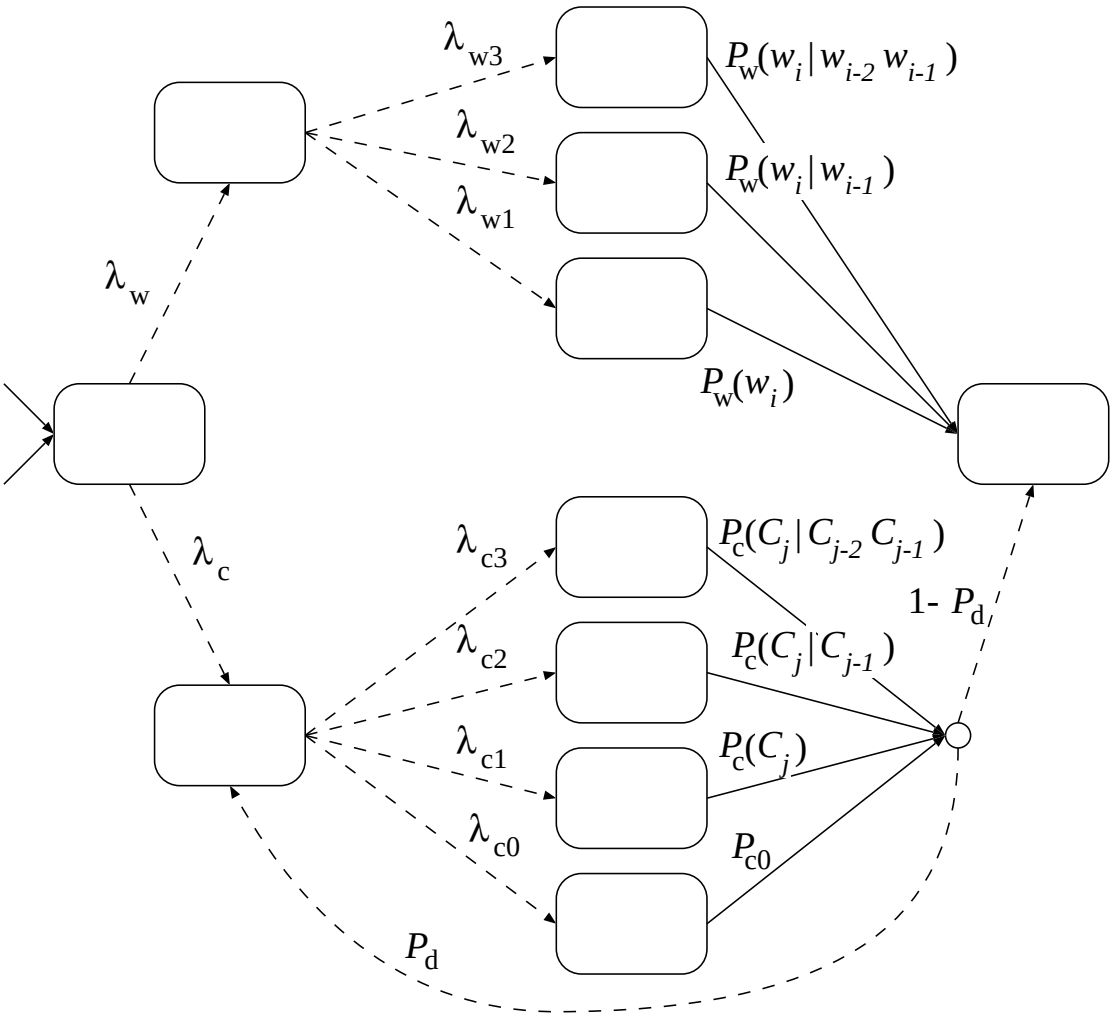

図 3.1: 複合モデルの状態遷移図。ここでは単語 trigram、文字 trigram の例

MAN の辞書やパラメータには修正は加えていない。

3.4.2　複合 n-gram モデルの学習

削除補間法による未知パラメータの学習は、まず訓練テキストをいくつかの部分に分割し、ついで各々の部分に対して訓練テキストからの削除を行ってから新しいパラメータを最尤推定によって求めるという手順で行われる。削除された部分が仮想的に未知テキストの役割を果たす。このため、この部分のテキストについては分かち書きは行われていないものと仮定せねばならない。もし初めから形態素解析の結果得られる単語境界が与えられていると、実際に未知文章を評価する時に未知語の存在が原因で発生する解析精度低下の影響を模倣できず、結果として単語モデルに対して不当に高い評価を与える恐れがあるからである。

パラメータの再推定プロセスではすべての可能な単語列に対して Forward-Backward アルゴリズムを実行する必要があるが、計算コストが大きい。そこで、本論文では未知文章の解析シミュレーションというパラメータ推定法を提案する。この方法では、削除補間法を高速に実行するため、可能な単語列すべてを考慮する代わりに、Viterbi アルゴリズムを適用して削除部分を単語列に分割してからパラメータを再推定する。Viterbi アルゴリズムは、次式で与えられる最適な分割 $w_1 \cdots w_l$ を高速に見出すアルゴリズムである。

$$\underset{w_1 \cdots w_l \in \boldsymbol{C}}{\operatorname{argmax}} \prod_{i=1}^{l} P_{\mathrm{w}}(w_i | w_{i-(n-1)} \cdots w_{i-2} w_{i-1}) \tag{3.12}$$

このパラメータ学習法を図 3.2 に示す。削除された単語列は、一度連結されて単語境界のない文字列となる。この文字列に対して Viterbi アルゴリズムを実行することで、未知テキストを解析した結果得られる単語列をシミュレートできる。文字 n-gram モデルの重み係数 $\lambda_{\mathrm{c}\,j}$ が既に求められているとすると、Viterbi アルゴリズムによって得た単語列 $w_1 w_2 \cdots w_l$ に対する文字モデルの寄与度 $\hat{\lambda}_{\mathrm{c}}$、うち長さ L の単語に関する文字モデルの寄与度 $\hat{\lambda}_{\mathrm{c}(\tau=L)}$、単語 j-gram モデルの寄与度 $\hat{\lambda}_{\mathrm{w}\,j}$ は次のように求められる。

$$\hat{\lambda}_{\mathrm{c}}(w_1 w_2 \cdots w_l) = \sum_{i=1}^{l} \frac{\lambda_{\mathrm{c}} \hat{P}_{\mathrm{c}\,m}(w_i | w_1 \cdots w_{i-2} w_{i-1})}{\hat{P}_n^m(w_i | w_1 \cdots w_{i-2} w_{i-1})} \tag{3.13}$$

$$\hat{\lambda}_{\mathrm{c}(\tau=L)}(w_1 w_2 \cdots w_l) = \sum_{\forall w_i = C_{i,1} \cdots C_{i,L}} \frac{\lambda_{\mathrm{c}} \hat{P}_{\mathrm{c}\,m}(w_i | w_1 \cdots w_{i-2} w_{i-1})}{\hat{P}_n^m(w_i | w_1 \cdots w_{i-2} w_{i-1})} \tag{3.14}$$

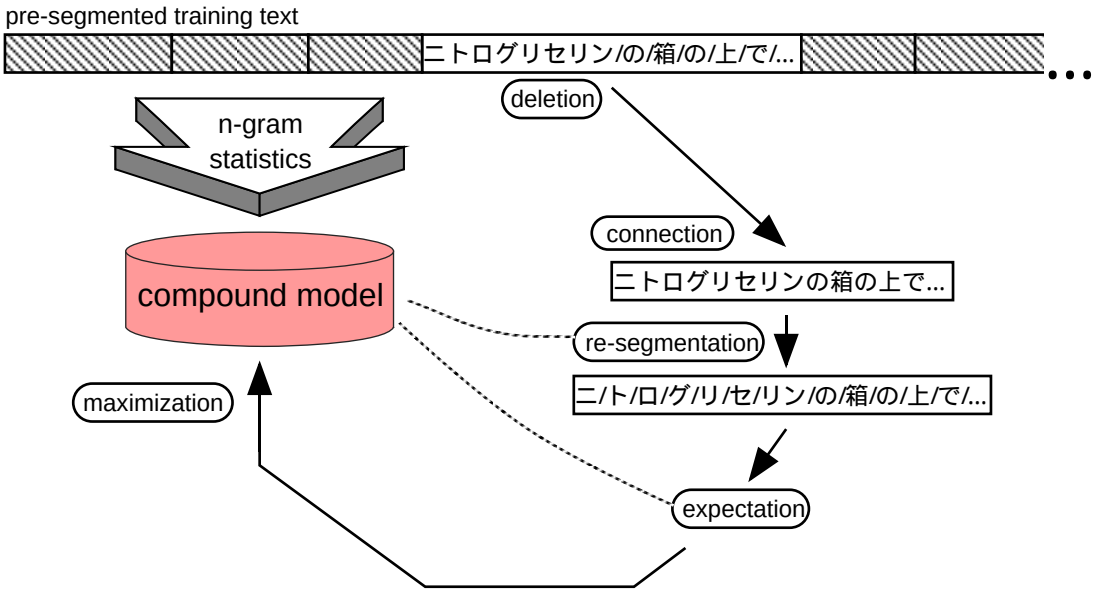

図 3.2: 未知文章の解析シミュレーションによるパラメータ学習法

$$\hat{\lambda}_{\mathrm{w}\,j}(w_1 w_2 \cdots w_l) = \sum_{i=1}^{l} \frac{\lambda_{\mathrm{w}\,j}\overline{P}_{\mathrm{w}\,j}(w_i|w_{i-(n-1)}\cdots w_{i-2}w_{i-1})}{\hat{P}_{\mathrm{w}}(w_i|w_{i-(n-1)}\cdots w_{i-2}w_{i-1})} \tag{3.15}$$

訓練テキスト $\mathcal{T}$ の全ての削除単位 $\mathcal{P}$ から Viterbi アルゴリズムによって得られる単語列 $\boldsymbol{w}(\mathcal{P})$ に対して式 (3.13)、式 (3.15) を計算し、単語あたりの寄与度を求めることで、λ_{c}、$\lambda_{\mathrm{w}\,j}$ の更新値 $\tilde{\lambda}_{\mathrm{c}}$、$\tilde{\lambda}_{\mathrm{w}}$ が得られる。すなわち、

$$\tilde{\lambda}_{\mathrm{c}} = \frac{\sum_{\mathcal{P}\in\mathcal{T}} \hat{\lambda}_{\mathrm{c}}(\boldsymbol{w}(\mathcal{P}))}{\sum_{\mathcal{P}\in\mathcal{T}} |\boldsymbol{w}(\mathcal{P})|}, \tag{3.16}$$

$$\tilde{\lambda}_{\mathrm{w}\,j} = \frac{\sum_{\mathcal{P}\in\mathcal{T}} \hat{\lambda}_{\mathrm{w}\,j}(\boldsymbol{w}(\mathcal{P}))}{\sum_{\mathcal{P}\in\mathcal{T}} |\boldsymbol{w}(\mathcal{P})|}. \tag{3.17}$$

また、P_{d} の更新値 $\tilde{P}_{\mathrm{d}}$ は、式 (3.13)、式 (3.14) から得られる単語の長さごとの寄与度の分布から、指数分布の最尤推定によって次のように得られる。

$$\tilde{P}_{\mathrm{d}} = 1 - \frac{\frac{\sum_{\mathcal{P}\in\mathcal{T}} \hat{\lambda}_{\mathrm{c}}(\boldsymbol{w}(\mathcal{P}))}{\sum_{\mathcal{P}\in\mathcal{T}} |\boldsymbol{w}(\mathcal{P})|}}{\sum_{L} L \frac{\sum_{\mathcal{P}\in\mathcal{T}} \hat{\lambda}_{\mathrm{c}(\tau=L)}(\boldsymbol{w}(\mathcal{P}))}{\sum_{\mathcal{P}\in\mathcal{T}} |\boldsymbol{w}(\mathcal{P})|}} \tag{3.18}$$

表 3.1: 評価テキスト

	単語数	(うち未定義語数)	字数	平均語長
新聞社説	558897	(5101)	981018	1.76
コラム記事	1128134	(13373)	1926513	1.71

3.5　評価実験

提案した文字と単語の複合 n-gram モデルを評価するため、未知テキストに対するパープレキシティを求めた。モデルの訓練テキストには、新聞社説記事 (朝日新聞、97–580 万字) を用いた。評価テキストとしては、訓練テキストと重複しないような新聞社説記事タスク、および新聞の連載コラム記事 (「天声人語」) タスクを用いた。評価テキストの諸元を表 3.1 に示す。表中、単語数と未定義語数は JUMAN による解析の結果得た参考値である。

評価の基準に用いたのは、次式で定義される文字を単位としたパープレキシティである。

$$PP \simeq \hat{P}(C_1 C_2 \cdots C_k)^{-1/k} \tag{3.19}$$

但し、$\hat{P}$ は評価する言語モデル、$C_1 C_2 \cdots C_k$ は評価テキストである。式 (3.19) は、第 2 章での評価に用いた式 (2.20) と形の上では同じものだが、単語モデルに対しては式 (3.5)、複合モデルに対しては式 (3.8) のように、可能な全ての分割可能性を考慮して求められる。

評価した言語モデルは次のとおりである。

文字モデル　式 (3.1)、式 (3.2)、式 (3.3) において、$n = 3$ としたもの

単語モデル　式 (3.5)、式 (3.6)、式 (3.7) において、$n = 3$ としたもの。なお、式 (3.7) 中の単語 zerogram 確率 P_{w0} は JUMAN の語彙数の逆数により与えた。

複合モデル　式 (3.8)、式 (3.9)、式 (3.6)、式 (3.7)、式 (3.10) において、$m = 3, n = 3$ としたもの。式 (3.7) 中の P_{w0} は、文字 n-gram モデルと組み合わせたことによって不要になるため 0 とした。

パープレキシティによる評価に先立って、単語モデルによって未知文章がどのように解析されているかを分析するため、式 (3.12) によって評価テキストを分割

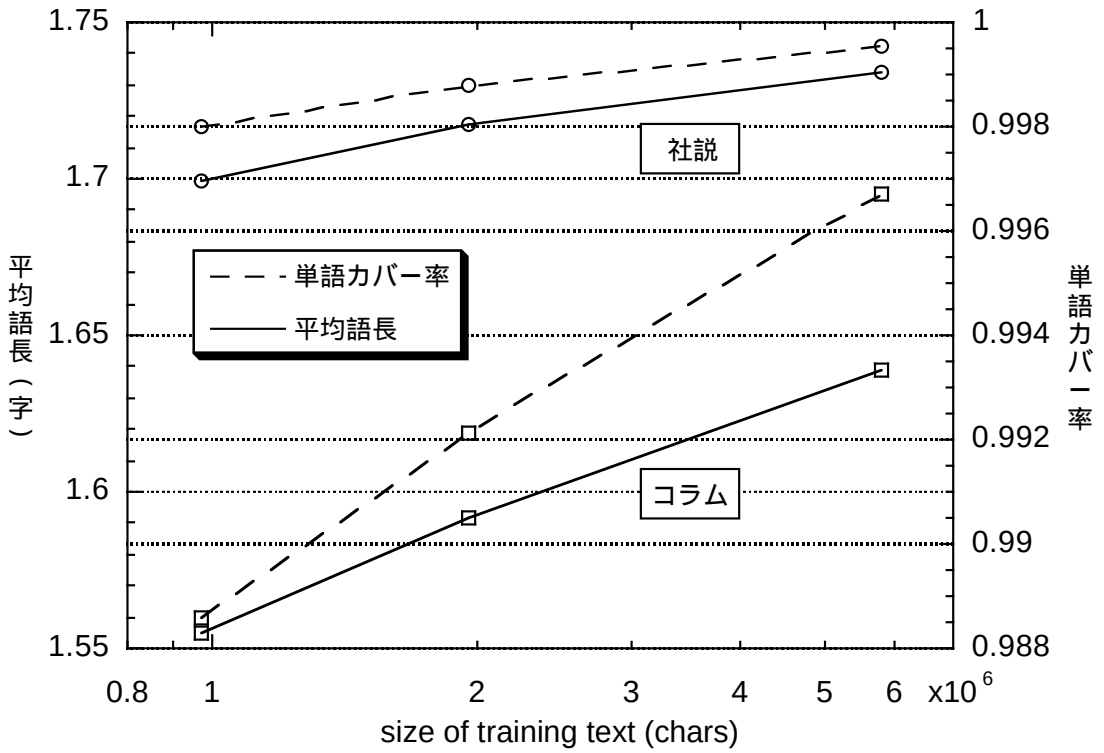

図 3.3: 単語モデルによる評価テキストの解析結果

した時の平均語長および単語カバー率を求めた。これを図 3.3 に示す。サンプルが少ない時に平均語長が短くなることがわかる。これは、学習時に単語として獲得できなかった文字列は、認識時にはもっと短い単語の連続となることが多いためである。コラム記事タスクは訓練テキストと性質が異なるため、この傾向はさらに強まる。また、語長の最頻値は常に 1 で、単語が長くなるとその頻度はほぼ単調に小さくなることが観察された。これは指数分布の仮定が妥当であることを示している。なお、指数分布の代わりに Poisson 分布を仮定した場合でも、後述するパープレキシティに変化は見られなかった。単語カバー率とは、単語モデルによる最適な分割の結果得られた単語がモデルの獲得単語集合に含まれる割合である。単語カバー率も、訓練テキストの量に依存する量であることがわかる。

まず、タスクの性質が訓練テキストと比較的似ている場合のモデルの能力を評価するため、新聞社説タスクに対するパープレキシティを求めた。その結果を図 3.4 に示す。横軸は訓練テキストの量である。複合モデルの性能が他のものに比べ高いことがわかる。単語モデルのパープレキシティと比べると、テキストが約 97

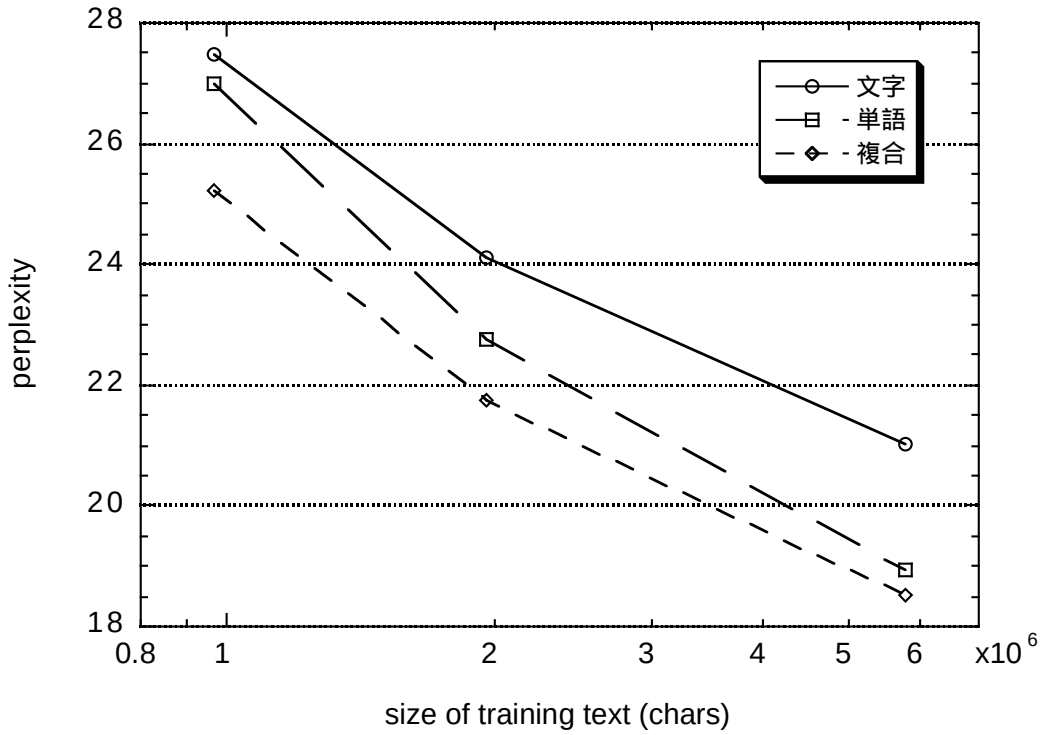

図 3.4: 新聞社説タスクに対するパープレキシティ

万字の場合、27.0 から 25.2 まで低下させることができている。訓練テキストが比較的少ない場合には、複合モデルがより有利であると言える。この場合のパープレキシティの改善率は、文字単位では 6.67%、表 3.1 に従って単語単位に換算すると 11.4%である。パターン認識においてこの差が認識率に及ぼす影響は、アプリケーションと認識系の実装によって異なり、評価が難しい。一般論として、言語モデルの改善による 10%程度の単語パープレキシティの減少は効果が少ないとされるが [47]、10000 単語の不特定話者連続音声認識システムにおいて、単語パープレキシティの 7.16%の減少が word accuracy を約 92%から約 93%に高めたという報告もある [39]。文字認識における言語モデルの改善による具体的な効果については、第 5 章で述べる。

次に、タスクの性質が訓練テキストと余り似ていない場合のモデルの能力を評価するため、訓練テキストと性質の異なるタスクであるコラム記事タスクに対するパープレキシティを求めた。これは、困難なタスクにおける性能を比較することで、言語モデルの汎化能力を議論するためである。その結果を図 3.5 に示す。単語モデルの性能の低下が著しく、パープレキシティで文字モデルを大きく上回る

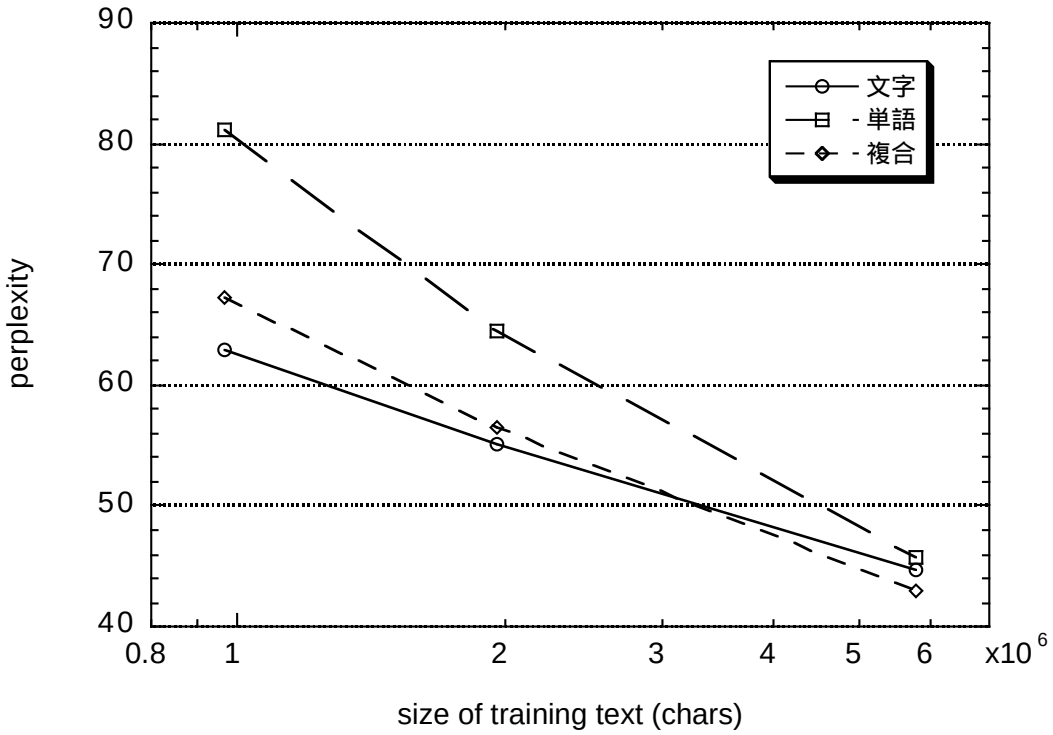

図 3.5: コラム記事タスクに対するパープレキシティ

結果となった。これは、コーパスとタスク間に共通する語彙が少なくなったためと考えられる。複合モデルはサンプルが少ない場合でもこのような性能の劣化が少ない。また訓練テキストを多くすると、文字モデル・単語モデルのどちらよりも低いパープレキシティが得られた。訓練テキストが少ないところでは文字モデルより劣る結果となったが、これは各モデルに対する重みの学習の段階で同一タスクの文章しか想定されていないのが原因と考えられる。

これらの結果は、次のように分析できる。未知文章に対して通常の単語 n-gram モデルによって確率を求めると、未知語などによる影響で解析精度が低下する部分での確率が非常に低くなり、結果として全体の確率が下がってしまう。複合モデルにおいては、解析精度が低下した部分においても式 (3.9) の右辺第 2 項で与えられる文字の n-gram 確率が保証されるので、全体の確率はあまり下がらない。また、タスクに左右されにくい一般的な表現は、依然として単語 n-gram モデルが有効に働くと考えられる。したがって、未知文章中のこのような部分では、文字 n-gram 確率よりも単語 n-gram 確率の方が高くなる。このように、2 種類のモデルを組み合わせたことによって、未知文章に対するパープレキシティを低下できたものと

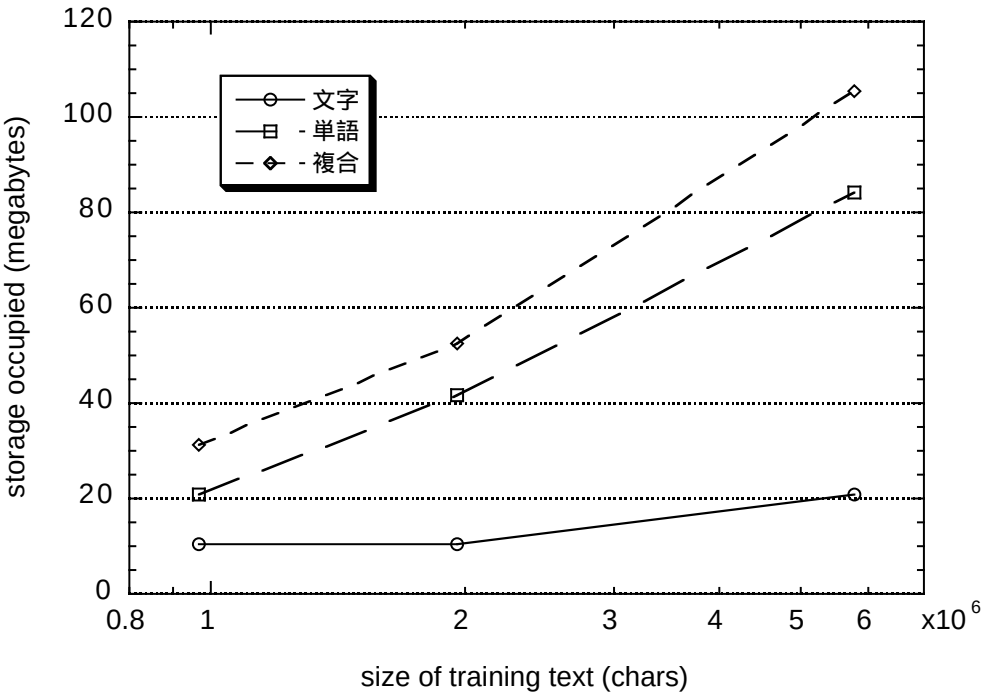

図 3.6: ディスク使用量

言える。

最後に、文字モデル・単語モデル・複合モデルのそれぞれの得失を、必要な記憶容量の観点を交えて論じる。図 3.6 に、各モデルの保持に必要なディスク容量を示す。n-gram の頻度情報は、動的ハッシュ法 [48] により管理している。頻度情報を完全二分木などにより主記憶上に展開する場合には、必要な記憶容量はもっと小さくなる。文字 trigram モデル単独のディスク使用量は他に比べ小さく、ディスクあるいはメモリに余裕がない場合にはこれが望ましい。また、単語モデルと複合モデルとの差は余り大きくなく、複合モデルは最大でも単語モデルの 1.5 倍程度である。このことから、少しでも高い認識性能を必要とする場合には、単語モデル単独よりはむしろ複合モデルを採用すべきだと言える。

3.6　まとめ

文字 n-gram と単語 n-gram を融合させた、べた書きテキストから学習可能な言語モデルの構成法を示し、提案したモデルが従来の文字 n-gram モデル、あるいは単語 n-gram モデルでは得られなかった高い予測性能を持つことを明らかにした。

このモデルのさらなる改善方法として、形態素解析システムが持っているにも関わらず、訓練テキスト中に存在しなかったことによって発生する未知語への対策が考えられる。例えば、形態素辞書中にある表現については、訓練テキストによらずある一定の確率を与える [49]、などである。しかし、日本語には活用する語が存在するため、単に形態素辞書を持つだけでは不十分で、活用規則など文法に近い知識を備えなくてはならない欠点がある。

また、現在の学習法では形態素解析システムがあらかじめ持っている形態素辞書にない単語を学習によって獲得することが困難である。これについては、形態素辞書に加え、コーパスから自動抽出した文字列 [41] を併用するなどの方法が考えられる。さらに、本章で述べた文字 n-gram と単語 n-gram の融合の手法は必ずしも形態素解析に基づくものではなく、次章で述べるような形態素解析を使わない方法にも適用可能である。

実験では、訓練テキストと性質が異なるタスクにおける性能が、文字 n-gram モデルよりも低い場合があることがわかった。認識するタスクに応じて、各モデルへの重み係数を適切に設定することが重要である。この重み係数は、比較的少ないサンプルからも学習できる。よって、事前に少量の文書を用意することによって、モデルをタスクに適応できる可能性がある。これは将来の課題としたい。

第4章　単語知識を必要としない高精度な言語モデル

4.1　はじめに

音声認識・文字認識の精度向上のための言語モデルとして、第 2 章では文字 n-gram モデルを、第 3 章では単語と文字の複合 n-gram モデルを取り上げ、詳細な検討を行った。n-gram が大語彙ないしタスク非依存のシステムのための統計的言語モデルとして今日もっとも有望視されており、強力かつ頑健性に優れていることは既に述べた。

英語などのヨーロッパ系言語においては、n-gram の単位として単語を用いることが多い。大語彙のシステムでは単語はカテゴリ数が非常に大きくなるため、単語の代わりに品詞を用いる [25]、または単語クラスタリングによって得られる単語クラスを用いることが多い。これらの言語においては単語は分かち書きされるため機械的に取り出すことができ、数え上げも容易に行える。

これに対し、日本語や中国語には分かち書きの習慣がない。朝鮮語は文節ごとに分かち書きをするが、その分かち方は一定しないうえ、n-gram の単位としては大き過ぎて汎化性に難がある。よって、これらの言語を n-gram によってモデル化する際には、テキストコーパスに何らかの前処理が必要である。これには次の可能性が考えられる。

- 人手によって分割されたタグ付きコーパスを使う
- 自動形態素解析システムによって単語に分割する
- 経験的な統計基準によって文字列に分割する

このうちタグ付きコーパスを使う方法には、コーパス自体の入手が質的・量的な困難を伴うという欠点がある。形態素解析に基づく方法は第 3 章でも採用しており有効であるが、モデルを学習するためにはまず形態素解析システムを用意せねば

ならないうえ、特定タスクに対して高い性能を得るためには予め辞書をチューニングする必要があると考えられ、メンテナンスのコストがかかる。また、形態素解析システムの文法規則によっては機能語が短めに分割される傾向があり、n-gram の性能を必ずしも最大にするものではない。

これらの手法に対して、伊藤ら [41] は統計的な基準によって文字列の集合を選定し、その文字列に分割されたテキストを使って n-gram を学習する方法を提案している。文字列を選定する基準としては、単純な頻度、および語彙の自動獲得のために提案されている正規化頻度 [50] の高いものから選ぶ方式が有効であったとされる。この方法は、形態素解析を必要としない点で優れている。しかし、抽出すべき文字列の最適な個数を見出す方法については述べられていない。また、用いられている基準と言語モデルの能力との理論的関係は浅く、最良の分割方法である保証はない。さらに、この手法ではテキストが明示的に分割される。このため、接辞を伴った語や複合語などの長い文字列が抽出された場合、その文字列を構成するもっと短い語は出現しなかったのと同様な扱いを受けることになる。有限のテキストから汎化性の高い言語モデルを構築したい場合に、このような明示的な分割が最良の結果を与えるとは限らない。

この章では、高い曖昧性削減能力を持つ新しい言語モデルを提案する。このモデルは、superword と呼ぶ文字列の集合の上の n-gram モデルとして定義される。superword は訓練テキスト中の文字列の再現性のみに基づいて定義される概念であり、与えられた訓練テキストに対して一意に定まる。長さを制限しない文字列の出現回数の統計を取るのは難しいとされていたが、再現性に着目することによって、superword 集合は入力テキスト長に対して線形時間で求めることができる。superword モデルの具体的な確率分布は、訓練テキストから Forward-Backward アルゴリズムによって求める。訓練テキストを明示的に分割せぬまま学習を行うため、長い文字列中の部分文字列を「再利用」することが可能となり、少量の訓練テキストでも効率の良いモデル化が期待できる。この章ではまた、いくつかのモデルの融合による汎化性の向上についても検討する。

4.2　superword モデルの定式化

第 3 章で述べた単語の n-gram や文字列の n-gram[41] では、与えられた系列を単語ないし文字列に分割するやり方が一意に決まらないため、これらのモデルは

直前の $n-1$ 個の単語や文字列を状態とする、隠れマルコフモデルの一種と考えられる。単語や文字列の集合は、語彙知識として人手で与えられるか、あるいは経験的な規則に基づいて訓練テキストから抽出されるものである。ここで定義する superword とはこれら単語や文字列を一般化したものであるが、それらと対照的なのは、訓練テキスト中の任意の文字列を含み得る点である。ただし、言語モデルとして意味を持つために必要最小限のヒューリスティクスは導入せねばならない。そこで、次の条件を満たす文字列を superword と定義する。

- 訓練テキスト中に最低 2 回出現する

または

- 長さ 1 の文字列である

訓練テキストにおける再現性の仮定は、ある文字列が何らかの言語的なまとまりを成すか否かに対する基準となるものであり、そのような基準として考え得る制約の中でもっとも緩い条件として与えてある。すなわち、ある文字列が訓練テキスト中で 1 回しか出現しない、または 1 回も出現しないならば、その文字列が何らかのまとまりを成すだろうという証拠は、他に人間が知識として与えない限り得られない。

また、再現性とは独立に、長さ 1 の文字列は全て superword と定義している。これにより、全ての文は少なくとも 1 通りの superword の系列として表現できることが保証される。superword n-gram 確率 $P(w_i|w_{i-(n-1)}\cdots w_{i-1})$ は、直前に $n-1$ 個の superword の列 $w_{i-(n-1)}\cdots w_{i-1}$ が生起したと仮定した時の superword w_i の条件付き生起確率である。

与えられた文 $\boldsymbol{C}=C_1C_2\cdots C_k$ が superword の列 $w_1w_2\cdots w_l$ に分割できるとき、$w_1w_2\cdots w_l\in\boldsymbol{C}$ と書く。superword n-gram モデルは、$\boldsymbol{C}$ の全ての可能な分割に関して計算した superword n-gram 確率の積の総和をもって $\boldsymbol{C}$ の発生確率を推定するものである。すなわち、その確率を次式で与える。

$$P(\boldsymbol{C})=\sum_{w_1\cdots w_l\in\boldsymbol{C}}\prod_{i=1}^{l}P(w_i|w_{i-(n-1)}\cdots w_{i-1}) \tag{4.1}$$

ここで $n=1$ の時、すなわち superword unigram モデルは、文全体の生起確率がそれぞれ独立な superword の生起確率の積で表されるとするものであり、multigram[53] と呼ばれる可変長単語列に基づく言語モデルと同一のものである。multigram は

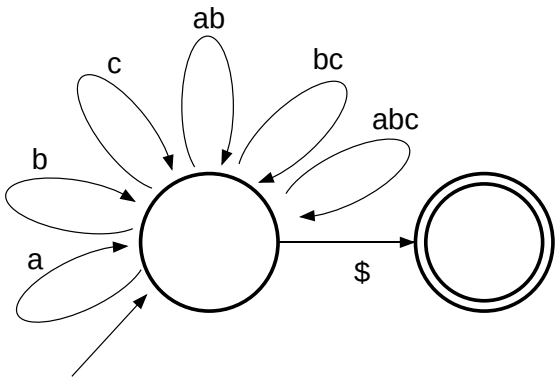

図 4.1: multigram の状態遷移図

記憶のない HMM の一種と考えることができ、1 回の状態遷移で単語ではなく単語列を出力する。文の出力確率は単語列の可能な組み合わせに対して各々の確率の総和または最大値で与える。総和を採用した場合には、図 4.1 のモデルで “abc” という文の出力確率は

$$P(\mathrm{a})P(\mathrm{b})P(\mathrm{c}) + P(\mathrm{a})P(\mathrm{bc}) + P(\mathrm{ab})P(\mathrm{c}) + P(\mathrm{abc})$$

と計算する。

superword n-gram モデルのクラスは、単語や文字列の n-gram モデルのクラスを包含する。この性質は、パラメータさえ適切に与えることができれば、superword に基づくモデルの性能が単語や文字列の n-gram モデルの性能と同等かそれ以上になることを保証する。

4.3　superword モデルの学習法

4.3.1　superword 集合の獲得

モデルの獲得にあたっては、パラメータの学習に先立ち、訓練テキストから superword の集合を求める必要がある。長さ 1 の superword については自明であるから、再現性のある文字列を集める作業が核心である。

長さ n の文字列は、すなわち文字 n-gram である。n-gram 統計はテキスト中に出現する回数を記録するという単純な処理のみで得ることができる。しかし、相異なる n-gram の数は、字種数に関して最悪指数オーダーで増加し、現実的には trigram($n = 3$) より大きな規模の n-gram 統計を取るのは難しいとされていた。

ところが、Nagao et al.[51] が提案した手法により、より大きな n に対しても現実的に n-gram 統計を取れることが示された。この手法は、任意の n に対する n-gram の出現回数を正確に求めることができる点で一般的であると言える。また、断片的な部分列の発生を抑制するような改良手法もある [52]。しかし、これらの手法は全ての処理に先立って入力文字列の全ての位置から始まる部分文字列のソートを必要とし、この処理に要する計算時間が大きい。

しかし、ここで興味があるのは再現性のある文字列である。そこで、一度しか発生しなかった文字列を無視することにより、Nagao らの方法によらずとも、大きな n に対する n-gram 統計を構成することができるようになる。

本論文で着目した点は、次の 2 つの事実である。

- ある n-gram が C 回出現するとき、テキスト中にはその n-gram の一部である $(n-1)$-gram が少なくとも C 回出現する
- ある定数 $n_{\max}$ が存在して、$n > n_{\max}$ では再現性のある n-gram の種類は n に対して単調に減少する

これらの事実に基づき、最終的に C 回以上出現する n-gram を全て求めるためには、図 4.2 の再帰手続き $\mathrm{find_superwords}(C)$ を実行すればよい。手続き中では $(n-1)$-gram の出現回数を検索する必要があるが、これは主記憶上でのハッシュ法によって高速に実現できる。この時占有する記憶域は $(n-1)$-gram の種類にほぼ比例する。したがって、高速な処理のためには C 回以上出現する $n_{\max}$-gram の種類があまり大きくならないことが求められる。図 4.3 に、新聞社説記事 (97–580 万字) 中に 2 回以上出現する n-gram の種類を示す。

このアルゴリズムの時間計算量は、ハッシュ検索が $O(1)$ で行えると仮定すると、テキスト長 L に対し $O(nL)$ である。後でわかるように、superword モデルを構築するという目的のためには $n = 20$ 程度で打ち切っても十分なので、短時間に処理することができる。

4.3.2　確率分布の Forward-Backward 学習

superword モデルでは、ある状態から別の状態に移る時に、ある確率で一つの superword を出力する。w_i の表記を $C_1 \cdots C_j \cdots C_L$ とし、w_i の長さ j のプレフィックスを $w_{i,j}$ とする。図 4.4 に示すように、確率 $P(w_i|w_{i-(n-1)} \cdots w_{i-1})$ で副状態

```
function find_superwords(L:integer):set of string;
    var table, superwords:set of string;
        count:array of integer indexed by strings;
begin
    table:=ϕ;    superwords:=ϕ;
    for ∀ string of length L in training text
        if L=1 or L>1 and chop(string)∈ find_superwords(L−1) then
            begin
                table:=table ∪ {string};
                count[string]:=count[string]+1
            end;
    for ∀ string∈ table
        if count[string]>1 then
            superwords:=superwords ∪ {string};
    return superwords
end
```

図 4.2: 再現性のある文字列を求める再帰的手続き

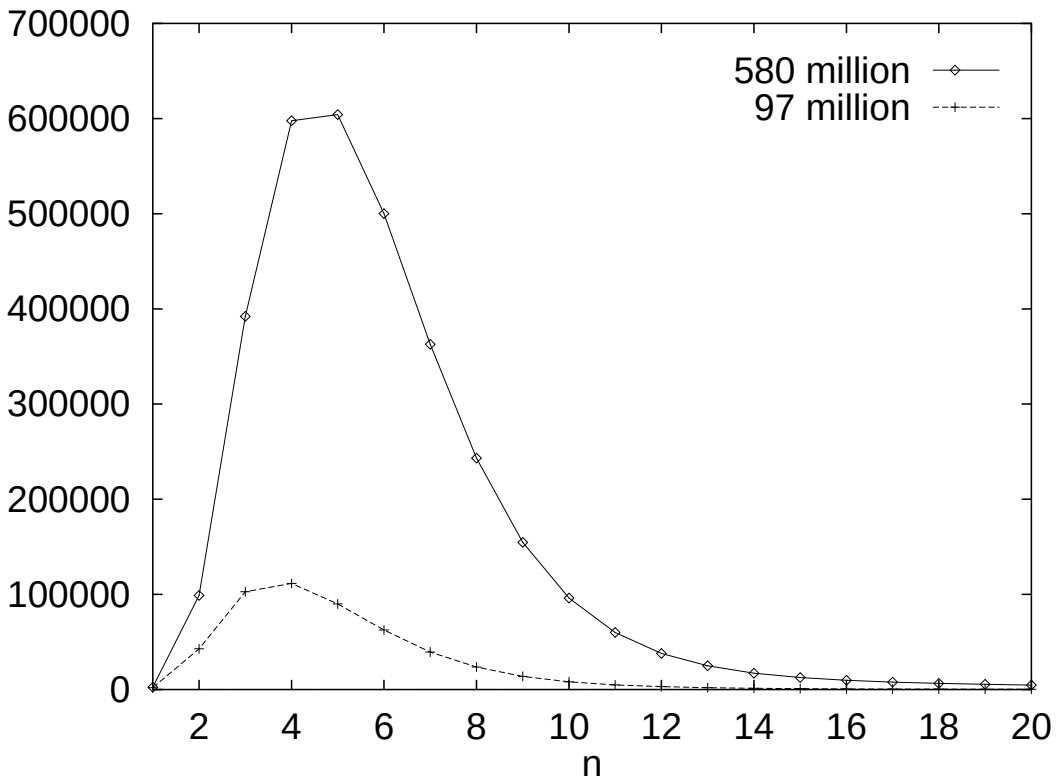

図 4.3: 2 回以上出現する n-gram の種類

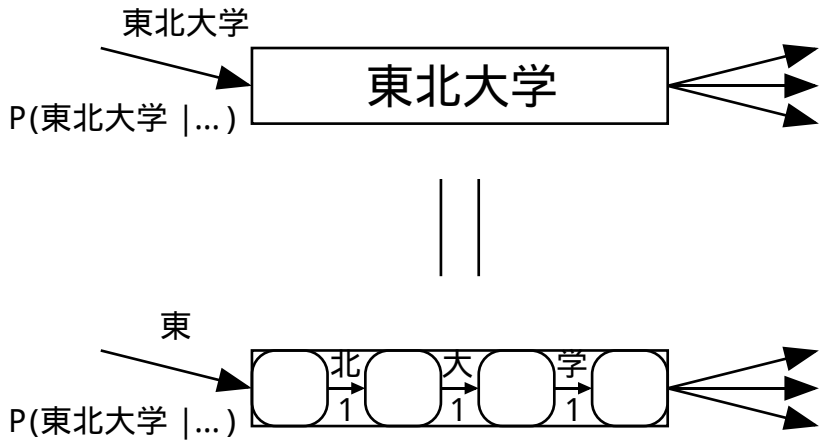

図 4.4:「東北大学」という superword の各文字に対応した副状態の系列

$w_{i-(n-2)} \cdots w_{i-1} w_{i,1}$ に移る時に C_1 を出力し、以後確率 1 で副状態 $w_{i-(n-2)} \cdots w_{i-1} w_{i,j}$ に移る時に C_j を出力し、最終的に状態 $w_{i-(n-2)} \cdots w_{i-1} w_i$ に至ると考える。すなわち、単位時間で 1 回の状態遷移をし、記号を 1 つ出力する通常の隠れマルコフモデルと同様に扱うことができる。

$n = 1$、すなわち superword unigram 確率の学習のための初期確率としては、全ての superword が等確率で発生するとして、superword の数の逆数を与える。$n > 1$ については、対応する superword の $(n-1)$-gram 確率で初期化する。

確率の再推定のために、図 4.5 のように訓練テキストから全ての superword を洗い出す。次に、連接可能な n 個の superword の組に関して、次式によって確率を更新する。

$$\tilde{P}(w_i | w_{i-(n-1)} \cdots w_{i-1}) = \frac{\sum_t \alpha_{t-1}(w_{i-(n-1)} \cdots w_{i-1}) P(w_i | w_{i-(n-1)} \cdots w_{i-1}) \beta_t(w_{i-(n-2)} \cdots w_i)}{\sum_t \alpha_t(w_{i-(n-2)} \cdots w_i) \beta_t(w_{i-(n-2)} \cdots w_i)} \quad (4.2)$$

ただし、α, β はそれぞれ Forward 確率、Backward 確率で、以下のように再帰的に定義する。

$$\alpha_1(w) = P(w|\#), \qquad \#\text{は文頭を表す状態} \quad (4.3)$$

時刻 $t(t > 1)$ で superword w_i の第 1 字目を出力するとき

$$\alpha_t(w_{i-(n-2)} \cdots w_i) = \sum_{w_{i-(n-1)}} \alpha_{t-1}(w_{i-(n-1)} \cdots w_{i-1}) P(w_i | w_{i-(n-1)} \cdots w_{i-1}) \quad (4.4)$$

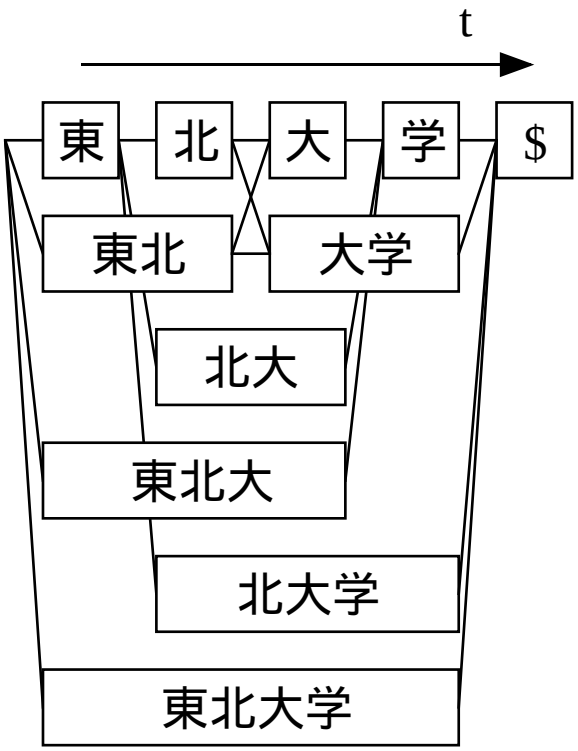

図 4.5:「東北大学」というテキストの解析。矩形は superword を、実線は可能なパスを表す。“$” は文の終端

時刻 $t(t > 1)$ で superword $w_i = C_1 \cdots C_j \cdots C_L$ の第 j 字目 $(j > 1)$ を出力するとき

$$\alpha_t(w_{i-(n-2)} \cdots w_{i-1} w_{i,j}) = \alpha_{t-1}(w_{i-(n-2)} \cdots w_{i-1} w_{i,j-1}) \tag{4.5}$$

ただし

$$\alpha_t(w_{i-(n-2)} \cdots w_{i-1} w_i) = \alpha_t(w_{i-(n-2)} \cdots w_{i-1} w_{i,L}) \tag{4.6}$$

同様に

$$\beta_T(\$) = 1, \qquad T \text{ は文末記号 “\$” を出力する時刻} \tag{4.7}$$

時刻 $t(t < T)$ で superword w_i の第 1 字目を出力するとき

$$\beta_{t-1}(w_{i-(n-1)} \cdots w_{i-1}) = \sum_{w_i} \beta_t(w_{i-(n-2)} \cdots w_i) P(w_i | w_{i-(n-1)} \cdots w_{i-1}) \tag{4.8}$$

時刻 $t(t < T)$ で superword $w_i = C_1 \cdots C_j \cdots C_L$ の第 j 字目 $(j > 1)$ を出力するとき

$$\beta_{t-1}(w_{i-(n-2)} \cdots w_{i-1} w_{i,j-1}) = \beta_t(w_{i-(n-2)} \cdots w_{i-1} w_{i,j}) \tag{4.9}$$

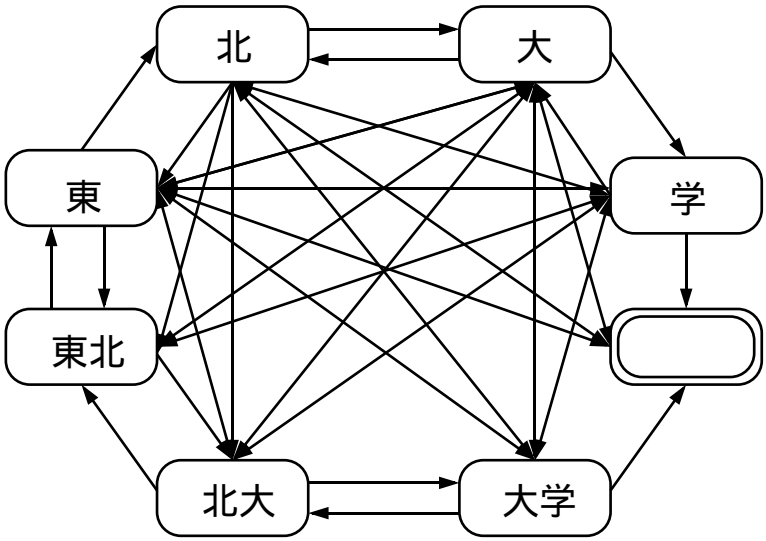

図 4.6:「東北大学」の部分文字列を superword の集合とする時の長さ制限モデル

ただし

$$\beta_t(w_{i-(n-2)} \cdots w_{i-1} w_i) = \beta_t(w_{i-(n-2)} \cdots w_{i-1} w_{i,L}) \tag{4.10}$$

4.4　長さ制限の導入

再現性のある文字列の長さを十分大きく取れば、前節までに述べたモデルは与えられた訓練テキストに対して一意に求まる。以下では、これを一般 superword n-gram モデルと呼ぶ。しかし、一般モデルのパラメータ数は大きい。特に、$n > 2$ では superword の組み合わせが爆発し、現実的ではない。さらに、あまりに長い superword は訓練テキストに特化してしまう恐れがあり、汎化能力の低下を招く。

これに対処するため、一般モデルに加えて長さ制限付きの superword モデルを導入する。これは、逐次的な再現性文字列の獲得を早い段階で打ち切って小さな superword の集合をつくり、その集合に基づいて Forward-Backward 学習を行うことで得ることができる。以下では、長さ l に制限された superword n-gram 確率を $P_{|w|\leq l}(w_i|w_{i-(n-1)} \cdots w_{i-1})$ と表記する。

長さが l に制限された superword n-gram モデルは、図 4.6 に示すような、状態数が高々字種の l 乗に制限されたエルゴーディック HMM となる。

4.5　複合モデル

n-gram に代表される確率モデルにおいては、モデルのパラメータを精度良く推定するに足るサンプルが得られないことが多く、パラメータ空間のさまざまなスムージング法が提案されている [39]。その一つに、いくつかのモデルの確率の重み付き線形和で表現する方法がある [38]。これは本来、詳細なモデルの値が信用できない場合に、パラメータの少ない安定したモデルの値を代用するものであるが、性質の異なる複数のモデルを組み合わせてより良いモデルを得るという積極的な利用も可能である。第 3 章で文字と単語の n-gram を融合した枠組も、この考えに基づくものである。本節では、線形補間に基づくいくつかの複合モデルを考える。

superword bigram($n = 2$) モデルに対しては、superword unigram 確率によって補間された確率は次式で与えられる。

$$\hat{P}(w_i|w_{i-1}) = \lambda_{\mathrm{g}} P(w_i|w_{i-1}) + (1 - \lambda_{\mathrm{g}}) P(w_i) \tag{4.11}$$

重み係数 λ_{g} は、訓練テキストとは別のサンプル (held-out データ) またはクロスバリデーションによって得られる仮想的な未知データの確率を最大にするように再推定する。

前述したように、一般 superword bigram はパラメータ量が多くなり過ぎるので、実際には superword の長さを最大 l に制限したモデルと組み合わせる。これは次式で与えられる。

$$\hat{P}_{|w|\leq l}(w_i|w_{i-1}) = \lambda_{\mathrm{b}} P_{|w|\leq l}(w_i|w_{i-1}) + (1 - \lambda_{\mathrm{b}}) P_{|w|\leq l}(w_i) \tag{4.12}$$

式 (4.12) のような制限されたモデルでは、長い語の表現に難があることも考えられる。そこで、長さ制限付き superword bigram モデルと一般 superword unigram モデルの複合モデルを導入する。複合 superword bigram 確率は次式で定義される。

$$P_{\mathrm{comp}}(w_i|w_{i-1}) = \lambda_{\mathrm{c}} \hat{P}_{|w|\leq l}(w_i|w_{i-1}) + (1 - \lambda_{\mathrm{c}}) P(w_i) \tag{4.13}$$

さらに、複合 superword bigram モデルを、文字の trigram モデルによってスムージングすることを考える。第 2 章で明らかにしたように、文字の trigram モデルはそれ自身で強力な曖昧性削減能力を持っているが、単語 n-gram モデルと融合させることにより、認識対象中の未知の文字列の存在による単語解析精度の低下の影響を低減させ、頑健なモデルとすることができる。このことは第 3 章で述べた。文字によって補間された複合 superword bigram 確率は次式で定義される。

$$\hat{P}_{\mathrm{comp}}(w_i|w_{i-1}) = \lambda_{\mathrm{w}} P_{\mathrm{comp}}(w_i|w_{i-1}) + (1 - \lambda_{\mathrm{w}}) \hat{P}_{\mathrm{c}}(w_i|w_{i-1}) \tag{4.14}$$

表 4.1: 訓練テキスト・評価テキストの量

	字　　(文)
訓練	969497　(21767)
held-out	85654　(1953)
評価	80098　(1779)

ただし、$\hat{P}_{\rm c}(w_i|w_{i-1})$ は superword w_i が生起する確率を、補間された文字 trigram 確率の積によって求めたものである。すなわち、w_i の表記を $C_1 \cdots C_{L(w_i)}$、w_{i-1} の最後の 2 文字を $C_{-1}C_0$ と書くとき

$$\hat{P}_{\rm c}(w_i|w_{i-1}) = \left(\prod_{j=1}^{L(w_i)} \hat{P}_{\rm c}(C_j|C_{j-2}C_{j-1}) \right) \cdot d(L(w_i)) \tag{4.15}$$

ただし $\hat{P}_{\rm c}(C_j|C_{j-2}C_{j-1})$ は bigram, unigram 等により補間された文字 trigram 確率である。また、$d(L(w_i))$ は文字モデルが生成する単語の長さに関する分布関数である。

4.6　評価実験

提案した言語モデルの能力を、文字を単位としたパープレキシティによって評価する。パープレキシティは、式 (4.1) において評価用テキストを $\boldsymbol{C}$ として次式で求められる。

$$PP \simeq \hat{P}(\boldsymbol{C})^{-1/k} \tag{4.16}$$

ただし、k は評価用テキストの全字数である。長さ 1 の superword に対しては、確率が設定した底値を下回る場合には底上げした。対象タスクは朝日新聞「社説」とした。実験に用いたテキストの量を表 4.1 に示す。表中、held-out とは式 (4.12), 式 (4.13), 式 (4.14) の重み係数を求めるために用いたテキストである。各々のテキストは、共通部分を持たない。

長さ制限の効果を見るため、superword unigram モデルについて最大長を変化させてパープレキシティを求めた。その結果を図 4.7 に示す。この結果から、長い superword を許してもパープレキシティは上がらないことがわかる。これは、superword の再現性の条件が適当であったことを示す。以下の実験では、$L = 20$ の場合を一般 superword unigram モデルとして扱う。

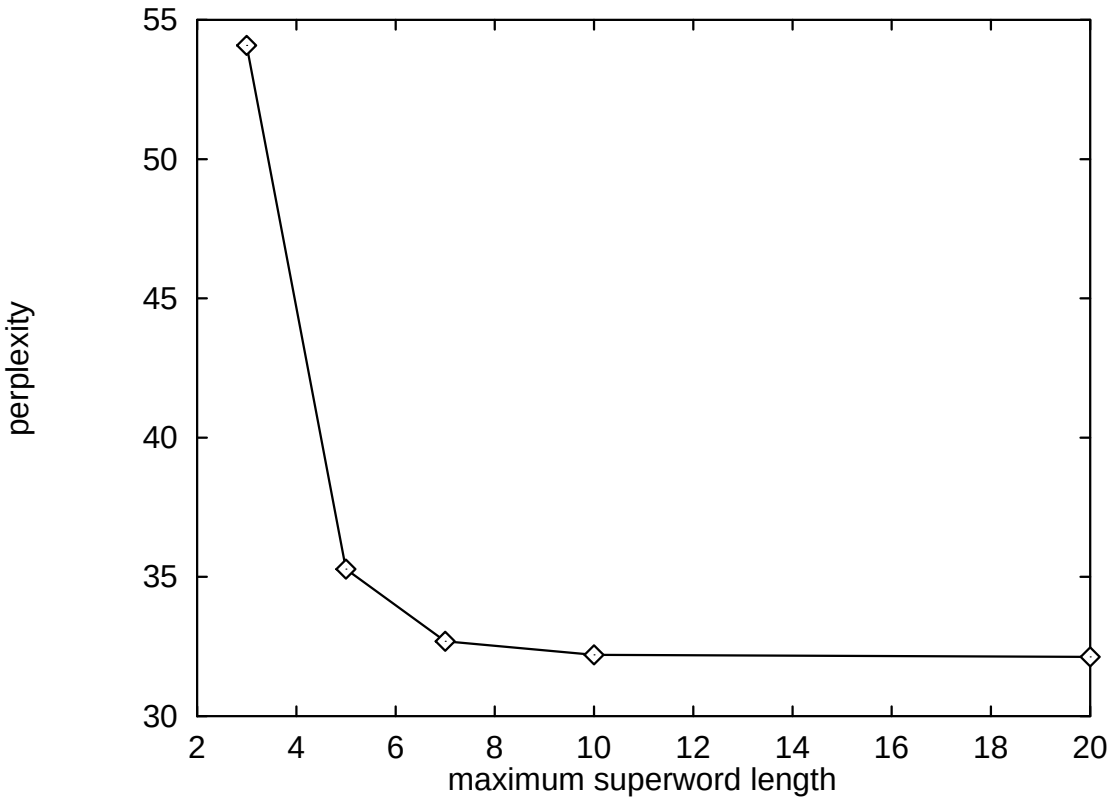

図 4.7: superword unigram モデルにおける長さ制限の効果

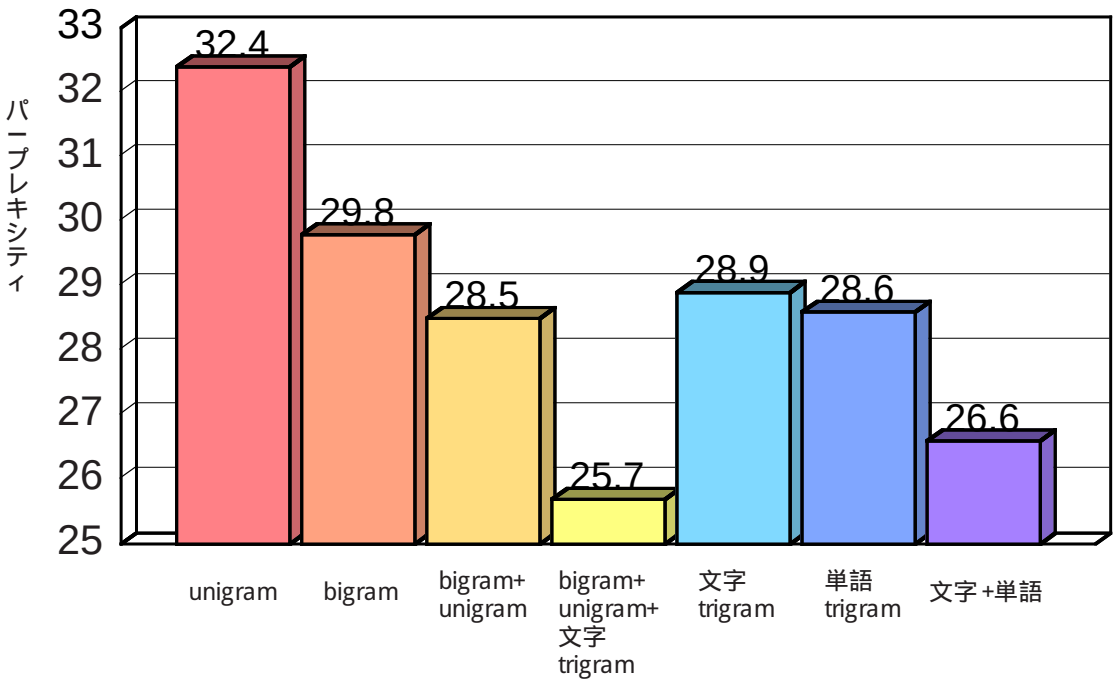

図 4.8: 各モデルの性能評価

図 4.8 に、提案したモデルのパープレキシティを示す。図中、左から 4 本が superword に基づくモデルである。bigram とあるのは式 (4.12) の長さ制限付きモデルである。ここでは最大長を 3 とした。bigram+unigram とあるのは式 (4.13) の複合モデル、bigram+unigram+文字 trigram とあるのは式 (4.14) をさらに文字 trigram で補間したモデルである。その場合の式 (4.15) の分布関数としては、指数分布を仮定した。右から 3 本は従来の n-gram モデルで、比較のために示してある。文字 trigram は式 (2.9) によるもので、第 2 章と同様である。単語 trigram は式 (3.5)、式 (3.6) において $n = 3$ としたもの、文字+単語とあるのは式 (3.8)、式 (3.9)、式 (3.6)、式 (3.7)、式 (3.10) において $m = 3, n = 3$ とした複合 n-gram モデルで、いずれも第 3 章と同様に学習したモデルである。

この結果から、次のことがわかる。まず、superword unigram モデルの性能が良くない。図 4.7 の結果をも考慮すると、これは superword の長さの問題ではなく、unigram では語と語の連接関係が本質的に表現できないものと考えられる。これは ATIS データベースの上での multigram の評価 [53] といくぶん矛盾する結果であるが、伊藤ら [41] も同様の結果を導いている。

長さ制限付き superword bigram モデルの導入によって、性能の向上が見られた。しかし、まだその性能は文字 trigram モデルに及ばない。

長さ制限付き superword bigram モデルと一般 superword unigram モデルを融合させることで、若干の性能向上が見られた。これは、長い superword は単独ではあまり性能に貢献しないが、語と語の連接関係だけでは表現しきれない部分を補う効果を持っているものと考えることができる。語と語の関係に関する知識と語彙知識とを独立に表現する枠組は、形態素解析の原理と類似している。

さらに、文字 trigram モデルでスムージングすることにより、大きく性能が向上した。その結果、形態素解析を用いたモデルを超える性能が得られた。superword に基づいたモデル単独では訓練テキストに対して過学習する傾向があり、未知テキストに対して脆弱な側面があるが、未知テキストに対して頑健な文字 trigram モデルとの融合によりそれが克服できることを意味する。

4.7　まとめ

本章では、superword の概念に基づいた新しい言語モデルを提案した。このモデルは従来の n-gram の枠組を包含したより一般的なものであり、コーパス以外の知識に全く依存しない。また、本章で導入した長さ制限モデルとスムージング手法により、現実的なコーパスの量の範囲でモデルの学習が可能となった。評価実験の結果、長さ制限を施した superword bigram モデルを文字 trigram モデルと組み合わせて頑健性を向上させたモデルの性能が高く、第 3 章で提案した形態素解析に基づく手法を超える能力が得られた。

superword に基づく言語モデルは、可搬性に優れた強力なものであるが、欠点として訓練テキストに比べモデルの規模が非常に大きいことが挙げられる。superword unigram モデルのパラメータ数は superword 集合の大きさにほぼ比例するが、図 4.3 からもわかるように、かなり大きい。特に、通常の n-gram ではモデルのサイズの上界がコーパスの量に対して線形のオーダーで与えられる (実際には図 2.3 のように次第にゆるやかに増大する) のに対し、superword の場合にはそれよりも大きくなる可能性がある。これは superword を可能な限り一般的に定義したためであり、特に大規模なコーパスを用いてモデルを学習する場合には、再現性の仮定を見直す必要があることが考えられる。

また、superword bigram モデルは長さ制限を加えた場合でも非常に大きくなる。

これは、与えられたテキストの superword による解析結果が極めて曖昧性が大きいものであることが原因である。

モデルのサイズを小さくし、実際のパターン認識システムで利用できるようにするためには、モデルの最適化が必要である。すなわち、学習の過程で非常に小さな確率を付与された状態遷移のアークは刈り取る、あるいは外から遷移してくる確率が十分小さな状態は削除する、などである。しかし、この種の枝刈りは、訓練サンプルに特化する危険がある。そのため、今後はパープレキシティを上げることなくモデルをコンパクトにするための枝刈り手法の開発が課題である。

第5章　文字認識システムへの統合と評価

5.1　はじめに

前章までに、パターン認識の曖昧性を解消するための言語モデルについて論じてきた。

本章では、これまでに提案した種々の言語モデルを用いた文字認識システムの構成法を示し、これまで得た曖昧性解消能力に関する結果と認識性能との関係を明らかにする。

言語処理を伴わない単純な文字認識においては、文字切り出しの問題が無視できるのならば、個別文字のマッチングにおいて最も高い評価を与えられた文字を個々の認識結果とすれば良い。しかし、本論文における文字認識のように文字認識を自然言語処理と融合する場合は、個々の文字の認識結果は前後の文字に依存するため、認識結果の決定は個別文字認識の場合ほど簡単ではない。文脈を用いた全ての文字認識システムにおいては、認識結果を決定するために何らかのスコア付けが必要となるが、スコア付けのための規則を人手で与えることの困難さは第1章で述べた通りである。また、現実にはスコア最大化のための探索をなるべく短い時間で行わなければならないという問題もある。

スコア付けの問題に対して、本論文では文字認識問題を情報理論的、通信理論的問題として解釈するという立場をとる。これにより、文字認識問題を誤り確率最小化という定式化の明確な枠組みにより取り扱うことが可能となる。したがって、認識結果候補のスコアを与えるのは確率そのものとなる。

本章で述べるように、スコアとしての事後確率は言語の性質を表す部分と特徴量空間の性質を表す部分という独立な2要素から成る。前者は、これまでに論じた言語の確率モデルに他ならない。本章では、まず文字認識系の確率的性質をパターン認識理論の立場から定式化する。ついで、この定式化に従い、既存の多フォント日本語文字認識系に第2章、第3章、第4章で述べた言語モデルを統合する2

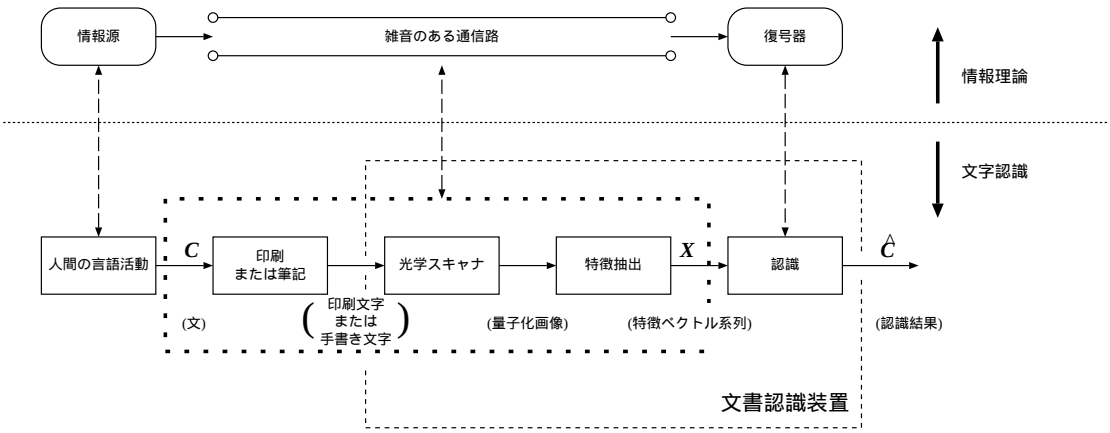

図 5.1: 情報理論と文字認識の対応関係。[14] を参考にした

種類の手法を述べる。第 1 は、文字 n-gram に特に適した枠組であり、ある程度高い個別文字認識精度が見込まれ、かつ高速な処理が必要な場合に向いている。第 2 は、個別文字認識の精度が低く、より高い性能を持つ言語モデルによる高精度な処理が求められる場合のためのものであり、個別文字認識による事前の大分類を必要としない。本章では、これらの文字認識システムの具体化をも行うとともに、印刷文書を用いた実験による性能評価を行う。

5.2　文字認識の情報理論的解釈

個々の文字でなく文の構成要素としての文字を認識するということの意味を少し形式的に考えてみると、それは図 5.1 にも示すように、人間の言語活動のあらわれである文あるいは文章を再現することと解釈できる。この文あるいは文章は雑音を含まない純粋な形で具体化するのではなく、紙などに印刷または筆記された文字の集合という歪んだ形で具体化する。我々が筆者の意図として知ることのできる直接的な情報は、このような紙上の文字だけである。したがって、実際に筆者が何を書こうとしたかを知ること、すなわち文字を認識することは、紙上の文字という歪んだ情報から元々の情報を推定することである。これを Shannon[15] の情報理論と対応付けるならば、人間の言語活動が文を生成する情報源であり、印刷あるいは筆記という行為が雑音を含む通信路にあたる。この意味での高精度な

文字認識とは、文字という雑音を含んだ情報から、できるだけ高精度に元の文を復元する過程である。

この文字認識過程を機械処理によって行おうとする場合には、まず文書画像を光学的に読み取り、機械によって扱うことのできるディジタル情報への変換が必要となる。この過程においても情報が欠落する。さらに、この後通常は、このディジタル画像を機械にとってもっと扱いやすい形にする特徴抽出の過程が入る。この過程は普通は非可逆な情報圧縮であり、当然全ての情報を保持しない。以上の 3 段階の過程、すなわち印刷 (筆記)、光学スキャニング、特徴抽出はそれぞれ雑音を含む通信路であるが、これらを縦続接続し、仮想的に一本の雑音を含む通信路とすれば、数式の上では扱いやすい形となる。

上でも述べた通り、ここでは文字認識を高精度に行うことを復号誤り確率を小さくすることであるとみなしている。つまり高精度な文字認識装置を実現するためには、復号誤り確率が最小となるような復号器を設計するようにすれば良い。このためには、雑音を含む通信路からの受信記号系列 $\boldsymbol{X} = \boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_l$ を観測した時、それが実は $\boldsymbol{C} = C_1C_2\cdots C_k$ という情報であった確率、つまり事後確率を最大にするような系列 $\hat{\boldsymbol{C}}$ に復号化すれば良いことが知られている [54]。すなわち、

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{X}) \tag{5.1}$$

情報理論の教えるところによれば、雑音を含む通信路において受信の誤りを小さくするためには、送信符号が何らかの冗長性を持っていなければならない。ここでいう冗長性は、後述するように自然言語に内在する制約と同等のものである。冗長性を仮定できない場合、すなわち自然言語としての性質を無視した場合には、式 (5.1) の右辺は次式で置き換えられる。

$$\underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{X}) = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, p(\boldsymbol{X}|\boldsymbol{C}) \tag{5.2}$$

ここで、情報源からの記号 C_i と通信路の受信端での記号 $\boldsymbol{x}_i$ が 1 対 1 に対応するという仮定を置く。このことは、文字認識で言えば文字切り出しが完全に行われることに対応する。この仮定の下では、式 (5.2) は次のようになる。

$$\underset{\boldsymbol{C}}{\operatorname{argmax}}\, p(\boldsymbol{X}|\boldsymbol{C}) = \underset{\boldsymbol{C}}{\operatorname{argmax}} \prod_{i=1}^{k} p(\boldsymbol{x}_i|C_i) \tag{5.3}$$

上式の最尤推定は、各時点での文字 C_i

$$\hat{C}_i = \underset{C_i}{\operatorname{argmax}}\, p(\boldsymbol{x}_i|C_i) \tag{5.4}$$

となる。式 (5.4) によって識別平面を決定する考えが、パターン認識の分野でいう統計的決定理論である。

以下では言語モデルを利用した認識について述べる。情報源記号のパープレキシティがシンボル数に比べて小さいことを冗長であるという。この場合、式 (5.1) における事後確率 $P(\boldsymbol{C}|\boldsymbol{X})$ は次のようになる [54]。

$$P(\boldsymbol{C}|\boldsymbol{X}) = \frac{P(\boldsymbol{C})p(\boldsymbol{X}|\boldsymbol{C})}{p(\boldsymbol{X})} \tag{5.5}$$

この関係は Bayes の定理と呼ばれる。式 (5.5) の分母は $\boldsymbol{C}$ に無関係な定数であるから、式 (5.1) は次式のようになる。

$$\begin{aligned}
\hat{\boldsymbol{C}} &= \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{X}) && (5.6)\\
&= \underset{\boldsymbol{C}}{\operatorname{argmax}}\, \frac{P(\boldsymbol{C})p(\boldsymbol{X}|\boldsymbol{C})}{p(\boldsymbol{X})} && (5.7)\\
&= \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C})p(\boldsymbol{X}|\boldsymbol{C}) && (5.8)\\
&= \underset{\boldsymbol{C}}{\operatorname{argmax}}\{\log P(\boldsymbol{C}) + \log p(\boldsymbol{X}|\boldsymbol{C})\} && (5.9)
\end{aligned}$$

式 (5.3) の場合と同様に、文字 C_i と特徴ベクトル $\boldsymbol{x}_i$ が 1 対 1 に対応するならば、

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\{\log P(\boldsymbol{C}) + \sum_{i=1}^{k} \log p(\boldsymbol{x}_i|C_i)\} \tag{5.10}$$

となる。式 (5.10) に見られるように、事後確率の大小を支配するのは、$\log P(\boldsymbol{C})$ と $\log p(\boldsymbol{x}|C)$ という 2 つの独立な要素である。

$P(\boldsymbol{C})$ は $\boldsymbol{C}$ の事前確率と呼ばれるものである。これは $\boldsymbol{C}$ の自然言語らしさを表わす値であり、本論文の主眼である言語モデルに相当する。第 2 章で論じたように、$P(\boldsymbol{C})$ は式 (2.7),(2.9),(2.12) によって推定される。

$p(\boldsymbol{x}|C)$ は図 5.1 に示した通信路の性質を表わしているもので、もし特徴ベクトルが離散的変数であれば、情報理論でいう通信路行列にあたる。この値を推定する方法については次節で詳しく述べる。

5.3　尤度関数の推定法

$p(\boldsymbol{x}|C)$ の特性を知るには、特徴ベクトル $\boldsymbol{x}$ のふるまいを調べなければならない。特徴量空間の性質を知ることはパターン認識においては非常に重要であり、文字認識においても文字の分布形状や分布の広がりについて研究されている [55, 56, 57, 58]。

ただ、式 (5.10) の算出にあたって必要となるのは、分布に関する断片的な性質よりは、むしろ具体的な $p(\boldsymbol{x}|C)$ の値である。これを直接求めるには、実際の文字からいかなる特徴ベクトルが得られるかについて統計を取ることによって推定せねばならない。理論的には、無限に多くのサンプルが得られるならば、$p(\boldsymbol{x}|C)$ を完全に知ることができる。しかし現実には、$\boldsymbol{x}$ が正規分布などの簡単な分布に従うと仮定した場合でさえも、それを推定するに足る十分な数のサンプルは得られない。そこで、分布形状に何らかの仮定を置き、分布のパラメータ数を制限する必要がある。

今、文字カテゴリ C に属する特徴ベクトル $\boldsymbol{x}$ が d 次元正規分布 $N(\boldsymbol{\mu}_C, \Sigma_C)$ に従って分布すると仮定する。ただし、$\boldsymbol{\mu}_C$ はカテゴリ C の重心のベクトルであり、Σ_C は共分散行列である。仮定から、

$$p(\boldsymbol{x}|C) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma_C|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_C)'\Sigma_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C)\right] \tag{5.11}$$

ただし $(\boldsymbol{x}-\boldsymbol{\mu}_C)'$ はベクトル $(\boldsymbol{x}-\boldsymbol{\mu}_C)$ の転置を表わす。上式の両辺の対数を取ると、

$$\begin{aligned}\log p(\boldsymbol{x}|C) &= \log\left\{\frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma_C|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_C)'\Sigma_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C)\right]\right\} \\ &= -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_C| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_C)'\Sigma_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C) \end{aligned} \tag{5.12}$$

上式の右辺第 3 項の 2 次形式の部分はマハラノビス距離と呼ばれるもので、この部分が式の値を実質的に支配する。$\boldsymbol{\mu}_C$ はパターン整合法におけるテンプレートに相当し、多くの文字サンプルから統計的に得ることができる。Σ_C の共分散行列も統計的推定が可能であるが、パラメータの数が $d(d+1)/2$ と非常に大きいため、カテゴリごとに求めるのは非常に難しい。対策の一つは、K-L 展開に基づいてパターンの性質を良く反映すると思われる特徴のみを抽出する (部分空間法 [59], 複合類似度法 [60] など)、あるいはそのような特徴に関する分散・共分散のみを精密に求める (修正ベイズ 2 次識別関数 [55]、改良型マハラノビス距離 [61]、簡素化マハラノビス距離 [62] など) ことである。もう一つは、各カテゴリにおける分布形状を同一とみなすことである。すなわち、

$$\Sigma_C \approx \Sigma \tag{5.13}$$

とする。後述するように、この仮定は必ずしも妥当とは言えないが、全カテゴリの特徴ベクトルをただ 1 つの共分散行列の推定に利用でき、誤差は低減できる。

このようにしてもなおパラメータ数が多いため、特徴量空間での各次元の相関を無視することにする。また、各次元の分散をも同一視することによって、共分散行列のパラメータ数を 1 にまで減少させることができる。すなわち、

$$\boldsymbol{\Sigma} \approx \sigma^2 \boldsymbol{I} \tag{5.14}$$

であるとする。ここで $\sigma > 0$、また $\boldsymbol{I}$ は単位行列を表わす。この仮定も必ずしも正しくなく、パラメータ推定のために便宜的に導入されるものである。

式 (5.13),(5.14) を用いると、式 (5.12) は次のように書き直せる。

$$\log p(\boldsymbol{x}|C) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{\mu}_C\|^2 \tag{5.15}$$

ただし、$\|\cdot\|$ はユークリッド ノルムを表わす。上式の右辺第 1,2 項は定数であり、C を変数とした時の $\log p(\boldsymbol{x}|C)$ の値の大小比較には影響を与えない。また第 3 項の係数は統計的に求めることもできるが、後述するように特に計算する必要はなく、式 (5.10) の確率の最大化が目的ならば、係数は任意の値で構わない。最終的に、$\log p(\boldsymbol{x}|C)$ を支配するのは次の値である。

$$-A\|\boldsymbol{x} - \boldsymbol{\mu}_C\|^2 \tag{5.16}$$

ただし、$A > 0$ とする。$\|\boldsymbol{x} - \boldsymbol{\mu}_C\|^2$ は特徴ベクトル $\boldsymbol{x}$ とカテゴリ C の標準パターン $\boldsymbol{\mu}_C$ とのユークリッド距離 (またはユークリッド自乗距離) と呼ばれ、統計的に比較的安定であるという利点を持つ。

式 (5.10) の右辺第 1 項 $\log P(\boldsymbol{C})$ を言語的尤度と呼び、前章までに述べた言語モデルにより求められる。また、右辺第 2 項は式 (5.16) により近似される。これらを用いることで式 (5.10) を最大にする系列 $\boldsymbol{C}$ が求められるが、両方とも大雑把な近似に過ぎないから、単純な足し合わせでは誤りを最小にできるとは限らない。

そこで、2 つの項の重み付き線形和で全体の尤度を表わすことにする。すなわち、尤度関数を次式のように定義する。

$$L(\boldsymbol{C}|\boldsymbol{X}) \stackrel{\text{def}}{=} w\log\hat{P}(\boldsymbol{C}) + (1-w)A\sum_{i=1}^{k}(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{C_i}\|^2) \tag{5.17}$$

ただし $0 \le w \le 1$ である。この尤度関数を用いて、式 (5.10) は次のように推定される。

$$\begin{aligned}
\hat{\boldsymbol{C}} &= \underset{\boldsymbol{C}}{\operatorname{argmax}}\{\log P(\boldsymbol{C}) + \sum_{i=1}^{k}\log p(\boldsymbol{x}_i|C_i)\} && (5.18)\\
&\approx \underset{\boldsymbol{C}}{\operatorname{argmax}}\, L(\boldsymbol{C}|\boldsymbol{X}) && (5.19)\\
&= \underset{\boldsymbol{C}}{\operatorname{argmax}}\{w\log\hat{P}(\boldsymbol{C}) + (1-w)A\sum_{i=1}^{k}(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{C_i}\|^2)\} && (5.20)
\end{aligned}$$

式 (5.20) の中括弧内を定数倍しても $\hat{C}$ は変わらない。したがって、式 (5.20) の値に影響を与える変数は、各項の重みの比 $(w/(1-w)A)$ のみである。最適な重みの比が与えられると、A をどのように選んでも同じ比を保つように w を選ぶことができる。したがって、A の値は結果に影響を与えない。最適な w の値は実験により求めることとする。

5.4　文字 trigram モデルを用いた認識システムの実現

5.4.1　認識アルゴリズム

言語モデルとして文字 n-gram モデルを用いると、文字認識システムの実現は容易になる。これは、処理が入力文字に完全に同期して行えるためである。

式 (2.12) を用いると、尤度関数は

$$
\begin{aligned}
L(\boldsymbol{C}|\boldsymbol{X}) &= L(C_1C_2\cdots C_k|\boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_k) && (5.21)\\
&= w\sum_{i=1}^{k}\log\hat{P}(C_i|C_{i-2}C_{i-1}) + (1-w)A\sum_{i=1}^{k}(-\|\boldsymbol{x}_i-\boldsymbol{\mu}_{C_i}\|^2) && (5.22)\\
&= w\log\hat{P}(C_k|C_{k-2}C_{k-1}) + (1-w)A(-\|\boldsymbol{x}_k-\boldsymbol{\mu}_{C_k}\|^2)\\
&\quad + L(C_1C_2\cdots C_{k-1}|\boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_{k-1}) && (5.23)
\end{aligned}
$$

となり、漸化式で表わすことができる。このような尤度関数を最大化する $\boldsymbol{C}$ は、Viterbi アルゴリズム [14] によって効率良く求めることができる。Viterbi アルゴリズムは、2 重マルコフ連鎖の各状態において、次のスコアを計算することである。

$$
\begin{aligned}
G_1(C_0C_1) &\stackrel{\text{def}}{=} w\log\hat{P}(C_1|C_{-1}C_0) + (1-w)A(-\|\boldsymbol{x}_1-\boldsymbol{\mu}_{C_1}\|^2) && (5.24)\\
G_i(C_{i-1}C_i) &\stackrel{\text{def}}{=} \max_{C_{i-2}}\{w\log\hat{P}(C_i|C_{i-2}C_{i-1}) + (1-w)A(-\|\boldsymbol{x}_i-\boldsymbol{\mu}_{C_i}\|^2)\\
&\qquad + G_{i-1}(C_{i-2}C_{i-1})\} \qquad (i\geq 2) && (5.25)
\end{aligned}
$$

各状態においては、max を取る計算の結果得られるスコア $G_i(C_{i-1}C_i)$ および選ばれた一つ前の状態 $C_{i-2}C_{i-1}$ を保持しておく。最終的な結果は、$\max_{C_{k-1}C_k} G_k(C_{k-1}C_k)$ を計算した後、最終状態から一つずつ状態を前の時点の方向へさかのぼっていくことにより得られる。

このアルゴリズムを高速に実行するためには、言語処理による詳細分類に先立ち、大分類によって C_i の領域をある程度制限する必要がある。本論文では、次の手法を用いて候補となる文字の数を制限する。文字集合を $\mathcal{C}$ と書く。パターン整合は、

特徴ベクトル $\boldsymbol{x}$ と全ての文字 $C \in \mathcal{C}$ の標準パターン $\boldsymbol{\mu}_C$ との距離 $\|\boldsymbol{x} - \boldsymbol{\mu}_C\|^2$ を求め、昇順にソートすることにより実現される。正解の文字が標準パターンと非常に類似していると仮定できるならば、個々の文字の認識結果としては、ソートの結果ある程度上位のものだけを考慮すれば良い。そこで、集合 $R = \{C^1, C^2, \cdots, C^{N_c}\}$ と各候補の評価値 $\|\boldsymbol{x} - \boldsymbol{\mu}_{C^j}\|^2$ を認識結果とする。ただし、C^j は j 位候補となった文字である。また N_c を出力候補数と呼び、前もって何らかの方法で決めておくこととする。このようにすると、最適化の過程におけるマルコフ状態の数は高々${N_c}^2$ となる。各状態における $\max$ 操作に必要な時間計算量は $O(N_c)$ であるから、系列全体の最適化に必要な時間計算量は $O({N_c}^3 k)$ となる。また、空間計算量は明らかに $O({N_c}^2 k)$ となる。N_c を定数とすれば、時間、空間いずれの計算量も系列長に対して線形のオーダーとなり、コストは非常に小さい。

また、計算コストをさらに低減するため、認識結果の確からしさの評価値を導入し、これによって認識結果が十分信頼できると判断された場合はその文字を言語処理部による修正の対象としないこととする。認識の確からしさの基準に関する研究としては西野ら [63] のものなどがあるが、ここでは良く用いられる値として、次の 1 位候補と 2 位候補の距離の比を用いる。

$$r = \frac{\|\boldsymbol{x} - \boldsymbol{\mu}_{C^2}\|^2}{\|\boldsymbol{x} - \boldsymbol{\mu}_{C^1}\|^2} \tag{5.26}$$

この値が十分に大きければ、認識結果は十分信頼できるとみなされ、2 位候補以下は考慮されない。すなわち、閾値 r_{th} に対し、

$$R = \begin{cases} \{C^1\} & \text{if } r \geq r_{\mathrm{th}} \\ \{C^1, C^2, \cdots, C^{N_c}\} & \text{otherwise} \end{cases} \tag{5.27}$$

とする。$r \geq r_{\mathrm{th}}$ となった文字を確定文字、それ以外を未確定文字と呼ぶ。r_{th} も N_c と同様に、予め決めておく。

Viterbi アルゴリズムによる最適化の様子を図 5.2 に示す。図からわかるように、確定文字が $(n-1) = 2$ 文字以上連続して出現した場合にはその前後の系列の処理を各々独立したものとして行うことができる。

5.4.2　認識実験

5.4.1 節で述べた文字 trigram モデルに基づく文字認識システムを構築し、認識実験を行った。個別文字認識は方向線素特徴量 [67] に基づくパターン整合法によ

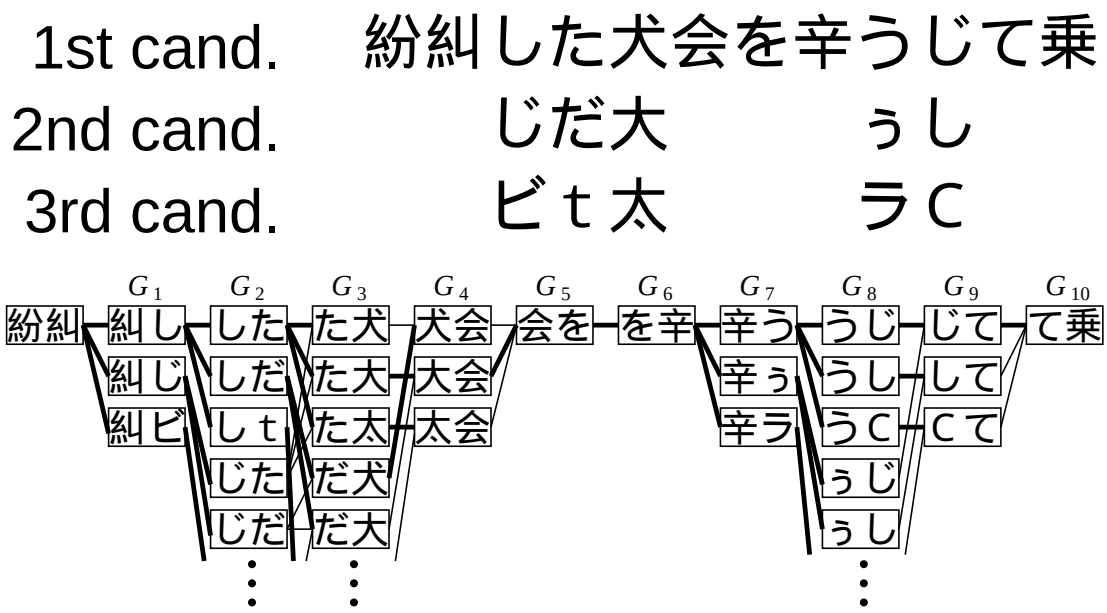

図 5.2: Viterbi アルゴリズムによる最尤系列の導出

り行った。本研究で辞書を作成するのに用いたのは、明朝体・教科書体・新聞の字体などの写真植字による印刷文字、およびドットインパクトプリンタ、レーザープリンタなどから出力された文字で、全体で 79 セットである。その他の個別文字認識系に関する実験条件を表 5.1 に示す。

言語モデル推定のための訓練テキストは、表 2.1 の「社説」(A) 7.7–580 万字である。認識実験の対象として選んだ文章は、2.5 節でパープレキシティの評価に用いたのと同じ、表 2.1 の「社説 (B)」、「天声人語 (D)」、「科研費概要 (E)」の 3 種類である。これらの文章を、レーザープリンタから 10 ポイント大のフォントにより印字し、テストサンプルとした。その際用いたフォントは、辞書を構成するために用いたフォント中にはないものである。文字切り出しには、誤りは含まれない。言語処理を用いない場合の誤認識率は、それぞれ 6.37%, 8.66%, 7.85%である。言語的尤度の重みなどのパラメータは、予備実験により選んだ。

訓練テキスト量、スムージング手法と認識性能の関係を図 5.3 に示す。図中、flooring とあるのは式 (2.7) のフロアリングによってスムージングされた日本語 trigram モデルを用いた場合の誤認識率、D.I. とあるのは式 (2.9) の削除補間法によってスムージングされた日本語 trigram モデルを用いた場合の誤認識率である。なお、ここで用いた trigram モデルの訓練テキストも同様に「社説」(A) である。

認識対象が「社説」(B) の場合には最大で約 99.5%の認識率が得られている。こ

表 5.1: 個別文字認識系の実験条件 (1)

正規化法	線形正規化
前処理	細線化後に輪郭抽出
特徴量	外郭加重方向線素特徴量
細線化回数	5
カテゴリ数	3303
サンプル数	79 セット
標準パターン	シングルテンプレート
識別関数	ユークリッド距離

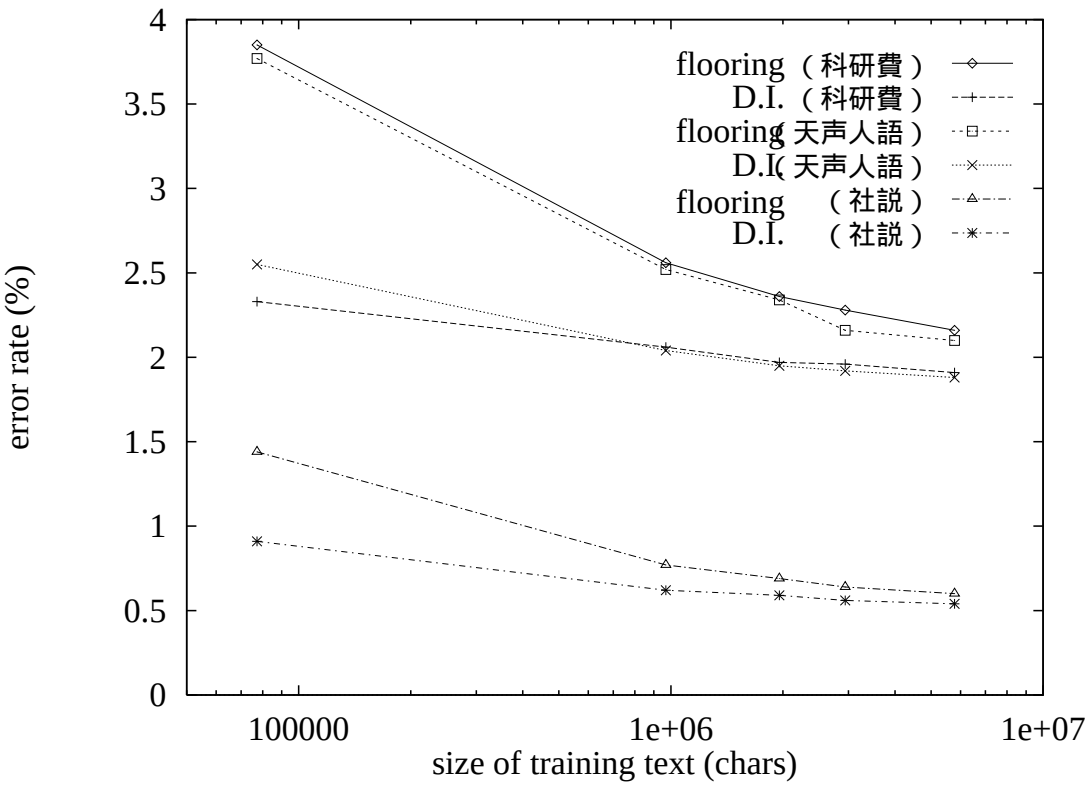

図 5.3: 訓練テキスト量、スムージング手法と誤認識率の関係

れは、言語モデルを用いない場合の認識誤りのうち 90%以上が本手法によって正解に転じたことを意味する。このように、訓練テキストと認識対象の性質が比較的近い場合には、本手法によって非常に高精度に認識を行え、第 2 章で論じた日本語文字 trigram モデルの持つ高い曖昧性解消能力を裏付けている。

一方、認識対象が「天声人語」(D) と「科研費概要」(E) の場合の認識性能はほぼ同等のもので、最大で約 98%である。これは「社説」の場合に比べると低い値ではあるが、それでも言語モデルを用いない場合に誤りであった文字の約 75%が救済されている。言語的性質が違う場合にもある程度言語処理による救済が成功しているのは、例え「社説」と「科研費概要」のように共通の語彙がほとんどない文書どうしであっても、日本語としての基本的な性質を共有している部分が多く、そのような部分では誤り訂正が有効に働くからだと考えられる。

スムージング手法に関しては、図からわかるように、全ての場合において、削除補間法によるモデルがフロアリングに比べ高い性能を示している。また、両者の差は訓練テキストの量が少ない時ほど、また認識対象と訓練テキストの言語的性質が違う時ほど顕著となっている。この事実は、2.5 節における言語モデル、認識タスクとパープレキシティの大小の議論と良く符合し、パープレキシティによる言語モデルの評価の妥当性を裏づけている。

5.5　低品質文書の認識のための単語・文字列認識システムの実現

5.5.1　単語・文字列認識のための探索アルゴリズム

前節で述べた文字 trigram モデルを用いた文字認識システムは、個別文字認識の結果がある程度信頼できる場合を前提とした方法であると言えるが、一般に文字認識における言語処理では候補外文字の問題がある。候補外文字とは、正解であるにもかかわらず認識結果の上位 N_c 位以内に入らなかった文字のことである。予備実験の結果から、N_c を増加させて行くと累積認識率は飽和してしまうことがわかっている。本章で述べたアルゴリズムの計算時間は N_c の 3 乗に比例して増加するため、あまり多くの候補を取ることはできない。この問題は、特に低品質文字を扱う場合に顕著となる。

文字認識の候補外文字の問題に対処するための手法としては、confusion 行列 (類

似文字の相互誤り確率) を用いて候補文字を補完する研究が知られている [64]。しかし、この方法で用いている confusion 行列は認識タスクに依存する先見知識であり、一般の多フォント認識には向いていない。また、仮に候補文字の再現率 (候補文字集合に正解を含んでいる割合) を十分高めることができたとしても、適合率 (候補文字集合に占める正解の割合) が低ければ処理時間が増大する。このため、大分類を必要としない言語処理、あるいは候補文字集合が非常に大きい場合でも極端な実行速度の低下がない言語処理が求められる。

筆者は過去に、低品質文書の候補外文字の問題に対処するため、文字 trigram モデルとビームサーチを用いた文字認識アルゴリズムを提案している [65]。ビームサーチとは、考えられる仮説を全て展開する代わりに、何らかのスコア付けによって展開する仮説数を減らす近似的探索アルゴリズムである。最適性は失われるが、スコアとビーム幅 (仮説展開の基準となる閾値) を適切に選ぶことができれば、少ない計算量で良い近似解を得ることが経験的に知られている。[65] では、個別文字認識の結果による大分類を行わず、全ての文字を認識結果の候補として考慮する代わりに、認識時点での尤度が高い状態のみを保存する。このような処理を行うことで、個別文字認識の精度が 70%台と極めて低い場合にも言語処理が大きな効果を上げる。

文字 n-gram に基づいた文字認識システムでは、切り出しの問題を無視すれば、入力の任意の時点で言語的尤度の計算が可能である。ところが、第 3 章で述べた単語 n-gram およびその改良、あるいは第 4 章で述べた superword n-gram モデルでは、単語仮説の途中の時点では尤度を計算することができない。

この問題を解決し、低品質文書の認識を高精度な言語モデルによって行うため、本論文ではセグメント同期型多重ビームサーチという認識アルゴリズムを提案する。このアルゴリズムでは、完成した単語仮説を保存するための主ビームと、部分列仮説を保存するための副ビームを別々に管理し、入力される文字に同期して枝刈りを行う。これにより、単語仮説が完成する前に枝刈りされる危険を極力抑えつつ、現実的な計算量の範囲で処理を遂行することが可能となる。

式 (3.5) や式 (4.1) などでは、文の生起確率は各々の解析結果ごとの確率の総和で表されるものとしたが、ここでは単純化のため、次式のように最適な解析に対する確率で近似する。

$$P(\boldsymbol{C}) \simeq \max_{w_1 \cdots w_l \in \boldsymbol{C}} \prod_{i=1}^{l} P(w_i | w_{i-(n-1)} \cdots w_{i-1}) \tag{5.28}$$

本アルゴリズムでの尤度関数を、式 (5.17) を元に次の漸化式で再定義する。

$$
\begin{aligned}
L(\boldsymbol{C}|\boldsymbol{X}) &= L(C_1C_2\cdots C_k|\boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_k) && (5.29)\\
&= \max_{w_1\cdots w_l\in\boldsymbol{C}} L(w_1\cdots w_l|\boldsymbol{X}_1^l) && (5.30)\\
&= \max_{w_1\cdots w_l\in\boldsymbol{C}} \left\{ w\sum_{i=1}^{l}\log P(w_i|w_{i-(n-1)}\cdots w_{i-1}) - (1-w)Ad_{\mathrm{sum}}(\boldsymbol{X}_1^l)\right\} && (5.31)\\
&= \max_{w_l}[w\log P(w_l|w_{l-(n-1)}\cdots w_{l-1}) - (1-w)Ad_{\mathrm{sum}}(\boldsymbol{X}_l^l)\\
&\qquad + \max_{w_1\cdots w_l\in\boldsymbol{C}}\{L(w_1\cdots w_{l-1}|\boldsymbol{X}_1^{l-1})\}] && (5.32)
\end{aligned}
$$

ただし、$d_{\mathrm{sum}}(\boldsymbol{X}_s^l)$ は s 番目から l 番目までの単語列の表記に対応する距離の和を表す。この式は完成した単語列に対してしか尤度を与えない。このため、入力文字に同期して処理を進める場合に適切なスコアが求まらず、ビーム幅を大きく取らないと単語仮説が完成する前にビームから漏れて行く危険が高まる。

そこで、任意の入力時点で途中の尤度が求められるようにする。n-gram に基づく連続音声認識では、単語辞書の構造をトライ [66] とし、根から葉にかけて n-gram 確率を分散させて保持する (factorization) ことで、単語仮説が完成した時点で急激に確率が減少するのを防ぐ方法が提案されている [39]。本論文では factorization の処理は行わず、次のような方法を採った。なお、両者を組み合わせて尤度を定義することも可能である。

現在の時刻で単語仮説が未完成である場合の尤度を次のように定義する。

$$
\begin{aligned}
L'(w_l\cdots w_{l-1}w_l'|\boldsymbol{X}_1^l) &= w\log P(w_l'|w_{l-(n-1)}\cdots w_{l-1}) - (1-w)Ad_{\mathrm{sum}}(\boldsymbol{X}_l^l)\\
&\qquad + L(w_1\cdots w_{l-1}|\boldsymbol{X}_1^{l-1}) && (5.33)
\end{aligned}
$$

ただし、w_l' は単語のプレフィックスとなり得る部分列である。式 (5.33) の右辺の計算においては第 1 項だけが w_l' を含み、これまで定義した言語モデルによっては計算できない。そこで、単語仮説が未完成である場合の尤度は、この確率を複合 n-gram 確率の代わりに次式で与えることとする。

$$
P(w_l'|w_{l-(n-1)}\cdots w_{l-1}) \equiv \lambda_{\mathrm{c}}\hat{P}_{\mathrm{c}\,m}(w_l'|w_{l-(n-1)}\cdots w_{l-1}) \qquad (5.34)
$$

この式は、$\lambda_{\mathrm{c}}=1$ の時に文字 m-gram 確率と一致し、上界は複合 n-gram 確率で与えられる。

式 (5.33) で定義した尤度は、式 (5.30) の $L(w_1\cdots w_l|\boldsymbol{X}_1^l)$ とは直接比較できないことに注意すべきである。このため、単語仮説が未完成の場合の状態は単語 n-gram

の状態とは別に扱い、単語仮説が完成した段階で初めて尤度の比較を行う必要がある。

提案するセグメント同期型多重ビームサーチの計算手順を以下に示す。

1. 前時刻の全ての単語終端から、可能な 1 文字語を全て展開する。式 (5.32) によって尤度を計算し、一定数を主ビームに残す。
2. 前時刻の全ての単語終端から、可能な 1 文字部分列を全て展開する。ここでいう 1 文字部分列とは、単語の先頭の 1 文字となり得る文字のことである。式 (5.33) によって尤度を計算し、一定数を (部分列長) $= 1$ の副ビームに残す。
3. 前時刻の全ての n 文字部分列から、可能な長さ $n+1$ の単語を全て展開する。式 (5.32) によって尤度を計算し、一定数を主ビームに残す。
4. 前時刻の全ての n 文字部分列から、可能な $n+1$ 文字部分列を全て展開する。式 (5.33) によって尤度を計算し、一定数を (部分列長) $= n+1$ の副ビームに残す。

1. では全ての文字について、2. では単語の先頭となり得る全ての文字について尤度計算を行う必要があり、計算量は (ビーム幅) $\times$ (字種) のオーダーである。3. および 4. では、単語辞書をトライ構造にしておくことにより、可能な単語および部分列を辞書から予測して展開できる。このため、3.,4. の処理量は 1.,2. に比べて小さい。

セグメント同期型多重ビームサーチの計算の様子を図 5.4 に示す。ここでは、主ビーム幅を 3、副ビーム幅を 3 として例示している。図中、「 ... 」の付いた文字列は単語の部分列を示し、「 / 」の前が各状態における直前の文脈である。各ビームの各時刻での状態は、上から下へ尤度の高い順に並べられている。時刻 3 の「 アジ... 」という部分列から時刻 4 で「 アジア 」という語を展開した結果、尤度が高くなって主ビームに入った様子がわかる。しかし、例えば時刻 2 の「 南千... 」という部分列から展開された「 南千住 」という単語は尤度が低かったため、主ビームからは漏れ、そのため表示されていない。また、時刻 5 で「 南アフリカ 」という語も展開され、尤度の比較が行われる。

5.5.2　認識実験

5.5.1 節で述べたセグメント同期型多重ビームサーチに基づいた文字認識システムを構築し、認識実験を行った。個別文字認識系に関する実験条件を表 5.2 に示す。

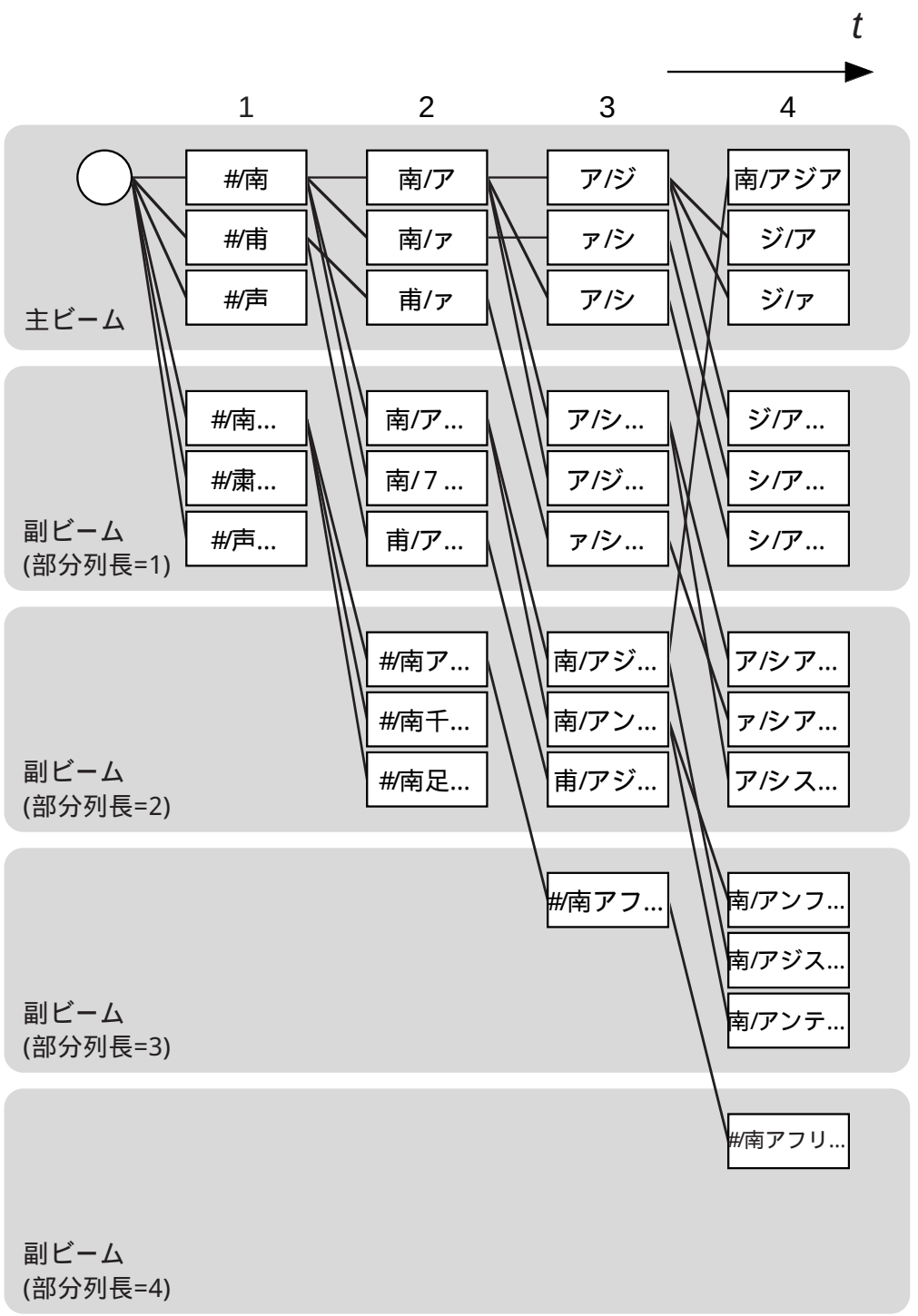

図 5.4: セグメント同期型多重ビームサーチ

表 5.2: 個別文字認識系の実験条件 (2)

正規化法	線形正規化
前処理	細線化後に輪郭抽出
特徴量	外郭加重方向線素特徴量
細線化回数	12
カテゴリ数	3303
サンプル数	99 セット
標準パターン	シングルテンプレート
識別関数	ユークリッド距離

言語モデル推定のための訓練テキストは、表 2.1 の「社説」(A) 97 万字である。言語モデルの性能比較のため、認識実験の対象としては 5.4.2 節のものよりも品質の低いものを用いた。具体的には、表 2.1 の「社説 (B)」から 19 文章 (29083 字) をレーザープリンタから 7 ポイント大のフォントにより印字した。その際用いたフォントは、辞書を構成するために用いたフォント中にはないものである。文字切り出しには、誤りは含まれない。言語処理を用いない場合の誤認識率は 11.38%である。ビーム幅は経験的に設定し、主ビーム幅を 3、副ビーム幅を 3 とした。言語的尤度の重みなどのパラメータは、予備実験により選んだ。

評価した言語モデルは文字 trigram モデル、単語 trigram モデル、文字 trigram と単語 trigram の複合モデル、第 4 章で最も性能の高かった superword bigram モデル (長さ制限、unigram と文字モデルで補間) で、学習法やスムージング法は第 3 章、第 4 章に準じた。

各モデルでの誤認識率を図 5.5 に示す。性能の良い順に、文字と単語の複合モデル、単語モデル、文字モデル、superword モデルとなっている。この結果を図 4.8 のものと比較すると、次のことが言える。第 1 に、第 3 章で論じたモデルについては、認識性能はパープレキシティの結果を良く反映したものになっている。複合モデルを導入することにより、単語モデルを単独で用いた場合に比べ、認識誤りを約 23%減少させることに成功している。これは、第 3 章で述べたように、提案した文字と単語を融合した言語モデルが、単語 n-gram の単純なスムージングだけでは得られない優れた曖昧性解消性能を持つことを裏付ける結果となっている。

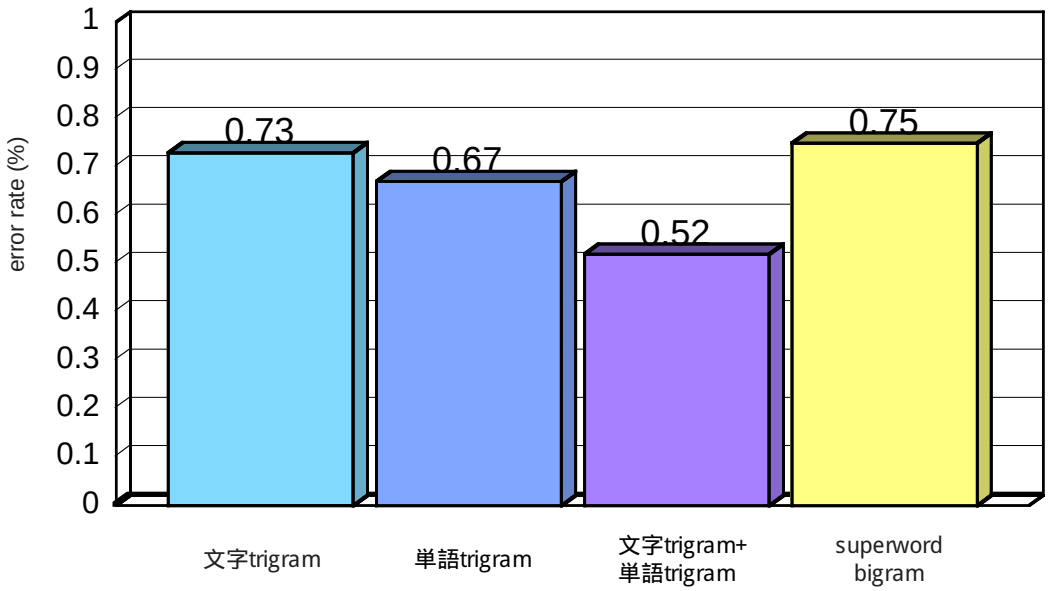

図 5.5: 各モデルによる低品質文書の認識性能比較

第 2 に、superword モデルの性能が他に比べて悪い。この結果は第 4 章の結論に矛盾したものであり、直観に反する。この原因としては、superword モデルの確率計算の仕方が考えられる。superword への解析は、言語知識を用いた辞書中の単語への解析に比べ、曖昧性が大きくなる。これから、文全体の確率に占める最適な解析結果の確率が占める割合が、superword では小さい。第 4 章でのパープレキシティの評価では、確率を全ての解析結果に対する総和として求めていたが、本章の認識では結果に対する単語あるいは superword への分割は 1 通りである。このことが原因で、本章での実験条件では最適な解析結果に対する 1 位解析結果の比重が大きい単語モデルや複合モデルが有利になったものと考えられる。superword モデルを有効に利用するためには、確率計算を複数の解析結果を考慮しながら行うようにする (トレリスサーチ [42])、またビーム幅を最適に調整するなどの対策が必要であろう。

5.6　まとめ

本章では事後確率最大基準を基礎とした文字認識の高精度化について検討し、情報理論的なアプローチによって言語の確率モデルを用いた文字認識アルゴリズムを定式化した。この手法の有効性は、2 種類の探索アルゴリズムに基づいた文書認識実験により確かめられ、計算量・訓練サンプル量のいずれについても現実的な範囲内で、高精度な文字認識システムが実現できることを明らかにした。また文字 trigram モデルのスムージング手法として削除補間法が優れていること、および文字と単語を融合した言語モデルが従来のモデルより優れていることが認識実験の上からも確認でき、第 2 章、第 3 章の結果を裏付けた。本章で提案したアルゴリズムによって、第 4 章で提案した superword モデルを用いた文字認識システムも実現することができたが、その性能を十分に発揮させるためには、今後も検討が必要である。

第6章　結論

6.1　本論文の成果

本論文で得られた成果をまとめると、以下のようになる。

第 2 章では、知識に依存しない言語モデルの基本となる日本語文の確率モデルとして、文字の n-gram モデルについて検討を行った。最初に、他の諸言語に比べて日本語において文字 n-gram を利用することの利点について述べた。また、曖昧性を解消する十分な制約を与えるため、$n = 3$、 すなわち trigram モデルを用いるのが適当であることを示した。同時に、trigram は何らかの手法によってスムージングされる必要があることも示された。スムージングの具体的手法を 2 種類挙げ、パープレキシティを基準とした評価実験によって線形補間による方法が優れていることが明らかとなった。

第 3 章では、第 2 章の文字 n-gram を発展させ、単語 n-gram による言語モデルについて論じた。まず、日本語においては単語 n-gram の構成が自明ではなく、従来のモデルでは未知語の存在が問題となることを示した。次に、第 2 章の文字 n-gram と同じく未知語を陽に扱わない形での単語 n-gram の定式化を行った。そして、少数サンプルが原因で起こる未知文章の単語解析精度の悪化を補うため、文字 n-gram と単語 n-gram を融合した言語モデルを提案した。ここでは、第 2 章で効果が高かった削除補間法が拡張して適用されており、n-gram のスムージングと同レベルでのパラメータ学習が可能となっている。またこのモデルは、第 4 章で述べるような形態素解析に基づかない言語モデルにも適用できる。最後に、本手法によって高い曖昧性削減能力を持つモデルが得られることを、パープレキシティを基準とした評価実験により示した。

第 4 章では、知識を全く必要としない高精度な言語モデルを提案した。このモデルは、superword と呼ぶ概念に基づいたものであり、最小限のヒューリスティックのみを用いて語の連鎖に確率を与える点が他のモデルと異なっている。また、この章で提案した新しいモデルは、従来の単語 n-gram を包含するものとなっている。

モデルのスムージング法として、一般 superword unigram モデルと長さ制限付き superword bigram モデルを組み合わせること、さらに第 3 章で提案した文字と単語の融合の手法を適用することについて検討を行った。実験の結果、提案したモデルが形態素解析に基づく手法を上回る能力を持つことがわかった。

第 5 章では、言語の確率モデルを用いた文字認識システムの実現法について述べた。まず、文字認識問題を雑音のある通信路における誤り確率最小化問題としてとらえ、文字認識系の確率的性質を言語処理部において扱いやすい形に定式化した。ついで、第 4 章までに述べた言語の確率モデルを用いた認識アルゴリズムを提案した。これらに基づいて文字認識システムの実装を行い、実際の文書を用いた認識実験により、言語モデルの性能と認識性能との関係を明らかにした。

これらの成果により、当初の目標であった、知識に依存しない高精度な言語モデルを実現、およびその評価を行うことができたと言える。

6.2　おわりに

本研究の動機は、知識に依存する言語モデルが様々な欠点を持つことへの反省であった。しかし、用例から自動的に言語モデルを得ることそのものが本研究の目指したものではない。パターン認識問題の困難さは第 1 で概観した通りであるが、その困難さは多くの要素が作用して発生するものである。したがって、困難を一つ一つ取り出して解決することの意味は薄い。ここで最も重要なものは、システムバランスであると考える。

その意味で、システムの中から言語モデルだけを取り出して改良することだけに甘んじるのは大きな視点を見失う可能性がある。第 5 章の認識実験は、言語モデルをシステムワイドで評価したことに意義があるが、言語モデルの研究においては、今後も広い視野を持った反省を忘れてはならぬと考える。

謝辞

本論文は筆者独りの所産にあらず、迷いが生じれば骨を惜しまず共に悩み考え、弱気になれば励まし力を与えてくださる周囲の方々の支え無くしては完成し得なかったものです。

本研究の機会を与えてくださると共に、大学院での研究生活全般に渡ってご指導、ご支援を賜わった東北大学大学院工学研究科 阿曽弘具教授に心から感謝の意を申し述べたいと思います。

東北大学大型計算機センター 牧野正三教授におかれては、日頃から労をいとわずご指導くださるばかりでなく、公私の別を超えてお心を配ってくださいました。心より感謝致します。

東北大学電気通信研究所 矢野雅文教授には、修士・博士と 2 度にわたり、論文をまとめるにあたっての大きなサジェスチョンを賜わりました。東北大学大学院工学研究科 阿部健一教授には、見落としていた視点からの貴重なご助言を頂きました。ここに記して感謝します。

東北大学大学院工学研究科 大町真一郎博士には、文字認識システム構築の上でご協力を賜わっただけでなく、専門的な内容から研究哲学に至るまでの議論に熱心にお付き合いくださいました。深く感謝致します。

山形大学工学部 伊藤彰則講師には、機会あるごとに研究のヒントを下さり、また無理な相談にも心良く乗ってくださいました。心から感謝の意を表します。

東北大学大学院工学研究科 木幡稔助教授からは、ゼミでの議論で多くのご意見を頂戴しました。広島大学総合科学部 黒岩丈介博士には、多方面でのご指導を頂戴し、また熱心に討論してくださいました。山口大学経済学部 成富敬助教授には、多岐に渡ってご指導くださいました。ここに感謝します。

東北大学情報処理教育センター 後藤英昭博士には、本研究の成果を組み込んだ文書朗読装置の開発にあたり、貴重なご意見を頂戴しました。東北大学大型計算機センター 鈴木基之氏には、音声認識の観点からの研究のヒントを数多く頂きま

した。

最後に、日頃熱心にご討論いただく東北大学工学部通信工学科阿曽研究室の皆様、東北大学大学院情報科学研究科牧野研究室の皆様、並びに東北大学大学院工学研究科電気・通信工学専攻通信システム工学講座コミュニケーション工学分野の皆様に、最大の感謝を込めて御礼を申し上げます。

参考文献

[1] 電子情報通信学会編，小川 英光編著“ パターン認識・理解の新たな展開 — 挑戦すべき課題— ，” p.94, 電子情報通信学会，1994.

[2] 若林 哲史，鄧　　巖，鶴岡 信治，木村 文隆，三宅 康二“ 非線形正規化と特徴量の圧縮による手書き漢字認識の高精度化 ，” 電子情報通信学会論文誌 (D-II)，vol.J79-D-II, no.5, pp.765–774, 1996.

[3] M. Suzuki, N. Kato, H. Aso, and Y. Nemoto, “A Handprinted Character Recognition System Using Image Transformation Based on Partial Inclination Detection,” IEICE Transactions on Information and Systems, vol.E79-D, no.5, pp.504–509, 1996.

[4] Y.L. Chow et al., “BYBLOS: the BBN continuous speech recognition system,” Proceedings of ICASSP 87, pp.89–92, 1987.

[5] 伊藤 彰則“ タスクに依存しない日本語文音声の認識に関する研究 ，” 東北大学審査博士学位論文，1991.

[6] 竹沢 寿幸，末松　博“ 音声・テキストコーパスとその構築技術，標準化動向 ，” 人工知能学会誌，vol.10, no.2, pp.4–16, 1995.

[7] 日本電子化辞書研究所“ EDR 電子化辞書 1.5 版仕様説明書 ，” 日本電子化辞書研究所，1996.

[8] 宇津呂 武仁，松本 裕治“ コーパスを用いた言語知識の獲得 ，” 人工知能学会誌，vol.10, no.2, pp.33–40, 1995.

[9] T. Kuhn, H. Niemann and E.G. Schukat-Talamazzini, “Ergodic Hidden Markov Models and Polygrams for Language Modeling,” Proceedings of ICASSP 94, vol.1, pp.357–360, 1994.

[10] 鷹見 淳一，嵯峨山 茂樹“，逐次状態分割法による隠れマルコフ網の自動生成，”電子情報通信学会論文誌 (D-II)，vol.J76-D-II, no.10, pp.2155–2164, 1993.

[11] 原田　淳，鈴木 基之，牧野 正三“，離散型 HMNet による新聞記事からの文節モデルの獲得，”電子情報通信学会技術報告, SP97-24, 1997.

[12] 竹内 孔一，松本 裕治“，隠れマルコフモデルによる日本語形態素解析のパラメータ推定，”情報処理学会論文誌，vol.38, no.3, pp.500–509, 1997.

[13] K. Lari and S. J. Young, “The Estimation of stochastic context-free grammars using the Inside-Outside algorithm,” Computer Speech and Language, no.4, pp.35–36, 1990.

[14] 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.

[15] C. E. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, vol.27, pp.379–423, 623–656, 1948.

[16] L.R. Bahl, F. Jelinek and R.L. Mercer, “A Maximum Likelihood Approach to Continuous Speech Recognition,” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.PAMI-5, no.2, pp.179–190, 1983.

[17] 村瀬　功，中川 聖一“，単語の共起確率を用いた音声認識の比較，”電子情報通信学会技術報告，SP90-25, 1990.

[18] T. Kawabata, T. Hanazawa, K. Itoh and K. Shikano, “Japanese Phonetic Typewriter Using HMM Phone Recognition and Stochastic Phone-Sequence Modeling,” IEICE Transactions on Information and Systems, vol.E-74, no.7, pp.1783–1787, 1991.

[19] 村上 仁一，荒木 哲郎，池原　悟“，日本文音節入力に対して 2 重マルコフ連鎖モデルを用いた漢字仮名交じり文節候補の抽出精度，”電子情報通信学会論文誌 (D-II), vol.J75-D-II, no.1, pp.11–20, 1992.

[20] 磯谷 亮輔，嵯峨山 茂樹“，自立語と付属語の連鎖統計モデルを用いた音声認識のための候補選択，”電子情報通信学会技術報告，SP93-33, 1993.

[21] 山田 智一，松永 昭一，川端　豪，鹿野 清宏“，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，”電子情報通信学会論文誌 (A), vol.J77-A, no.2, pp.198–205, 1994.

[22] 大附 克年，森　岳至，松岡 達雄，古井 貞熙，白井 克彦“，新聞記事を用いた大語彙連続音声認識の検討，”電子情報通信学会技術報告，NLC95-55, SP95-90, 1995.

[23] K.W. Church, “A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text,” Proceedings of ANLP ’88, pp.136–143, 1988.

[24] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, “A Practical Part-of-Speech Tagger,” Proceedings of ANLP ’92, pp.133–140, 1992.

[25] M. Nagata, “A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A^* N-Best Search Algorithm,” Proceedings of COLING ’94, pp. 201–207, 1994.

[26] P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai and R.L. Mercer, “Class-Based n-gram Models of Natural Language,” Computational Linguistics, vol.18, no.4, pp.467–479, 1992.

[27] 荒木 哲郎，村上 仁一，池原　悟“，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，”情報処理学会論文誌, vol.30, no.4, pp.467–476, 1989.

[28] 伊東 伸泰“，Bigram によるオンライン漢字認識の文脈後処理手法，”電子情報通信学会技術報告，PRU97-6, pp.37–44, 1993.

[29] 池原　悟，白井　諭“，単語解析プログラムによる日本文誤字の自動検出と二次マルコフモデルによる訂正候補の抽出，”情報処理学会論文誌，vol.25, no.2, pp.298–305, 1984.

[30] T. Araki, S. Ikehara and J. Tutihashi, “A New Method of Finding Provisional Boundaries of “bunsestu” using 2nd-Order Markov Model,” Proceedings of 2nd IEEE International Workshop on Robot and Human Communication, pp.114–119, 1993.

[31] 荒木 哲郎，池原　悟，土橋 潤也，笹島 伸一，“ 3 重マルコフモデルによるべた書きかな文の仮文節境界の推定法，” 情報処理学会研究報告，NL102-14, pp.105–112, 1994.

[32] H.-C. Kwon, H.-J. Hwang, M.-J. Kim and S.-W. Lee, “Contextual Postprocessing of a Korean OCR System,” Proceedings of ICDAR ’95, pp.557–562, 1995.

[33] S.K. Katz, “Estimation from sparse data for the language model for a speech recognition,” IEEE Transactions on Acoustics, Speech & Signal Processing, vol.ASSP-35, no.3, pp.400–401, 1987.

[34] M.K. McCandless, “Automatic Acquisition of Language Models for Speech Recognition,” Massachusetts Institute of Technology LCS Techninal Report TR-627, 1994.

[35] E.M. Riseman and A.R. Hanson, “A Contextual Postprocessing System for Error Correction Using Binary n-Grams,” IEEE Transactions on Computer, vol.C-23, no.5, pp. 480–493, 1974.

[36] 杉村 利明，斎藤 珠喜，“ 文字連接情報を用いた読取り不能文字の判定処理 — 文字認識への応用 — ，” 電子通信学会論文誌 (D), vol.J68-D, no.1, pp.64–71, 1985.

[37] 鈴木 基之，“ HMM を用いた音素認識の高精度化に関する研究，” 卒業論文, 1993.

[38] F. Jelinek and R.L. Mercer, “Interpolated estimation of Markov source parameters from sparse data,” in Pattern Recognition in Practice, eds. E.S. Gelsema and L.N. Kanal, pp.381–397, North-Holland, Amsterdam, 1980.

[39] M. Federico, M. Cettolo, F. Brugnara and G. Antoniol, “Language modelling for efficient beam-search,” Computer Speech and Language, vol.9, pp.353–379, 1995.

[40] 森　大毅，阿曽 弘具，牧野 正三，“ 文字認識後処理における trigram モデルの文書適応化手法，” 平成 6 年度電気関係学会東北支部連合大会講演論文集，p.143, 1994.

[41] 伊藤 彰則，好田 正紀“，かな・漢字文字列の連鎖統計による言語モデル，”電子情報通信学会論文誌 (D-II), vol.J79-D-II, no.12, pp.2062–2069, 1996.

[42] 村上仁一“，フレーム同期型フルサーチアルゴリズムを用いた連続音声認識と自由発話への応用，”電子情報通信学会技術報告，SP95-32, 1995.

[43] 高尾 哲康，西野 文人“，日本語文書リーダ後処理の実現と評価，”情報処理学会論文誌，vol.30, no.11, pp.1394–1401, 1989.

[44] M. Nagata, “Context-Based Spelling Correction for Japanese OCR,” Proceedings of COLING-96, pp.806–811, 1996.

[45] 永田 昌明“，単語頻度の期待値に基づく未知語の自動収集，”情報処理学会研究報告，96-NL-116, pp.13–20, 1996.

[46] 松本 裕治，黒橋 禎夫，宇津呂 武仁，妙木　裕，長尾　真“，日本語形態素解析システム JUMAN 使用説明書 2.0，”奈良先端大技報，NAIST-IS-TR94025, 1994.

[47] 中川聖一“，音声言語情報処理研究の動向と研究課題，”情報処理，vol.36, no.11, pp.1012–1019, 1995.

[48] M. Seltzer and O. Yigit, “A New Hashing Package for UNIX,” Proceedings of Winter 1991 USENIX, pp.173–184, 1991.

[49] P. Clarkson and R. Rosenfeld, “Statistical Language Modeling Using the CMU-Cambridge Toolkit,” Proceedings of Eurospeech, 1997.

[50] 中渡瀬 秀一“，統計的手法による単語の切出しについて，”電子情報通信学会技術報告，NLC95-68, 1995.

[51] M. Nagao and S. Mori, “A New Method of N-gram Statistics for Large Number of N and Automatic Extraction of Words and Phrases from Large Text Data of Japanese,” Proceedings of 15th International Conference on Computational Linguistics, pp.611–615, 1994.

[52] 池原　悟，白井　諭，河岡　司“，大規模日本語コーパスからの連鎖型および離散型共起表現の自動抽出法，”電子情報通信学会技術報告, NLC95-3, 1995.

[53] S. Deligne and F. Bimbot, “Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams,” Proceedings of ICASSP 95, pp.169–172, 1995.

[54] 瀧　保夫 “, 情報論 I ,” 岩波書店 , 1978.

[55] 若林 哲史 , 鶴岡 信治 , 木村 文隆 , 三宅 康二 “, 手書き文字認識における特徴量の次元数と変数変換に関する考察 ,” 電子情報通信学会論文誌 (D-II), vol.J76-D-II, no.12, pp.2495–2503, 1993.

[56] 孫　　方 ,大町 真一郎 ,阿曽 弘具 “, カテゴリー間分布を考慮した文字認識用辞書のマルチテンプレート化の一手法 ,” 電子情報通信学会技術報告 ,PRU94-11, pp.41–48, 1994.

[57] 阿曽 弘具 , 越後 和徳 , 木村 正行 “, 文字特徴量空間の性質と特徴抽出法の性能評価法 ,” 電子情報通信学会論文誌 (D-II), vol.J76-D-II, no.11, pp.2285–2294, 1993.

[58] 細越 淳一 , 阿曽 弘具 “, 文字特徴量空間の構造 ,” 1994 年電子情報通信学会春季大会講演論文集 , D-485, p.221, 1994.

[59] S. Watanabe, “Knowing and Guessing,” John Wiley & Sons, 1969.

[60] 飯島 泰蔵 “, パターン認識理論 ,” 森北出版, 1989.

[61] 加藤　寧 , 安倍 正人 , 根元 義章 “, 改良型マハラノビス距離を用いた高精度な手書き文字認識 ,” 電子情報通信学会論文誌 (D-II), vol.J79-D-II, no.1, pp.45–52, 1996.

[62] F. Sun, S. Omachi and H. Aso, “Precise Selection of Candidates for Handwritten Character Recognition Using Feature Regions,” IEICE Transactions on Information and Systems, vol.E79-D, no.5, pp.510–515, 1996.

[63] 西野 文人 , 杉本 尚美 “, 擬似文字認識による文字認識後処理実験環境 ,” 電子情報通信学会技術報告 , PRU94-44, pp.41–48, 1994.

[64] 杉村 利明 “, 候補文字補完と言語処理による漢字認識の誤り訂正処理法 ,” 電子情報通信学会論文誌 (D-II), vol.J72-D-II, no.7, pp.993–1000, 1989.

[65] 森　大毅，阿曽 弘具，牧野 正三“，trigram モデルを用いた低品質文書の高精度認識，”平成 7 年度電気関係学会東北支部連合大会講演論文集，p.376, 1995.

[66] 青江 順一“，トライとその応用，”情報処理，vol.34, no.2, pp.244–251, 1993.

[67] 孫　　寧，田原 透，阿曽 弘具，木村 正行“，方向線素特徴量を用いた高精度文字認識，”電子情報通信学会論文誌 (D-II), vol.J74-D-II, no.3, pp.330–339, 1991.

研究業績

論文

[1] Hiroki Mori, Hirotomo Aso, and Shozo Makino, "Robust n-Gram Model of Japanese Character and Its Application to Document Recognition," IEICE Transactions on Information and Systems, vol.E79-D, no.5, pp.471–476, 1996.

[2] 森　大毅，阿曽 弘具，牧野 正三，" 文字・単語 n-gram の融合に基づく言語モデル，" 電子情報通信学会論文誌 (D-II) (条件付採録)．

国際会議

[1] Hiroki Mori, Hirotomo Aso and Shozo Makino, "Japanese Document Recognition Based on Interpolated n-gram Model of Character," ICDAR '95: Third International Conference on Document Analysis and Recognition, Montréal, Canada, pp.274–277, 1995.

研究会

[1] 森　大毅，阿曽 弘具，牧野 正三，" 2 重マルコフモデルを用いた日本語文書認識後処理，" 情報処理学会研究報告，NL102-12, pp. 89–96, 1994.

[2] 森　大毅，阿曽 弘具，牧野 正三，" 文字・単語 n-gram の融合に基づく言語モデル，" 電子情報通信学会技術報告，NLC96-24, 1996.

[3] 森　大毅，阿曽 弘具，牧野 正三，" 再現性を考慮した文字列に基づく統計的言語モデル，" 電子情報通信学会技術報告，NLC97-47, 1997.

全国大会

[1] 森　大毅, 阿曽 弘具 “, 2 重マルコフモデルを用いた日本語文書認識 ,” 1994 年電子情報通信学会春季大会講演論文集 , vol.7, p.223, 1994.

[2] 森　大毅 , 阿曽 弘具 , 牧野 正三 “, trigram モデルによる文字切り出しと認識の統合処理 ,” 1995 年電子情報通信学会情報・システムソサイエティ大会講演論文集 , p.192, 1995.

[3] 森　大毅 , 阿曽 弘具 , 牧野 正三 “, 再現性 n-gram 統計の効率的な構成法 ,” 1996 年電子情報通信学会情報・システムソサイエティ大会講演論文集 , p.56, 1996.

[4] 森　大毅 , 阿曽 弘具 , 牧野 正三 “, Forward-Backward アルゴリズムに基づくテキストコーパスからの擬似単語抽出法 ,” 1997 年電子情報通信学会総合大会講演論文集 , p.91, 1997.

支部大会

[1] 森　大毅 , 阿曽 弘具 , 牧野 正三 “, 文字認識後処理における trigram モデルの文書適応化手法 ,” 平成 6 年度電気関係学会東北支部連合大会講演論文集 , p.143, 1994.

[2] 森　大毅 , 阿曽 弘具 , 牧野 正三 “, trigram モデルを用いた低品質文書の高精度認識 ,” 平成 7 年度電気関係学会東北支部連合大会講演論文集 , p.376, 1995.

索 引

1 文字語, 40, 81

ARPA, 8

back-off スムージング, 18, 37
Backward 確率, 59
Baum-Welch アルゴリズム, 25
Bayes の定理, 26, 71
bigram, 17
binary n-gram, 23

confusion 行列, 78

DP マッチング, 9
DTW, 9

EM アルゴリズム, 25, 26

Forward-Backward アルゴリズム, 26, 44, 53
Forward 確率, 59

held-out データ, 27, 62
held-out 補間法, 27
HMM, 9, 10, 25
HMNet, 9

multigram, 54, 65

n-gram, 10, 14, 24, 36, 39, 52, 53, 55, 56, 62, 66, 80
  単語の, 11, 53, 55
  文字の, 13
  文字列の, 40, 53, 55
n-gram 確率, 15, 27, 80
  文字の, 49
n-gram モデル, 10, 15, 22, 36, 37, 39, 53, 65
  文字の, 14, 37

OCR, 15

Poisson 分布, 47

SCFG, 10
superword, 13, 53–56, 59, 61, 62, 65, 66
superword n-gram 確率, 54
superword n-gram モデル, 54, 55, 79
superword bigram モデル, 62, 83
superword unigram 確率, 59, 62
superword unigram モデル, 54, 63, 65
superword モデル, 56

trigram, 17, 55
trigram モデル
  文字の, 62

unigram, 19, 65

Viterbi アルゴリズム, 44, 74

word accuracy, 48

zerogram, 32

後処理, 17
誤り確率, 70
誤り確率最小化, 68

一般 superword n-gram モデル, 61
一般 superword bigram, 62
一般 superword unigram モデル, 62, 63, 66
意味, 36
意味解析, 6, 7
意味論, 4
印刷文書, 32

エルゴーディック HMM, 61

音響処理部, 9, 14
音声言語, 2, 6
音声認識, 9, 10, 12, 14, 36, 37
音節, 12, 14, 17, 18
音節文字, 15, 17
音素, 12, 14
音素 n-gram, 38
音素認識, 23

解析結果, 40, 41, 79, 84
改良型マハラノビス距離, 72
過学習, 39
係り受け, 6, 11
格, 7
確定文字, 75
獲得単語集合, 40, 47
格フレーム, 7
格文法, 7
確率文脈自由文法, 10
確率モデル, 9, 38
  言語の, 68
隠れマルコフモデル, 9, 25, 40, 54, 59
活用, 15, 51
活用規則, 51
活用形, 40
かな trigram, 17
可搬性, 4, 8, 12, 66
仮文節, 17
頑健性, 4, 11
簡素化マハラノビス距離, 72

機能語, 15, 53
寄与度, 44, 45

句構造文法, 7
クロスバリデーション, 24, 62
訓練テキスト, 11, 12, 17

形態素, 4, 14, 36
形態素解析, 4, 7, 9, 14, 17, 36–40, 42, 44, 52, 53, 66
形態素解析システム, 37, 50, 52
形態論, 4
言語処理部, 9, 14, 75
言語的尤度, 73, 79
言語モデル, 3, 9, 12, 14, 15, 31, 37, 39, 46, 53, 63, 66, 68, 71, 73, 74, 80
  確率的, 9, 12, 29
  コーパスに基づく, 7–9, 36
  知識に基づく, 4

語彙, 4, 66
語彙論, 4
構文, 36
構文解析, 6
候補外文字, 78, 79
候補文字集合, 79
コーパス, 4, 8, 10, 17, 36, 38, 48
個別文字認識, 68
語用論, 4

再現性, 13, 18, 53–56, 61, 63, 66
再現率, 79
最大尤度基準, 24
最尤推定, 18, 22, 26, 27, 32, 39, 44, 45, 70
削除補間法, 24, 27, 31–33, 39, 41, 42, 44

事後確率, 68, 70, 71
辞書, 7, 44, 80
  形態素, 51
指数分布, 42, 45, 47, 65
事前確率, 32, 71
自然言語処理, 2, 3, 5, 10, 12, 36, 68
自然言語理解, 12
実質語, 15
修正ベイズ 2 次識別関数, 72
柔軟性, 4, 7
主記憶, 50, 56
主ビーム, 79, 81
少数データ問題, 11–13, 15, 17, 19, 22, 29
冗長性, 70
冗長度, 29
素性, 7
素性構造, 7
スムージング, 13, 15, 22, 23, 29, 31, 33, 37–40, 62, 66

正規化頻度, 53
正規分布, 72
セグメント同期型多重ビームサーチ, 79, 81
接辞, 53
線形補間, 41, 62

大分類, 69, 74, 79
タグ付きコーパス, 8, 11, 12, 36, 52
単一化文法, 7
単語, 10, 14, 36, 40, 51, 52
単語 n-gram, 13, 36–38, 40, 41, 49, 62, 79, 83
単語 n-gram 確率, 40, 41, 49
単語 trigram, 83
単語解析精度, 38, 62
単語クラス, 12, 52
単語辞書, 4
単語集合, 40
単語照合, 4, 38

知識獲得のボトルネック, 9, 12

通信路行列, 71

適応, 51
  n-gram モデルの, 33
  教師なし, 33
適合率, 79
テキストコーパス, 8, 36, 52

統計的決定理論, 71
同形文字, 2
統語論, 4
動的時間伸縮, 9
特徴ベクトル, 71, 72
トライ, 80, 81

長さ制限, 63
長さ制限付き superword n-gram モデル, 61
長さ制限付き superword bigram モデル, 62, 66
長さ制限付き superword モデル, 65
生コーパス, 8, 11, 14

ヌル遷移, 25, 42

パープレキシティ, 12, 13, 24, 29, 31, 32, 46–48, 63, 71
　単語, 48
パターン整合法, 72
パターン認識, 1, 3, 6, 9, 48, 68
ハッシュ法, 56
汎化性, 53
汎化能力, 48, 61

ビームサーチ, 79
被覆率, 18
表音文字, 14
品詞, 4, 8, 10, 11, 14, 17, 40, 52
品詞 n-gram, 38

複合 n-gram 確率, 80
複合 n-gram モデル, 41, 42, 46, 52, 65
複合 superword bigram 確率, 62
　文字で補間された, 62
複合 superword bigram モデル, 62
複合語, 53
複合類似度法, 72
副状態, 56
副ビーム, 79
部分空間法, 72
フロアリング, 23, 31, 33
分析済みコーパス, 8–10, 36
文節, 6, 9
文法, 4, 7, 10, 17, 36, 51, 53
文脈, 15, 68, 81
文脈自由文法, 5–7, 10

べた書きテキスト, 36, 37

マハラノビス距離, 72
マルコフ過程, 10, 25
　エルゴード性の, 29
マルコフ性, 15
マルコフ情報源, 24
マルコフモデル, 15, 17, 25
マルコフ連鎖, 15

未確定文字, 75
未知語, 37, 40, 41, 44, 49
未知語テンプレート, 38
未知テキスト, 40
未知文章の解析シミュレーション, 13, 44
未定義語, 40, 46

文字 n-gram, 14, 38, 44, 52, 55, 69, 74, 79
文字 n-gram 確率, 39, 49

文字 n-gram **モデル**, 15, 17
文字 trigram, 83
文字 trigram **確率**, 63
文字 trigram **モデル**, 29, 31, 50, 66, 78, 79
文字切り出し, 68, 70
文字言語, 24
文字認識, 2, 12, 14, 36, 37, 68
文字認識システム, 12, 13, 68, 74, 79

ユークリッド距離, 73
有限オートマトン, 5, 9, 10
尤度関数, 26, 74, 80

類似文字, 2

連続音声認識, 4, 5, 8–10, 15, 24, 31, 38, 48, 80

分かち書き, 11, 36, 44, 52