修士学位論文

言語の確率モデルを利用した文字認識に関する研究

東北大学大学院工学研究科電気及通信工学専攻

森　大毅

目 次

1 序論 **1**

1.1 研究の背景 1

1.2 研究の目的 7

1.3 本論文の構成 9

2 n-gram による日本語モデル **11**

2.1 はじめに 11

2.2 文字 n-gram モデルの定式化 11

2.3 trigram モデルと少数データ問題 14

2.4 trigram モデルのスムージング 19

2.4.1 フロアリング 19

2.4.2 削除補間法 20

2.5 スムージング法の比較及び検討 24

2.5.1 パープレキシティ 27

2.6 まとめ 31

3 n-gram モデルを用いた文字認識後処理の効果 **33**

3.1 文字認識の情報理論的解釈 34

3.2 尤度関数 37

3.2.1 パターンの生起確率密度関数の推定 37

3.2.2 重み付け 38

3.3 最適化アルゴリズム 39

3.4 文字認識後処理系の実現 41

3.4.1 個別文字認識系 41

3.4.2 予備実験 43

3.4.3 認識対象 . . . 43
3.4.4 認識実験 . . . 46
3.5 まとめ . . . 48

4 切り出しと認識を同時に行う文字認識方式の提案 49
4.1 文字切り出し問題 . . . 50
4.1.1 分離文字 . . . 50
4.1.2 接触文字 . . . 52
4.1.3 日本語 trigram モデルの導入 . . . 52
4.2 切り出し関数 . . . 53
4.3 切り出し ・認識アルゴリズム . . . 54
4.3.1 平均文字サイズ算出 . . . 54
4.3.2 接触文字強制分離 . . . 54
4.3.3 分離文字統合と最尤系列導出 . . . 54
4.4 日本語文字 trigram モデルを用いた文書認識系の実現 . . . 55
4.4.1 個別文字認識系 . . . 55
4.4.2 認識対象 . . . 55
4.4.3 言語モデル . . . 57
4.4.4 認識率 . . . 57
4.4.5 実験結果 . . . 57
4.5 低品質文書への対処 . . . 59
4.5.1 ビームサーチ法の導入 . . . 62
4.5.2 ビームサーチ法を用いた認識実験 . . . 62
4.6 まとめ . . . 63

5 n-gram モデルの改善 64
5.1 言語モデルの適応化 . . . 64
5.2 trigram 確率再推定アルゴリズム . . . 66
5.3 実験 . . . 69
5.4 まとめ . . . 69

6 結論 71
6.1 本論文の成果 . . . 71
6.2 おわりに . . . 72

謝辞 **73**

参考文献 **74**

研究業績 **78**

索引 **79**

図 目 次

1.1 実際の日本語の文章 (天声人語より) 3
1.2 同じ文章を文字ごとにばらばらに並べ換えたもの 3

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 13
2.2 trigram の出現回数 16
2.3 訓練テキストの量による異なり trigram 数の増加傾向 17
2.4 n-gram の被覆率 18
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 22
2.6 削除補間法の処理の概要 25
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 26
2.8 「社説」モデルのパープレキシティ 28
2.9 タスク間の関係と言語モデル 30
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 . . . 31

3.1 情報理論と文字認識の対応関係。[7] を参考にした 34
3.2 Viterbi アルゴリズムによる最尤系列の導出 41
3.3 方向線素特徴量の算出過程 (線素の数え上げ) 42
3.4 確定文字の認識率、および未確定率 44
3.5 累積認識率 45
3.6 ガイドライン付きサンプル (部分) 45
3.7 言語的尤度の変化と誤認識率の関係 46
3.8 スムージング手法と誤認識率の関係 47

4.1 画像情報だけでは正しく切り出しが行えない例 51
4.2 接触文字の例 52
4.3 行イメージと列ヒストグラム 55

4.4 分離文字統合と最尤系列導出のためのグラフ 56
4.5 切り出し方式と誤認識率の関係 58
4.6 印字品質と誤認識率の関係 60
4.7 出力候補数 N_c と誤認識率の関係 61
4.8 ビーム幅と誤認識率 62

5.1 適応化がうまく行く例 67
5.2 適応化がうまく行かない例 68
5.3 改善例 70

表 目 次

1.1 文字認識技術とその応用 [18] 3
1.2 言語の確率モデルの分類 7

2.1 実験に用いる日本語テキストデータ (単位 字) 15

3.1 個別文字認識系の実験条件 43

4.1 個別文字認識系の実験条件 56

第 1 章

序論

1.1　研究の背景

人類が 1940 年代に初めて電子計算機を発明して以来、電子計算機に人間の知的活動の代行をさせようとする試み、すなわち「人工知能」の研究は、今日に至るまで常に多くの計算機科学者の興味をひきつける関心事であり続けている。それは同時に、「人工知能」の実現が当初考えられていたよりもはるかに難しい問題であったことを示唆するものである。その間のハードウェアの進歩、計算機の「物理的」な性能の向上に議論の余地はないが、本来期待された「論理的」「知的」な能力の実現を目指す研究の歴史は、それ自身の本質的な困難さが明らかになる過程であったと言っても過言ではない。

人工知能の分野において、パターン認識は一つの代表的な課題である。パターンを認識することは、人間や動物の頭脳において基本的な能力であり、また最も得意とする情報処理作業であることが知られている [1]。こうしたことから、パターン認識は人工知能の部分問題として広く研究されてきた。とりわけ文字認識は、その応用の重要性もあいまってパターン認識の中心的研究対象であり、この問題を解決することは「知能」の謎を解き明かす最初の一歩と期待されてきた。事実、手書き数字の認識に始まる文字認識研究は、数十年を経て大きな成果をあげた。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 [2]、多くの **OCR**(Optical Character Reader, 光学的文字読み取り装置) を入手できるほどに実用化されている。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認識精度の向上は目ざましく、最近では実用レベルに達しつつある。

しかし、決して文字認識問題が克服されたのではない。それどころか、これまでの成果は人工知能が目標とする知的活動の具現とは異質のものに思われる。これらは単に、ある限られた状況の下での「文字当て技術」[2] の進歩に過ぎず、想定されていない状況下では

全く機能しないからである。現実に、低品質文字の認識、複雑背景下での認識、文字切り出しの問題など、人間にとってはたやすく思えるにも関わらず未だ解決されていない多くの問題が残されている。文字認識が実用の域に入ったと言われるようになった今日、もう一度人間の認知メカニズムの模倣という視点に立ち戻り、「認識とは何か」を問い直すことは、真の人工知能の実現に近づくためには必要不可欠な姿勢であろう。

このように考えると、文字認識の問題を自然言語処理の問題と結び付けるのはごく自然であると言える。多くの場合、文字はそれ自身が単独で使われるのではなく、文あるいは文章の構成要素としての位置を占めている。人間が文字を認識する過程において、個々の文字認識が他の文字に全く影響されずに行われているとは考えにくい。むしろ、文字認識は内容の理解と同時に行われていると見るべきである。もちろん人間は個別文字の認識能力においても優れており、文字を文の一部として見なすことができない場合でもほぼ完全に文字の同定ができる。だが、人間にとって各々の文字を独立と考えて認識することは極めて効率の悪い仕事である。このことは、図 1.1, 図 1.2 に示すような実際の文章とそれをランダムにかきまぜた文字の集まりとを「読む」速度を比べてみれば明らかである。また現実の文書中には、形状が酷似していてそれ自身だけでは区別が難しい文字グループ (類似文字集合) に属するような文字が多く存在する。さらに、後の章で示すよう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。人間がこのような文字をも正確に判読できる大きな要因の一つが、背景の言語知識であることに疑いはない。このように、真に知的な文字認識の実現は、文を単なる文字の集まりでなく、あくまで文としてとらえるための技術、つまり自然言語処理との融合なしにはあり得ない。

自然言語処理とは、人間の行う知的情報処理のうちでもとりわけ人間らしい活動である「ことば」を電子計算機に扱わせようとする試み全般を指す用語である。自然言語処理が目標としたのは主に機械翻訳、かな漢字変換、対話などであり、電子計算機の登場以来数十年の間一貫して進められてきたこと、また後述するような困難のために多くの分野で未だ研究途上であることなど、パターン認識と事情を同じくする点が多い。

では、文字認識のためにはどのような種類の自然言語処理が望まれるのだろうか。文字認識をその応用される分野によって分類したものを表 1.1に引用しておく。文字認識における自然言語処理は、それが認識過程の後段に位置することが多いことからしばしば文字認識後処理の名前で呼ばれるが、この表の分類のように、文字認識の技術そのもののみならず、後処理の対象もその適用分野によって異なると考えるのが適当であろう[1]。本論文で扱

[1]もっとも、アプリケーションが広がるにつれて認識対象の違いに関係なく共通な後処理技術が必要となる、とする考えもある [18]。

１９６４年、ニューヨークのハーレムで黒人暴動が起こった時、人びとは興奮の極にあった。「武装して警官と戦え」という騒ぎの中で、一人の指導者がいった。「われわれは非暴力主義を貫いてこの流血の惨事を終わらせよう」。いきりたった群衆が指導者に殴りかかろうとすると、七、八十人の聴衆がごく自然に指導者を取りまき、敵意に満ちた男たちを退けた。七、八十人はその後街頭にでて傷ついた人を助け、警官にも暴力中止を訴え続けた。指導者Ｂ・ラスティンは「あの時、私は非暴力主義の強さを知った」と回想している。

図 1.1: 実際の日本語の文章 (天声人語より)

。き警一動極群てわた４官たでンュ中」ちとす、６暴。指は然ぎの者想のハつい血義せ流。りご者れたさか私き人た敵で自者。人殴にっ時と十・にりー意まレ人スの、と人主しけし、官の傷がの。七こＢて、っに年っーラ後ろあったと警にたら奮、はで力衆衆義」を暴指ニけ武はを取強黒あのの非導力惨「い力の回頭、ヨ続も終りい「び導者事、」導いに貫の聴人をうてくーうる１暴中を人指たえをムちそ八が起け。かた「と、助わ導非テ訴はと主装指興満戦えはィ十よ七をがたをいた９れ男のる退八知騒こ、街ク暴っうてにい止がわ時た

図 1.2: 同じ文章を文字ごとにばらばらに並べ換えたもの

表 1.1: 文字認識技術とその応用 [18]

文字認識技術	代表的応用	主な読み取り対象
活字文字認識	文書の読み取り	一般文
手書き文字認識	帳票の読み取り	住所、氏名、商品名
オンライン手書き文字認識	ペンコンピュータ	住所、氏名、商品名

う認識対象は、表 1.1の活字文字、またはもっと広く印刷文字 (写植文字、ワープロの出力などを含む) とする。また、特に一般文書中の文字の認識を想定している。したがって、書式が固定している帳票や葉書の宛名を対象としたものと違い、文書の内容にはできるだけ制限を加えるべきでない。本研究では、日本語で書かれた文章全てを対象とする。実際の文書中には実に多彩な表現が現われる可能性があり、実用性を重視するならばこれらをなるべく許容するような柔軟な処理が必要である。

一般に、自然言語処理に共通の最大課題は曖昧性解消の問題であると言われる。自然言語処理における推論の過程では大量の探索が必要であるが、この探索の結果が多義であることを曖昧であるという。曖昧性を減らすとは探索空間を小さくすることである。パターン認識における自然言語処理とは、結局ここでいう曖昧性を解消するための処理そのものである。自然言語処理が実際に扱わねばならないものは、極めて複雑な人間の活動全体から生じる言語であり、これらをできるだけ広く包含するような語彙ベースを計算機上に実現しなければならない。しかし、語彙を増やすことによって探索空間が極端に増大するというジレンマに陥る。

この問題に対して従来取られてきたアプローチは、大きく 2 つに分けることができる。一つは知識に基づく方法、または規則主導型の方法である。曖昧性は、同一の処理対象に複数の規則が適用され得るために発生するものであるから、規則を細分化する、すなわちより多くの知識を導入することによって複数の規則の適用を抑え、曖昧性を減らすことができる。規則の細分化の例としては、品詞の細分類、文法の精密化が考えられる。また、このようにしてもなお曖昧性が残る場合には、複数の適用可能な規則を何らかの手段で評価し、その評価値の高い規則を適用する、という過程が必要となる。この評価基準の代表例が、形態素解析の分野における最長一致法 [3] や文節数最小法 [4] などのヒューリスティックである。しかし、規則自身や規則評価関数の作成・管理は人手で行われなければならず、規則が大規模になるにしたがってそれは非常に困難な作業となる。特に品詞分類や文法規則の枠組みに手を加えるのは難しい。実用に耐える適切な規則体系をまとめ上げるためには言語学的知識を動員した大がかりな調査・研究が必要となる。また仮にそれが可能だったとしても、実際の自然言語処理システムを構築するためには、規則体系を電子化して機械可読な形式にしなければならない。実際にこのような試みもいくつか見受けられる [5]。しかし言葉というものは常に変化し続けるものであり、人手によるメンテナンス (例えば、新語の登録) は将来も続けて行く必要がある。こうした知識獲得の困難さのため、規則主導の方式には限界がある。この種の問題はエキスパート・システムの分野ではファイゲンバウム・ボトルネック [6] として古くから指摘されている。

規則・知識に基づく方法に代わるもう一つの曖昧性解消の手段として近年注目を浴びて

いるのが、用例に基づく手法である。これはエキスパート・システムの分野でいう事例ベース推論に対応するものであり、自然言語処理においてはコーパスを用いた手法とも呼ばれる。この手法では、知識を人手で与えるかわりに、実際の用例を集めた大規模データベース (コーパス) から知識を自動または半自動で抽出する。用例を用いた手法は高集積・大容量の時代の到来と共に現実的になってきており、規則主導方式の限界を越えるものとして期待を集めている。

確率モデルを用いた手法は、この用例ベースの手法の特殊ケースと位置づけることができる。この手法では、自然言語の営みを確率的な事象とみなし、何らかの確率モデルを用いてこれを近似する。自然言語を良く近似するモデルを用いることで、曖昧性を自然言語が本来持つそれに近い程度まで解消することができる。確率的言語モデルにおいては、通常の文法規則に対応するものはモデルのパラメータによって表現され、パラメータの値は普通コーパスの統計的処理 (学習) によって得る。言語の確率モデルは、非確率的・知識主導・構造的な文法を一般化したモデルととらえることができ、適用される規則に対する評価を確率という扱いやすい尺度によって与えることで、曖昧性の問題を自然に解決する。

確率的言語モデルの代表例は確率文脈自由文法 [7] である。文脈自由文法は自然言語の統語構造を良く表現できることから自然言語処理において基本的である。確率文脈自由文法は文脈自由文法の書き換え規則を確率付きに拡張することで得られる。書き換え確率の推定および文解析を統計的に行う手法としては、Inside-Outside アルゴリズム [9], CYK 法に基づくアルゴリズム [7] 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、想定される言語のサイズを制限し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、規則や知識に基づく自然言語処理の場合と同じ問題が残る。このように、確率文脈自由文法は原理的には明らかに通常の文脈自由文法に比べ探索空間が小さく有利であるにも関わらず、応用のためにはいくつかの乗り越えなければならない課題がある。

確率文脈自由文法の場合と同様に、正規文法の書き換え規則に確率の概念を導入することにより確率正規文法が定義される。これは確率有限オートマトンが受理する言語のクラスと等価であり、また音声認識の分野で広く用いられている隠れマルコフモデル (Hidden Markov Model, **HMM**) で表現できる。正規文法は Chomsky 階層 [8] においては文脈自由文法の下位に位置する。HMM による探索空間は確率文脈自由文法のそれよりも大きく曖昧性を解消する能力が劣ることが知られているが、HMM の学習アルゴリズムの一つである Baum-Welch のアルゴリズム [10] の計算量は非終端記号数の 2 乗のオーダであり、確率文脈自由文法よりは扱いやすく、むしろ実用的な場合がある。日本語の文節構造は有限

オートマトンで規定できると言われ [7] [11]、これに確率を導入した応用例が形態素解析の分野に見られる [12]。

また、別の確率的言語モデルとして、良く知られている n-gram モデルがある。n-gram とは n 個の組を意味し、n-gram モデルは $(n-1)$ 重マルコフモデルと等価である。n-gram の単位としては、単語、品詞、文字などが考えられる。このモデルが自然言語の性質を良く反映することは、古くは Shannon[13] によっても言及されている。自然言語のマルコフ性が仮定できる範囲では、n を無限大とした場合に自然言語の完全なモデル化ができる。通常 n-gram モデルという場合には、n を有限で打ち切ったものである。確率文法と n-gram モデルはどちらも n が無限大である場合を近似したものと考えられるから、その近似の手法が異なるだけで本質的にどちらが優れているといったことは言えない。実際、連続音声認識の分野では、n-gram モデルが音響的曖昧性の解消に有用であることが広く知られている [14][15]。

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。確率文法の場合、学習アルゴリズムは多少複雑であり計算コストが大きい。
- 構文解析が不要であるため処理が高速である。

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サンプルは用意できない (少数データ問題)。訓練サンプルの量が小さいと、未知データに対する頑健性が失われる。

以上で説明した言語の確率モデルをまとめて表 1.2に示す。なお、ニューラルネットワーク [16] も一種の確率的学習機械と考えることができるが本論文では考慮しない。ニューラル

表 1.2: 言語の確率モデルの分類

探索空間	学習	確率モデル	
小	困難		n 大
↑	↑		↑
		確率文脈自由文法	
			n-gram モデル
		確率正規文法または HMM	
↓	↓		↓
大	容易		n 小

ネットワークの言語モデルへの応用としては、ニューラルネットワークの持つ汎化能力を n-gram モデルによる英単語品詞列予測に適用した例がある [17]。

これまでにも文字認識と自然言語処理との統合について多くの研究努力がなされてきた [18]。それらの手法間に細かい相異点はあるものの、対象が印刷文書である限り、土台となっているのはほとんどが単語辞書との照合と文法に基づく規則主導型の言語処理である。規則主導型の後処理方式が成果をあげてきたのは、言いよどみ、言い直し、「あー」などの間投詞の挿入や省略が頻繁に起こる音声言語と違い、構造が比較的はっきりしている文書が対象であったためと思われる。これらの手法に共通の主たる問題点は、先に触れた規則・知識ベースの自然言語処理の欠点をそのまま引き継いでいることである。知識の他に、制約として確率正規文法の概念を取り入れ、品詞遷移確率をコストに加味することも試みられている [19][20] が、知識の獲得がボトルネックになっている点では同じである。

1.2　研究の目的

本論文では、このような従来の規則主導型の文字認識後処理手法の本質的問題点に着目し、これを解決するような新しい知的文字認識の枠組をつくることを目標とする。そのために、コーパスからの確率的言語モデルの自動獲得、およびその文字認識への応用を研究の対象とする。

処理の単位としては、従来の手法で主として用いられてきた単語や品詞ではなく、最小単位である文字を選んだ。これは、次のような理由による。

- 単語を単位とするためには単語の定義をしなければならないが、日本語の場合これは単語辞書を用意することと同等である。単語辞書を人手によって与えることは知識を与えることに他ならないから目標に反して望ましくない。また単語知識をべた書きされた文から自動獲得するための効率的な手法はまだ知られていない。

- 単語 (あるいは品詞) を基本とした確率的言語モデルの学習のためには、ラベル付けされた文のコーパスが必要である。英語の場合には実際にこのような目的で提供されているコーパスが存在する [21][22] が、日本語文では実用に耐える品質・規模を持つコーパスの入手が難しいのが現状である。ラベル付けされていないべた書きのテキストから間接的に得る方法もあるが、単語境界が明確なラテン語系言語と違い、日本語の場合には形態素解析が必須となる。形態素解析を精度良く行うためには人手による知識の投入か用例による知識獲得が必要であり、結局堂々巡りとなる。

単位を文字とした以上、言語モデルとしては n-gram モデルを選ぶのが妥当であろう。確率文法も可能であるが、採用する意義があまり感じられない。文字の n-gram モデルは、校正支援 [23]、音声認識 [24] [25] などで有用性が確かめられている。類似したものでは、文字連接の可否のみを問題とする binary n-gram と呼ばれる特殊な統計量の英語 [26] 及び日本語 [27] 文字認識への応用例があるが、コストに見合うだけの制約とならないと考えられるので扱わない。n-gram モデルを採用したことによって、先に示したように多彩な言語表現に柔軟に対応した、学習が容易で高速な文字認識の実現が期待できる。

前述したように、n-gram モデルの最大の問題点は、探索空間を小さくするために n を大きくしようとすると、パラメータ数が急速に増大して学習が困難になることであり、実際の文字認識システムを構築するためにはどうしてもこの点に何らかの対策を講じなければならない。本論文では **trigram** モデル、すなわち $n = 3$ として生成した言語モデルが曖昧性を減少させる有効な手段となると予想し、パラメータ数に対する訓練サンプルの少なさを、パラメータ空間のスムージングの導入によって補う。このようにして得られたモデルは非文法的表現や未知語をある程度許容するため、目標とする柔軟な文字認識システムの実現が期待できる。

本論文では、このような言語の確率モデルをまず定式化が簡単で文字切り出しの不要な文字認識系に適用し、類似文字の判定に関する有効性を実験により検証する。この時、訓練サンプルと認識対象との性質の相違に起因する性能の劣化が論じられるが、第 5章において、n-gram モデルの学習容易性を活かした動的なモデル更新手法を導入し、この問題の改善について検討する。また、文字認識の未解決問題である文字切り出し問題にも自然言語処理の導入が必要であることを示し、この問題を統計的パターン認識の立場から扱える

よう定式化する。さらに、本論文では文字認識のもう一つの未解決問題である低品質文書の認識を取り上げる。従来の単語辞書ベースの文字認識では計算量的に困難な低品質文書認識を、n-gram モデルを用いて現実的に処理するための手法を提案し、実験により有効性を確認する。

文字単位の n-gram モデルは、人間の認識機構の一部としての言語活動とは若干異なる振る舞いをするであろうことが予想される。これは本来望ましいことではない。本研究における文字 n-gram モデルは、あくまでコーパスに基づく手法のうち現時点で最も実現が容易かつ有望なモデルとして選択された。どのような言語モデルが究極の姿かという問いに今のところ解答は見いだされていないが、本研究はそうした目標へ到達するための基礎研究の一環と位置付けられる。

1.3　本論文の構成

本論文の構成は以下のごとくである。

第 2章では本研究の主要構成要素である文字の n-gram に基づく日本語のモデル化について定式化を行う。ここではまた少数データ問題について、実際のテキストデータの統計結果を示しながら言及し、これに対処するために 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の尺度としてパープレキシティと呼ばれる量を導入し、この量を用いて評価実験を行う。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章および第 4章では、第 2章によって定式化された日本語 n-gram モデルを用いた文字認識について議論する。第 3章ではまず、文字認識問題を雑音のある通信路における復号誤り確率最小化問題とみなし、文字切り出し位置が与えられている場合について、文字認識後処理を定式化する。次に、誤り確率最小化のために必要となる文字認識結果の確率的評価の方法について説明する。さらに n-gram にもとづく文字認識後処理を高速に実行する手法である Viterbi アルゴリズムによる処理手順を示す。また実際の印刷文字認識系を用いた後処理実験を行い、種々の言語モデルと文書セットを用いながら本手法の有効性を明らかにする。

第 4章では、実用的な文書認識システムの核となる行内文字認識部を実現する。まず行内文字切り出し問題が画像情報単独では本質的に解決できないことを示し、この問題に日本語文字 n-gram モデルを適用することを提案する。次に、n-gram モデルを用いて文字切り出しと認識を同時に行うアルゴリズムを示し、認識実験によって有効性を実証する。最後に、候補外文字の問題を回避するための認識アルゴリズムとしてビームサーチ法に基づ

く手法を提案し、低品質文書の認識精度改善を図る。

第 5章では、認識対象中の未知の語彙に対応するための n-gram モデルの教師なし適応化手法について検討する。このためのアルゴリズムを提案し、性能を評価する。

第 6章は結論である。

第 2 章

n-gram による日本語モデル

2.1 はじめに

n-gram モデルを用いて確率過程を推定することは良く行われていることであるが、本論文のように対象が自然言語である場合には特に注意を払わねばならないことがいくつかある。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

まず最初に、情報源記号の集合である日本語の文字セットを定義する。

定義 **2.1(1)** 空白は文字ではない。

ラテン系言語と違い、日本語文中には基本的に空白は必要ではない。段落空けを含むレイアウト上の問題で空白を用いることもあるが、レイアウトの認識は本研究の対象外である。定義にしたがい、空白文字が認識結果となることはない。

定義 **2.1(2)** JIS X0208-1983[28] で規定されているものだけが文字である。

ワープロで作成された文書中には、「全角文字」「半角文字」などと称して同種の文字に対して複数の形状を与えている場合がある。これらはそれぞれ違う文字セットに属しており、

書き手が意識して使いわけていることもある。定義 2 は、このようなものを区別せずに同一の文字とみなすことを明らかにするためのものである。また文字集合 JIS X0208-1983 に含まれない文字も日本語文中に出現するが、このような文字をも扱える文字認識系は実現が難しいので考慮の対象外とする。

定義 **2.1(3)** 定義 2.1(1), 定義 2.1(2) を満たすものは全て文字である。

これは、句読点や括弧、記号なども文字とみなすこと、また字形が同じ文字 (例えばひらがなの「ぺ」とカタカナの「ペ」) を区別すること、などを意味する。実際に文中に出現する文字は、大ざっぱな実験によれば約 3000 字種である。

文および文章は以下のように定義する。

定義 **2.2** 文とは、最初の文字または句点「。」の次の文字からその次の句点までのことである。

定義 **2.3** 文章とは、順序付けられた文の集合である。

次に、言語モデルの定式化を行う。まず日本語文字系列のマルコフ性を仮定する。ここでいうマルコフ性とは、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。

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、

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

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

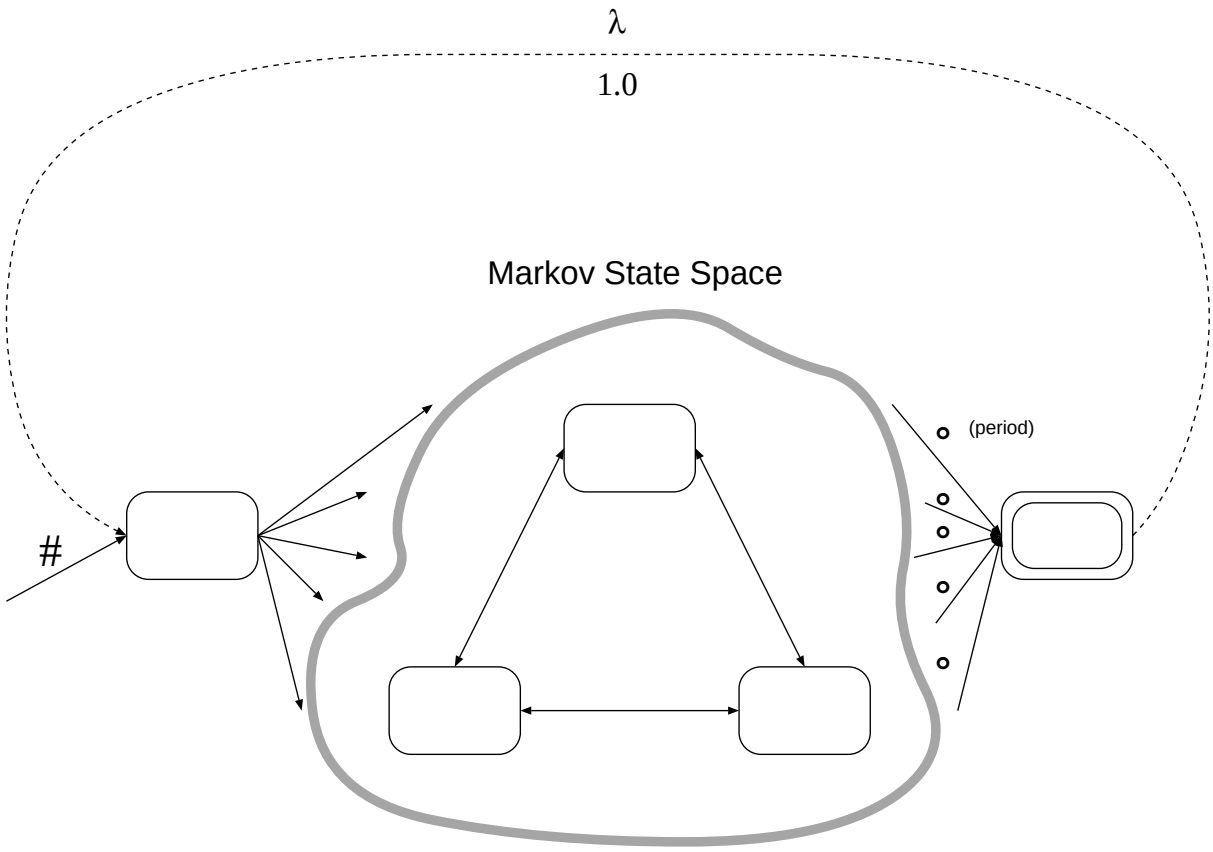

図 2.1: 文の生成モデル。破線はヌル遷移 (出力のない状態遷移) を表す

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 [29]。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に実現されることが目的であり、通常の OCR ならもっと計算資源を利用して認識精度の向上を目指すべきであろう。

かな文字については、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 [23]。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 [24][25]。

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)

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[32] 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 [33] は、音節 (「あ」「か」など単独で発声できる最小単位) 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。そこで本研究では、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

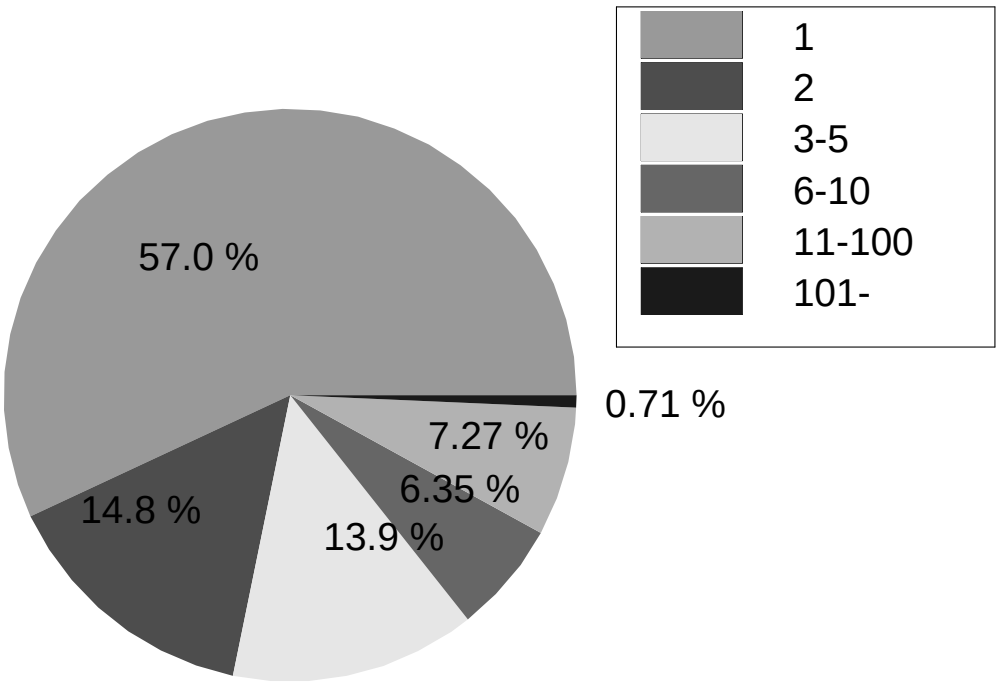

図 2.2: trigram の出現回数

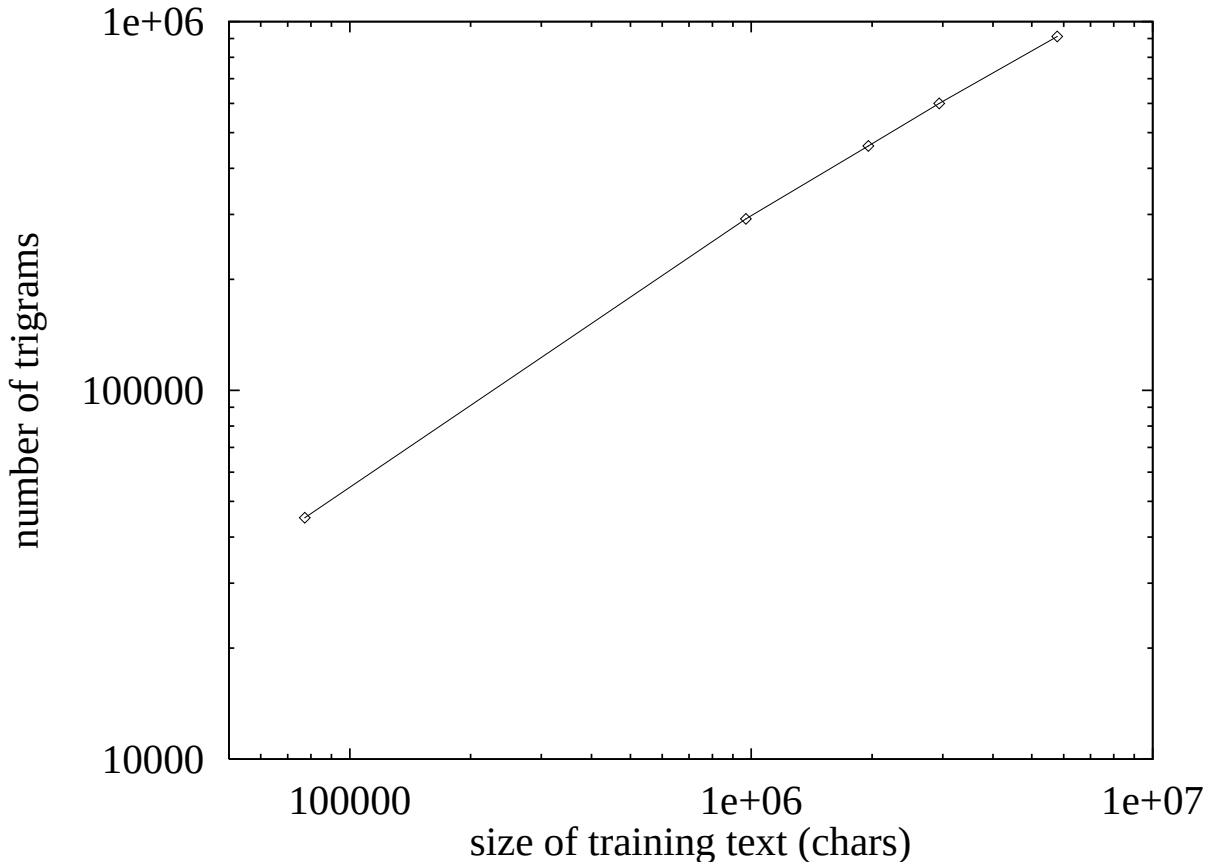

図 2.3: 訓練テキストの量による異なり trigram 数の増加傾向

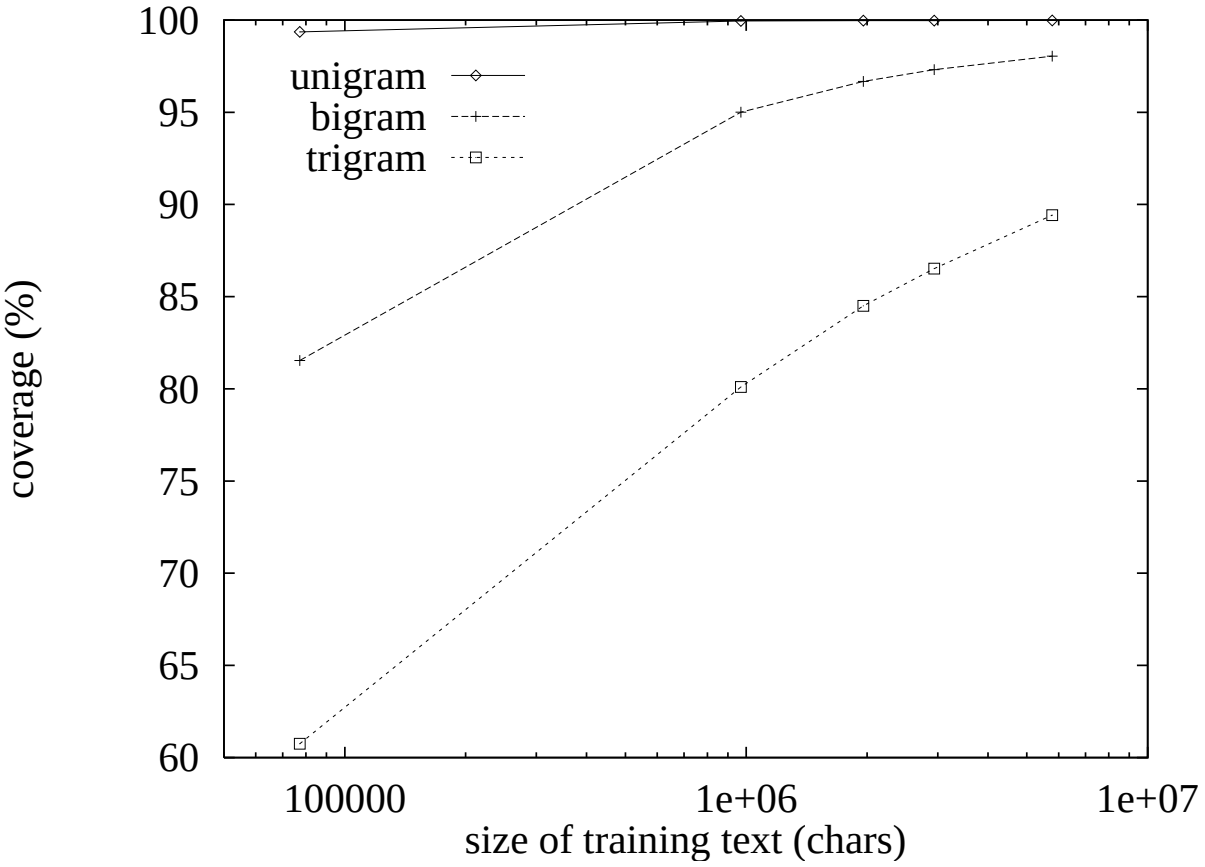

図 2.4: n-gram の被覆率

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

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。

2.4　trigram モデルのスムージング

$\mathcal{A}$ を情報源記号の集合 (すなわち本論文では 2.2節で定義した日本語文字セット) 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 [7]、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 [7]。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 [7]。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[26] [27] 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 [35]、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 [33]。

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 [36] 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削除補間法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。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 のアルゴリズムが知られている。Baum[10] 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。

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

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、

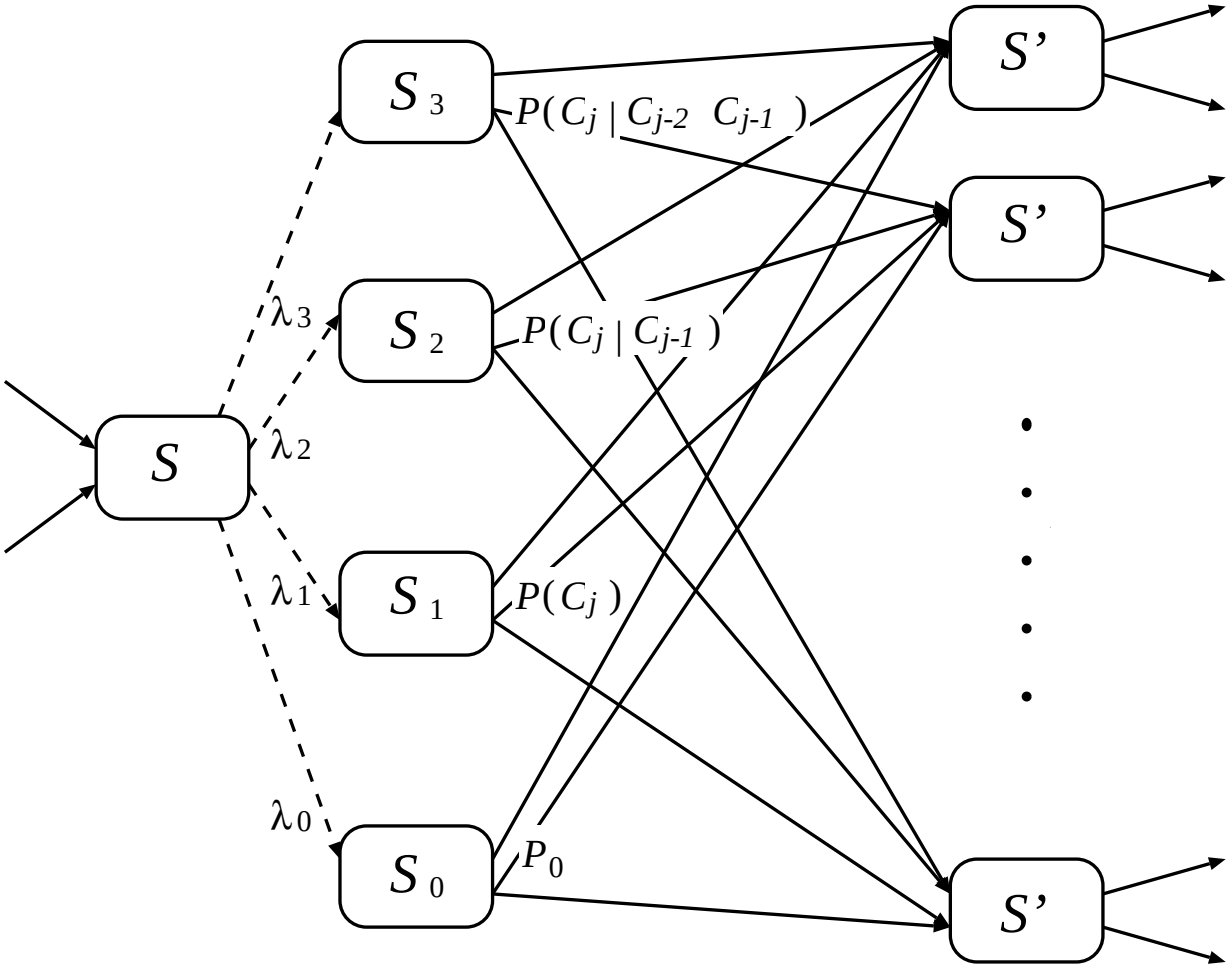

図 2.5: trigram モデルを bigram, unigram, zerogram によって内挿した隠れマルコフモデル

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$\boldsymbol{C} = C_1 C_2 \cdots C_k$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 確率を推定しなければならない。これには、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 2.2節で定義した文とし、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。

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

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 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。

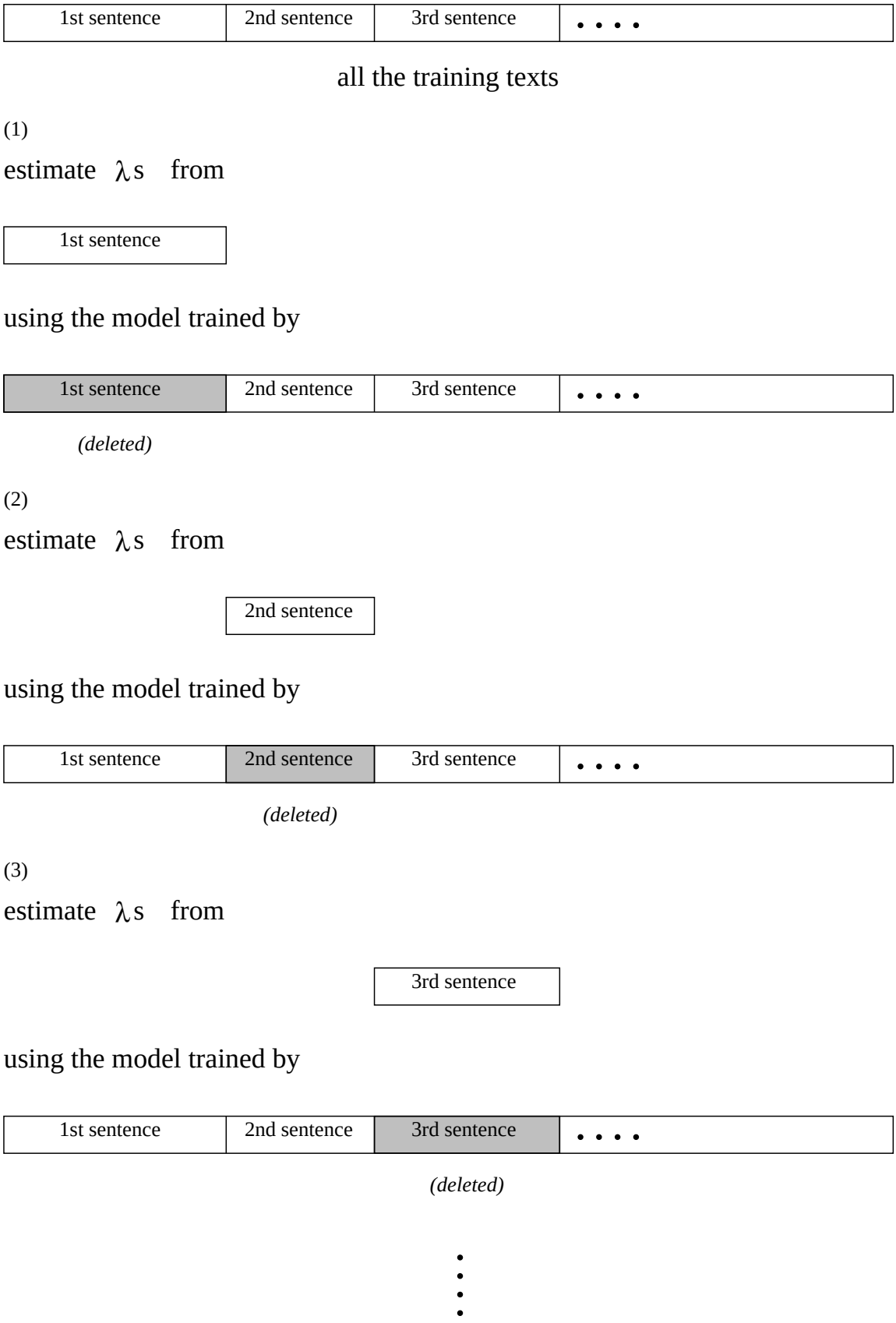

図 2.6: 削除補間法の処理の概要

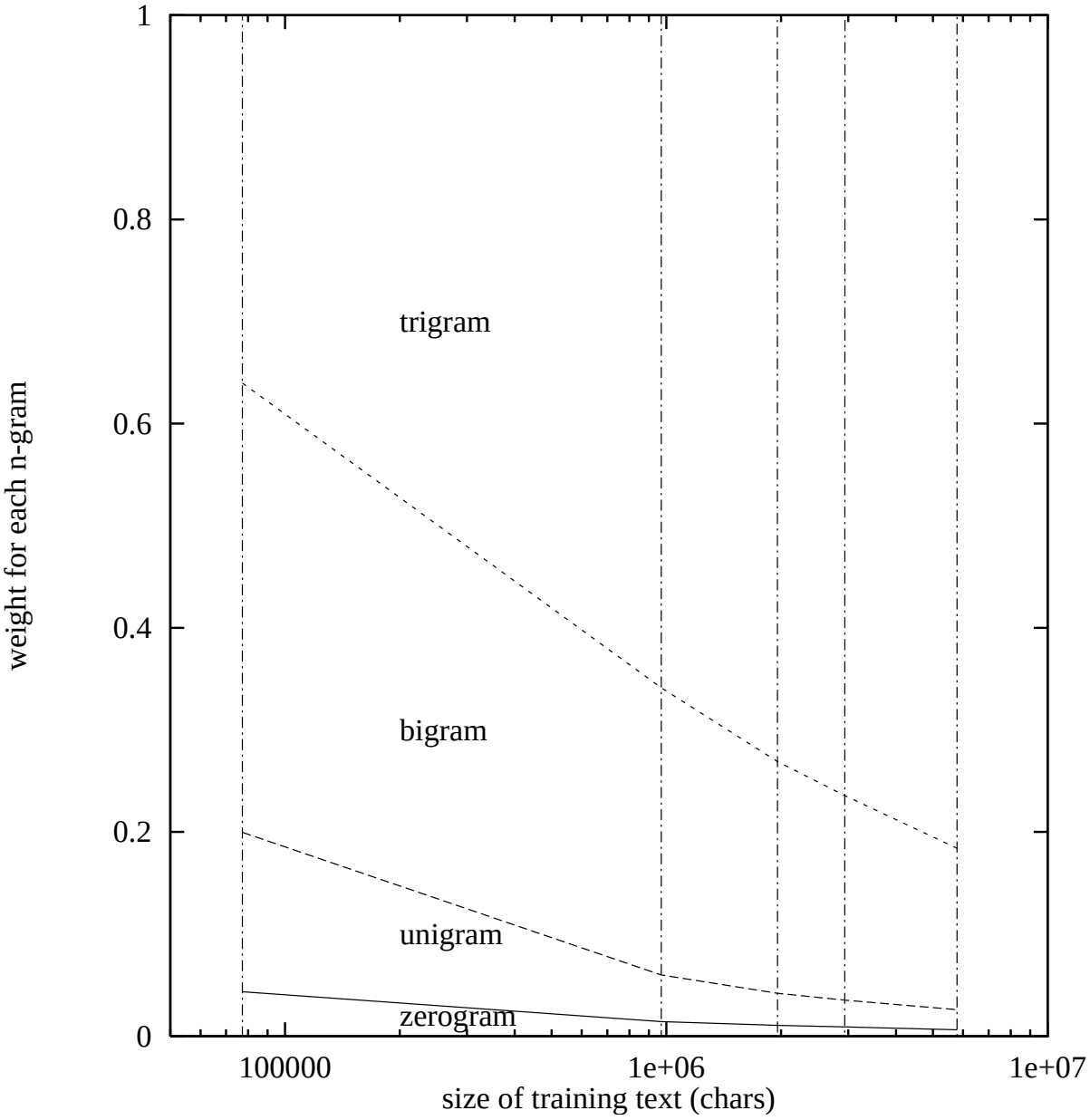

図 2.7: 訓練テキストの量と削除補間法による重み係数の関係

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 [7]、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$\boldsymbol{C} = C_1C_2\cdots C_k$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。

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

本論文で扱うスムージングされた 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 文字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 (タスク) 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。)

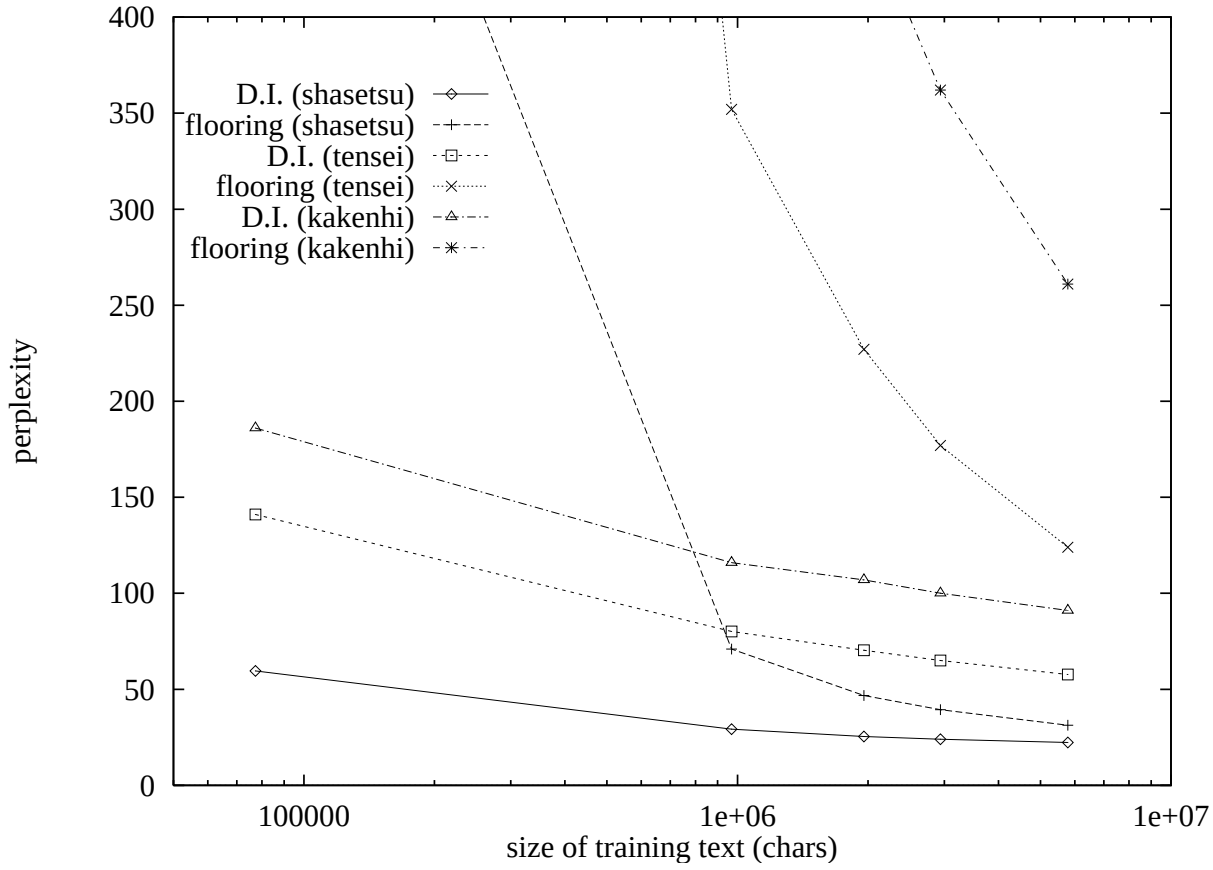

図 2.8: 「社説」モデルのパープレキシティ

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 モデルとも呼ばれる。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 [7]。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訓練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。このことに対する改善策は、第 5章において詳しく検討する。

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。

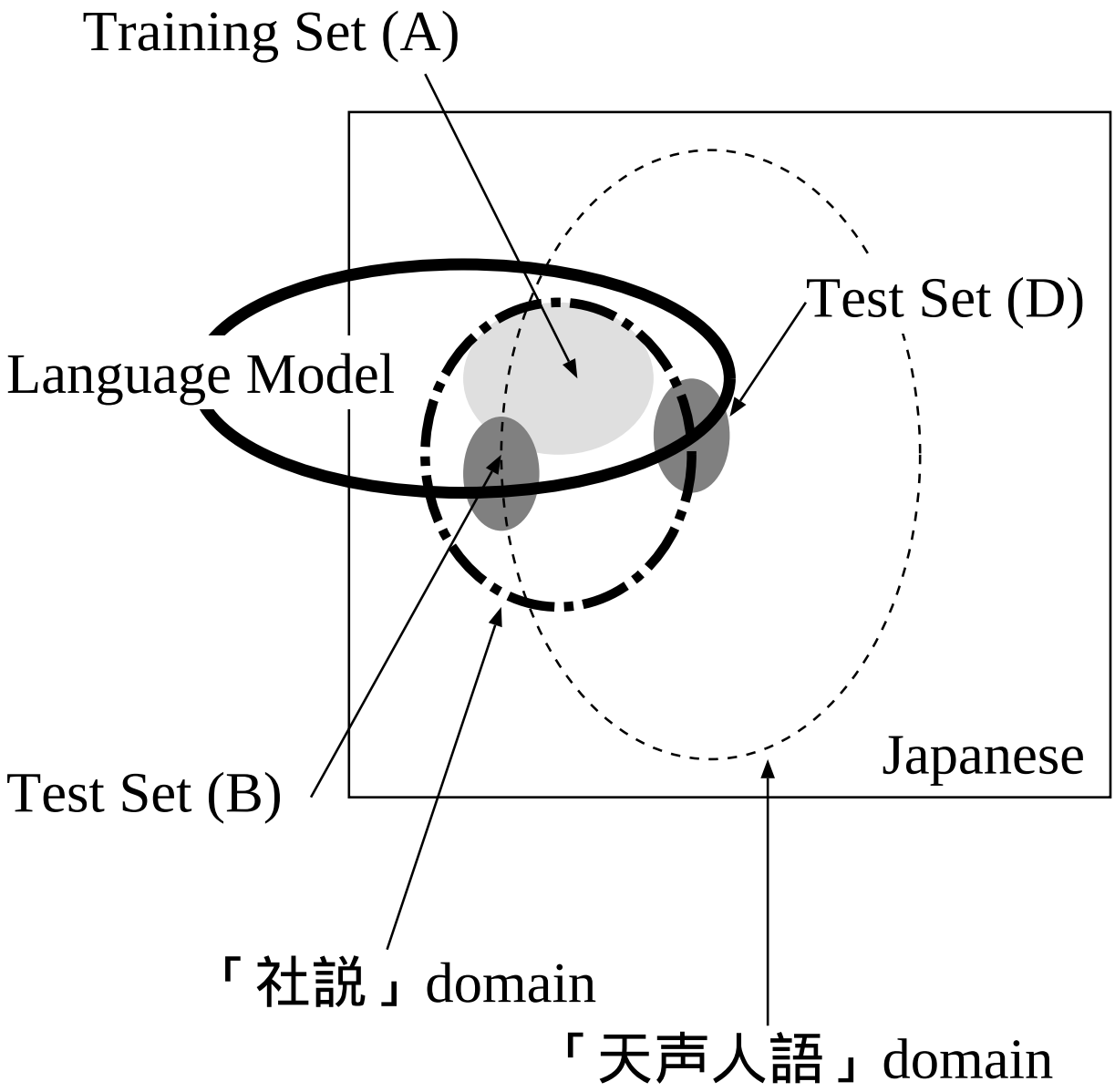

図 2.9: タスク間の関係と言語モデル

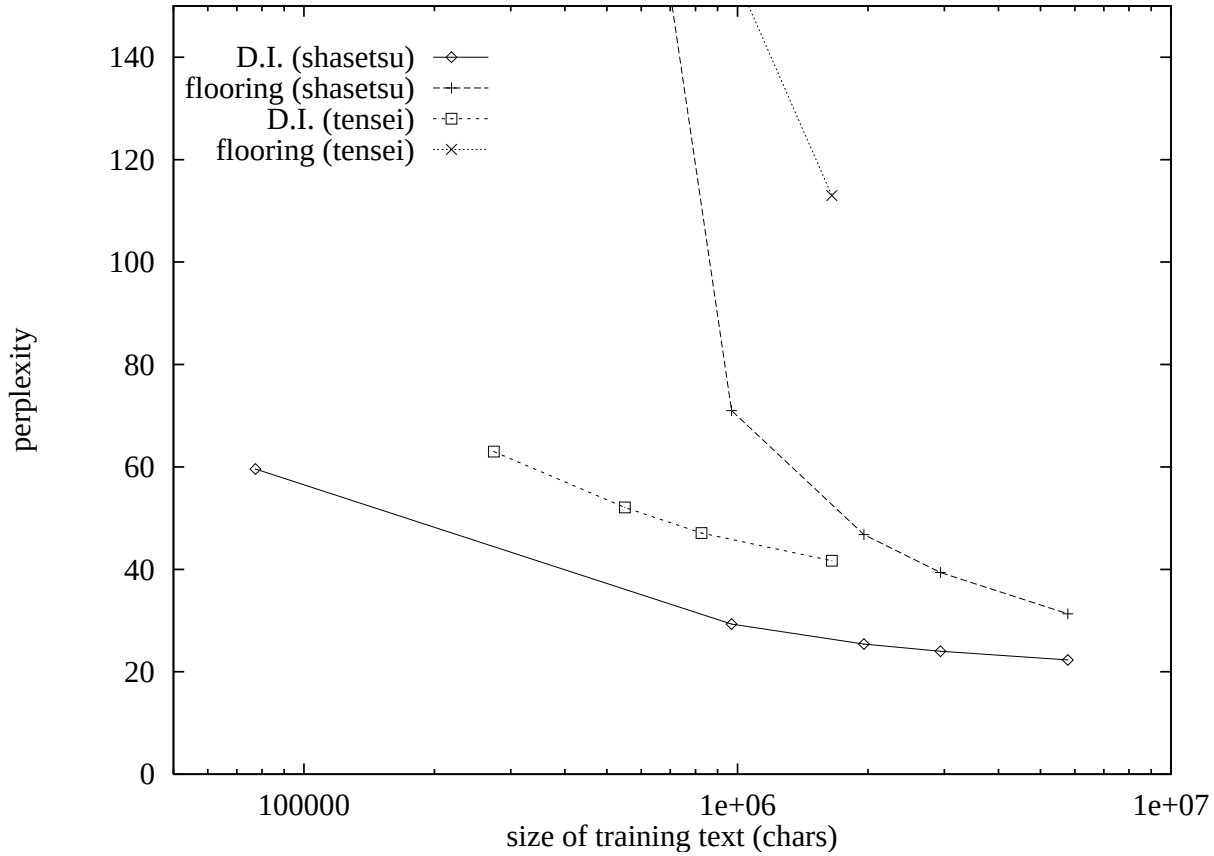

図 2.10: 「社説」モデルと 「天声人語」モデルのパープレキシティによる比較

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[32] 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。他にも、trigram 辞書の削減法や、単語 n-gram への拡張などが考えられるが、これらは将来の課題とする。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、第 5章で述べる。

第 3 章

n-gram モデルを用いた文字認識後処理の効果

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系においては、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。前者は第 2章で論じた言語の確率モデルに他ならない。本章では、まず一般の文字認識系に関して、後者の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章の日本語 n-gram モデルによる文字認識後処理部を付加する手法を述べる。さらに、n-gram モデルの場合の誤り確率最小化を非常に高速に行うアルゴリズムを紹介し、処理速度の観点からの n-gram モデルを採用することの優位性を主張する。最終的にはこれらに基づく文字認識後処理系を具体化するとともに、実験による性能評価

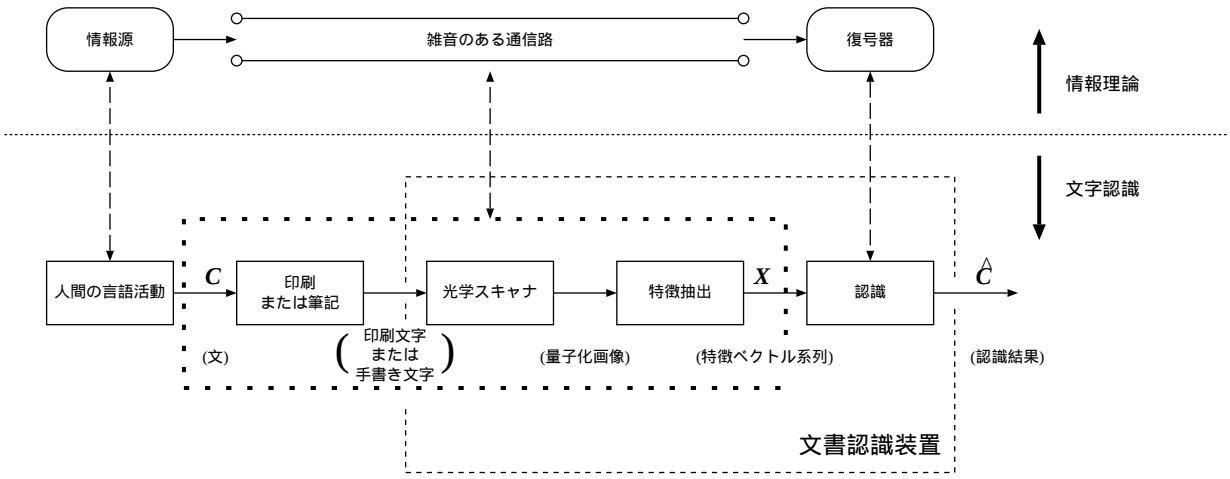

図 3.1: 情報理論と文字認識の対応関係。[7] を参考にした

を行う。

3.1　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 3.1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[13] 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 $P(\boldsymbol{C}|\boldsymbol{X})$ 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 [37]。すなわち、

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{X}) \tag{3.1}$$

これは最大事後確率推定法と呼ばれるものである。

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 (3.1) の右辺は次式で置き換えられる。

$$\underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{X}) = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, p(\boldsymbol{X}|\boldsymbol{C}) \tag{3.2}$$

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 (3.2) は次のようになる。

$$\underset{\boldsymbol{C}}{\operatorname{argmax}}\, p(\boldsymbol{X}|\boldsymbol{C}) = \underset{\boldsymbol{C}}{\operatorname{argmax}} \prod_{i=1}^{k} p(\boldsymbol{x}_i|C_i) \tag{3.3}$$

上式の最尤推定は、各時点での文字 C_i

$$\hat{C}_i = \underset{C_i}{\operatorname{argmax}}\, p(\boldsymbol{x}_i|C_i) \tag{3.4}$$

となる。式 (3.4) 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。

事後確率最大基準の枠組みにおいて認識の高精度化を図るとするならば、改良の余地があるのは次の 2 か所である。第 1 には、通信路そのものの改良である。図 3.1 において、

通信路を構成するのは印刷 (筆記) の過程、スキャニングの過程、特徴抽出の過程である。このうち印刷 (筆記) の過程の存在は本質的なものであって、この部分を改良することは無意味である。スキャニングの過程は文字認識では前処理と呼ばれる画像処理の過程を含む。ただ、文字認識における課題としては、むしろ次の特徴抽出の技術が極めて重要だと考えられており、特徴抽出に関してはこれまでに多くの研究がなされてきた [38][39]。第 2 は復号器、すなわち識別を行う部分の改良である。これにはテンプレートの作成、識別関数 (または距離尺度、類似度) の設計といった問題が含まれる。

これまでの文字認識に関する研究では、上記の 2 点に重点が置かれてきた。その結果、OCR は実用レベルまで進歩するに至ったが、序論でも述べた通り、さらに知的な文字認識を指向するのならば、自然言語の持つ性質を認識機構に反映させなければならない。情報理論と対応させて考えれば、これは自然言語の持つ制約を利用して誤り訂正を行うことである。そこで、以下では自然言語の性質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 (3.1) における事後確率 $P(\boldsymbol{C}|\boldsymbol{X})$ は次のようになる [37]。

$$P(\boldsymbol{C}|\boldsymbol{X}) = \frac{P(\boldsymbol{C})p(\boldsymbol{X}|\boldsymbol{C})}{p(\boldsymbol{X})} \tag{3.5}$$

この関係は **Bayes** の定理と呼ばれる。式 (3.5) の分母は$\boldsymbol{C}$に無関係な定数であるから、式 (3.1) は次式のようになる。

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{X}) \tag{3.6}$$

$$= \underset{\boldsymbol{C}}{\operatorname{argmax}}\, \frac{P(\boldsymbol{C})p(\boldsymbol{X}|\boldsymbol{C})}{p(\boldsymbol{X})} \tag{3.7}$$

$$= \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C})p(\boldsymbol{X}|\boldsymbol{C}) \tag{3.8}$$

$$= \underset{\boldsymbol{C}}{\operatorname{argmax}}\{\log P(\boldsymbol{C}) + \log p(\boldsymbol{X}|\boldsymbol{C})\} \tag{3.9}$$

式 (3.3) 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、

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\{\log P(\boldsymbol{C}) + \sum_{i=1}^{k} \log p(\boldsymbol{x}_i|C_i)\} \tag{3.10}$$

となる。式 (3.10) 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の確率モデルに相当する。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 3.1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。

3.2　尤度関数

3.2.1　パターンの生起確率密度関数の推定

$p(\boldsymbol{x}|C)$ 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 [40][41][42][43]。

ただ、式 (3.10) 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 $N(\boldsymbol{\mu}_C, \boldsymbol{\Sigma}_C)$ 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、$\boldsymbol{\Sigma}_C$は共分散行列である。仮定から、

$$p(\boldsymbol{x}|C) = \frac{1}{(2\pi)^{d/2}\sqrt{|\boldsymbol{\Sigma}_C|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_C)'\boldsymbol{\Sigma}_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C)\right] \tag{3.11}$$

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、

$$\begin{aligned}\log p(\boldsymbol{x}|C) &= \log\left\{\frac{1}{(2\pi)^{d/2}\sqrt{|\boldsymbol{\Sigma}_C|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_C)'\boldsymbol{\Sigma}_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C)\right]\right\} \\ &= -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_C| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_C)'\boldsymbol{\Sigma}_C^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_C)\end{aligned} \tag{3.12}$$

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。$\boldsymbol{\Sigma}_C$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 [44], 複合類似度法 [1] 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 [40] 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、

$$\boldsymbol{\Sigma}_C \approx \boldsymbol{\Sigma} \tag{3.13}$$

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、

$$\boldsymbol{\Sigma} \approx \sigma^2 \boldsymbol{I} \tag{3.14}$$

であるとする。ここで$\sigma > 0$、また$\boldsymbol{I}$は単位ベクトルを表わす。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 (3.13),(3.14) を用いると、式 (3.12) は次のように書き直せる。

$$\log p(\boldsymbol{x}|C) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{\mu}_C\|^2 \tag{3.15}$$

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 (3.10) 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。

$$-A\|\boldsymbol{x} - \boldsymbol{\mu}_C\|^2 \tag{3.16}$$

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。

3.2.2 重み付け

式 (3.10) の右辺第 1 項 $P(\boldsymbol{C})$ を言語的尤度と呼ぶ。既に述べた通り、言語的尤度は第 2 章の言語の確率モデルにより近似する。右辺第 2 項は式 (3.16) により近似される。これらを用いることで式 (3.10) を最大にする系列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。

$$L(\boldsymbol{C}|\boldsymbol{X}) \stackrel{\text{def}}{=} w \log \hat{P}(\boldsymbol{C}) + (1-w)A \sum_{i=1}^{k} (-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{C_i}\|^2) \tag{3.17}$$

ただし $0 \leq w \leq 1$ である。この尤度関数を用いて、式 (3.10) は次のように推定される。

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}} \{\log P(\boldsymbol{C}) + \sum_{i=1}^{k} \log p(\boldsymbol{x}_i|C_i)\} \tag{3.18}$$

$$\approx \underset{\boldsymbol{C}}{\operatorname{argmax}}\, L(\boldsymbol{C}|\boldsymbol{X}) \tag{3.19}$$

$$= \underset{\boldsymbol{C}}{\operatorname{argmax}} \{w \log \hat{P}(\boldsymbol{C}) + (1-w)A \sum_{i=1}^{k} (-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{C_i}\|^2)\} \tag{3.20}$$

式 (3.20) の中括弧内を定数倍しても$\hat{\boldsymbol{C}}$は変わらない。したがって、式 (3.20) 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。

3.3　最適化アルゴリズム

式 (2.12) を用いると、尤度関数は

$$L(\boldsymbol{C}|\boldsymbol{X}) = L(C_1 C_2 \cdots C_k|\boldsymbol{x}_1 \boldsymbol{x}_2 \cdots \boldsymbol{x}_k) \tag{3.21}$$

$$= w \sum_{i=1}^{k} \log \hat{P}(C_i|C_{i-2}C_{i-1}) + (1-w)A \sum_{i=1}^{k} (-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{C_i}\|^2) \tag{3.22}$$

$$= w \log \hat{P}(C_k|C_{k-2}C_{k-1}) + (1-w)A(-\|\boldsymbol{x}_k - \boldsymbol{\mu}_{C_k}\|^2) + L(C_1 C_2 \cdots C_{k-1}|\boldsymbol{x}_1 \boldsymbol{x}_2 \cdots \boldsymbol{x}_{k-1}) \tag{3.23}$$

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Cは、**Viterbi** アルゴリズム [7] と呼ばれる動的計画法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。

$$G_1(C_0 C_1) \stackrel{\text{def}}{=} w \log \hat{P}(C_1|C_{-1}C_0) + (1-w)A(-\|\boldsymbol{x}_1 - \boldsymbol{\mu}_{C_1}\|^2) \tag{3.24}$$

$$G_i(C_{i-1}C_i) \stackrel{\text{def}}{=} \max_{C_{i-2}} \{w \log \hat{P}(C_i|C_{i-2}C_{i-1}) + (1-w)A(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_{C_i}\|^2) + G_{i-1}(C_{i-2}C_{i-1})\} \qquad (i \geq 2) \tag{3.25}$$

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、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。定義 2.1 が定める文字集合を $\mathcal{C}$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となり、コストは非常に小さい。

また、通常個別文字認識の精度は比較的高く、大部分の文字は明らかに 1 位候補が正解であると考えられる場合が多い。そこでさらに計算コストを低減するため、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後処理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 [45] 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。

$$r = \frac{\|\boldsymbol{x} - \boldsymbol{\mu}_{C^2}\|^2}{\|\boldsymbol{x} - \boldsymbol{\mu}_{C^1}\|^2} \tag{3.26}$$

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、

$$R = \begin{cases} \{C^1\} & \text{if } r \geq r_{\mathrm{th}} \\ \{C^1, C^2, \cdots, C^{N_c}\} & \text{otherwise} \end{cases} \tag{3.27}$$

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、または曖昧な認識結果と呼ぶ。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 3.2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。

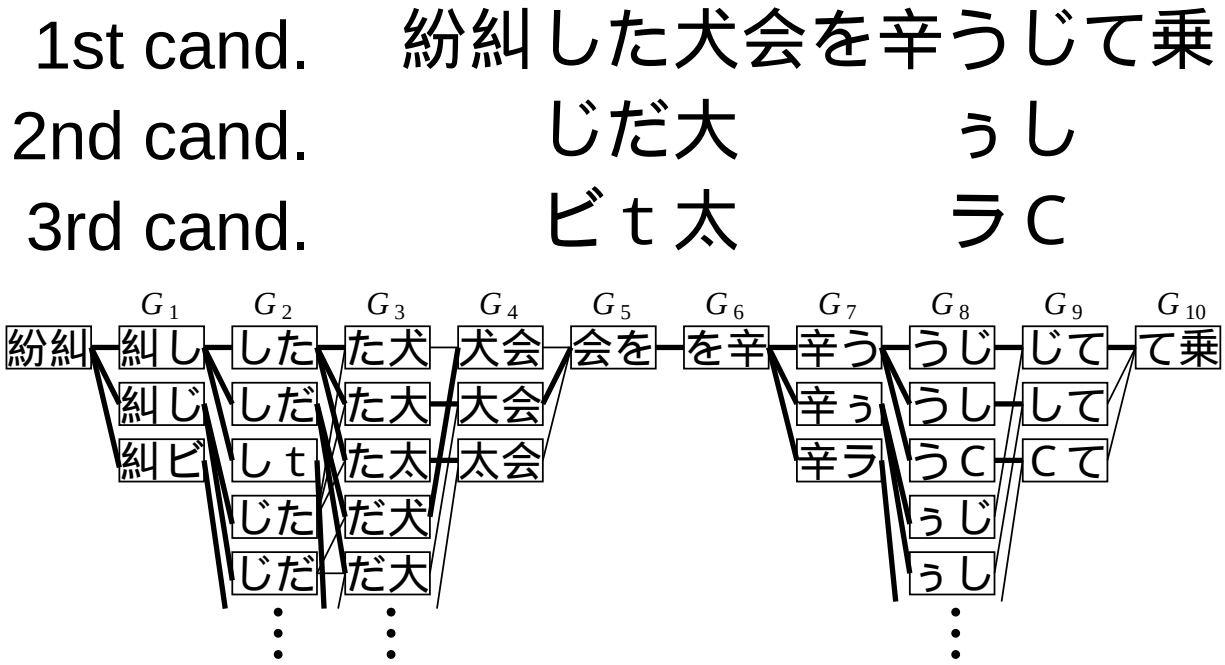

図 3.2: Viterbi アルゴリズムによる最尤系列の導出

3.4　文字認識後処理系の実現

3.4.1　個別文字認識系

前節で述べた最尤系列の導出は、個別文字認識の結果と言語モデルを用いて行われる。処理のタイミングが個別文字認識の後になることから、前節の処理を今後後処理と呼ぶことにする。後処理の有効性を確かめるため、既存の文字認識系を用いた文書の認識実験を行う。

ここで対象として選ばれたのは、認識性能に影響を与える最も重要な処理である特徴抽出部に方向線素特徴量 [38] を用いた文字認識系である。方向線素特徴量の詳細をここで論ずる余裕はなく、文献に譲ることとするが、大略は次の通りである。量子化された文字画像は正規化 ･細線化の処理を施された後、4 方向の向きを持つ短い線の集合に変換される。この短い線を線素と呼ぶ。次に、正方形の文字領域を 8×8 の桝目に区切る。隣接する桝目を 4 つ取ってできる正方形を一つの小領域とする。このような小領域は全体で $7 \times 7 = 49$ 個ある。小領域を重複させながら取るのは、線分の位置ずれによる影響を低減するためである。各々の小領域について、4 方向それぞれの線素の数を数え上げ、それぞれを 1 つの

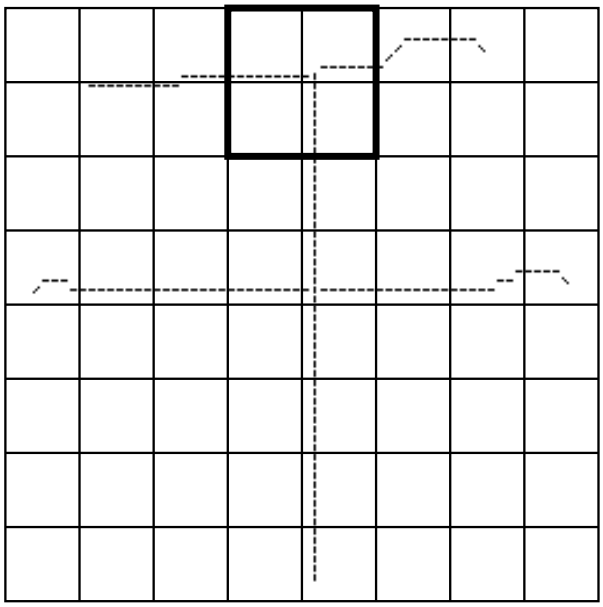

図 3.3: 方向線素特徴量の算出過程 (線素の数え上げ)

特徴とする。図 3.3の太枠で囲まれた小領域の例では、「 | 」が 9 個、「—」が 15 個、「/ 」と 「\ 」がそれぞれ 0 個である。このようにして、文字全体は $49 \times 4 = 196$ の特徴で表わされることになる。この定義からも明らかなように、方向線素特徴量は各次元間に相関があり、その意味では式 (3.14) の仮定は満たされない。しかし、この特徴量は印刷漢字の識別において非常に有効であることがわかっており、識別にユークリッド距離を用いた場合でも全角漢字に対して 99.8%以上の認識率が得られる [38]。ただ、拡大・縮小に不変な特徴量であるため、実際の文書を扱うシステムでは単純小図形や正規化によって大きく変形する文字を考慮する必要がある。本章では簡単のため、正規化による影響は無視している。

ユークリッド距離を算出するのに予め必要な量は各文字の標準パターンだけである。標準パターンの集合を辞書と呼ぶ。サンプルを収集する上での都合で、辞書の要素となる字種は文字集合 JIS X0208-1983 の部分集合で、いわゆる JIS 第 1 水準漢字・非漢字と呼ばれる 3303 字に限定した。本研究では 「認識不能」という出力を許さないため、文書中に上記の 3303 字に含まれない文字が出現した場合は、仮定 2.1 にかかわらず必ず誤認識することになる。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 3.1に示す。

表 3.1: 個別文字認識系の実験条件

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

3.4.2 予備実験

候補文字を制限するためのシステムパラメータ $r_{\rm th}$および N_cを決めるために、予備実験を行った。その結果を図 3.4 および図 3.5に示す。図 3.4は $r_{\rm th}$を決めるための予備実験である。図中、実線は既知文書に対して確定文字の個別認識率を求めたものであり、破線は未確定文字の割合を求めたものである。横軸は $r_{\rm th}$である。この値が大きくなると、確定文字はより正解の文字だけに限定されていくが、あまり大きくすると未確定文字の割合が増加して処理量の点からは好ましくない。この図から、確定文字の認識率が飽和し始める $r_{\rm th} = 1.8$ を選んだ。

図 3.5は N_cを決めるための予備実験である。横軸は N_c, 縦軸は N_c位候補までの累積認識率である。図から、$N_c = 8$ とした。

3.4.3 認識対象

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デザインのものである。

この章では文字切り出しに誤りがないという前提で議論したので、認識対象は図 3.6のような切り出しガイドラインのある専用のフォーマットで出力する。

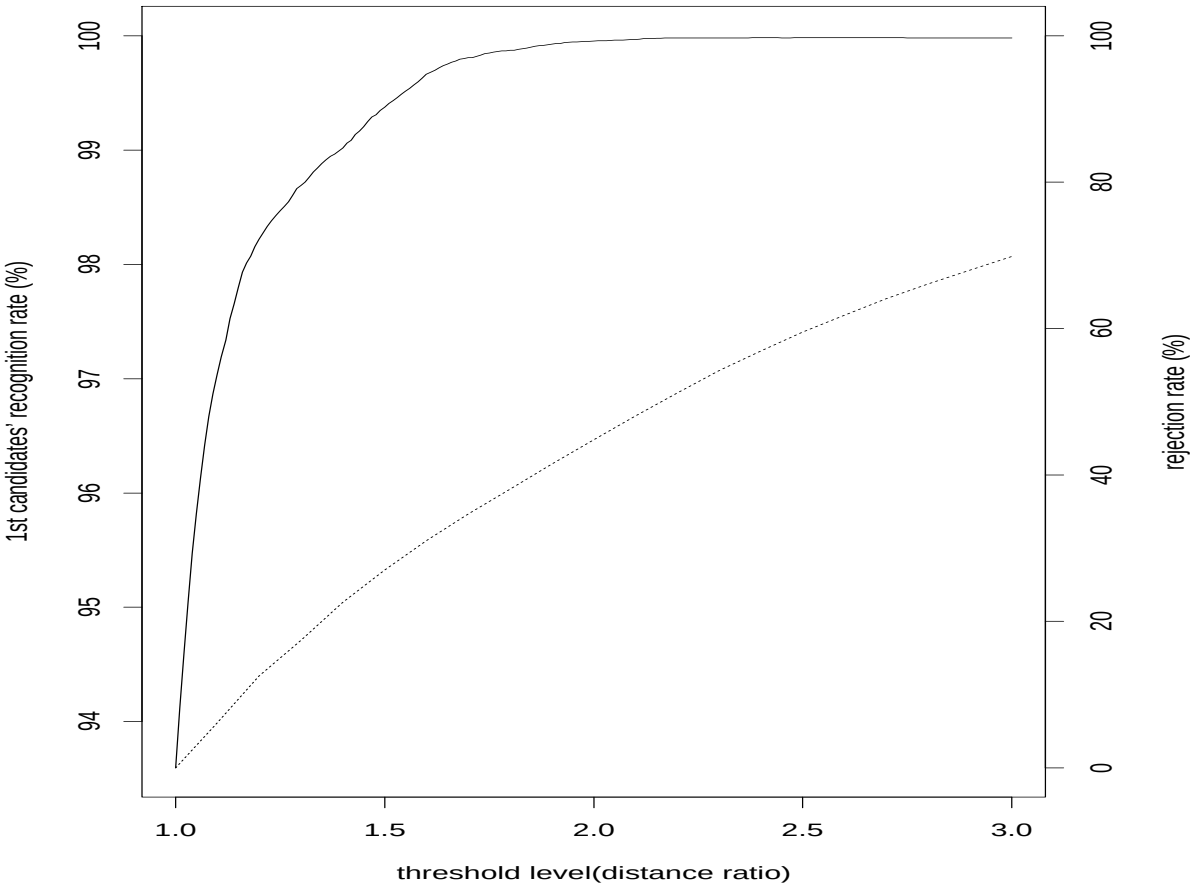

図 3.4: 確定文字の認識率、および未確定率

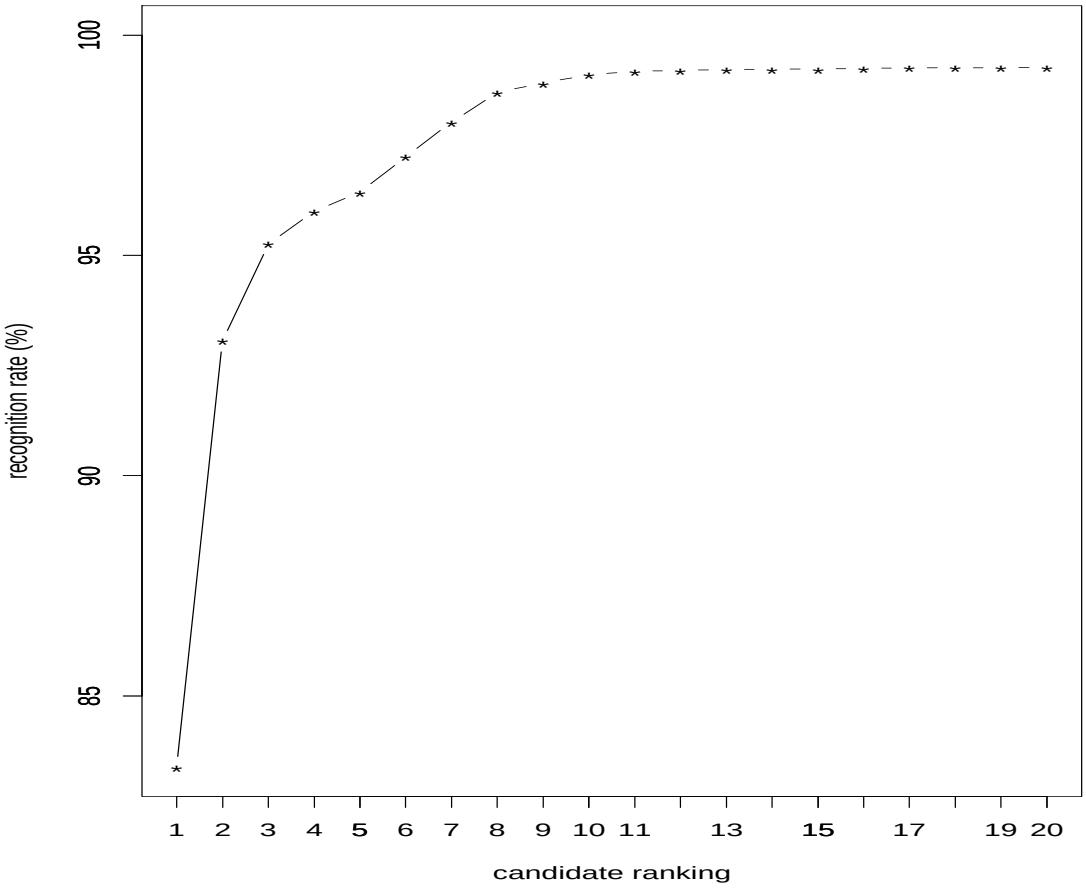

図 3.5: 累積認識率

リクルート事件に対する検察の捜査は大づめを迎え、今週いよいよ核心の政治家ルートに迫る見通しだ。リクルート疑惑の黒白は、間もなく明確になろう。こうした状況下で、国会はいぜん、野党の「中曽根喚問」要求と、これに反対する自民党との対立により、長期空転を続け

図 3.6: ガイドライン付きサンプル (部分)

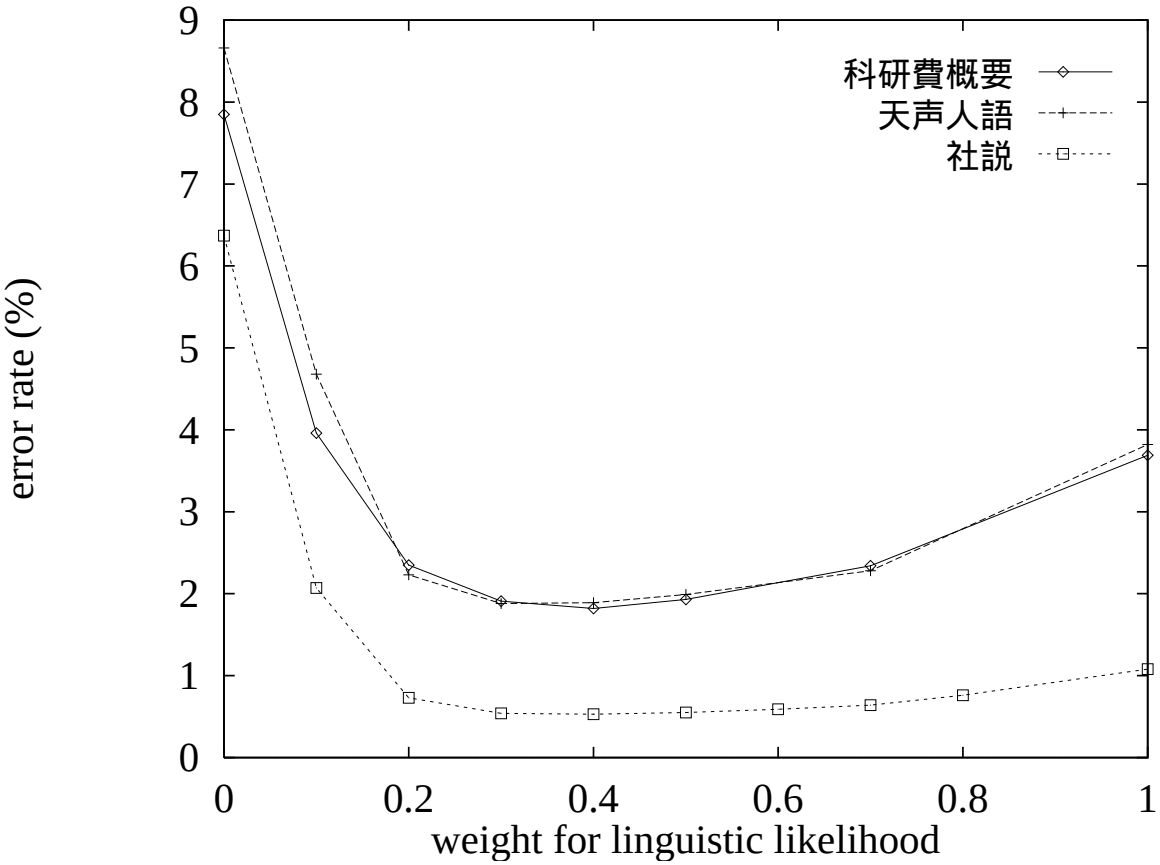

図 3.7: 言語的尤度の変化と誤認識率の関係

3.4.4　認識実験

以上に基づき、日本語文字 n-gram モデルに基づく文字認識系を構築し、認識実験を行った。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) 580 万字である。また、trigram モデルのスムージングには式 (2.9) の削除補間法を用いた。

言語的尤度の重み wを変化させた時の誤認識率を図 3.7に示す。横軸は式 (3.20) における言語的尤度の重み wである。ただし、ここでは $A = \frac{1}{1000}$として表示している。wの定義から、$w = 0$ の軸上の値が言語モデルによる後処理を伴わない個別文字認識の結果と同じである。

どの文書においても、$w = 0.3$ の付近で最も誤認識率が低くなっている。特に認識対象が「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

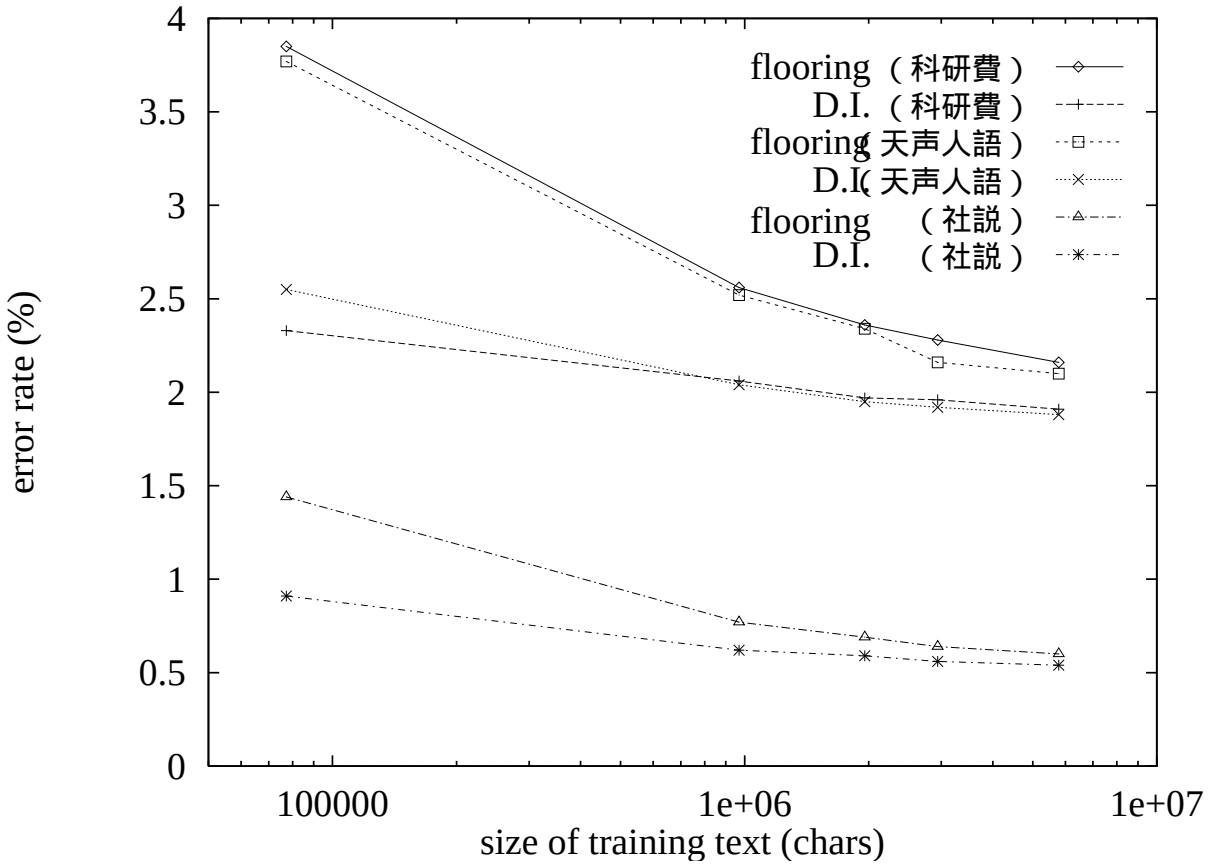

図 3.8: スムージング手法と誤認識率の関係

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後処理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。以降、この章では最適な言語的尤度の重みとして $w = 0.3$ を用いる。

次に、スムージング手法と認識性能の関係を図 3.8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。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。

なお、これらの処理の実行速度は、S-4/2 ワークステーション上で約 1000 文字/ 秒と極めて高速である。

3.5 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、高精度な後処理が実現できることを明らかにした。また 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が、認識実験の上からも再確認できた。

問題点として、次のような点が残っている。まず、式 (3.15) によって $p(\boldsymbol{x}|C)$ を求めることの妥当性である。サンプルが少ない場合にも頑健な、特徴量の分布を考慮した識別関数を用いることで、この部分の近似が改善される可能性がある。印刷文字に関しては特徴量の分布の正規性を仮定することは疑問で、むしろニューラルネットワークのようなノンパラメトリックな識別器を検討すべきかも知れない。

次に、候補外文字の問題が挙げられる。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。図 3.5からもわかるように、N_c を増加させて行くと累積認識率は飽和してしまう。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、次章で述べるように低品質文字を扱う場合に顕著となる。この問題に対する対策は、次章で詳しく述べる。

最後に、訓練テキストと認識対象の言語的性質の違いの問題がある。これは第 2章でも触れた問題である。本章の認識実験で明らかにされたように、言語的性質の相違による影響は削除補間法の導入により小さく押さえることができる。しかし、さらに高精度化を望むならば、言語モデルの認識タスクへの適応化が必要である。これについては第 5章で検討する。

第 4 章

切り出しと認識を同時に行う文字認識方式の提案

第 3章では、個別文字認識系の文字切り出しが誤りなく行われることを前提に、言語の確率モデルを用いた文字認識の高精度化を議論した。このような前提は、個別文字認識の精度を評価する場合にも良く用いられる。

しかし、現実の文書を対象とした文字認識装置を設計する場合にはこのような前提を採用することはできず、文書画像中のどの領域が文字なのかを判断することそれ自体が困難な問題である。

文書認識系は通常、認識に先立って文書構造理解を行うレイアウト解析と呼ばれる処理を実行し、これによって抽出された文字行に対して、さらに個々の文字に分割する文字切り出しの処理を行う。

レイアウト解析は、非常に単純な構造を持った文書や、初めからレイアウトがわかっている文書に対してはほぼ自明に行えるが、文書構造に対する先見知識なしに一般文書の解析をすることは、それが人間にとってはいともたやすいことであるにもかかわらず未解決の問題である。

一方、文字行からの文字切り出しについてもレイアウト解析と同様に人間にとっては簡単であるが、この問題は後に示すように本質的な曖昧さを含んでおり、文字認識の未解決問題の一つである。本章では、この文字切り出しの問題に対し、第 3章で認識結果の曖昧さを解消するのに大きな効果があった日本語文字 trigram モデルを利用することを提案する。ここでは第 3章と同様に事後確率最大基準を基礎とし、切り出しまで含めた最終的な認識結果の誤りが最も少なくなるような定式化を行う。この結果、通常の切り出しの後に認識を行う文字認識方式では解消できなかった認識誤りがどの程度解消できるかを実験により

確かめる。

また、この方法ではあまり効果の上がらない低品質文書の認識問題について、本章では通常の Viterbi アルゴリズムに、候補外文字の問題を見かけ上回避するビームサーチ法を導入し、精度の改善を図る。

ところでレイアウト解析は近年盛んに研究されているパターン認識・理解の一分野であり、本論文で取り扱うべき範囲を超える。よって以降ではレイアウトが自明である文書を対象とする。高精度なレイアウト解析系の出力である行画像を対象としても以降の議論は同様に行える。

4.1　文字切り出し問題

印刷文字について言えば、組版の過程で各文字間にはほとんどの場合多少の空白が設けられる。余白なしに敷きつめられた文字を読むのは人間にとっても非常に苦痛だからである。したがって、一つの黒連結図形を 1 文字とみなすことによって、多くの場合には切り出しはうまく行く。一つの連結図形はラベリング [46] の処理によって抽出できるが、日本語の場合、文字はほぼ正方形の中に収まるようにフォントがデザインされているため、ラベリングの代わりにヒストグラムを計算するだけで済む場合が多い。

しかし、連結図形のラベリングやヒストグラムだけでは元々の文字と行内の文字イメージを 1 対 1 に完全に対応付けることはできない。これは、文字切り出しが本質的に持つ次のような問題が存在するからである。

4.1.1　分離文字

複数の連結図形によって構成される文字を広義の分離文字と呼ぶ。通常問題となるのは行方向に切れ目がある文字 (横書き時の「加」「ル」、縦書き時の「言」など) のみであるので、この意味で、射影の切れ目が行方向にある文字を狭義の分離文字と呼ぶ。本章では連結図形の抽出のためにヒストグラムを用いているので、以後分離文字とはここでいう狭義の分離文字を指すこととする。

分離文字の存在のために、文字と画像の対応付けは曖昧なものとなる。手書き文字においては、この問題を定ピッチ帳票を用いることで物理的に解消することがある。これは郵便番号のように 1 文字ごとに枠内に記入させるための専用の用紙を用いる方法である。印刷文字の場合にはこれは不可能であるが、日本語の文書は文字ピッチがほぼ一定であるという特徴を持っている。そこで、ピッチの情報を利用することで切り出しの曖昧性をなく

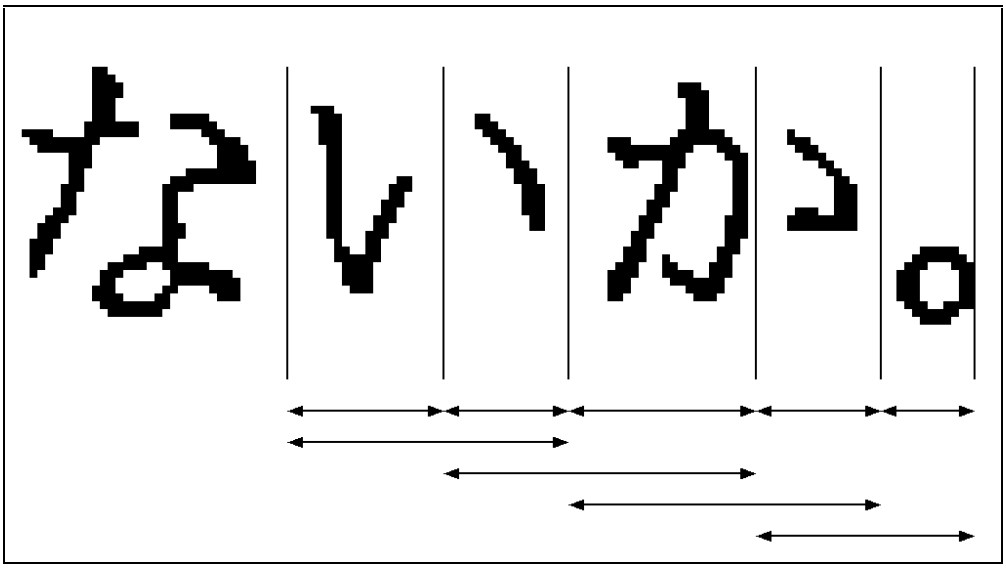

図 4.1: 画像情報だけでは正しく切り出しが行えない例

そうとする手法が古くから使われてきた。

しかし実際の文書、特に横書きの文書中には、英数字や括弧などの記号類が含まれる。組版の際、これらの文字は漢字や仮名のようには扱われず、前後に若干の空白が挿入されたり、漢字 1 字分のスペースを与えられずに詰められたりする。このように局所的にピッチ一定の仮定が破れた文書については、ピッチ情報だけでは文字切り出しを完全に行えないことが近年の研究により明らかになってきた。

そこで、考え得る全ての文字切り出しの可能性を仮定して実際に認識を行い、最も高い評価の認識結果をもたらすような切り出し位置を結果とする手法が提案されている [47]。切り出しを誤るとどの文字にも似ないという仮定が成り立つ範囲では、この手法によりピッチ情報だけでは判断できなかった場合にも高精度な切り出しが可能である。

しかし、実際にはこの手法をもってしても切り出しの曖昧性は残る。その例を図 4.1に示す。この例では切り出しの可能性が矢印で示すように多く存在する。これを 「ないか。」であると判断するのは我々にはまことに簡単である。しかし、認識の可能性としては 「ないカ ↘o 」というものも考えられる。画像から得た情報、すなわち文字の形やピッチからだけでは、 「ないカ ↘o 」が不正解であると結論付けるに足る根拠はない。このような曖昧性は、分離文字が連続して出現すると容易に発生してしまう。

分離文字の問題を難しくしている原因として、同一の文字が常に同じように分離するとは限らないことがある。分離するかどうかはフォントのデザインと印字品質による。した

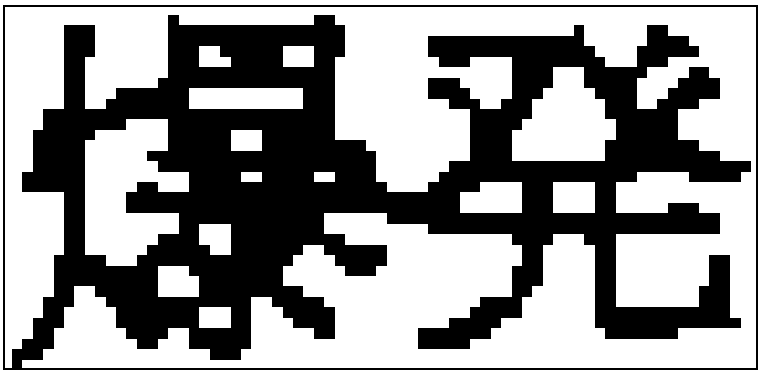

図 4.2: 接触文字の例

がって、認識対象がどのようなフォントを用いて印刷されているかに関する事前の知識がない場合には、分離情報 [46](どの字がどのように分離しているかに関する事前情報) を用いて曖昧性を解消しようとすることには無理がある。

4.1.2　接触文字

分離文字とは文字と単一の連結図形が 1 対多に対応するものであったが、これが多対 1 に対応するのが接触文字である。一例を図 4.2に示す。接触の起こる原因としては、にじみによる黒部分の広がり、ノイズの混入、ゴミの付着などがある。接触が起こっていると考えられる場合には、何らかの方法により分離すべき箇所を特定しなければならない。

定ピッチ文書であれば、ピッチ情報から分割すべき位置を比較的簡単に知ることができるが、不定ピッチ文書の場合には簡単ではない。

図 4.2の例は接触文字への対応の観点からすると比較的簡単な部類に属するもので、ピッチ情報とヒストグラムを利用することで正しい切り出し位置を推定できる。しかし、接触文字の中には接触した位置が非常に推定しづらいものが多く、考えられる可能性を全て網羅しようとすると探索が爆発してしまいかねない。

4.1.3　日本語 trigram モデルの導入

図 4.1のような例を我々がたやすく読むことができるのは、序論で述べたのと同様に、単なる文字の集まりではなく意味を持つ言語として読んでいるからだと考えられる。したがって、より知的な文字切り出しのを実現するためには、第 3章でそうしたのと同様に、言語的制約を導入することが効果的であろう。そこで、本論文では第 3章で述べた日本語 trigram

モデルを再び導入し、文字切り出しの高精度化に与える効果を検証する。

4.2　切り出し関数

文字切り出しとは、文書画像から抽出された文字行画像$\boldsymbol{i}$から文字画像の系列への写像であると定義される。文字行画像$\boldsymbol{i}$を入力とした場合、文字認識とは$\boldsymbol{i}$から文字系列$C = C_1C_2\cdots C_k$への写像である。文字認識の誤りをできるだけ小さくしたいならば、事後確率を最大にするような次の系列$\hat{C}$を求めれば良い。

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, P(\boldsymbol{C}|\boldsymbol{i}) \tag{4.1}$$

第 3章では、切り出しが正しく行われる、つまり文字と文字画像が 1 対 1 に対応することを前提に議論した。この場合には、文字行画像$\boldsymbol{i}$から特徴ベクトル系列$\boldsymbol{X}$への対応は一意に決まるので、式 (4.1) の計算を式 (3.10) の計算に置き換えられる。

しかし、ここではそのような前提を置くことはできない。行から文字への対応付けが複数通りあり得る上、文字画像の系列長が必ずしも元の文字系列長と一致しない。そこで、次のような切り出し関数 fを定義する。この関数は、文字行画像$\boldsymbol{i}$を個々の文字に切り出し、それぞれの特徴を抽出してベクトル系列$\boldsymbol{X} = \boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_l$とするような任意の変関数である。すなわち、

$$\boldsymbol{X} = f(\boldsymbol{i}) \tag{4.2}$$

この関数が決まると、$\boldsymbol{i}$から$\boldsymbol{X}$が求まり、式 (4.1) を使える形になる。これらを用いて、次の式により求まる系列Cを認識結果とする。

$$\underset{\boldsymbol{C},f}{\operatorname{argmax}}\, P(\boldsymbol{C}|f(\boldsymbol{i})) \tag{4.3}$$

定性的には、全ての切り出し可能性について実際に事後確率を求め、それらの中で最も確率の高いものを選ぶことを意味している。切り出し結果は全ての切り出し可能性に関する認識結果を総合して得られる。したがって、言語モデルによる誤り訂正のタイミングは切り出しと同時であって、もはや後処理と呼ぶにはふさわしくない。

式 (4.3) は Bayes の定理によって $P(\boldsymbol{C})$ と $p(f(\boldsymbol{i})|\boldsymbol{C})$ から計算できる。だが、$p(f(\boldsymbol{i})|\boldsymbol{C})$ の値を知ることは難しい。しかし、切り出し関数 fを適切に選ぶことができれば、特徴ベクトル系列$\boldsymbol{X} = f(\boldsymbol{i})$ は式 (3.9) 中の$\boldsymbol{X}$と等しくなる。そこで、ここでは $p(f(\boldsymbol{i})|\boldsymbol{C})$ を次式により推定することとする。

$$p(f(\boldsymbol{i})|\boldsymbol{C}) = p(\boldsymbol{X}|\boldsymbol{C}) = \sum_{i=1}^{l} \log p(\boldsymbol{x}_i|C_i) \tag{4.4}$$

以降の最適化処理は、第 3章と同様に行う。

4.3　切り出し ・認識アルゴリズム

以下では光学的に取り込まれた行画像から切り出しと認識の同時処理を行う具体的なアルゴリズムを示す。

4.3.1　平均文字サイズ算出

文字行として取り込まれたイメージに対し、以下の処理を行う。まずイメージの各列を縦方向にサーチしていったときに黒画素の全くない領域を区切りとし、各区間ごとに行ヒストグラムを計算する。これにより黒画素の塊が検出できるので、これをもとに平均文字サイズ (幅 ・高さ) を求める。すなわち各黒画素の塊の横方向縦方向の長さを 1 行の高さと比較し極端に違わない場合は、この塊をひとつの文字とみなす。文字とみなされた塊の横方向と縦方向の長さの平均を、この行の文字の平均文字幅 ・平均文字高さとみなす。

4.3.2　接触文字強制分離

行内の領域のうち、横幅が平均文字幅と比較して極端に大きいものがある場合は文字間接触があるとみなし、まず接触文字分割を行う。これを行うには黒画素の塊ごとの行ヒストグラムの他に列ヒストグラムが必要なので、まずこれを求める。図 4.3の上段の画像に対して得られた列ヒストグラムが下段である。黒画素の塊のうち、右側の大きい方が文字間接触があるとみなされた部分である。最初に列ヒストグラムの各列ごとにその近傍を見て微分係数を求め、極小となる位置を探す。極小となる位置全てにおいて、列ヒストグラムの値がある閾値よりも小さければ、その列の位置を強制分離の位置とする。図 4.3では 「争」と 「の」の間の縦線の位置で文字領域を 2 分している。以上により文字の区切りを定めてもまだ横幅が極端に大きい領域が残る場合があるが、現在はこれ以上の分割は行わない。

4.3.3　分離文字統合と最尤系列導出

平均文字幅を大きく超えないという条件の下で考えられる全ての統合の組み合わせについて、図 4.4のようなグラフをつくる。このグラフの全てのノードについて個別文字認識を

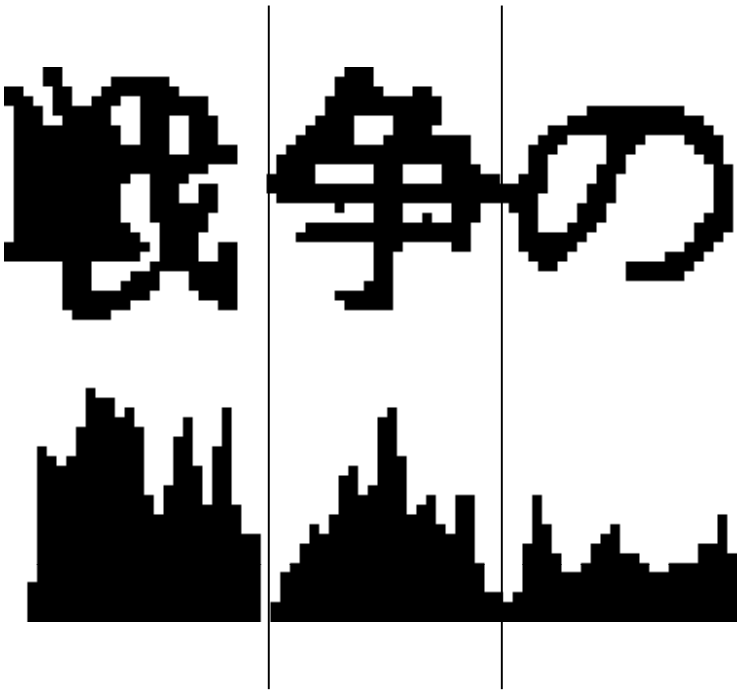

図 4.3: 行イメージと列ヒストグラム

行い、3.3節と同様の方法で認識結果集合 Rを求める。

最後に式 (4.3) を Viterbi アルゴリズムによって求め、認識結果として出力する。

4.4　日本語文字 trigram モデルを用いた文書認識系の実現

4.4.1　個別文字認識系

以上のアルゴリズムに基づいた文書認識系を構成し、性能を評価する。個別文字認識系は 3.4節で用いたものとほぼ同じである。実験条件を表 4.1に示す。また、確定文字を決める閾値 $r_{\rm th} = 1.7$, 候補文字数 $N_{\rm c} = 5$ とした。

4.4.2　認識対象

認識実験の対象となる文章は、第 3章で改善が顕著であった「社説」(B) のうち前半 30451 字を用いた。認識対象画像の品質による性能の違いを見るため、この文章をそれぞれ 9 ポイント (中品質) ・7 ポイント (低品質) ・5 ポイント (極めて低品質) の 3 種類の大きさの未

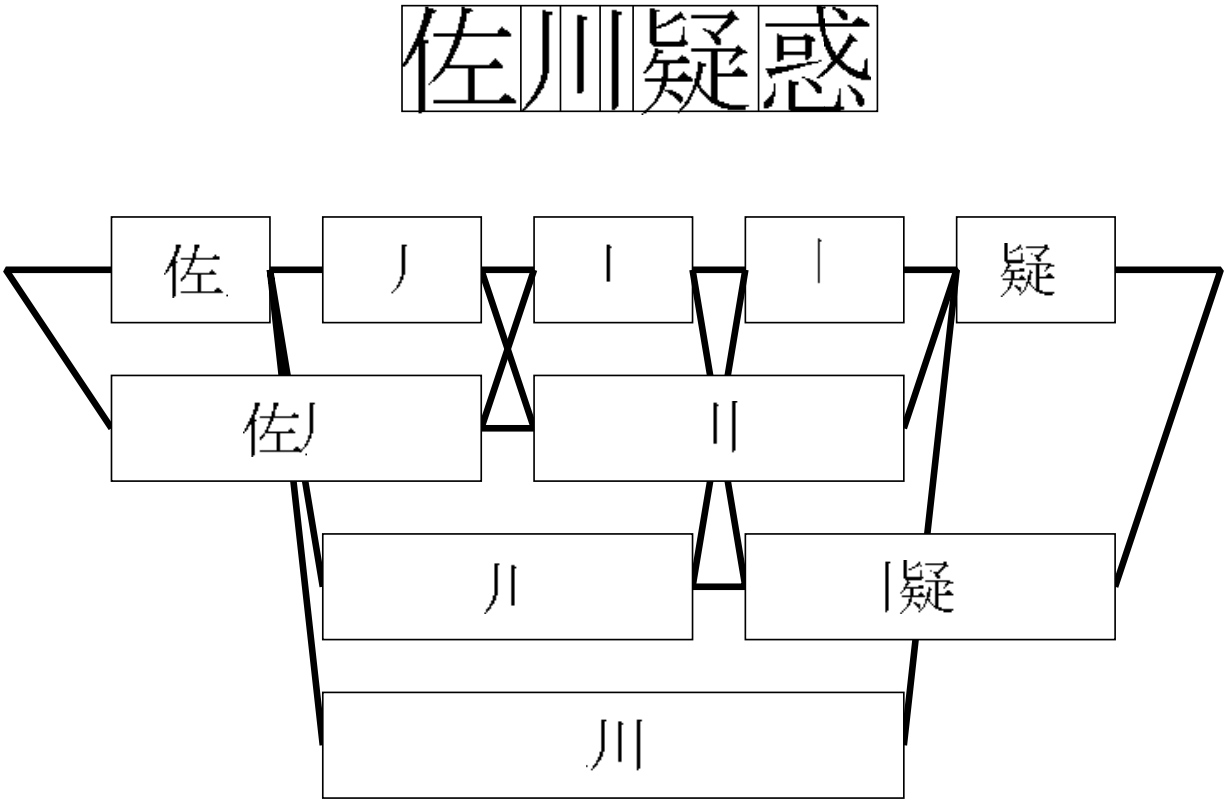

図 4.4: 分離文字統合と最尤系列導出のためのグラフ

表 4.1: 個別文字認識系の実験条件

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

知フォントを用いてレーザープリンタより印字し、テストサンプルとした。

4.4.3　言語モデル

日本語文字 trigram モデルを訓練するためのテキストとして、第 3章で用いたのと同じ「社説」(A) のうち約 97 万字を用意した。スムージング法としては、第 3章でその効果が確かめられた削除補間法を用いた。

4.4.4　認識率

切り出しが未知の場合には正解の文字数と認識結果の文字数が一般には一致しない。このため、次によって認識率を定義する。正解を$\boldsymbol{C} = C_1C_2\cdots C_k$, 認識結果を$\hat{\boldsymbol{C}}$としたとき、両者の DP マッチングによって得られる最大一致文字数を $D(\boldsymbol{C},\hat{\boldsymbol{C}})$ と表記する。このときの認識率は $D(\boldsymbol{C},\hat{\boldsymbol{C}})/k$で定義される。また誤認識率は $1-D(\boldsymbol{C},\hat{\boldsymbol{C}})/k$で定義される。

4.4.5　実験結果

以下、本手法による文書認識実験の結果を示す。まず 9 ポイントと 7 ポイントの大きさで印刷された認識対象サンプルに関する結果を図 4.5に示す。横軸は、式 (3.20) で $A = \frac{1}{1000}$ とした時の言語的尤度の重み wであり、縦軸は誤認識率である。凡例に proposed とあるものが本提案手法である。9 ポイント、7 ポイントのいずれにおいても $w = 0.4$ の付近で最も高い認識率が得られており、言語モデルを利用しない時に誤認識する文字がそれぞれ 80%, 85%程度減少している。

また、本手法の特徴である、切り出しに trigram モデルを用いることの有効性を確かめるために、次のような比較実験を併せて行った。すなわち、切り出しを決定するために文字画像の情報だけを使い、言語モデルの情報を使わない。式 (4.3) の代わりに、次式によって切り出し関数$\hat{f}$を求める。

$$\hat{f} = \operatorname*{argmax}_{f} \max_{\boldsymbol{C}} P(\boldsymbol{C}|f(\boldsymbol{i})) \tag{4.5}$$

$$= \operatorname*{argmax}_{f} \max_{\boldsymbol{C}} p(f(\boldsymbol{i})|\boldsymbol{C}) \tag{4.6}$$

$$= \operatorname*{argmax}_{f} \max_{\boldsymbol{C}} \prod_{i=1}^{l} p(\boldsymbol{x}_i|C_i) \tag{4.7}$$

ただし$\boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_l = f(\boldsymbol{i})$. これは村瀬ら [47] などのものと似た手法によって得られた文字認識結果に、第 3章の後処理を施すことを意味する。この手法を、以下では従来法と呼ぶ。

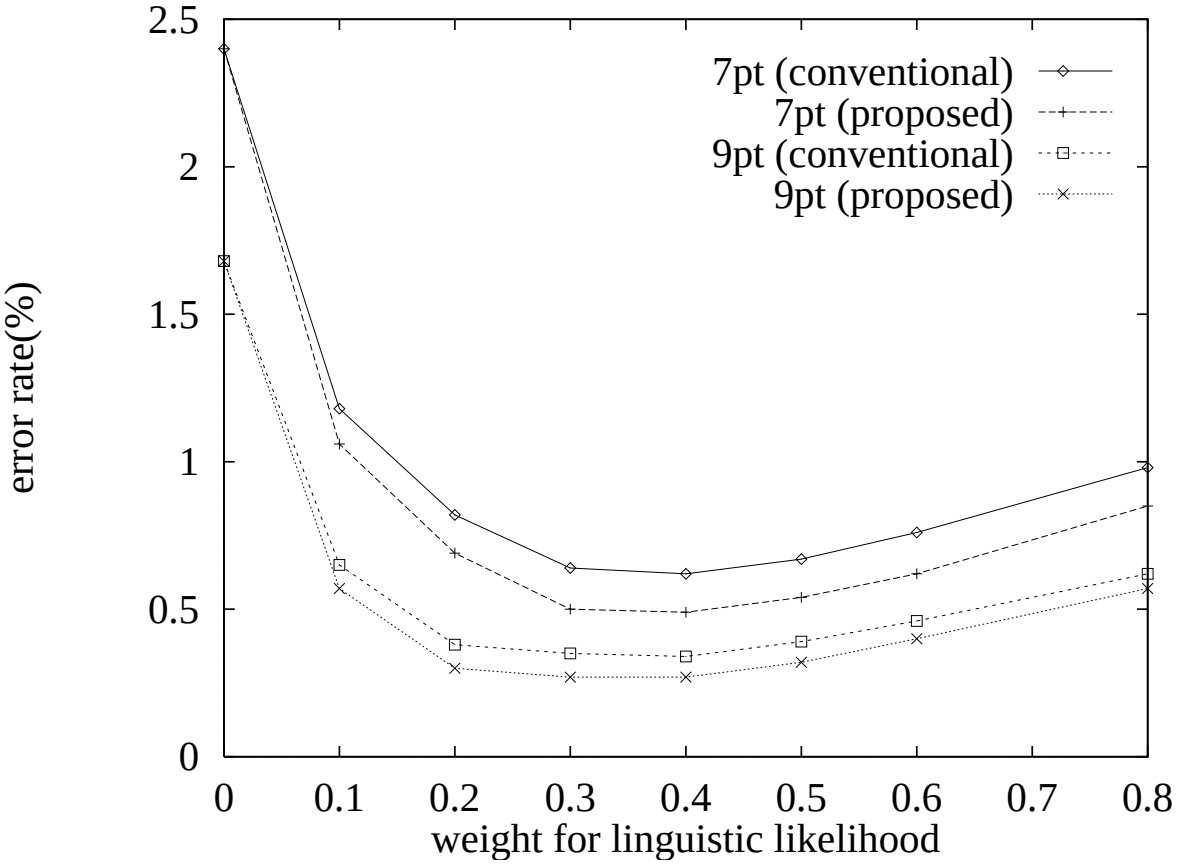

図 4.5: 切り出し方式と誤認識率の関係

従来法による誤認識率は、図 4.5の凡例で conventional と記してあるものである。提案手法と従来法の認識精度を比較すると、提案手法の方が優れている。従来法の認識誤りのうちの 20%程度が提案手法により解消されている。また、実験した範囲内では提案手法の導入による悪影響は全く観察されなかった。もし切り出し結果が同一ならば 2 つの手法による結果も同一となる。したがって、2 つの手法による結果に差があることから、提案手法によって切り出し誤りが減少したと言える。このことは、日本語 trigram モデルが個別文字認識の曖昧性だけでなく文字切り出しの曖昧性の削減にも効果があるという当初の予想と一致した結果となっている。

4.5　低品質文書への対処

次に、本手法による認識実験の結果を、5 ポイントのフォントで印刷された認識対象サンプルに対するものと併せて図 4.6に示す。5 ポイントのサンプルはつぶれが極端に激しく、認識率が 80%以下と非常に悪い。言語モデルを用いることで認識率を 86%程度まで上げることができるが、まだ実用になる精度ではない。

このような低品質文字の認識において言語モデルの効果が小さいことの主要原因の一つが候補外文字である。自然言語処理を用いた文字認識においては、探索空間の爆発を防ぐために、認識結果として出力される候補文字の数に上限が設けられる。ところが、文書の品質が低下して画像から得られる情報が少なくなると、正解が上位の候補に存在するという前提が成り立たなくなってくる。図 4.6は出力候補数 $N_c = 5$ としたときの結果であった。そこで、N_cを変化させることによって低品質文書に対する認識精度がどの程度変化するかを調べる。$w = 0.6$ とし、N_cを変化させながら 5 ポイントのサンプルを認識した結果を図 4.7に示す。この図からわかるように、N_cが 5 以上になると N_cの増加に比べ誤認識率の低下の度合が非常に緩やかなものとなっている。第 3章で既に述べた通り、最適化演算に必要な時間計算量は N_cの 3 乗のオーダであり、候補外文字を減らすためにこれ以上 N_cを増加させることは困難である。

文字認識における候補外文字の問題に対処するための手法として、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 [48]。しかし confusion 行列はサンプルに依存する先見知識であるため、過学習の危険がある。

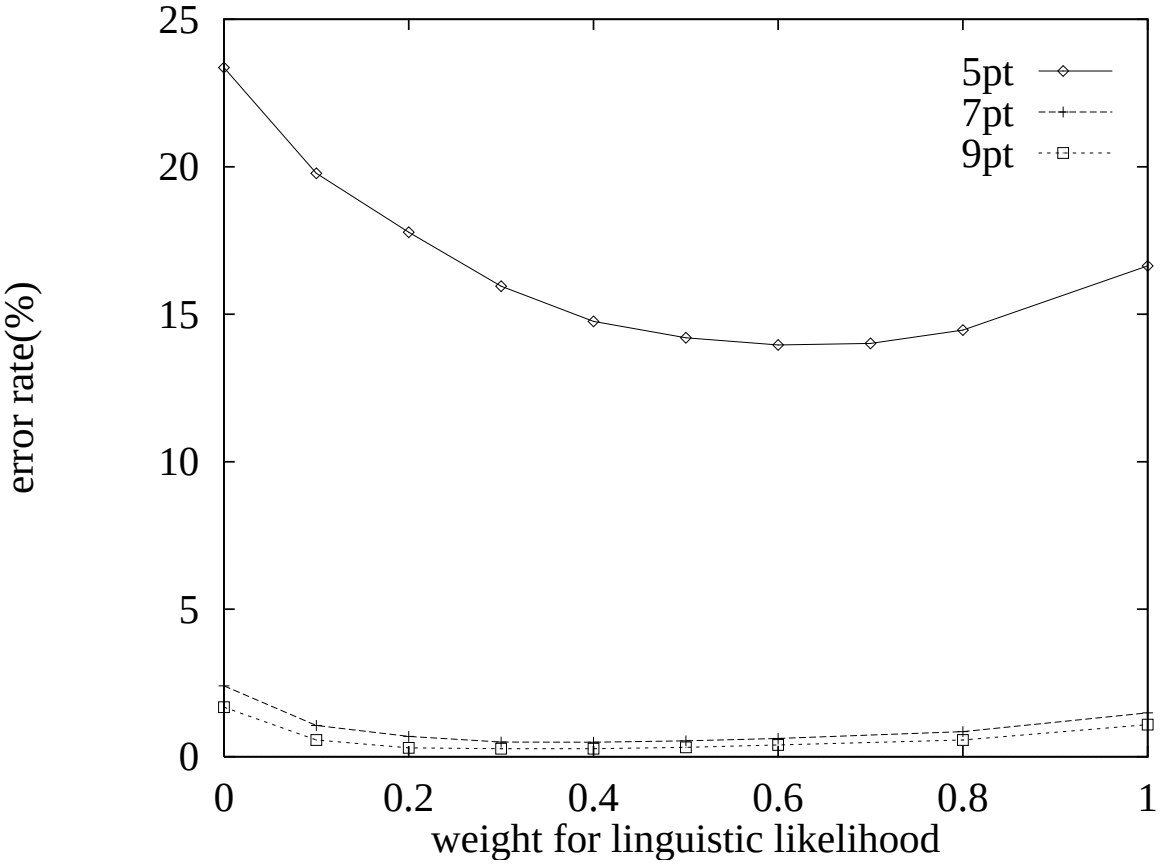

図 4.6: 印字品質と誤認識率の関係

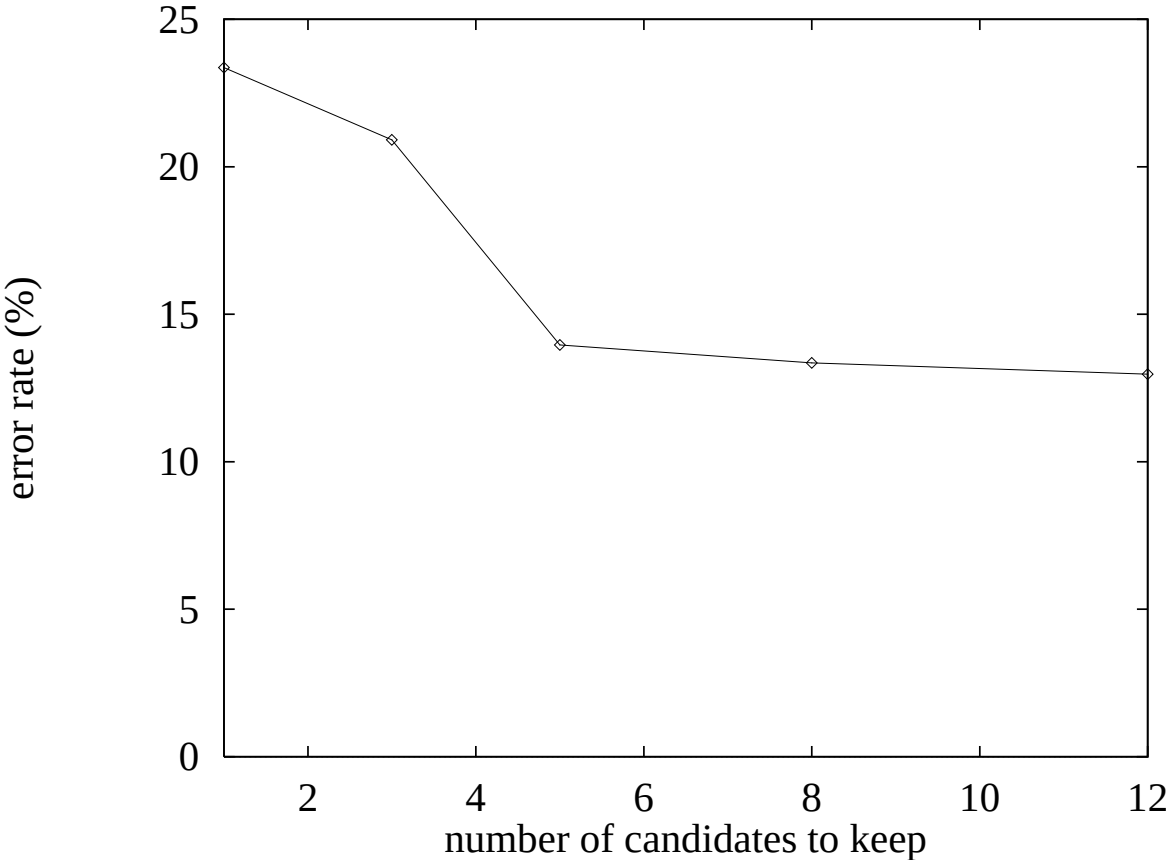

図 4.7: 出力候補数 N_c と誤認識率の関係

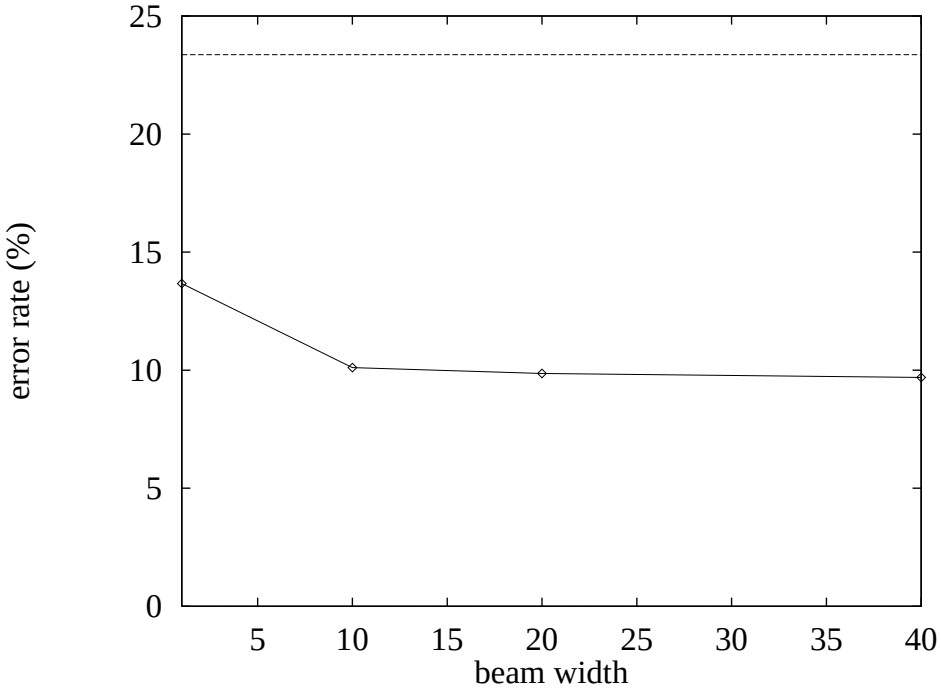

図 4.8: ビーム幅と誤認識率

4.5.1　ビームサーチ法の導入

本論文では、低品質文書の候補外文字の問題に対処するため、個別文字認識の結果に関わらず、全ての文字を候補として考慮することを試みる。その際、探索空間の増加を抑えるためにビームサーチ法を導入する。

Viterbi アルゴリズムは式 (3.24), (3.25) で表わされるスコアを各時点での全てのマルコフ状態について計算し、保存するものである。しかし状態数は ${N_c}^2$であり、全ての文字を候補とすると計算コストは非常に大きいものとなってしまう。そこで、スコアを保存する状態の数をある定数 Wに制限し、各時点でのスコアが高い上位 W個の状態を保存する。この定数 Wをビーム幅という。ビームサーチの導入によって Viterbi アルゴリズムによる解の最適性は失われるが、Wを十分大きく選べば最適解が得られる可能性が高くなる。

4.5.2　ビームサーチ法を用いた認識実験

ビームサーチ法の導入による候補外文字問題の解消効果を調べるため、前述のものと同じ 5 ポイントのサンプルを用いて認識実験を行った。条件は $w = 0.5$, $r_{\rm th} = 5.0$, 言語モデルは前述のものと同様である。結果を図 4.8に示す。破線は言語モデルを用いない場合の認

識率 (76.64%) を表わす。本手法により、認識率は 90%以上となり、言語モデルなしの場合に比べ約 13%向上した。また、ビームサーチの導入以前に比べ約 3%の認識率向上が見られた。これにより、ビームサーチの導入前は候補外文字が影響していたことがわかる。W が比較的小さいところでも誤認識率はほぼ一定になっていることから、既にビームサーチ法の限界に達していることが予想される。

4.6　まとめ

本章では文字認識の未解決問題である切り出しの曖昧性の問題を、日本語文字 trigram モデルを用いることにより、認識の曖昧性と同時に解決する手法を提案し、その有効性を実験により確かめた。また、もう一つの未解決問題である低品質文字の認識の高精度化を行うため、候補外文字への対処法としてビームサーチ法の導入を試み、有効性を確かめた。

切り出しの問題は、接触文字の分割が適切に行われていれば、本手法によりほぼ解決されると考える。ただ、今回実験した限りでは、切り出しに言語情報を用いないものとの差があまり大きく出なかった。この一つの原因としては、横書き文書における切り出しの曖昧性がそれほど大きくなく、画像情報だけでほとんど解決できてしまうからということが考えられる。これに対して、縦書きの文書の場合は、「言」と「一」「二」「三」など分離の影響が甚大な文字が多いために、横書き文書に比べて切り出しの本質的曖昧性が大きいことが考えられる。よって、今後は縦書き文書を対象に本手法の有効性を確かめることが必要である。

一方、本文中では触れなかったが、低品質文書の問題点としては、文字それ自身が低品質で得られる情報が少ないことの他に、分離が難しい接触文字が非常に多いことが挙げられる。本論文では簡単のためにヒストグラムから接触文字の分割を行ったが、より高次の画像理解の手法の導入により、接触文字対策を強化することでさらに認識精度が向上できると考える。しかし、画像情報だけでは分離が不可能だと思われる例もある。このため、例えば接触が疑われる文字領域をある一定の幅で短冊状に切断し、後の統合処理で文字を復元するといった手法も今後検討されるべき事柄である。

第 5 章

n-gram モデルの改善

第 4章までに、日本語文字 trigram モデルが前後の文脈から可能性のある文字を限定する強い曖昧性削減能力を持っており、文字認識の高精度化に有効であることを明らかにした。同時に、日本語文字 trigram モデルの曖昧性削減効果は、モデルを推定するための訓練テキストの言語的性質と、認識対象文書の言語的性質との関係に依存しており、訓練テキストと相違するタスクに関する認識性能が低下することが示された。

序論で述べたように、本研究では認識対象文書のタスクを特定せず、一般の文書に対する文字認識の高精度化を目標としている。したがって、タスクによって認識性能が劣ることは望ましくない。

そこで本章では、タスクの相違が認識性能に与える悪影響を低減するため、日本語文字 trigram モデルを認識対象文書に合わせて適応化させることを検討する。本論文で提案するのは、曖昧な認識結果から得られる対象文書に関する情報を用いて trigram 確率を更新する手法である。

5.1 言語モデルの適応化

図 2.9に示したように、自然言語の要素である文は全てがいつも同じような頻度で出現するわけではなく、文が生成される環境・条件によって出現頻度が増加したり、あるいは全く出現しなかったりということが起きると考えられている。このように、文の出現頻度を支配する特定の環境・状況をタスクと呼ぶ。

属する文の数が最も少ないタスクの例が定型帳票の類である。手書き文字での例を挙げれば、郵便物の住所の項がそれにあたる。住所のタスクは、文として存在するかどうかの線引きが非常に明確であり、認識の対象としては与しやすい部類に属する。逆に、 天声人

語」などの随筆は、読者の目先を変えるために様々な話題を提供する必要があり、非常に広い領域のタスクであると考えられる[1]。また、漫画の登場人物のせりふ、擬音などを構成する文の文体は「社説」や「天声人語」のような新聞記事とは大きく異なる独特のものである。

特定のタスクが認識対象であれば、それを最も良く認識するための確率モデルは、同じタスクに属する文のみから構成された大量サンプルを用いて訓練したものであろう。しかし、ここで扱っているのは特定タスク向けではない一般文を認識するための言語モデルである。一般の文の認識のためには全ての考え得るタスクに属するサンプル文を収集し、そこから言語モデルを構築すれば良い。しかし、そのようにして得られた言語モデルは想定する文が特定タスクに比べて大量であるため曖昧性削減効果が小さくなり、特定タスクを用いて訓練された言語モデルにを用いた場合と比べて認識性能の低下が予想される。

このような問題に対処するために、言語モデルを認識タスクに近づけることを適応化という。適応化のための一つの方法は、予めいくつかのタスクに関してそれぞれを最も良く表現するような言語モデルを用意しておき、認識タスクに最も近いモデルを選択することである。この種の手法には、どのタスクが最も近いかを人間が指示する教師ありのものと、認識対象文書を手がかりとして自動的に判断する教師なしのものとがある。過去の研究としては、不特定話者連続音声認識のための HMM による音素モデルの教師なし話者適応に関するもの [49] がある。

適応化のもう一つの考え方は、認識対象そのものの持つ情報を利用するというものである。Kuhn and De Mori[50] は、英単語の trigram に関して、最近出現したテキストをキャッシュに記録することによって trigram 確率を更新する方法を考案した。このときの trigram 確率は、一般向け大規模テキストコーパスから得られたモデルによるものと、キャッシュから得られたものとを線形結合し、重みは削除補間法を用いて決定している。これによってパープレキシティが $1/3$ に減少したと報告されている。また Matsunaga ら [51] は、上述の 2 通りの適応化を統合した教師あり適応化手法を提案している。

しかし現実のパターン認識において問題となってくるのは、前述したようなタスクの相違そのものよりは、むしろ少数データ問題であることが多い。つまり、一般向けの言語モデルを構成したくとも、現実には全てのタスクに属するテキストデータを収集することは不可能なのである。この場合、言語モデルの認識タスクへの適応化は、少数データの影響を低減する意味合いを持つ。

言語モデルが認識対象文書をいかに良く模倣するかの尺度として、第 2章では被覆率を

[1] 気が付いたところでは、方言、「さもありなん」などの文語的表現、古文の引用、会話調などが見られる。

定義して検討した。その結果、認識対象文書が言語モデルを訓練したテキストと比較的近い性質を持つ場合であっても、異なる trigram の数は飽和しないことがわかった。したがって、タスクが異なる場合にはこの被覆率がさらに低下することが予想される。

少数データ問題は、削除補間法などのスムージングの手法によって影響を低減できる。しかし、認識タスクの相違までは吸収できないことが第 2章のパープレキシティによる評価、および第 3章の認識実験による評価によって明らかである。このことから、本論文では特に被覆率の向上を目的とした認識タスクへの適応化によって異なるタスクに対する少数データの影響の低減を図り、認識精度に与える影響を調べることとする。本論文が指向する知的な文字認識の実現のためには、適応化の過程にあっても人間が介在しないのが好ましい。そこで、trigram モデルの適応化は教師なしとする。

認識対象文書は未知のものであるから、未知の語彙を獲得するためには一度対象を認識し、その結果を利用するしかない。しかし、認識結果は誤りを含んでいるから、認識結果を直接モデルの適応化に用いると、誤認識した文字が悪影響を及ぼすことが考えられる。そこで、何らかの方法で誤認識した文字、およびその周辺の文字を排除することが必要である。

一つの方法は、認識結果を一度後処理することである。第 3章で明らかになったように、削除補間法によってスムージングされた trigram モデルを用いた後処理は、未知の語彙を多く含む対象に対しても誤り訂正能力を持つ。これは、日本語としての基本的な部分が共有されているのと同時に、未知の語彙も潜在的な日本語らしさを保っているからだと考えられる。したがって、後処理を施すことにより誤認識した文字を減少できる。

認識結果はこれでもなお誤りを含むが、残った誤りを知る方法を我々は知らない。よって、正解テキストの代わりに、このような一度後処理を施した認識結果を用いることとする。

このようにして得られる不完全な適応テキストがモデルの確率の再推定に有効であるためには、1 度目の後処理結果が次のような仮定を満たさなければならない。

- 局所的には誤りを含んでいて良いが、ある語彙が誤って認識されたならば、同じ語彙が別の箇所に正解として存在し、しかも正解の数の方が誤りよりも多く存在する。

以下ではこの仮定が成り立つという前提で議論する。

5.2 trigram 確率再推定アルゴリズム

trigram モデルの訓練用テキストを $\mathcal{T}$ と書く。また、式 (3.17) で定義される尤度が訓練用テキスト $\mathcal{T}$ に基づく trigram モデルから求められる時、この尤度を $L_{\mathcal{T}}$ と書く。

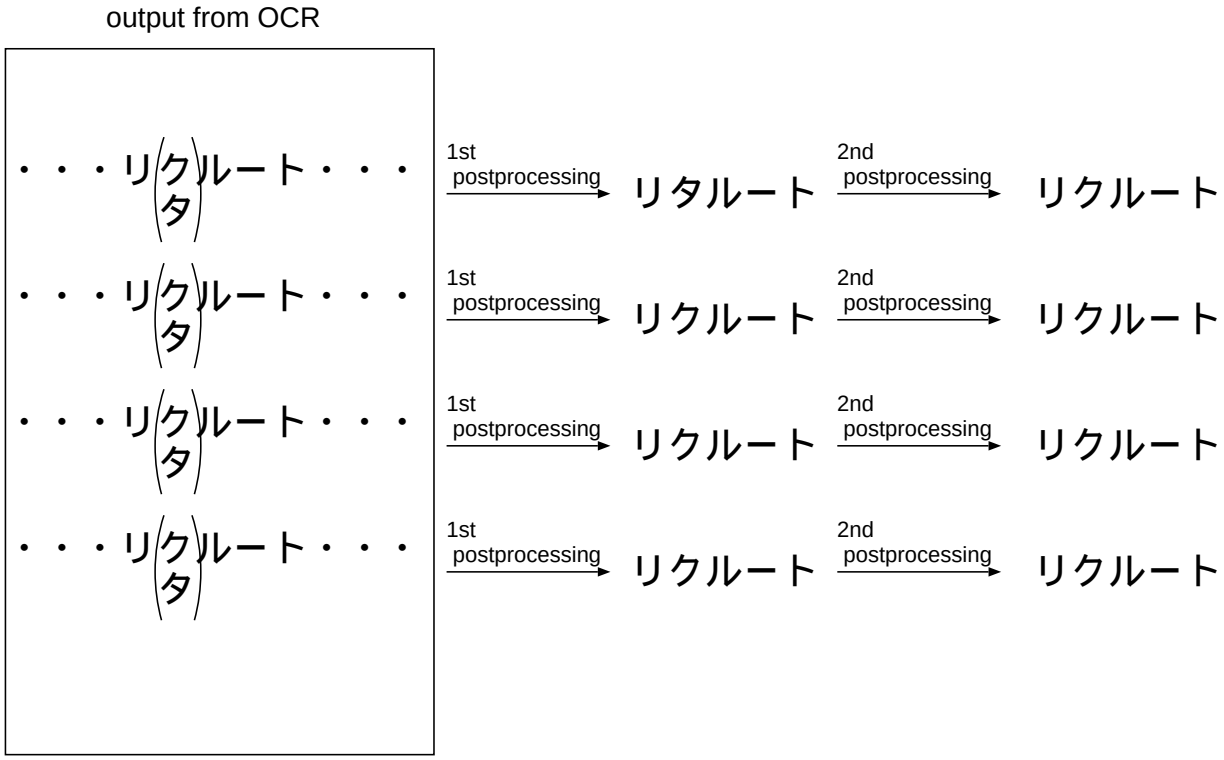

図 5.1: 適応化がうまく行く例

$\mathcal{T}$による後処理の結果得られた最尤系列を、連続した確定文字による処理単位ごとに$\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_m$と表記する。仮定によって、これらの系列の全体としての正解率は高い。すると、このテキスト自身を trigram 訓練用テキストデータと考えることができ、認識対象文書の性格を反映した trigram 確率の更新が行える。更新した trigram モデルを用いて再度後処理を行うと、一度目の後処理では不正解であったものが正解に転ずる可能性がある。一例を図 5.1に示す。左側は後処理を施していない認識結果で、括弧付きの文字は未確定文字、つまり曖昧な認識結果である。trigram モデルの訓練テキスト中にない語彙 『リクルート」が文書中に 4 回出現し、このうち 1 つが 『ク」を 『タ」に誤認識したとする。1 度目後処理の過程においては 『リクルート」という語彙は未知であり、 『リタルート」と同等に低い言語的尤度を持っている。その時は画像情報が後処理結果を支配するため、1 度目の後処理の結果はやはり 1 つが 『ク」を 『タ」に誤認識したとする。ここで trigram 確率の再推定が行われるが、その際に trigram 確率ベースには 『リタルート」が 1 回出現、 『リクルート」が 3 回出現と記録される。このようにして更新された trigram モデルを用いて 2 度目の後処理を行うと、今度は 『リクルート」という語彙が trigram 確率ベースに存在し、

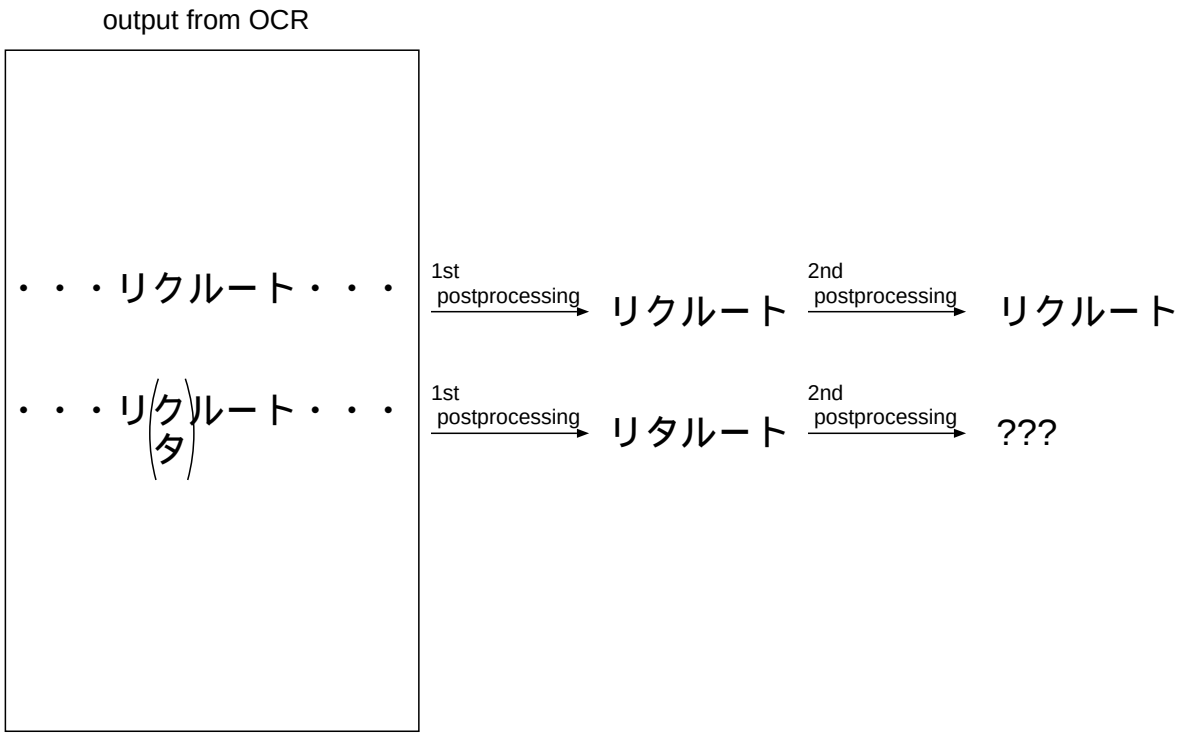

図 5.2: 適応化がうまく行かない例

しかもその頻度が 「リタルート」より多いので、1 度目には 「リタルート」と誤認識した部分が今度は 「リクルート」と正しく認識されるようになる。このようにして、未知語彙への適応化が実現される。

ところが、j番目の処理単位に対して、$\mathcal{S}_j$を含む全系列によって再推定した trigram 確率によって再び後処理を行うことは、$\mathcal{S}_j$ が誤りを含んでいた場合に誤りそのものを肯定する結果となってしまう。この例を図 5.2に示す。この例では、未知の語彙 「リクルート」が認識対象中に 2 回出現し、そのうち 1 つが未確定文字を含んでいる。1 度目の後処理によって未確定文字が 「タ」に誤認識されたとしよう。誤認識された 「リタルート」も出現回数 1 回となり、「リクルート」と甲乙付けがたくなってしまう。したがって、このような認識結果を用いて trigram モデルを適応させても、2 度目の後処理で訂正される望みは薄くなってしまう。この例では未知語彙が 1 度は正しく認識され、しかも確定されているにもかかわらず適応化に失敗しており、適応化の手法として不適当と思われる。

したがって、j番目の処理単位に再び後処理を適用する場合には、$\mathcal{S}_j$以外の系列を用いて修正した trigram 確率を用いることが妥当と考えられる。図 5.2では、未確定文字の部分に

対して 2 度目の後処理を実行する際には、1 度目の認識結果の中から自分自身の認識結果「リタルート」を除き、残りを用いた trigram モデルによって 2 度目の処理を行う。これによって、この部分を処理する場合には「リクルート」が 1 回出現していることを利用して正しい結果が得られる。

この考えにもとづいて、対象文書に適応化した文字認識を、次の 2 回の後処理過程により実現することを提案する。

1. 処理単位 $j = 1, \cdots, m$ に対し $L_{\mathcal{T}}$を最大にする $\mathcal{S}_j$ を求める
2. 処理単位 $j = 1, \cdots, m$ に対し $L_{\mathcal{T}+\mathcal{S}_1+\cdots+\mathcal{S}_{j-1}+\mathcal{S}_{j+1}+\cdots+\mathcal{S}_m}$を最大にする新しい $\mathcal{S}_j$ を求める

5.3 実験

個別文字認識系は第 3章で用いたものとほぼ同様だが、知識処理によって正規化の悪影響を低減したものを用いた。認識対象としては、「科研費概要」(E) を選んだ。後処理のための日本語文字 trigram モデルは、「社説」(A) のうちの 97 万字によって訓練した。また trigram モデルのスムージングを削除補間法により行った。後処理を施す前の認識率は 97.81%、1 度目の後処理を施した後の認識率は 98.82%である。なお、この訓練テキストに認識対象文書の正解そのものを加えて構築した trigram モデルによる後処理後の認識率は 99.37%であり、この値が適応化処理によって向上し得る上限を与える。

提案する手法により 2 度目の後処理を行った後の認識率は 98.89%となった。これは、従来法での誤り文字が本手法により約 5.9%減少したことを意味する。認識率に大きな差は見られなかったが、改善したものの中には「センサ」「正規化」などの専門用語が多く観察され、当初の目標であった未知語彙の獲得が行えていることを示している。適応化処理で改善が見られたものを図 5.3に示す。図中、鉤括弧内は 1 度目の後処理を施された後の認識結果で、下線部が 1 度目には誤認識された文字である。矢印の先が本手法によって得られた正解文字である。

5.4 まとめ

日本語文字 trigram モデルにより一度後処理を施した認識対象文書から未知の語彙を獲得して適応化を行う手法を提案し、実験によりその効果を確かめた。その結果、適応化処

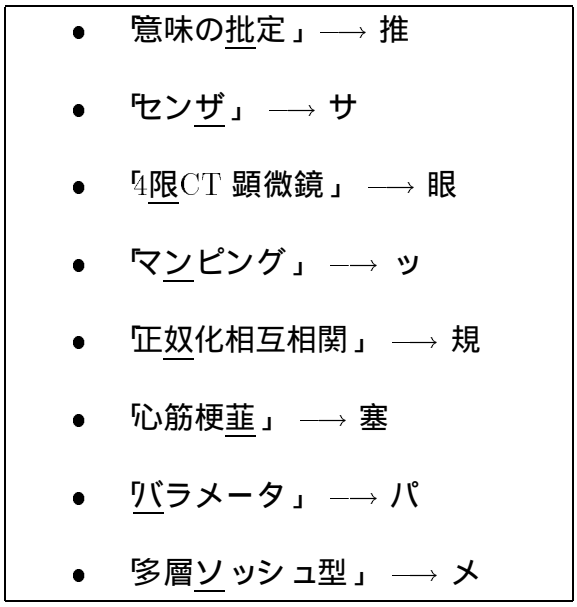

図 5.3: 改善例

理によっていくつかの未知語彙の誤認識を救済できた。認識率向上への貢献はわずかであり、本手法はさらに改良される必要がある。

本手法は後処理の精度が十分高いことが前提となっているから、条件によっては改善が見込まれない。特に、現実の文字認識では頻繁に起こる習慣性のある誤りのついては、本手法によって改悪することも考えられる。したがって、今後の課題として、結果に信頼のおける部分を同定する手法が望まれる。

また、本手法は未知語彙の獲得のみを行うことが目的であり、タスクに依存する大局的な確率分布に影響を与えることはない。そこで、汎用の大規模 trigram ベースと適応化によって得られたタスク向き trigram ベースを重み付きで組み合わせることにより、大局的な適応化が可能か否かも興味のある問題である。

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、文字認識に言語知識を融合させるため、日本語文字 n-gram モデルを導入し、定式化を行った。また、曖昧性を解消する十分な制約を与えるため、$n = 3$ 以上、したがって trigram モデル以上が用いられなければならないことを示した。同時に、trigram モデル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な手法としては削除補間法が優れていることが、パープレキシティを基準とした評価により明らかになった。

第 3 章では、文字の切り出し位置が与えられている場合について、事後確率最大基準による文字認識後処理系が検討された。そのために、まず文字認識系のモデル化を含めた必要な定式化が行われた。ついで、日本語文字 trigram モデルを用いた最適化アルゴリズムを実際の文字認識結果に適用し、非常に高い割合で誤認識された文字が救済されることを示した。またこの実験結果によって、削除補間法が少数サンプル状況下でその影響を軽減することも明らかにされた。このことは第 2 章のパープレキシティによる評価に対応するものである。

第 4 章では、日本語文字 trigram モデルを用いた類似文字の判定に加え、切り出し位置まで同時に判定する認識アルゴリズムを示し、実験により検討を行った、その結果、切り出し位置を画像情報のみから判定する方式に比べより高精度な認識系を実現でき、本手法の有効性が確認された。

第 5 章では、訓練テキストと異なる言語的性質を持つ認識対象に関する認識精度を向上させるため、trigram モデルの認識対象への適応化が検討された。ここではその具体的な手

法として、一度後処理された結果を言語モデル訓練用テキストに加え、再び後処理を実行するというアルゴリズムが提案された。切り出し境界が与えられた文字認識系の後処理に本手法を適用した結果、通常の後処理では正解とならなかった文字が正解に転じ、認識精度の向上にわずかながら寄与した。この処理において改善したものの中には、専門用語の一部をなす文字が多く見られ、trigram モデルの確率分布が未知語を多く含むタスクへ適応していることが示された。

6.2　おわりに

序論で述べた通り、本研究の目的は、従来のイメージ情報だけに頼った一面的な文字認識技術の限界を打破する知的な認識技術の開発であった。その目的を達成するために不可欠の要素と見られる自然言語処理の形態として、本論文では確率モデル、具体的には文字の n-gram モデルについて詳細な検討が行われた。筆者がこれまでに目にした過去の文字認識のための自然言語処理研究は、そのほとんどが「まず単語辞書ありき」の立場に立つものであった。これらの研究から受ける印象は、高精度な言語処理とはマルコフモデルより高次の言語情報を活用する構文解析や意味解析である、とするパラダイムに強く支配されているというものである。筆者も、そういった高次の言語情報の必要性は感じている。それにも関わらず本研究が敢えて単語辞書という出発点を選ばなかった理由の一つは、常に変化し成長し続ける「ことば」をある単一の枠組の中へ押し込めてしまおうとする考えに対する違和感であったように思える。

本研究は、自然言語の自動学習モデルを本格的に日本語文字認識に応用した初めての試みである。trigram モデルは簡単なモデルであるが、本論文はこれが従来の知識主導型の言語モデルに代わる存在となり得ることを示した。一般に n-gram モデルは非効率的であり、近年のハードウェアの進歩によって初めて現実的なものとなった感がある。これは、今後のさらなるハードの性能向上により、n-gram モデルを上回る精度を持つ確率的言語モデルがもたらされる可能性をも示唆するものである。

用例を用いた手法、特に確率モデルを用いた手法が今後の自然言語処理研究の最も重要な流れを形成して行くことは間違いない。本研究の成果が、このような確率モデルを用いた自然言語処理技術の礎として、新たな技術への足がかりとなることが期待される。

謝辞

本研究を進めるにあたり、2 年間に渡って全般的にご指導くださった東北大学工学部 阿曽弘具教授に心から感謝致します。本論文をまとめるにあたり貴重なご助言をくださった東北大学情報科学研究科 西関隆夫教授、東北大学電気通信研究所 矢野雅文教授に深く御礼申し上げます。

日頃から労をいとわずご指導くださる東北大学情報科学研究科 牧野正三助教授に心から感謝致します。

東北大学情報処理教育センター 大町真一郎博士には、文字認識全般に渡って非常に有益なご助言を頂きました。東北大学情報処理教育センター 伊藤彰則博士には、自然言語処理研究の立場から貴重なご意見を頂きました。東北大学工学部 成富敬博士には、多岐にわたってご指導くださいました。東北大学大型計算機センター 孫寧博士には、私の疑問に丁寧にお答えくださいました。ここに感謝の意を表します。

八戸工業高等専門学校 細越淳一氏には、文字認識全般に渡ってご討論くださいました。日本電信電話株式会社 粟津辰功氏には、音声認識の分野からの貴重なご助言をいただきました。

最後に、日頃から熱心にご指導を賜わり、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、並びに東北大学情報科学研究科情報生物学研究室の皆様には非常にお世話になりました。ここに記して深く感謝致します。

参考文献

[1] 飯島 泰蔵：“パターン認識理論”, 森北出版 (1989).

[2] 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).

[3] 牧野, 大澤：“べた書き文の仮名漢字変換システムとその同音語処理”, 情報処理, **22**, 1, pp. 59–67 (1981).

[4] 吉村, 日高, 吉田：“文節数最小法を用いたべた書き日本語文の形態素解析”, 情報処理学会論文誌, **24**, 1, pp. 40–46 (1983).

[5] 日本電子化辞書研究所：“EDR 電子化辞書仕様説明書”, 日本電子化辞書研究所 (1993).

[6] E. A. Feigenbaum : “The art of artificial intelligence I: themes and case studies of knowledge engineering”, Proceedings of International Joint Conference of Artificial Intelligence, pp. 1014–1029 (1975).

[7] 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).

[8] N. Chomsky : “Formal properties of grammers”, Handbook of Mathematical Psychology II, pp. 323–418, John Wiley (1963).

[9] K. Lari and S. J. Young : “The Estimation of stochastic context-free grammars using the Inside-Outside algorithm”, Computer Speech and Language, 4, pp.35–36 (1990).

[10] L. E. Baum : “An inequality and associated maximization technique occurring in the stastical estimation for probabilistic functions of Markov process”, Inequalities, 3, pp. 1–8 (1972).

[11] 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).

[12] 松延, 日高, 吉田 : “確率文節文法による構文解析”, 情報処理学会自然言語処理研究会資料, **NL56**-3 (1986).

[13] C. E. Shannon : “A Mathematical Theory of Communication”, Bell System Technical Journal, **27**, pp. 379–423, 623–656 (1948).

[14] L. R. Bahl, F. Jelinek and R. L. Mercer : “A Maximum Likelihood Approach to Continuous Speech Recognition”, IEEE Transactions on Pattern Analisys and Machine Intelligence, **PAMI-5**, 2, pp.179–190 (1983).

[15] 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).

[16] 星宮, 石井, 塚田, 井出 : “生体情報工学”, 森北出版 (1987).

[17] 中村 雅巳, 鹿野 清宏 : “ニューラルネットワークによる英単語品詞列予測モデル”, 電子情報通信学会論文誌 (D-II), **J74-D-II**, 1, pp. 1–7 (1991).

[18] 西野 文人 : “文字認識における自然言語処理”, 情報処理, **34**, 10, pp. 1274–1280 (1993).

[19] 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, **30**, 11, pp. 1394–1401 (1989).

[20] 紺野 章子, 本郷 保夫 : “日本語 OCR の後処理に関する一手法”, 電子情報通信学会技術報告, **PRU92**-21, pp. 23–30 (1992).

[21] Brown University : “Brown Corpus”, Technical Report, Brown University (1967).

[22] S. Johansson, E. Atwell, R. Garside and G. Leech : “The Tagged LOB Corpus User's Manual”, Norwegian Computing Centre for the Humanities, Bergen, Norway (1986).

[23] 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, **25**, 2, pp. 298–305 (1984).

[24] 村上, 荒木, 池原 : “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), **J75-D-II**, 1, pp. 11–20 (1992).

[25] 山田, 松永, 川端, 鹿野 : “音声認識における仮名 ・漢字文字連鎖確率に基づく統計的言語モデルの利用”, 電子情報通信学会論文誌 (A), **J77-A**, 2, pp. 198–205 (1994).

[26] E. M. Riseman and A. R. Hanson : "A Contextual Postprocessing System for Error Correction Using Binary n-Grams", IEEE Transactions on Computer, **C-23**, 5, pp. 480–493 (1974).

[27] 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), **J68-D**, 1, pp. 64–71 (1985).

[28] 日本規格協会編 : "JIS ハンドブック 情報処理 ソフトウェア ・符号編 情報交換用漢字符号 X0208-1990", 日本規格協会 (1990).

[29] 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).

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

[31] 荒木, 池原, 土橋, 笹島 : "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] S. K. Katz : "Estimation from sparse data for the language model for a speech recognition", IEEE Transactions on Acoustics, Speech & Signal Processing., **ASSP-35**, 3, pp. 400-401 (1987).

[33] 荒木, 村上, 池原 : "2 重音節マルコフモデルによる日本語の文節音節認識候補の曖昧さの解消効果", 情報処理学会論文誌, **30**, 4, pp. 467–476 (1989).

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

[35] 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).

[36] F. Jelinek and R. L. Mercer : "Interpolated Estimation of Markov Source Parameters from Sparse Data", Pattern Recognition in Practice, pp. 381–397, ed. E. S. Gelsema, L. N. Kanal, North-Holland (1980).

[37] 瀧 保夫 : "情報論 I", 岩波書店 (1978).

[38] 孫, 田原, 阿曽, 木村 : "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), **J74-D-II**, 3, pp. 330–339 (1991).

[39] 萩田, 内藤, 増田 : “外郭方向寄与度特徴による手書き漢字の識別”, 電子情報通信学会論文誌 (D), **J66-D**, 10, pp. 1185–1192 (1983).

[40] 若林, 鶴岡, 木村, 三宅 : “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), **J76-D-II**, 12, pp. 2495–2503 (1993).

[41] 孫, 大町, 阿曽 : “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).

[42] 阿曽, 越後, 木村 : “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), **J76-D-II**, 11, pp. 2285–2294 (1993).

[43] 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, 7, **D-485**, p. 221 (1994).

[44] S. Watanabe : “Knowing and Guessing”, John Wiley & Sons (1969).

[45] 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).

[46] 孫 寧 : “文字認識の高速化と高精度化に関する研究”, 東北大学審査博士学位論文 (1991).

[47] 村瀬, 若原, 梅田 : “候補文字ラティス法による枠無し筆記文字列のオンライン認識”, 電子通信学会論文誌 (D), **J68-D**, 4, pp. 765–772 (1985).

[48] 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), **J72-D-II**, 7, pp. 993–1000 (1989).

[49] 粟津 辰功 : “不特定話者音声認識に関する基礎的研究”, 修士学位論文 (1994).

[50] R. Kuhn and R. De Mori : “A cache-based natural language model for speech recognition”, IEEE Transantions on Pattern Analisys and Machine Intelligence, **PAMI-12**, 6, pp. 570–582 (1990).

[51] S. Matsunaga, T. Yamada and K. Shikano : “Task Adaptation in Stochastic Language Models for Continuous Speech Recognition”, Proceedings of ICASSP-92, **I**, pp. 165–168 (1992).

研究業績

[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, 7, **D-487**, p. 223 (1994).

[2] 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).

[3] 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, **2D15**, p. 143 (1994).

[4] 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 (submitted).

索 引

Bayes の定理, 36, 53
bigram, 14, 18
bigram モデル, 14

EM アルゴリズム, 23

HMM, 5, 20, 21, 65

n-gram 確率, 12, 19
n-gram モデル, 6, 8, 12

OCR, 1, 14, 36

trigram, 15, 18, 66
trigram モデル, 8, 14, 46, 47, 53, 57

unigram, 18

Viterbi アルゴリズム, 39, 55

zerogram 確率, 21
zerogram モデル, 29

曖昧, 49
曖昧性, 4–6
曖昧な認識結果, 40
後処理, 2, 41, 53, 57, 66

印刷文字, 4

係り受け, 6
確定文字, 40, 67
確率正規文法, 5, 7
確率文脈自由文法, 5
確率モデル, 5, 6, 33, 36, 65
隠れマルコフモデル, 5, 21

規則主導, 4, 7
教師あり, 65
教師なし, 65
共分散行列, 37
切り出し関数, 53

形態素解析, 4, 6, 8, 14

言語的尤度, 38, 57

コーパス, 5, 7
候補外文字, 48, 59

最大事後確率推定法, 35
最大尤度基準, 20
最尤推定, 19, 23, 35
削除補間法, 20, 27, 46, 47, 57, 65, 66

自然言語処理, 2
出力候補数, 40, 59
少数データ問題, 6, 15, 19, 24, 65

事後確率, 35, 36, 53
事後確率最大基準, 35
辞書, 42
事前確率, 36
状態, 12, 19

冗長, 36
人工知能, 1

スムージング, 8, 19, 46, 47, 66

正規分布, 37
正規文法, 5
線素, 41

タスク, 27, 64

知識, 4, 7

低品質文字, 59
適応化, 65

統計的決定理論, 35
特徴抽出, 35, 36

パープレキシティ, 21, 27, 36, 48, 66
パターン整合法, 37
パターン認識, 1
パラメータ, 5

ヒストグラム, 50
被覆率, 15, 20, 21, 65

ビームサーチ法, 62
ビーム幅, 62

ファイゲンバウム ・ボトルネック, 4
復号誤り確率, 35
フロアリング, 20, 27, 47

文, 12, 24, 64
文章, 4, 12, 24
文脈自由文法, 5
分離文字, 50

方向線素特徴量, 41

前処理, 36
マハラノビス距離, 37
マルコフ性, 6, 12
マルコフモデル, 6

未確定文字, 40, 67

文字, 11
文字切り出し, 2, 8, 35, 43
文字認識, 1

ユークリッド距離, 38, 42

用例, 5

類似文字, 59
類似文字集合, 2