テキストコーパスからの確率的言語モデルの推定

森　信介

1997年12月

テキストコーパスからの確率的言語モデルの推定

森　信介

1997年12月

まえがき

情報理論を理論的支柱とする確率的言語モデルは、1950年代には、言語の文字あたりの情報量を推定するという学問的興味から研究されていた。しかし、1970年代に音声認識の一部分としてその重要性が再認識されると、計算機性能の飛躍的な向上と大規模なコーパスの出現に裏打ちされた確率的言語モデルは、確率的手法による音声認識の精度を実用レベルに押し上げる原動力の一つとなった。音声認識の両輪である言語モデルと音響モデルの双方は、精度を追求した場合には人手で記述できる範囲を遥かに超越している。このため、確率的手法は音声認識の主流となっている。加えて、双方が確率という意味のある尺度を共有しているので、後の統合を保証したまま独立に研究することができる。この性質は、応用に依存しない確率的手法の普遍的な長所である。したがって、確率的な手法による自然言語処理は、個人の言語直感に頼った場当たり的な方法から離れ、分析・分割・枚挙・統合という近代科学の方法論を適用することが可能となる。

確率的言語モデルは、対象を共有する言語学よりもモデルに対する制限が厳しいので、モデルの構成が困難であり、現状のモデルで扱える言語現象は単語の連接程度である。実際、現在実用となっている音声認識器には、単語列の頻度に基づく確率的モデルが用いられており、間隔をおいた単語の共起関係などをとらえることはできず、しばしばこれが原因と考えられる認識誤りを生じる。また、辞書に含まれない未知語などを認識することもできない。文字認識や読み付与に関しても同様である。このような限界を越えるには、確率的言語モデルをその言語現象に対応させることが必要かつ十分である。これにより、自然言語処理は確

実な進歩を遂げ、文脈処理などの前近代的な方法論によって研究されている問題を体系的に解決するための基礎が確保される。そして、これらを完全な確率的言語モデルで記述しさえすれば、複数の解析段階の統合という問題は自然に解決される。

本論文では、ある言語の任意の文字列の生成確率を計算する確率的言語モデルをコーパスから推定する方法について論じる。これは、特定の応用のための近似的な確率的言語モデルではなく、確率的言語モデルの条件を少しも逸脱しない完全なモデルである。具体的には、最も簡単な文字 n-gram モデルと、現在の応用で一般的に用いられる形態素 n-gram モデルと、今後の応用が十分見込める係り受けモデルである。文字 n-gram モデルは 1950 年代から論じられているが、先行事象の長さに関する実験は計算機資源の不足から不十分であった。これを十分な先行事象の長さまでについて実際に計算した結果を提示し、その振る舞いについて論じる。これはまた、より複雑なモデルの未知語を扱うモデルという重要な位置を占める。形態素 n-gram モデルは 1970 年代の音声認識においてすでに利用されている。近年になってこれに未知語モデルを付加することが提案されており、この論文では、未知語モデルに一般的な辞書を付加し、モデルの改善を図ることを提案する。これは、コーパス以外の情報を、確率的言語モデルの条件を逸脱することなく利用する例として重要である。このような改善に加えて、形態素をクラスと呼ばれるグループに分類することで、予測精度と記憶領域の双方を改善する方法を提案する。記憶領域の改善としてクラスを用いることはすでに提案されているが、予測精度との両立に成功した例は報告されていない。提案手法を実装し実験した結果、予測精度と記憶領域の改善が観測された。この結果得られた言語モデルを形態素解析に応用した結果、解析精度の有意な向上が観測された。係り受けモデルは、構文解析のためのモデルとして提案されているが、確率的言語モデルの条件を逸脱していない完全な係り受けモデルは、筆者の知る限り、本論文で提案するモデルのみである。本論文では、これについて詳述するとともに、形態素のクラス分類によるモデルの改善を提案する。これを実装し実験した結果、予測力

の有意な向上とともに、応用の一例としての構文解析の精度向上も観測された。前述したように、確率的言語モデルは応用に全く依存しない。また、対象言語に依存する部分も極めて少ない。したがって、本論文で述べる確率的言語モデルおよびその改善手法は、自然言語処理の多岐に渡る応用の精度を向上させる。

係り受けモデルは、文節の属性を終端記号とする確率文脈自由文法であるが、文節の実際の形態素列を記述するモデルとして形態素n-gramモデルを、未知語の文字列を記述するモデルとして文字n-gramモデルを内包している。このように、より複雑な言語現象をモデル化するとしても、完全な言語モデルであるためには、簡単なモデルを内包している必要がある。今後、さらに複雑な言語現象をモデル化し、確率的手法による認識系や解析系の精度向上が望まれる。このためには、世界知識などを確率的なネットワークとして保持しておくことや、照応などの現象に対応するために、文脈に応じてこれを動的に変更することなどが考えられる。係り受けモデルは、このような未知語処理や形態素解析から文脈処理までを一括して扱う確率的言語モデルの一部をなす。また、モデルに可変の部分を設け、本論文で提案しているアイデアを用いて、この部分の具体的な値を推定するという方法でモデルの改善を図ることも重要である。本論文は、このようなより複雑な確率的言語モデルへの進歩の基礎としても意義深い。

謝辞

本研究を進めるにあたり、終始適切な御指導と御教示を賜りました京都大学工学研究科の長尾眞教授に謹んで御礼申し上げます。また、黒橋禎夫助手を始めとして長尾研究室の皆様に感謝致します。

東京大学滞在中や同大学訪問の折などに、示唆に富んだ御意見を頂いた東京大学理学研究科の辻井潤一教授及び鳥沢健太郎助手を始めとして情報科学科辻井研究室の皆様に感謝致します。

日本アイ・ビー・エム株式会社の西村雅史氏、伊東伸泰氏、荻野紫穂氏、山崎一孝氏、金子宏氏、野美山浩氏、丸山宏氏、武田浩一氏、渡辺日出雄氏、那須川哲哉氏、浦本直彦氏には、東京基礎研究所滞在中を含めて多岐にわたる御教示を賜りました。ここに感謝の意を表します。

日本電信電話株式会社の永田昌明氏と徳島大学工学部の北研二助教授と九州大学工学部の富浦洋一助教授と株式会社国際電気通信基礎技術研究所の柏岡秀紀氏には確率的言語モデルの基礎の教授や文献紹介などを通して一方ならぬお世話になりました。ここに謝意を表します。

奈良先端科学技術大学院大学情報科学研究科の松本裕治教授、北陸先端科学技術大学院大学情報科学研究科の佐藤理史教授、筑波大学電子情報工学系の中村裕一講師、奈良先端科学技術大学院大学情報科学研究科の伝康治助教授、奈良先端科学技術大学院大学情報科学研究科の宇津呂武仁助手、東京大学医科学研究所の角田達彦氏及び日本電信電話株式会社の春野雅彦氏には長尾研究室在籍中のみならず様々な機会に貴重な御批判と建設的な御意見を頂きました。ここに謹んで御礼申し上げます。

最後に、研究を含めた日常において御助力を頂いた皆様に心より感謝致します。

もくじ

第 1 章

はじめに

自然言語を研究対象とする言語学は、伝統的に研究者の内省に基づく定性的かつ主観的な議論に終始する傾向があり、近代科学の条件を満たしていなかった。これは、言語現象が複雑であるだけでなく、言語がそれ自身のみでは本質的に普遍的でないことに起因する。このような背景のもと1948年にShannonは情報理論[1)]を確立し、この理論に基づいて自然言語を確率的現象としてとらえる極めて数理的な立場を提案した[2)]。つまり、自然言語をある情報源からの記号列とみなし、言語に対する仮説をその情報源に対する予測力という基準で評価するという方法である。このため、言語に対する仮説は必然的にその言語のアルファベット列を定義域とする確率分布として表現される。このような仮説は、確率的言語モデルと呼ばれる。

この枠組では、言語に対する仮説はアルファベット列から確率値への写像として表現されることだけが条件である。最初のモデルは、Shannonによる連続するn文字(n-gram)の頻度を利用するモデルである[2)]。具体的には、ある自然言語に属する文を大量に集めたコーパス中に、その言語を記述するために用いられる文字がどのように出現するかを観測し、その結果に基づいてモデルのパラメーターを決定する。このとき、注目する文字列の長さnが大きくなる程よい近似となることが示せる。

一方、1956年にChomskyは、自然言語の記述を目的として、3つの形式的な言語モデルを提案した[3)]。これらは、Shannonの立場とは異なり、自然言語を集合としてとらえている。この立場でのある言語の記述

は、その言語のアルファベット列を要素とする集合を、その言語に属するアルファベット列(文)と属さないアルファベット列(非文)とに峻別する。これは、言語のアルファベット列を定義域とし、真偽を値域とする関数とみなすこともできる。このようなモデルは現在、形式言語理論として知られている[4]。これらには、Shannonの確率的言語モデルでは捕らえられなかった離れた要素の関係を記述することができる文脈自由文法や文脈依存文法などが含まれる。さらに、このようなモデルに確率の概念を導入するという拡張が提案されている[5,6]。

本論文では、自然言語を確率的現象としてとらえる立場を踏襲し、様々な確率的言語モデルの予測力の評価と自然言語処理への応用について述べる。確率的言語モデルとしては、確率正規文法(n-gramモデル)と確率文脈自由文法を用いる。文字や形態素や文節を予測単位として、これらの文法を用いた日本語の確率的言語モデルを推定する。さらに、予測単位をクラスと呼ばれるグループに分類し、予測力を向上する方法を提案する。これにより、確率的言語モデルの応用としての自然言語の認識や解析の精度が向上する。応用例として、形態素解析と構文解析について述べる。

第 2 章

確率的言語モデル

自然言語を確率的現象としてとらえると、その記述はアルファベット列と確率値の対応として定義される確率的言語モデルとなる。この章では、まず、確率的言語モデルの定義と評価基準について述べる。次に、確率的言語モデルの応用について述べる。

2.1 定義

確率的言語モデルは、アルファベット列が出現する確率値を記述する。ある言語のアルファベットを$\mathcal{X}$とすると以下のように表される。

$$P : \mathcal{X}^* \mapsto [0,\ 1]$$

確率的モデルであるので、確率値をすべてのアルファベット列に渡って合計すると1以下になる必要がある。

$$\sum_{x \in \mathcal{X}^*} P(x) \leq 1$$

このことから分かるように、確率的言語モデルはデータ圧縮に用いることができる。この事実は、モデルを構築する際に、仮想的に符号器と復号器を考えて、一意に復号できるか否かを考えることでモデルが検証できることを意味する。

2.2 評価基準

確率的言語モデルの良否は、モデルが記述しようとする対象の真の分布にどの程度近いかで測られる。この近さには、クロスエントロピーという尺度を用いるとができる。以下では、まずエントロピーについて説明し、次いでクロスエントロピーについて述べる。

2.2.1 エントロピー

エントロピーは、確率分布 $P(x)$ に対して以下のように定義される。論文を通じて log の底は 2 であるとする。

$$H(P) = \sum -P(x) \log P(x)$$

ここで、和は $P(x)$ の定義域に渡って計算される。エントロピーは、確率分布 $P(x)$ の不確かさを表す。これは、一様分布 $P(x) = \frac{1}{|X|}$ のときに最大となり、ある x に対して $P(x) = 1$ のとき最小値となる。

$$0 \leq H \leq \log |X|$$

エントロピーは、確率分布 $P(x)$ で表現される情報源からの記号を符号化する際の符号長の下限を表す[7]。

2.2.2 クロスエントロピー

クロスエントロピーは、確率分布 $P_1(x)$ と $P_2(x)$ に対して以下のように定義される。

$$\begin{aligned} H(P_1,\, P_2) &= \sum -P_1(x) \log P_2(x) \\ &= \sum -P_1(x) \log P_1(x) + \sum P_1(x) \log \frac{P_1(x)}{P_2(x)} \\ &= H(P_1) + \mathrm{D}(P_1 || P_2) \end{aligned} \tag{2.1}$$

ここで、和は $P(x)$ の定義域に渡って計算される。また、$\mathrm{D}(P_1||P_2)$ は確率分布 P_1 と P_2 の Kullback-Leibler 情報量 (divergence) をあらわす。これ

は次の性質を持つ。

$$\mathrm{D}(P_1||P_2) \geq 0 \quad \text{for } \forall P_1,\ \forall P_2$$

$$\mathrm{D}(P_1||P_2) = 0 \ \Leftrightarrow \ P_1(x) = P_2(x)$$

したがって、クロスエントロピーは、確率分布 $P_1(x)$ と $P_2(x)$ をそれぞれ真の分布とそのモデルとすると、Kullback-Leibler 情報量で測ったモデルの良さを表す。符号化という視点では、このモデルによる符号長の下限と考えることもできる。Kullback-Leibler 情報量は非負なので、式 (2.1) からクロスエントロピーは真の確率分布のエントロピーの上限と考えることもできる。

2.2.3 評価基準

自然言語を確率的な現象としてとらえ、確率的言語モデルを用いてモデル化する立場では、クロスエントロピーをモデルの評価基準とすることが自然である。しかし、クロスエントロピーの計算に必要な真の分布こそが究極的な目標であり、決して知ることはできない。この理由から、あるコーパスの文字列の分布で真の分布を近似し、クロスエントロピーの計算に用いる。このためのコーパスはテストコーパスと呼ばれる。これを $L_{test} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ とすると、テストコーパスにおける文の最尤推定による確率分布は以下のようになる[1]。

$$P_{test}(\boldsymbol{x}_i) \stackrel{MLE}{=} \frac{1}{n}$$

ここで関係 $\stackrel{MLE}{=}$ は、右辺と左辺が等しいのではなく、右辺は左辺の最尤推定の結果であることを示す。確率的言語モデルを $M(\boldsymbol{x})$ とすると、このモデルによるテストコーパスのクロスエントロピーは以下のようになる。

$$H_s(P_{test},\ M) \ = \ \sum_{i=1}^{n} -P_{test}(\boldsymbol{x}_i) \log M(\boldsymbol{x}_i)$$

[1]テストコーパスに同一の文が存在しないことを仮定しているが、同一の文が存在する場合もクロスエントロピーの計算式 (2.2) は同一である。

$$= -\frac{1}{n}\sum_{i=1}^{n}\log M(\boldsymbol{x}_i) \tag{2.2}$$

これは文を予測単位とした場合のクロスエントロピーである。複数の確率的言語モデルの比較に用いるテストコーパスは共通であることが望ましいが、これが不可能な場合には、文の長さの影響を排除した文字あたりのクロスエントロピーを評価基準とする。

$$H_c(P_{test},\ M) = \frac{1}{\sum_{i=1}^{n}|\boldsymbol{x}_i|}\sum_{i=1}^{n}-\log M(\boldsymbol{x}_i)$$

本論文では、確率的言語モデルの予測力の評価基準として、この文字あたりのクロスエントロピーを用いる。

音声認識などの研究では、単語あたりのクロスエントロピーから一意に換算されるパープレキシティー[8]を評価基準とすることが多い。

$$\begin{aligned} PP_w &= 2^{H_w} \\ H_w(P_{test},\ M) &= \frac{1}{\sum_{i=1}^{n}|\boldsymbol{x}_i|_w}\sum_{i=1}^{n}-\log M(\boldsymbol{x}_i) \end{aligned}$$

ここで、$|\boldsymbol{x}|_w$ は文字列 $\boldsymbol{x}$ に含まれる単語数を表す。パープレキシティーは、各単語が等確率に選ばれると仮定した場合の後続可能単語数の幾何平均を表している。

確率的言語モデルが複雑になると、文字列の生成方法が複数になる可能性がある。これらの集合を T とすると、このようなモデルでの文字列の生成確率 $M(\boldsymbol{x})$ は、以下のように全ての生成方法に渡る確率の和となる。

$$M(\boldsymbol{x}) = \sum_{t\in T} M(\boldsymbol{x},\ t)$$

クロスエントロピーの計算には、この確率値を用いることが理想的であるが、確率的言語モデルが複雑になると、生成方法は組み合わせ的に増加するので、特に長い文に対して、この和のための計算量は膨大となることがある。この問題を避けるため以下の近似値を用いる。

- Viterbi 近似：生成確率が最大となる生成方法による確率値

$$M_V(\boldsymbol{x}) = \max_{t\in T} M(\boldsymbol{x},\ t)$$

- コーパス近似：コーパスに付与された生成方法による確率値

$$M_C(\boldsymbol{x}) = M(\boldsymbol{x},\, t_{test})$$

定義から明らかなように、これらの間には以下の関係が成り立つ。

$$M_C(\boldsymbol{x}) \leq M_V(\boldsymbol{x}) \leq M(\boldsymbol{x})$$

したがって、式(2.2)のクロスエントロピーの大小関係は以下のようになる。

$$H(P_{test},\, M_C) \geq H(P_{test},\, M_V) \geq H(P_{test},\, M)$$

よって、これらの近似の結果得られるクロスエントロピーは、真の確率分布のエントロピーの上限の推定値としての意味を失わない。

2.3 簡単な言語モデル

この節では、確率的言語モデルの例として文字0-gramモデルとその改善である文字1-gramモデルについて説明し、確率的言語モデルの構成から評価までを概観する。

2.3.1 文字0-gramモデル

最も簡単な確率的モデルは各事象に対して等確率を与えるモデルであろう。自然言語の場合、事象はアルファベットのクリーネ閉包であり、定義域の要素数は無限であるので、等確率を与えるモデルを構成することはできない。次に簡単なモデルとして、各アルファベットを等確率で生成するモデルが考えられる。

$$M_{x,0}(x_1,\, x_2,\, \ldots,\, x_n) = \left(\frac{1}{|\mathcal{X}|+1}\right)^{n+1}$$

ここで、指数部の+1は文末を表す記号の生成による。これを導入することによって、すべての可能な記号列に対する確率の和が1となることが保証される。データ圧縮という観点では、2つの文とその連接に等しい

文とを区別するために必要である。より形式的には、日本語のアルファベットに含まれない記号(BT)を文末に付加した記号列を予測しているといえる。分母の+1は、生成可能な記号がアルファベットかまたはこの文区切り記号であることによる。データ圧縮の観点から、以下では文の文字数には文末の区切り記号も含める。

例として、次の日本語の文の生成確率を計算してみる。

今日、京都大学に行く。BT

日本語のアルファベットを、我々の計算機環境で表示可能であった全角文字とする。したがって $|\mathcal{X}| = 6878$ であり、この文の文字数は11なので、この文の文字1-gramモデルによる生成確率は以下のようになる。

$$M_{x,0}(\text{今日、京都大学に行く。BT}) = \left(\frac{1}{6878+1}\right)^{11+1}$$

この文をテストコーパスとすると、このモデルによる文字あたりのクロスエントロピーは以下のようになる。

$$\begin{aligned} H_c(L_{test},\, M_{x,0}) &= -\frac{1}{12}\log M_0(\text{今日、京都大学に行く。BT}) \\ &= -\log\frac{1}{6878+1} \\ &= 12.7480 \end{aligned}$$

したがって、日本語の文字あたりのエントロピーの上限は12.7480ビットと推定される。

2.3.2 文字1-gramモデル

上述した確率的言語モデルは、全てのアルファベットに一様な出現確率を与えるという単純なモデルである。言語のアルファベットの出現確率には偏りがあるので、このモデルには改善の余地がある。この偏りをとらえる最も単純なモデルである文字1-gramモデルでは、各アルファベットの出現確率を文字に依存させる。この確率値を内省によって与えることも不可能ではないが、これが高い予測力を持つモデルとなるとは

考えられない。そこで、テストコーパスと異なるが、確率的に類似した振る舞いをすると考えられるコーパスから、アルファベットの出現確率を最尤推定することが考えられる。このようなコーパスは学習コーパスと呼ばれる。

$$M_1(x) \stackrel{MLE}{=} \frac{N(x)}{N()}$$

ここで$N(\boldsymbol{x})$は文字列$\boldsymbol{x}$の学習コーパスでの頻度であり、以下の性質がある。

$$\sum_{x \in \mathcal{X}} N(\boldsymbol{x}x) = N(\boldsymbol{x})$$

学習コーパスの各文の末尾には、文区切り記号が付加されているとする。後述する実験に用いたコーパスから推定した結果、たとえば$M_1(\text{に}) = \frac{182858}{7439580}$であった。このようにして前述の例文の出現確率を計算し、クロスエントロピーを計算すると次のようになった。

$$H_c(L_{test},\ M_{x,1}) = 7.9623$$

これを文字0-gramモデルのクロスエントロピーと比較すると、このモデルのクロスエントロピーの方が低いので、このテストコーパスと学習コーパスに対しては、文字1-gramモデルは文字0-gramモデルよりも優れていると結論できる。また、日本語の文字あたりのエントロピーの上限の推定値は、新たに7.9623ビットであると結論できる。

このように、確率的言語モデルの構成から評価までは以下の段階からなる。

1. 確率的モデルのクラスの設定

2. 学習コーパスからのパラメータ推定

3. テストコーパスに対するクロスエントロピーの計算

ここで、注意しなければならないのは以下の2点である。

- 確率的モデルの条件を満たしていること

- クロスエントロピーの計算までテストコーパスを参照しないこと

確率的モデルの条件を逸脱すれば、際限なくクロスエントロピーを下げられる。つまり、あるモデル$M(\boldsymbol{x})$に対して$M'(\boldsymbol{x}) = 2M(\boldsymbol{x})$となるモデル$M'$のクロスエントロピーは、1ビット小さくなる。また、モデルの構成のときやパラメータの推定のときにテストコーパスを参照すれば、容易にクロスエントロピーを下げられる。文字1-gramモデルをテストコーパスから推定すると、テストコーパスに出現する文字の確率が$\frac{1}{12}$となりクロスエントロピーは$\log 12 = 3.5850$となり、前述の文字1-gramモデルのエントロピーを大きく下回る。また、モデルクラスを変更し$M(L_{test}) = 1$とすると、$H(L_{test},\ M) = 0$となる。このように、上述の条件を満たさなければ、クロスエントロピーによるモデルの評価は意味をなさなくなる。

確率的モデルの条件ではないが、以下の式を満たすことは実用上重要な性質である。

$$M(\forall \boldsymbol{x}) > 0 \tag{2.3}$$

これにより、いかなる文を含むテストコーパスに対してもクロスエントロピーは有限の値となり、以下で述べる応用における頑強性が保証される。つまり、認識系では全ての文字列を認識できるということが保証され、解析系では任意の入力に対して解析結果を出力できることが保証される。前述の文字1-gramモデルでは、学習コーパスに出現しないアルファベットの出現確率が0となり、それを含むテストコーパスの出現確率が0となる。したがって、これを確率的言語モデルとして持つ認識系ではそのような文字は決して認識されない。また、解析系では、入力文にそのような文字を含むと、全ての解の候補の生成確率が0になるので、比較する意味がなくなる。式(2.3)で表される性質は、このような問題が生じないことを保証する。頑強性を保証するための一つの方法として、フロアリングがある[9]。これは、学習コーパスでの頻度をかさ上げすることであり、文字1-gramモデルでは、以下の式ようになる。

$$M_1(x) = \frac{N(x) + \alpha}{N() + (|\mathcal{X}| + 1)\alpha}$$

ここでαの具体的な値は、実装上の都合から最小正数とする$(\alpha = 1)$のが一般的である。

2.4 応用

自然言語の確率的モデルの応用は、テキスト圧縮と自然言語認識と自然言語解析に大別できる。この節では、これらについて述べる。

2.4.1 テキスト圧縮

エントロピーとクロスエントロピーは、テキスト圧縮という視点からも重要な値である。エントロピーは、この情報源からのアルファベット列を一意に復号できる符合で記述するために必要な平均ビット数の期待値の下限を与える[7]。これを式で表わすと以下のようになる。

$$\lim_{n\to\infty} \frac{1}{n} \mathrm{E}_P l(X_1, X_2, \ldots, X_n) \geq H(P)$$

ここで、$l(X_1, X_2, \ldots, X_n)$はアルファベット列$X_1, X_2, \ldots, X_n$に対応する符合のビット数を表わす。同様に、情報源PをモデルMでモデル化した場合、クロスエントロピーは、情報源Pからのアルファベット列をモデルMを用いて一意に復号できる符合で記述するために必要な平均ビット数の期待値の下限を与える[7]。これを式で表わすと以下のようになる。

$$\lim_{n\to\infty} \frac{1}{n} \mathrm{E}_P l_M(X_1, X_2, \ldots, X_n) \geq H(P, M)$$

等号が成り立たないのは、実際に符号化する際に整数個の符号を用いなければならないという制約による。しかし、算術符号[7, 10]を用いれば上式の等号が成り立つことが示される。このように、テキスト圧縮は確率的言語モデルの最も直接的な応用である。

2.4.2 認識系

認識系とは、音響特徴量などの言語の文字列に対応する信号を言語の文字列に変換する関数である。以下では、例として音声認識について説

表 2.1: 確率的言語モデルの応用(認識系)

認識系	入力
音声認識	音響特徴量
文字認識	画像特徴量
仮名漢字変換	平仮名列
誤り訂正	文字列
⋮	⋮

明するが、他の認識系でも同様の議論が成立し、異なるのは入力の信号のみである(表2.1参照)。

確率的モデルによる音声認識では、音響特徴量を入力とし、言語の文字列を出力する。このとき、複数ある解の中から尤らしい文字列を出力する。この尤らしさを与えるのは、音響特徴量 s が与えられたときの文字列 x の条件付き確率である。したがって、出力はこれを最大にする文字列 $\hat{x}$ であり、以下の式で与えられる。

$$\begin{aligned}\hat{x} &= \operatorname*{argmax}_{x\in\mathcal{X}^*} P(x|s) \\ &= \operatorname*{argmax}_{x\in\mathcal{X}^*} \frac{P(s|x)P(x)}{P(s)} \quad (\because \text{ベイズの公式}) \\ &= \operatorname*{argmax}_{x\in\mathcal{X}^*} P(s|x)P(x) \quad (\because P(s)\text{は}x\text{によらない})\end{aligned}$$

この式から、音声認識のためのモデルは、確率的音響モデル $P(s|x)$ と確率的言語モデル $P(x)$ に分割されることが分かる。本論文で問題にするのは、後者の確率的言語モデルである。この応用における言語モデルの良否は、音響モデルを一定にした上で認識精度を比較する。この式から分かるように、認識の問題における確率値は、絶対的な値ではなく複数の候補の比較という相対的な値が問題となる。さらに、認識の精度は正解文字列と認識結果との比較に基づいて算出されるので、確率的言語モデルの予測力と認識系の精度との関係は、解析的に導出できるような確固とした関係ではない。実験的に得られた関係として、西村ら[11]は相関

表 2.2: 確率的言語モデルの応用(解析系)

解析系	出力
形態素解析	形態素列
構文解析	構文木
読み付与	読み
⋮	⋮

係数0.6を報告している。

2.4.3 解析系

解析系とは、言語の文字列を入力として、それに構造を付与する関数である。以下では、例として形態素解析について説明するが、他の解析系で異なるのは出力として付与される情報のみである(表2.2参照)。

確率的モデルによる形態素解析では、言語の文字列を入力とし、形態素$\mathcal{M}$の列を出力する。認識系と同様に、このとき複数ある解の中から尤らしい文字列を出力する。この尤らしさを与えるのは、文字列$\boldsymbol{x}$が与えられたときの形態素列$\boldsymbol{m} \in \mathcal{M}^*$の条件付き確率である。したがって、出力はこれを最大にする形態素列$\hat{\boldsymbol{m}}$であり、以下の式で与えられる。

$$\begin{aligned}
\hat{\boldsymbol{m}} &= \operatorname*{argmax}_{\boldsymbol{m}\in\mathcal{M}^*} P(\boldsymbol{m}|\boldsymbol{x}) \\
&= \operatorname*{argmax}_{\boldsymbol{m}\in\mathcal{M}^*} P(\boldsymbol{m}|\boldsymbol{x})P(\boldsymbol{x}) \quad (\because P(\boldsymbol{x})\text{は}\boldsymbol{m}\text{によらない}) \\
&= \operatorname*{argmax}_{\boldsymbol{m}\in\mathcal{M}^*} P(\boldsymbol{x}|\boldsymbol{m})P(\boldsymbol{m}) \quad (\because \text{ベイズの公式})
\end{aligned}$$

この式により、形態素解析のためのモデルは、形態素列を条件として文字列の確率を記述する部分($P(\boldsymbol{x}|\boldsymbol{m})$)と形態素列に対する確率的言語モデル($P(\boldsymbol{m})$)に分割されることが分かる。誤りのある文の文字訂正と同時に形態素解析を行う場合を除いて形態素列は文字列に一意に変換できるので、一般の形態素解析では$P(\boldsymbol{x}|\boldsymbol{m}) = 1$である。言語モデル$P(\boldsymbol{m})$は、形態素という概念を内包する必要がある。解析系における言語モデ

ルの予測力と認識系の精度との関係については、前項の認識系と全く同じ議論が展開される。我々の知る限り、解析系における精度と言語モデルの予測力の相関の報告は少ない。関根[12]は確率的構文解析の報告として精度に加えて予測力を示している。本論文の第6章および第8章では、それぞれ形態素解析と構文解析の精度と予測力を示している。これらの結果から予測力がより高い確率的言語モデルを用いることで、解析精度の改善が見込めることが実験的に示される。

2.5 結論

この章では、確率的言語モデルの定義を与え、その評価基準としてのクロスエントロピーについて説明した。確率的言語モデルの課題は、テストコーパスを参照することなく、テストコーパスのクロスエントロピーがより小さくなるモデルを推定することである。このようにして構成されたモデルは、自然言語を対象とする様々な応用の共通の部分であり、これを改善することで、これらの応用の全てに渡る精度向上が期待できる。

第 3 章

文字単位のモデル

この章では、前章で説明した文字を単位とするモデルの改善として、文字2-gramモデルを説明する。次にこれを題材として、確率的言語モデルのパラメータ推定における一般的な手法を説明する。さらに、より一般的なモデルとして文字n-gramモデルを説明する。

3.1 文字2-gramモデル

前章のモデルでは、各時点での文字の確率分布が先行する文字列に対して独立であった。文字の出現は直前の文字に影響されることが分かっているので、この現象をモデル化することで予測精度がより高いモデルが得られる。これは、各時点での文字の出現確率を、先行する文字列(履歴)の条件付き確率とすることで実現できる。この節で述べるモデルは、先行する文字列を直前の1文字で代表する。このモデルによる文$\boldsymbol{x}\cdot\mathrm{BT} = x_1x_2\cdots x_{l+1}$の出現確率は以下の式で表される。

$$P(x_1x_2\cdots x_{l+1}) = \prod_{i=1}^{l+1} P(x_i|x_{i-1}) \tag{3.1}$$

ここでx_0は文頭に対応する特別な記号であり、これを導入することによって式が簡便になる。

式(3.1)の$P(x_i|x_{i-1})$の値は実際の文を大量に集めたコーパスにおける

文字列の頻度から最尤推定される。

$$P(x_i|x_{i-1}) \stackrel{MLE}{=} \frac{N(x_{i-1}x_i)}{N(x_{i-1})}$$

このように、コーパスにおける文字2-gramの頻度に基づいているので、文字2-gramモデルと呼ばれる。文字が状態に対応すると考えると、これは単純マルコフモデルと等価である。また、正規言語の各生成規則に確率値を付加した確率正規言語とみなすこともできる。実際にこの式からなるモデルを推定したとすると、以下の二つの問題を生じる可能性がある。

1. コーパスに現れない1-gramがあれば、分母が0になる。

2. コーパスに現れない2-gramがあれば、確率値が0になる。

これらの問題はアルファベットの数が大きい日本語などの場合には必ず生じると言ってよい。これらの問題に対処するためには、以下のように前章で説明したフロアリングを用いることができる。

$$P(x_i|x_{i-1}) = \frac{N(x_{i-1}x_i) + \alpha}{N(x_{i-1}) + (|\mathcal{X}| + 1)\alpha}$$

この方法は、アルファベットが有限集合であることを利用しており、モデルの単位を単語などの無限集合に変更した場合に適用できない。ここでは、より汎用性のある以下の二つの方法を用いる。

1. 未知文字の導入

2. 文字1-gramとの補間

以下では、これらについて説明する。

3.1.1 未知文字の導入

学習コーパスに出現しない文字がテストコーパスに存在するという問題に対処する基本的なアイデアは、アルファベットを互いに素な既知文

字集合と未知文字集合に分割し、既知文字部分と未知文字部分の二段のモデルを用いることである。つまり、未知文字はこれを表す一つの記号(未知文字記号)で代表させて、文字2-gramモデルの段階では既知文字とこの記号までを予測する。未知文字が現れた場合は、まず未知文字記号を予測し、実際の未知文字は別のモジュール(未知文字モデル)を用いて予測する。既知文字集合を$\mathcal{X}_k$とし、未知文字集合を$\mathcal{X}_u$とすると、ここで説明したモデル$M_{x,2}$は以下の式で表される。

$$M_{x,2}(x_i|x_{i-1}) = \begin{cases} P(x_i|x_{i-1}) & \text{if} \quad x_i \in \mathcal{X}_k \wedge x_{i-1} \in \mathcal{X}_k \\ P(x_i|\mathtt{UX}) & \text{if} \quad x_i \in \mathcal{X}_k \wedge x_{i-1} \in \mathcal{X}_u \\ P(\mathtt{UX}|x_{i-1})M_{ux}(x_i) & \text{if} \quad x_i \in \mathcal{X}_u \wedge x_{i-1} \in \mathcal{X}_k \\ P(\mathtt{UX}|\mathtt{UX})M_{ux}(x_i) & \text{if} \quad x_i \in \mathcal{X}_u \wedge x_{i-1} \in \mathcal{X}_u \end{cases} \tag{3.2}$$

ここで、UXは未知文字記号であり、M_{ux}は未知文字モデルである。右辺の第一因子は、コーパスにおける文字列の頻度を、未知文字を未知文字記号に置き換えてから計数し、最尤推定する。この式から未知文字モデルを除くと、既知文字集合に未知文字記号を加えた集合$(\mathcal{X}_k \cup \{\mathtt{UX}\})$をアルファベットとする言語の確率的モデルとなっていることが分かる。したがって、未知文字モデルが確率的であれば、これを未知文字記号に代入することで得られる全体としてのモデルも確率的言語モデルとなる。

前述の分母が0になる問題を避けるためには、既知文字集合を学習コーパスに現れる文字集合の真部分集合とすることが必要かつ十分である。既知文字集合を学習コーパスに現れる文字集合とすると、未知文字記号の頻度が0になり、テストコーパスに未知文字が出現する場合に分母が0になることに注意しなければならない。

未知文字モデルは、確率的であることが唯一の条件である。未知文字の出現に関する情報はないので、以下の式のように等確率分布とした。

$$M_{ux}(x) = \frac{1}{|\mathcal{X}_u|} \tag{3.3}$$

3.1.2 文字1-gramとの補間

学習コーパスに出現しない文字2-gramがテストコーパスに存在すると、テストコーパスの生成確率が0になり、クロスエントロピーが無限大になるという問題には、補間と呼ばれる方法で対処する[13]。これは、一般には、異なる複数のモデルによる確率を一定の割合で混合することを言う。ここでは、以下の式のように、文字2-gramモデルを文字1-gramと補間する。

$$\begin{aligned} &M'_{x,2}(x_i|x_{i-1}) = \lambda_1 M_{x,1}(x_i) + \lambda_2 M_{x,2}(x_i|x_{i-1}) \\ &\text{ただし } 0 \leq \lambda_j \leq 1 \ (j=1,2) \text{ かつ } \lambda_1 + \lambda_2 = 1 \end{aligned} \tag{3.4}$$

$$M_{x,1}(x_i) = \begin{cases} P(x_i) & \text{if} \quad x_i \in \mathcal{X}_k \\ P(\mathtt{UX}) M_{ux}(x_i) & \text{if} \quad x_i \in \mathcal{X}_u \end{cases} \tag{3.5}$$

文字1-gramは、学習コーパス中の文字の頻度から最尤推定される。したがって、文字2-gramと同様に未知文字モデルをもつ文字1-gramモデルは、全ての既知文字と未知文字記号に対する出現頻度が正数となり、これらの生成確率が正数となる。結果として、補間を行った後のモデルは、$\lambda_1 > 0$である限り、全ての既知文字と未知文字記号に対する生成確率が0より大きくなり、このモジュールでのクロスエントロピーは有限の値を取る。

上述の補間は、対象とする事象の頻度が低い場合には、推定値の信頼性が低くなるという最尤推定の問題に対する対策にもなっている。つまり、一般的に同じコーパスにおいてはn-gramの頻度よりも$(n-1)$-gramの頻度の方が大きいので、最尤推定の結果得られる値は2-gramモデルよりも1-gramモデルの方が信頼性が高い。したがって、文字2-gramモデルを文字1-gramと補間することは、文字2-gramモデルの信頼性が低い場合に対処しているといえる。

残された問題は、λ_1とλ_2の決定方法である。これらの値に対するクロスエントロピーの値は、広義の下に凸なグラフとなる。実際、後に詳述する実験で用いた文字2-gramモデルと文字1-gramモデルの補間によるテストコーパスのクロスエントロピーを補間係数の値(λ_2)に対して計

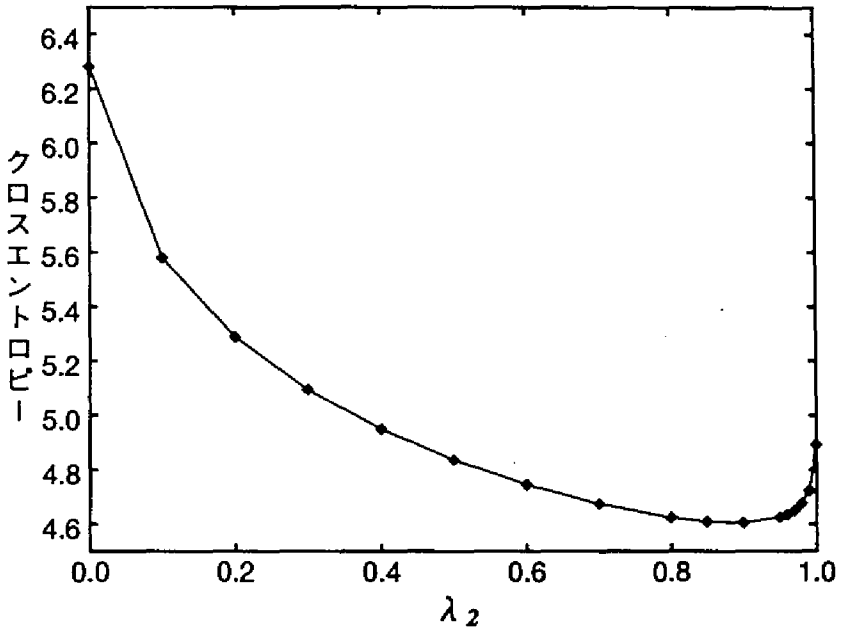

図 3.1: 補間係数とクロスエントロピーの関係

算すると図3.1のようになった。ここでの問題は、このグラフにおいて最小値を与える λ_1 と λ_2 を、テストコーパスを参照することなく求めることである。これを次の節で説明する。

3.2 補間係数の推定

補間係数は、EMアルゴリズム[14]を用いて、あるコーパスの出現確率が最大となるように推定される。テストデータに対して最適となる補間係数に近い値を得るためには、このコーパスの性質がテストデータの性質に近いことが必要である。このようなデータの選び方に関して、Held-out法とその改良である削除補間法がある。以下では、これらについて説明する。

3.2.1 Held-out 法

Held-out 法では、学習コーパスを文字 1-gram と文字 2-gram の計数用と補間係数の推定用に分割することで、テストデータを模擬する。補間係数の推定用のデータは Held-out データと呼ばれ、学習コーパスの残りの部分から推定された確率モデルにとってのテストデータとみなすことができる。つまり、まず学習コーパスの残りの部分から文字 1-gram モデルと文字 2-gram モデルを推定し、次にこの Held-out データ $L_{held\text{-}out}$ の生成確率が最大になる補間係数の値を求める。

$$(\lambda_1,\ \lambda_2) = \underset{(\lambda_1,\lambda_2)}{\mathbf{argmax}}\, P(L_{held\text{-}out})$$

こうして得られた補間係数を用いれば、テストコーパスのクロスエントロピーが最小に近い値となることが期待される。Held-out データの生成確率が最大になる補間係数の値は、適当な初期値から以下に示す式を用いた繰り返しのアルゴリズムにより用いて求められる。

$$\lambda_1' = \frac{1}{N}\sum_{j=1}^{|L_{held\text{-}out}|}\sum_{i=1}^{|\boldsymbol{x}_j|}\frac{\lambda_1 P(x_i)}{\lambda_1 P(x_i)+\lambda_2 P(x_i|x_{i-1})} \tag{3.6}$$

$$\lambda_2' = \frac{1}{N}\sum_{j=1}^{|L_{held\text{-}out}|}\sum_{i=1}^{|\boldsymbol{x}_j|}\frac{\lambda_2 P(x_i|x_{i-1})}{\lambda_1 P(x_i)+\lambda_2 P(x_i|x_{i-1})} \tag{3.7}$$

これは、EM アルゴリズムの特殊な場合であるが、一般的な EM アルゴリズムとは異なり、初期値に関係なく最適値に収束する。このことは、図 3.1 からも直感的に分かる。

3.2.2 削除補間法

補間係数を求めるためのより優れた方法である削除補間法では、まず学習コーパス L を m 個の互いに素な部分 $L_1, L_2, \ldots, L_m$ に分割する。そうしておいて、文字列の頻度の計数を L_i を除いた学習コーパスに対して行ない、L_i を用いて補間係数を推定するということを i を変えて m 通り行ない、それぞれの補間係数の平均値を実際の補間係数とする。補間係数の推定には Held-out 法と同様に EM アルゴリズムを用いる。平均を取

る操作はEMアルゴリズムの繰り返し毎に適用される。このようにして得られた補間係数と、学習コーパス全てに対して再推定した文字1-gramモデルと文字2-gramモデルを最終的なモデルとする。

この方法がHeld-out法に対して優位な点は、最終的なモデルに組み込まれる文字1-gramモデルと文字2-gramモデルが、学習コーパスの全てを用いて推定されているので、結果的にHeld-out法よりも大きな学習コーパスから推定されたモデルとなる点である。この理由から、本論文では一貫して削除補間法を前提とする。

削除補間法の考え方を応用して、既知文字集合が自然に定義できる。削除補間法では、文字列の頻度の計数の対象としてL_iを除いたコーパスと、Held-outデータとしてL_iを用いる。補間係数の推定の際、式(3.6)(3.7)の分母が0にならないためには、Held-outデータに出現する全ての既知文字の頻度が0より大きいことが必要である。この条件は、以下のように表すことができる。ただし、$\mathcal{X}_i$はL_iに含まれる文字を表す。

$$\mathcal{X}_k \cap \mathcal{X}_i \subseteq \mathcal{X}_k \cap \bigcup_{j \neq i} \mathcal{X}_k$$

これが全ての$i\,(1 \leq i \leq m)$に対して成り立つ必要がある。これを満たす最大の文字集合はm個の部分学習コーパスの2つ以上に出現する文字の集合である。したがって、このような文字集合の部分集合を既知文字集合とする。

以上に説明した削除補間法において、学習コーパスをいくつに分割するか(分割数)が不定である。この影響を調べるために、分割数を変えてクロスエントロピーを計算した。詳しい実験の条件は後に詳述するが、既知文字を分割数3の場合に部分学習コーパスの2つ以上に出現する文字の集合に固定している点が異なる。分割数を3から243に指数関数的に変化させて得られた結果を表3.1に掲げた。この結果、分割数が最大の243のときに、補間係数はテストコーパスで最適化した値に最も近かった。この理由は、このときが補間係数の推定のための文字列頻度の計数の対象となるコーパスの大きさと、最終的に文字列頻度の計数の対象となる学習コーパス全体の大きさが最も近くなることであろう。しかしな

表3.1: 削除補間における学習コーパスの分割数とクロスエントロピーの関係

分割数	比率	λ_1	λ_2	H
3	0.6667	0.0326	0.9674	5.4110
9	0.8889	0.0277	0.9723	5.4105
27	0.9630	0.0264	0.9736	5.4105
81	0.9877	0.0260	0.9740	5.4104
243	0.9959	0.0259	0.9741	5.4104
最適値	1.0000	0.0257	0.9743	5.4104

$$\text{比率} = \frac{\text{分割数} - 1}{\text{分割数}}$$

がら、クロスエントロピーの差異はかなり小さい。このことは図3.1の曲線の最小値付近の形状からも了解される。一方、補間係数の推定のための計算時間と記憶容量はともに分割数に比例する。これらの点から、分割数は10程度あれば十分であると結論する。

既知文字集合の選択方法とクロスエントロピーの関係についても実験を行った。一般に、未知文字に対する予測方法は、既知文字のそれよりも学習コーパスの性質を反映していないので、テストコーパスに近いと考えられる学習コーパスを用いている限り、テストコーパスの未知文字は少ない方がよいと考えられる。これを表す尺度として、テストコーパスの既知文字の割合をのべで計算したカバー率と呼ばれる値を用いる。既知文字集合の選択方法としてはこのカバー率が問題となる。図3.2は、上述した既知文字集合(このカバー率は99.99%)の文字をカバー率が約5%刻みに変化するように既知文字を学習コーパスにおける頻度の降順に選択し、それぞれのテストコーパスのクロスエントロピーを計算した結果である。このことから、カバー率の上昇はクロスエントロピーを一次関数的に減少させることが分かる。また、削除補間法の考え方を応用した

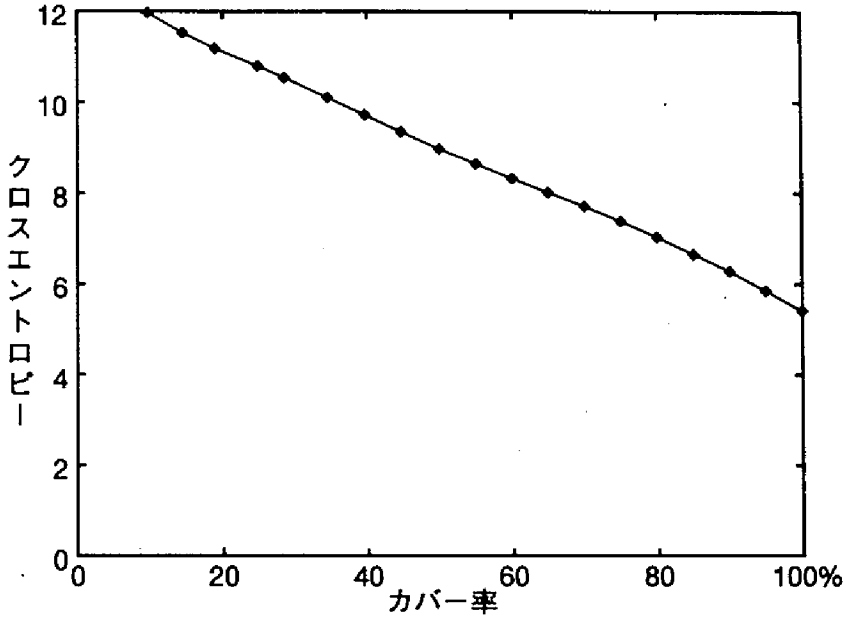

図 3.2: カバー率とクロスエントロピーの関係

自然な既知文字集合の定義が十分に良好であることも分かる。

3.3 文字 n-gram モデル

上述した文字 2-gram モデルの自然な一般化の一つである文字 n-gram モデルでは、履歴を直前の $(n-1)$ 文字で代表する。これは $(n-1)$ 重マルコフモデルと等価であり、以下の式で表される。

$$P(x_1 x_2 \cdots x_{l+1}) = \prod_{i=1}^{l+1} M_{x,n}(x_i | x_{i-k} \cdots x_{i-2} x_{i-1}) \tag{3.8}$$

ここで x_i $(i \leq 0)$ は、式を簡便にするために便宜的に導入された文頭に対応する特別な記号である。また、$M_{x,n}$ はアルファベット $\mathcal{X}_k \cup \{\mathtt{UX}\}$ 上の n-gram モデルであり、未知文字記号 UX を予測するときには、未知文字モデル M_{ux} による確率も含むとする (式 3.2参照)。

文字 2-gram モデルの場合と同様に、式 (3.8) の $P(x_i | x_{i-k} \cdots x_{i-2} x_{i-1})$ の

値は実際の文を大量に集めたコーパスにおける文字列の頻度から最尤推定する。未知文字は未知文字記号に置き換えられていることを仮定している。

$$M_{x,n}(x_i|x_{i-k}\cdots x_{i-2}x_{i-1}) \stackrel{MLE}{=} \frac{N(x_{i-k}\cdots x_{i-2}x_{i-1}x_i)}{N(x_{i-k}\cdots x_{i-2}x_{i-1})}$$

このように、このモデルはコーパスにおける$n=k+1$個の記号列の頻度統計の結果に基づくのでn-gramモデルと呼ばれる。低頻度事象に対する推定値の信頼性の問題に対処するため、より信頼性が高いことが期待される、より低次のn-gramモデルと補間することができる。これは、次の式で表される。

$$M'_{x,n}(x_i|x_{i-k}\cdots x_{i-2}x_{i-1}) = \sum_{j=1}^{n} \lambda_j M_{x,j}(x_i|x_{i-j+1}\cdots x_{i-2}x_{i-1})$$

$$\text{ただし } 1 \le \lambda_j \le 1\ (1 \le j \le n) \text{ かつ } \sum_{j=1}^{n} \lambda_j = 1$$

ここで、$j=1$のときは$P(x_i|x_{i-j+1}\cdots x_{i-2}x_{i-1}) = P(x_i)$であるとする。補間係数は状態の関数とすることも可能である。本論文では、先行文字列の学習コーパスにおける頻度が0の場合と1以上場合で補間係数を以下のように区別した。

$$M'_{x,n}(x_i|x_{i-k}\cdots x_{i-2}x_{i-1}) = \sum_{j=1}^{h+1} \lambda_j^{h+1} M_{x,j}(x_i|x_{i-j+1}\cdots x_{i-2}x_{i-1}) \qquad (3.9)$$

ただし、$h<n$はそれぞれの先行事象について学習コーパスにおける頻度が1以上となる最長の先行文字数である。

$$N(x_{i-h}\cdots x_{i-2}x_{i-1}) > 0 \ \wedge \ N(x_{i-h-1}\cdots x_{i-2}x_{i-1}) = 0$$

以上のようにすることで、式(3.9)の右辺の確率の推定値が不定となる場合を参照することを避けられる。このとき、文字n-gramモデルの補間係数の数は$1+2+\cdots+(n-1)$となる。補間係数の値の求め方は、文字2-gramと同じように、状態頻度の計数に用いたコーパスとは別のコーパスの出現確率が最大になるように決定する。

表 3.2: 実験に用いたコーパス (文字 n-gram モデル)

用途	文数	文字数	文字数 / 文数
学習	187,022	7,252,558	38.78
評価	20,780	802,576	38.62

3.4 評価

以上で説明した文字 n-gram モデルを実装し、この評価を行なうために以下の項目を調べる実験を行なった。

- 学習コーパスの大きさとクロスエントロピーの関係
- 先行事象の長さ (n の値) とクロスエントロピーの関係

以下では、これらの結果を提示し、考察を加える。

3.4.1 実験の条件

式(3.3)(3.9)に基づいて未知文字モデルをもつ文字 n-gram を構成した。補間係数の推定の繰り返し計算は、小数点以下 5 桁が不変となることを停止条件として行なった。削除補間における学習コーパスの分割数は 9 である。既知文字集合は、これら 9 個の部分学習コーパスの 2 個以上に出現する文字の集合である。

実験には EDR コーパス[15] を用いた。まず、これを 10 個の部分コーパスに分割し、このうちの 9 個を学習コーパスとし、残りの 1 個をテストコーパスとした。表 3.2はコーパスの大きさである。文字数は文区切り記号を含んでいない。

3.4.2 学習コーパスの大きさとクロスエントロピーの関係

図 3.3は、文字 n-gram モデル ($n = 1, 2, 3, 4$) における学習コーパスの大きさ (常用対数値) とクロスエントロピーの関係である。グラフから分

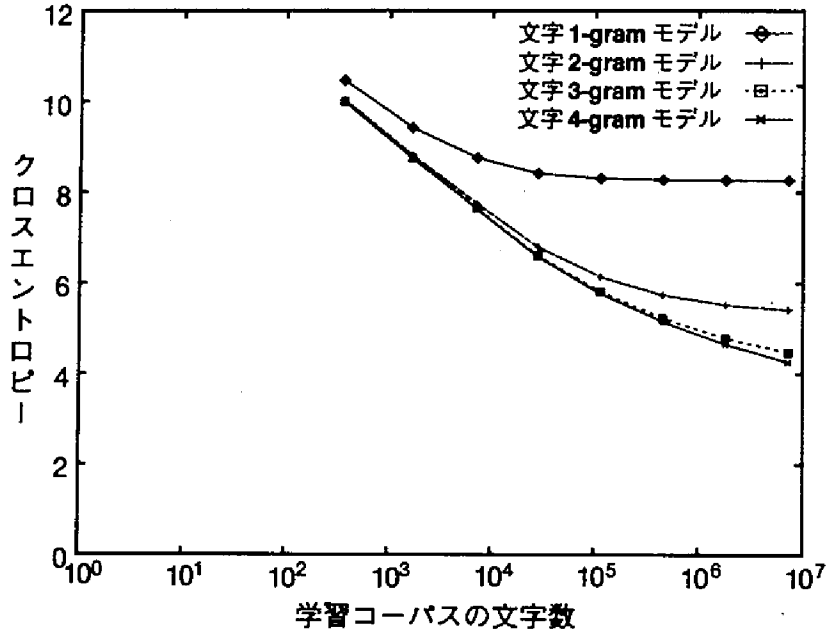

図 3.3: 学習コーパスの大きさとクロスエントロピーの関係 (文字 n-gram モデル)

かるように、文字 1-gram モデルのクロスエントロピーは、学習コーパスの文字数が 10^5 の付近で横ばいとなっている。また、文字 2-gram モデルのクロスエントロピーは、学習コーパスの文字数が 10^7 付近でほぼ横ばいとなっている。しかし、文字 3-gram モデルおよび文字 4-gram モデルのクロスエントロピーは、学習コーパスの文字数が 10^7 付近でも減少傾向を保っている。このことは、これらのモデルでは、学習コーパスを大きくするだけでより良い言語モデルが得られることを意味する。ただし、グラフの横軸は学習コーパスの文字数の常用対数値であり、横軸を一目盛右に移動した結果を得るには 10 倍の学習コーパスが必要であるという点に注意しなければならない。文字 3-gram モデルと文字 4-gram モデルのクロスエントロピーはあまり差がない。学習コーパスがさらに大きくなると、これらの差が顕著になると考えられる。

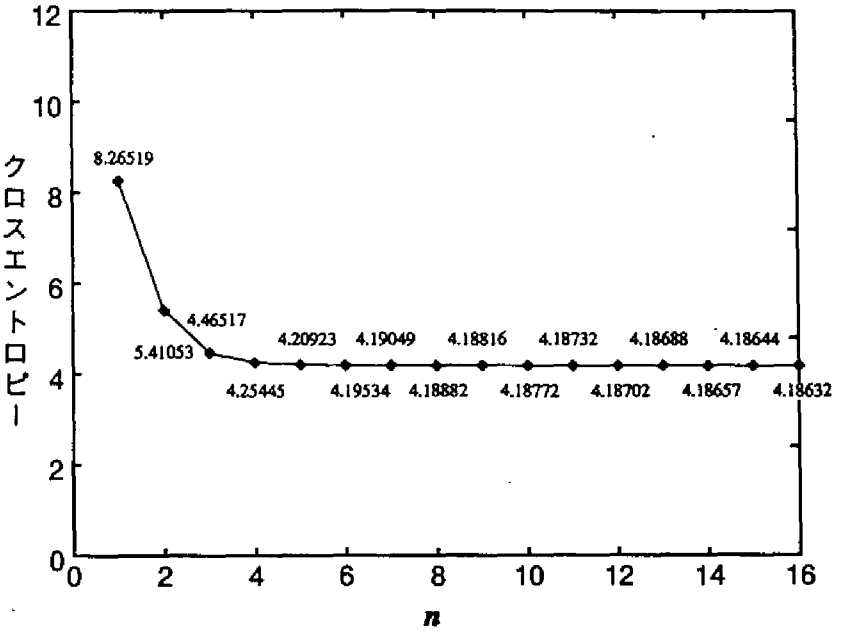

図 3.4: 先行事象の長さとクロスエントロピーの関係(文字 n-gram モデル)

3.4.3 先行事象の長さとクロスエントロピーの関係

図 3.4は先行事象の長さとクロスエントロピーの関係である。この結果から、先行事象をさらに長くすることでモデルの予測力が改善されることがわかる。しかし、クロスエントロピーの変化量は極めて微小であり、変化量自身も減少している。既に述べた学習コーパスの大きさとクロスエントロピーの関係を考慮すると、先行事象を長くすることによる減少量よりも、より大きな学習コーパスを用いることによる減少量の方が十分大きい。また、実装を考えた場合、先行事象を長くすることはパラメータの数を増加させるので、より大きい記憶容量が必要となりあまり好ましくない。これらのことを考慮すると、先行事象を長くすることよりは、学習コーパスを大きくすることが言語モデルの改善により貢献すると考えられる。

第2章で述べたように、クロスエントロピーには真のモデルのエントロピーの上限の推定値という意味もある。文字 16-gram の結果である

4.18632 ビットは、本論文を通してのクロスエントロピーの最小値であり、この値を日本語のエントロピーの上限とする[1]。

3.5 結論

この章では、文字n-gramモデルを題材として、確率的言語モデルのパラメータ推定における一般的な手法について説明した。この中で最も重要なのは削除補間法であり、この方法から既知文字集合の決定方法も自然に導出できる。削除補間法では可変であるパラメータを決定するために、文字n-gramモデルを実装し様々な実験を行った結果を報告した。この結果得られた結論は以下の通りである。

1. 削除補間における学習コーパスの分割数は10程度で良い。
2. 削除補間法から導出される既知文字集合の決定方法は十分なカバー率となる。
3. 先行事象を長くすることよりは、学習コーパスの大きさを増大することが言語モデルの改善により貢献する。

これらは、文字n-gramモデルの実験結果として得られた知見であるが、本章以降の類似のモデルに対しても十分有効であると考えられる。

[1]後述する形態素や文節を予測単位とした実験の結果はこの値を下回らなかった。これは、計算した値がCorpus近似であることに起因する可能性がある。

第 4 章

形態素単位のモデル

前章までの言語モデルは文字を予測単位としていたが、言語学が提案する形態素という単位を予測単位としたモデルを構築することもできる。ここでは、形態素を品詞と表記(単語)の組と定義する。第2章で述べたように、予測単位として形態素という概念を内包する確率的言語モデルを用いれば、入力文を形態素に分割することができる。この処理は、一般的に形態素解析と呼ばれており、書き言葉に対する自然言語処理の第一段階という重要な位置を占めている。この章で述べる形態素n-gramモデルを用いれば、その応用の一つとして形態素解析器を容易に実現できる。

4.1 形態素n-gramモデル

第3章で説明した文字n-gramモデルの予測単位を文字から形態素に変更することで、自然言語の文を形態素の連接とみなす確率的言語モデルが構成できる。これを形態素n-gramモデルと呼ぶ。文字n-gramモデルと同様に、アルファベット(既知形態素)の選択方法が問題となる。文字n-gramモデルの場合とは異なり、形態素は無限にあると考えられるので、既知形態素としてどのような形態素の集合を選択したとしても、テストコーパスに出現する可能性のあるすべての形態素が、学習コーパスに出現することは望めない。このため、未知語の扱いが避けられない問題となる。この問題に対処するため、未知語に対応する特別な記号を

用意し、既知形態素以外はこの記号から未知語モデルにより与えられる確率で生成されるとする。未知語に対応する特別な記号は、必ずしも唯一である必要はなく、品詞などの情報を用いて区別される複数の記号であってもよい。以下の説明では、各品詞に対して未知語に対応する記号を設ける。未知形態素に対しては、形態素n-gramモデルでその品詞の未知語記号を生成してから、未知語モデルで個々の表記をある確率で生成する。こうすることで、未知語モデルが確率的モデルの条件を満たす限りにおいて形態素n-gramモデルが確率的モデルの条件を満たすことが保証される。ただし、未知語モデルが既知形態素も生成する場合は、既知形態素を含む文は複数の導出を持つので、このような文の生成確率は、これら複数の導出に渡る和を計算した結果となる点に注意しなければならない。

以上に述べた形態素n-gramモデル$M_{m,n}$による、形態素列$m_1 m_2 \cdots m_h$の出現確率は以下の式で表される。ただし$\mathcal{M}_k$は既知形態素の集合を表わす。

$$P(m_1 m_2 \cdots m_h) = \prod_{i=1}^{h+1} M_{m,n}(m_i | m_{i-k} \cdots m_{i-2} m_{i-1})$$

ここでm_i $(i \leq 0)$は、式を簡便にするために便宜的に導入された文頭に対応する特別な記号である。また、$M_{m,n}$はアルファベット$\mathcal{M}_k \cup \{\mathtt{UM}\}$上の$n$-gramモデルであり、未知語記号UMを予測するときには、以下のように未知語モデル$M_{um,pos}$による確率も含むとする。

$$M_{m,n}(m_i | m_{i-k} \cdots m_{i-2} m_{i-1}) = \begin{cases} P(m_i | m_{i-k} \cdots m_{i-2} m_{i-1}) & \text{if } m_i \in \mathcal{M}_k \\ P(\mathtt{UM}_{pos} | m_{i-k} \cdots m_{i-2} m_{i-1}) M_{um,pos}(m_i) & \text{if } m_i \notin \mathcal{M}_k \end{cases}$$

ただし、posはm_iの品詞を表わす。未知語モデルについては次節で述べる。

文字n-gramモデルと同様に、形態素n-gramモデルの確率値も、コーパスの頻度から最尤推定するのが一般的である。文字n-gramモデルと異なる点は、予測単位に分割されたコーパスが必要なことである。予測単位に分割されたコーパスが無い場合でも、アルファベットを定義した後

にEMアルゴリズムの特殊形であるForward-Backwordアルゴリズム[14]を用いることでパラメータを推定することができる。しかし、この場合の解は局所的な最適解でしかなく、この方法で推定されたモデルは、予測単位に分割されたコーパスから推定されたモデルよりも良くないことが報告されている[16,17,18,19]。本論文では、予測単位に分割されたコーパスを用いた結果のみを報告する。

文字n-gramモデルと同様に、アルファベットはパラメータ推定の前に決定しなければならない。これには、何らかの辞書の見出し語を用いることや、学習コーパスに高頻度で出現する形態素とすることなどが考えられる。既知形態素集合を定義した後は、これに未知語に対応する特別な記号を加えてアルファベットとし、学習コーパスの未知語をこれらの記号に置き換えて頻度を計数することで形態素n-gramモデルの確率値を推定する。補間係数の推定なども同様に行なうことができる。

4.2 低頻度事象への対処

形態素n-gramモデルの場合も、文字n-gramモデルと同様に複数のモデルの補間が定義できる。これは、以下の式のように、文字n-gramモデルの式(3.9)の文字を形態素と考えるだけでえられる。

$$M'_{m,n}(m_i|m_{i-k}\cdots m_{i-2}m_{i-1}) = \sum_{j=1}^{h+1} \lambda_j^{h+1} M_{m,j}(m_i|m_{i-j+1}\cdots m_{i-2}m_{i-1}) \tag{4.1}$$

ただし、$h < n$はそれぞれの先行事象について学習コーパスにおける頻度が1以上となる最長の先行形態素数である。

$$N(m_{i-h}\cdots m_{i-2}m_{i-1}) > 0 \quad \wedge \quad N(m_{i-h-1}\cdots m_{i-2}m_{i-1}) = 0$$

補間係数の推定方法には、Held-out法と削除補間法がある。EMアルゴリズムの繰り返しの式では、コーパスに付加された導出方法(形態素分割)を用いる。

4.3 未知語モデル

未知語モデルは、表記から確率値への写像として定義され、既知形態素以外のあらゆる形態素の表記を0より大きい確率で生成し、この確率をすべての表記に渡って合計すると1以下になる必要がある。このような条件を満たすモデルのひとつとして、第2章で説明した文字n-gramモデルがある。このような文字n-gramモデルとして実現された未知語モデルM_xは、未知形態素だけでなく既知形態素の表記も0より大きい確率で生成する。この場合には、以下の式が示すように、未知語の生成確率の合計は1未満となる。

$$\sum_{m\in\mathcal{M}_u} M_{um}(m) + \sum_{m\in\mathcal{M}_k} M_{um}(m) = \sum_{m\in\mathcal{X}^*} M_{um}(m) = 1$$

$$\Leftrightarrow \quad \sum_{m\in\mathcal{M}_u} M_{um}(m) = 1 - \sum_{m\in\mathcal{M}_k} M_{um}(m) < 1 \quad \left(\because \sum_{m\in\mathcal{M}_k} M_{um}(m) > 0\right)$$

これは、言語モデルとしての条件を満たしてはいるが、クロスエントロピーという点で改善の余地がある。つまり、既知形態素の生成確率を何らかの方法で未知形態素に分配することで、未知形態素の生成確率が大きくなり、テストコーパスにそのような未知形態素が出現した場合に、テストコーパスの出現確率が大きくなり、クロスエントロピーが低くなる。既知形態素の生成確率の分配には、様々な方法が考えられるが、以下の式が表すように、すべての未知語にその生成確率に比例して分配する方法が一般的であろう。

$$M'_{um}(m) = \frac{M_{um}(m)}{\sum_{m\in\mathcal{M}_u} M_{um}(m)} = \frac{M_{um}(m)}{1 - \sum_{m\in\mathcal{M}_k} M_{um}(m)} \quad (m \in \mathcal{M}_u) \qquad (4.2)$$

形態素n-gramモデルと未知語モデルの関係は、文字n-gramモデルと未知文字モデルの関係と同じである。したがって、形態素n-gramモデルがその未知語モデルとして文字n-gramモデルを持っている場合、合計3種類の確率的モデルが含まれる。

未知語モデルとしての文字n-gramモデルのパラメータは、未知語の実例における文字列の頻度から推定される。換言すると、未知語の実例を

あたかも文であるかのように扱い、パラメータを推定する。未知語の実例の収集の方法として、削除補間法を応用した以下の方法を提案する。

> 学習コーパスをk個の部分コーパスに分割し、i番目の部分コーパスの未知語の実例を、i番目の部分コーパス以外を学習コーパスとしi番目の部分コーパスをテストコーパスと見た場合の未知語とする。

対案としては、学習コーパスに含まれるすべての形態素とすることや、学習コーパスにおける頻度が1である形態素とする[20]などが考えられるが、我々が提案する方法は削除補間法を応用して実際のテストコーパスにおける未知語と類似した実例を得ているので、他の方法よりも優れていると予測される。

未知語モデルのアルファベットの定義には、何らかの辞書の見出し語の文字を用いることや、学習コーパスまたは未知語の実例に高頻度で出現する文字とすることなどが考えられる。既知文字集合を定義した後は、これに未知文字に対応する特別な記号を加えてアルファベットとし、未知語の実例の未知文字をこれらの記号に置き換えて頻度を計数することで文字n-gramモデルの確率値を推定する。補間係数の推定なども同様に行なうことができる。

4.4 外部辞書の付加

この節では、未知語モデルの改善方法の一つとして、既知語の生成確率を辞書の見出し語などとして与えられる未知語の部分集合に等しく配分することを提案する。つまり、ある形態素の集合が与えられたとして、ここから既知形態素を除いた集合を$\mathcal{M}_d$として、この要素の生成確率を文字n-gramモデルによる確率と既知語の生成確率の合計を$\mathcal{M}_d$の要素数で割った値の和とする。

$$M'_{um}(m) = M_{um}(m) + \frac{1}{|\mathcal{M}_d|} \sum_{m \in \mathcal{M}_k} M_{um}(m) \quad (m \in \mathcal{M}_d) \qquad (4.3)$$

これは、既知形態素の生成確率を、学習コーパスには現れないが辞書などから形態素であると考えられる文字列に優先的に分配し、それらの生成確率を相対的に高くすることを意味する。このような文字列の集合を外部辞書と呼ぶ。品詞毎に未知語モデルを持つ場合には、外部辞書には文字列に加えてその品詞が記述されている必要がある。

以上に述べた未知語モデルによる未知形態素の出現確率は、以下の式で表される。

$$M'_{um}(m) = \begin{cases} 0 & \text{if } m \in \mathcal{M}_k \\ M_{um}(m) & \text{if } m \in \mathcal{M}_u \wedge m \notin \mathcal{M}_d \\ M_{um}(m) + \dfrac{1}{|\mathcal{M}_d|} \displaystyle\sum_{m \in \mathcal{M}_k} M_{um}(m) & \text{if } m \in \mathcal{M}_u \wedge m \in \mathcal{M}_d \end{cases}$$

このような改善は、確率的言語モデルの条件を満たしながら、辞書などのコーパス以外の情報を組み込むことを可能にしている。

4.5 評価

以上で説明した形態素n-gramモデルを構成し、この評価を行なうために以下の項目を調べる実験を行なった。

- 未知語モデルの評価実験
- 形態素n-gramの評価実験

以下では、これらの結果を提示し、考察を加える。

4.5.1 実験の条件

実験に用いたコーパスは、分割方法も含めて、第3章の文字n-gramの実験と同じである。表4.1は各コーパスの大きさと1文あたりの平均形態素数である。アルファベット数も文字n-gramの実験と同じ6,879としている。

表 4.1: 実験に用いたコーパス (形態素 n-gram モデル)

用途	文数	形態素数	形態素数 / 文数
学習	187,022	4,595,786	24.57
評価	20,780	509,261	24.51

テストコーパスの形態素区切りは、コーパスに予め付加されたものを用いた。したがって、テストコーパスの文字列の出現確率は、その文字列のすべての導出方法による確率を合計した値ではなく、コーパスに示された導出方法のみによる値である (コーパス近似)。なお、形態素区切りが明示されていない文に対しては、動的計画法を用いたアルゴリズムにより、出現確率が最大となる形態素区切りとその確率値 (Viterbi 近似) を、文に含まれる文字数に比例した時間で求めることができる。これは、第6章で述べる形態素解析である。

4.5.2 未知語モデルの評価実験

本章で説明した未知語モデルを文字 2-gram モデルで実装し、この部分でのテストコーパスの生成確率の対数値を計算した。文字 2-gram モデルは、第3章で述べた方法で構成した。ただし、パラメータ推定の対象が学習コーパスの文ではなく、未知語の実例である点が異なる。本章で提案した方法の有効性を確かめるため、以下の点に関して他の方法との比較実験を行なった。

1. パラメータ推定のための未知語の実例の収集方法

2. 既知形態素の生成確率の分配方法

以下では、それぞれの結果を提示し検討を加える。

未知語の実例の収集方法

実験を行なった未知語の実例の収集方法は以下の通りである。

表 4.2: 未知語の実例の収集方法の比較

方法	補間係数の値		クロスエントロピー	
	1-gram	2-gram	学習セット	テストセット
1	0.193	0.807	6.303	6.075
2	0.007	0.993	4.113	6.905
3	0.191	0.809	6.307	6.040

方法1 学習コーパスにおける頻度が1である形態素

方法2 学習コーパスに含まれるすべての形態素

方法3 分割された学習コーパスの1個にのみ出現する形態素

方法1は、永田[20]が未知語の収集に用いた方法である。方法2の長所は、非常に多くの実例が得られることである。方法3は、本論文で提案する方法である。この方法の長所は、実際の未知語に近い性質を持つ実例が得られることである。

これらの方法を実装し、学習に用いた実例(方法により異なる)に対するクロスエントロピーと、テストコーパスの未知語(方法によらず一定)に対するクロスエントロピーを計算した。表4.2は、この結果である。なお、ここでの実験に用いた未知語モデルは、品詞を区別していない。この結果から、この実験では方法3が最もよい未知語モデルであることがわかる。方法2は、他の方法よりも2-gramの補間係数が非常に高く、学習に用いた実例のクロスエントロピーは非常に低い。その一方、テストコーパスの未知語のクロスエントロピーは高い。これは、未知語の性質とすべての形態素の統計的振る舞いが大きく異なることを意味する。方法1と方法3では結果に大差はないが、学習に用いた実例のクロスエントロピーでは方法1がより良く、テストコーパスの未知語のクロスエントロピーでは方法3がより良いという結果である。これも、学習に用いた形態素とテストコーパスの未知語の性質の類似性によると考えられる。

既知形態素の生成確率の分配方法

実験を行なった既知形態素の生成確率の分配方法は以下の通りである。未知語の実例の収集方法としては、上述の方法3を用いた。

方法A すべての未知語にその生成確率に比例して分配(式(4.2))

方法B 特定の未知語に等しく配分(式(4.3))

学習コーパスには出現しないが、辞書等に記載されている形態素は多数ある。これらが、テストコーパスに未知語として多数出現する場合には、方法Bが方法Aよりも良い結果となることが容易に想像される。実際に問題になるのは、既知形態素の生成確率を分配する対象となる形態素集合の選択である。テキストの分野を反映した形態素が多く含まれているほど良い結果を導くと考えられる。また、未知語モデルを品詞等で分ける場合には、このための情報が記述されていなければならない。ここで述べる実験では、以下の二つの形態素集合の和集合とした。

- EDR日本語形態素辞書[15]の見出し語
- 学習コーパスに出現する未知語

この結果、テストコーパスの未知語の文字あたりのクロスエントロピーは、方法Aでは5.9338であり、方法Bでは5.1403であった。この差は十分有意であるので、既知形態素の生成確率をすべての表記に分配するモデルよりも形態素と考えられる特定の表記に配分するモデルが優れていると結論できる。

4.5.3 形態素n-gramの評価実験

この章で述べた未知語モデルをもつ形態素n-gramモデルを実装し、文字n-gramモデルと同様に、以下の項目を調べる実験を行なった。

- 学習コーパスの大きさとクロスエントロピーの関係
- 先行事象の長さ(nの値)とクロスエントロピーの関係

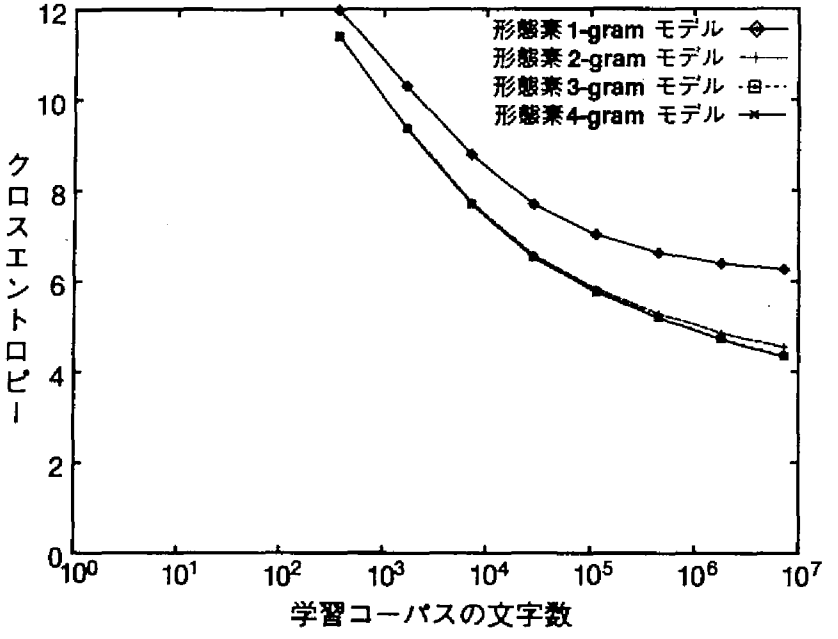

図4.1: 学習コーパスの大きさとクロスエントロピーの関係(形態素n-gramモデル)

以下で提示するクロスエントロピーの値は、特に断らない限りコーパス近似であり、全ての導出に対する合計の確率値ではない。以下では、これらの結果を提示し、考察を加える。

学習コーパスの大きさとクロスエントロピーの関係

図4.1は、形態素n-gramモデル($n = 1, 2, 3, 4$)による学習コーパスの大きさ(常用対数値)とクロスエントロピーの関係である。グラフから分かるように、形態素1-gramモデル以外のクロスエントロピーは、学習コーパスの文字数が10^7の付近でもかなり減少している。このことは、形態素1-gramモデル以外は、学習コーパスを大きくするだけでより良い言語モデルが得られることを意味する。ただし、グラフの横軸は学習コーパスの文字数の常用対数値であり、横軸を一目盛右に移動した結果を得るには10倍の形態素に分割された学習コーパスが必要であるという点に

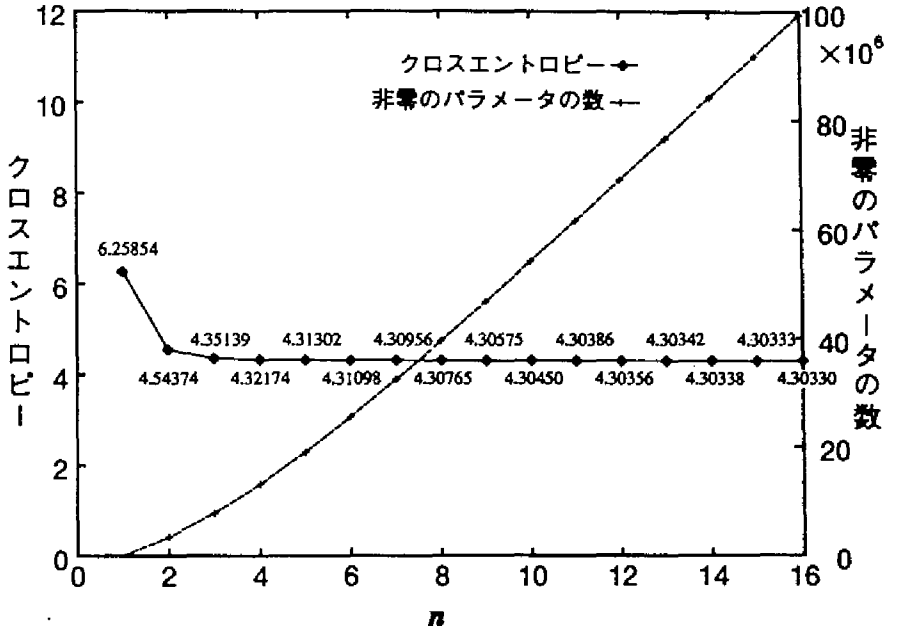

図 4.2: 先行事象の長さ (n の値) とクロスエントロピーの関係 (形態素 n-gram モデル)

注意しなければならない。

先行事象の長さとクロスエントロピーの関係

図 4.2は、先行事象の長さとクロスエントロピーの関係である。ただし、未知語モデルは一定である。この結果から、先行事象を長くすることでモデルの予測力が増すことがわかる。しかし、その変化量は極めて微小であり、変化量自身も減少している。未知語モデルを文字 2-gram モデルからより高次の文字 n-gram モデルに変更することで、この値が減少することは容易に予測される。しかし、既に述べた学習コーパスの大きさとクロスエントロピーの関係を考慮すると、先行事象を長くすることによる減少量よりも、より大きな学習コーパスを用いることによる減少量の方が十分大きい。このことから、先行事象を長くすることよりは、学習コーパスを大きくすることが言語モデルの改善により貢献すると考

表 4.3: クロスエントロピーの内訳

モデルの部分	クロスエントロピー
形態素予測	3.92692
未知語の文字予測	0.37638
合計	4.30330

えられる。

形態素 n-gram モデルは、形態素解析や音声認識などに応用されている。実用に際しては、記憶容量とクロスエントロピーの関係が重要である。形態素 n-gram モデルの実装に必要な頻度(確率)表は、配列で実装するのが簡潔かつ高速であるが、この場合には、$O(e^n)$ の記憶領域が必要である。このため、リストやハッシュなどのデータ構造を用いて、学習コーパスに出現する文字列の頻度だけを記憶するという方法が採用される。この場合の記憶領域の目安となる、実際に出現する形態素 n-gram の種類数を図4.2に加えてある。このグラフから、リストやハッシュなどのデータ構造を有効利用すればおおよそ $O(n)$ の記憶領域で形態素 n-gram モデルが実装できることがわかる。これは、形態素 n-gram モデルを応用する際に、適切に n の値を選ぶ指標となる。例えば、形態素 2-gram モデルを形態素 3-gram モデルに変更した場合、クロスエントロピーという基準で約 4.23% の改善となるが、頻度表の記述に必要となる記憶領域は約 2.27 倍となる。

クロスエントロピーの分枝性[21]から、代入によって多段になっている確率的モデルによるクロスエントロピーは、各部分の寄与に分解されるので、独立に計算することができる。形態素 n-gram モデルは、既知形態素と未知文字記号を予測する部分と未知語の文字列を予測する部分に分解できる。表4.3は、このようにして計算したクロスエントロピーの各部分の内訳である ($n = 16$)。この結果を見ると、テストコーパスに対するクロスエントロピーには、形態素を予測する部分がかなり大きく寄与し

表 4.4: テストコーパスにおける各品詞の既知語と未知語の数

品詞	助詞	名詞	語尾	動詞	記号
既知語の数	135,634	127,799	60,156	59,670	49,122
未知語の数	2	9,048	4	938	13
未知語率	0.001%	6.612%	0.007%	1.548%	0.026%

品詞	数字	副詞	形容動詞	助動詞	接尾語
既知語の数	7,510	6,868	5,797	30,470	12,671
未知語の数	358	216	259	15	85
未知語率	4.550%	3.049%	4.277%	0.049%	0.666%

品詞	形容詞	連体詞	接続詞	接頭語	感動詞
既知語の数	5,395	3,773	2,234	2,137	25
未知語の数	84	15	17	25	6
未知語率	1.533%	0.396%	0.755%	1.156%	19.355%

$$\text{未知語率}(\%) = 100 \times \frac{\text{未知語の数}}{\text{未知語の数+既知語の数}}$$

ていることが分かる。よって、短期的により良い言語モデルを構成するためには、この部分を改良することが近道であると考えられる。未知語モデルの改善例としては、文字 2-gram モデルに代わって、文字 4-gram を用いることが考えられるが、これらの間のエントロピーの差異が、第3章で述べた文字 n-gram モデルのクロスエントロピーの差異と同程度である考えると、未知語の文字予測部分の値は 78.6% になるが、これによる全体のエントロピーの減少は 1.86% にすぎない。形態素を予測する部分の改善には、品詞や文節などの文法的概念を用いることが有効であると考えられる。また、形態素 n-gram モデルよりも抽象度の高いモデルを用いることも考えられる。長期的には、未知語の文字列を予測する部分も改良することが望ましい。表 4.4に掲げた品詞別の未知語数を考慮す

ると、名詞に対する未知語モデルを優先的に改良することが効果的である。この場合にも、文字のクラスなど文法的概念や確率文脈自由文法などの n-gram モデルよりも抽象度の高いモデルを用いることが考えられる。また、より根本的に、文字予測という観点から形態素の定義を見直すことも興味深い課題である。

4.6 結論

この章では、まず予測単位を形態素に変更した形態素 n-gram モデルを説明し、次に未知語モデルを中心に、従来の形態素 n-gram モデルに対する改善を提案し、この改善案の妥当性を実験的に確かめた。最後に、学習コーパスの大きさと先行事象の長さのクロスエントロピーに対する影響を実験的に示し、実用化するための指針を与えた。クロスエントロピーの内訳を計算すると、未知語の文字予測よりも形態素の予測のほうが寄与が大きいことが分かった。したがって、短期的にはこの部分を改善することが全体の改善への近道である。次の章では、この改善としてのクラス n-gram モデルについて説明し、このための最適なクラス分類を推定する方法を提案する。

第 5 章

形態素クラスタリング

形態素n-gramモデルでは、ある時点iの形態素の予測に直前の長さ$n-1$の形態素列を用いる。このとき、i番目の形態素の確率分布を、長さ$n-1$の形態素列のすべての組合せに対して別々に推定しておき、予測に用いる。しかし、これらの直前の形態素列のいくつかは、次の形態素を予測するという目的においては区別する必要がないという場合がある。このような場合には、直前の事象を一定の長さのすべての形態素列に分類することは、不必要に直前の事象を区別していることになる。その結果、限られた学習コーパスにおける出現回数を減少させ、推定される確率値の信頼性の低下を招く。このような問題に対処するために、あらかじめ形態素をクラスと呼ばれるグループに分類しておき、先行するクラスの列を直前の事象とみなして分類することが提案されている[22]。このようなモデルは、クラスn-gramモデルと呼ばれている。クラスとしては、品詞などの人間の直感による分類を用いることができるが、概してこのような分類は確率的言語モデルという視点では良くない。したがって、予測力という観点でクラス分類をコーパスから推定することが課題となる。この章では、まずクラスn-gramモデルを説明し、次に最適なクラスを推定する方法を提案する。

5.1 クラス n-gram モデル

クラス n-gram モデル $M_{c,n}$ による~~おける~~形態素列 $\boldsymbol{m}\cdot\mathrm{BT} = m_1 m_2 \cdots m_{l+1}$ の出現確率は、以下の式で与えられる。ただし、c_i は m_i が属するクラスとする。

$$P(m_1 m_2 \cdots m_{l+1}) = \prod_{i=1}^{l+1} M_{c,n}(m_i | c_{i-k} \cdots c_{i-2} c_{i-1})$$

$$M_{c,n}(m_i | c_{i-k} \cdots c_{i-2} c_{i-1}) = \begin{cases} P(c_i | c_{i-k} \cdots c_{i-2} c_{i-1}) P(m_i | c_i) & \text{if } m_i \in \mathcal{M}_k \\ P(c_i | c_{i-k} \cdots c_{i-2} c_{i-1}) M_{um,pos}(m_i) & \text{if } m_i \notin \mathcal{M}_k \end{cases}$$

この式の中の $M_{um,pos}$ は形態素 n-gram と同じ未知語モデルである。未知形態素のクラスは、その品詞に対応する未知語記号 UM_{pos} とする。

形態素 n-gram モデルの場合と同様に、確率 $P(c_i | c_{i-k} c_{i-k+1} \cdots c_{i-1})$ の値および、確率 $P(m_i | c_i)$ の値は、コーパスから最尤推定することで得られる。

$$P(c_i | c_{i-k} c_{i-k+1} \cdots c_{i-1}) \overset{MLE}{=} \frac{N(c_{i-k} c_{i-k+1} \cdots c_i)}{N(c_{i-k} c_{i-k+1} \cdots c_{i-1})}$$

$$P(m_i | c_i) \overset{MLE}{=} \frac{N(m_i, c_i)}{N(c_i)}$$

形態素 n-gram モデルと異なる点は、各形態素にクラスが付与されたコーパスが必要なことである。ただし、クラスとして品詞を用いた場合は、形態素 n-gram モデルと同様、形態素に分割されたコーパスから上の式を用いて確率値が推定できる。

5.2 低頻度事象への対処

あるクラス n-gram モデルも、より低次のクラス n-gram モデルとの補間により低頻度事象に対処することが提案されている[23]。これは、以下の式のように、文字 n-gram モデルの式 (3.9) の文字をクラスと考えるだ

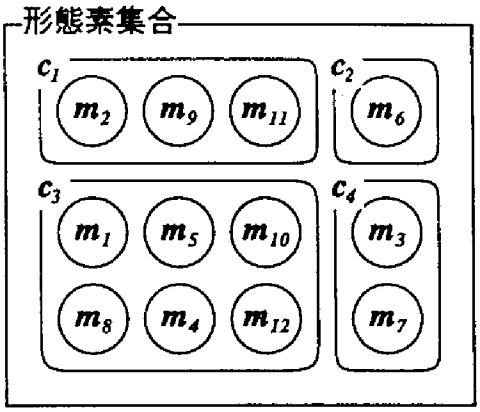

図 5.1: 形態素のクラスへの直和分解

けである。

$$M'_{c,n}(c_i|c_{i-k}\cdots c_{i-2}c_{i-1}) = \sum_{j=1}^{h+1} \lambda_j^{h+1} M_{c,j}(c_i|c_{i-j+1}\cdots c_{i-2}c_{i-1}) \qquad (5.1)$$

ただし、$h < n$ はそれぞれの先行事象について学習コーパスにおける頻度が1以上となる最長の先行クラス数である。

$$N(c_{i-h}\cdots c_{i-2}c_{i-1}) > 0 \quad \wedge \quad N(c_{i-h-1}\cdots c_{i-2}c_{i-1}) = 0$$

補間係数の値を求める方法も全く同じである。

5.3 クラス分類の推定

この節では、クラス n-gram モデルにおいて可変であるクラス分類を、確率的言語モデルの最適化という観点から求める方法を提案する。

5.3.1 形態素とクラスの対応関係

すでに述べたように、クラスは形態素の集合である。本論文ではさらに、全ての形態素は唯一のクラスに属することを仮定する。このとき、クラスの集合は形態素の集合の直和分割となっている(図5.1参照)ので、形態素とクラスの対応関係 F は、M, C をそれぞれ形態素の集合とクラ

スの集合とすると、関数$f : M \mapsto C$を用いて表すことができ、この関数は以下の条件を満たす[1]。

$$M = \bigcup_{m \in M} f(m)$$

$$\forall m \in M \text{ に対し } m \in f(m)$$

$$f(m_1) \neq f(m_2) \Rightarrow f(m_1) \cap f(m_2) = \phi$$

形態素とクラスの対応関係に対して、以下の関数を定義する。

- 形態素の移動を表す関数

$$move : F \times M \times C \mapsto F$$

$move(f, m, c)$は、形態素とクラスの関係fに対して形態素mをクラスcに移動した結果得られる形態素とクラスの関係を返す関数であり、以下のように定義される[2]。

```
define move(f, m, c)
    f(m) := f(m) − {m}
    c := c ∪ {m}
    return f
```

5.3.2 目的関数

すでに述べたように、形態素クラスタリングの目的は、クロスエントロピーという観点でより良い言語モデルを構成することである。したがって、最適なクラス分類は、テストコーパスのクロスエントロピーを最小にするクラス分類である。

$$\hat{f} = \underset{f \in F}{\mathbf{argmin}}\, H(L_{test}, M)$$

---

[1] fの値は形態素の集合である (例: $f(m_1) = \{m_1, m_2, m_3\}$)。

[2] 正確には、同じクラスに属する全ての形態素に対してfの値を改めなければならない。

このようなクラス分類に近いクロスエントロピーとなるクラス分類を、テストコーパスを参照することなく推定することが課題である。これは、補間係数の推定の場合と本質的には同じ問題である。したがって、以下の式のように削除補間を応用することで得られる平均クロスエントロピーを目的関数とすることを提案する。

$$\overline{H} = \frac{1}{m}\sum_{i=1}^{m} H(L_i, M_i) \tag{5.2}$$

ここで、M_i は i 番目以外の $m-1$ の部分コーパスから推定されたクラス n-gram モデル (補間係数の推定も含む) であり、L_i は i 番目の部分コーパスを表す。

この章で問題としているのは、確率的言語モデルとしてクラス n-gram モデルを用いた場合の形態素のクラスタリングである。この場合、コーパス (文の列) は一定であり、確率的言語モデル M は形態素とクラスの関係 F にのみ依存する。従って、平均クロスエントロピーは、形態素とクラスの関係の関数とみなすことができる。クロスエントロピーの値域は正の実数であるから、平均クロスエントロピーの値域も正の実数であり、これにより形態素とクラスの関係に全順序関係を与えることができる。この値がより小さいほうが、未知のコーパスに対してより良い言語モデルであることが予測される。

Brown ら[22] や Ney ら[9] は、クラスタリングの基準として、確率値の推定に用いるコーパスのエントロピーを用いている。我々は、これらの先行研究と異なり、クラスタリングのためのコーパスを確率値の推定用のコーパスとは別に用意することとした。この利点は、単語とクラスの関係を変えた場合に、クロスエントロピーが増化することもあれば減少することもあるので、閾値を設ける代わりに減少する場合のみクラスの変更を施すことができるということである。

5.3.3 アルゴリズム

クラスタリングの解空間は形態素とクラスの対応関係である。しかし、この数はある程度の大きさの語彙数に対しては非常に大きいため、

これら全てに対して平均クロスエントロピーを計算し、これが最小化となるクラス関係を選択するということは、計算量という観点から不可能である。平均クロスエントロピーの値はクラス関係の一部分の変更が全体に影響するという性質をもっているので、分割統治法や動的計画法を用いることもできない。以上のことから、我々は最適解を求めることを諦め、貪欲アルゴリズムを用いることにした。このアルゴリズムは以下の通りである(図5.2参照)。なお、$\overline{H}$ は式(5.2)で与えられる平均クロスエントロピーである。

$\mathcal{M}_k$ を頻度の降順に並べ $m_1, m_2, \ldots, m_n$ とする
foreach i $(1, 2, \cdots, n)$
  $c_i := \{m_i\}$
  $f(m_i) := c_i$
foreach i $(2, 3, \cdots, n)$
  $c := \mathbf{argmin}_{c \in \{c_1, c_2, \ldots, c_{i-1}\}} \overline{H}(move(f, m_i, c))$
  if $(\overline{H}(move(f, m_i, c)) < \overline{H}(f))$ **then**
    $f := move(f, m_i, c)$

計算量は、二番目の **foreach** での繰り返しの回数は形態素数 $|M|$ に比例し、**argmin** での繰り返しの回数はクラス数 $|C|$ に比例するので、全体で $O(|M| \cdot |C|)$ である。クラス数 $|C|$ は、全ての形態素が独立したクラスに分けられる場合に最大 $(|C| = |M|)$ となり、全ての形態素が同一のクラスとなる場合に最小 $(|C| = 1)$ となる。従って、初期化における全体の計算量は、最良の場合が $O(|M|)$ であり、最悪の場合が $O(|M|^2)$ である。ただし、形態素の並べ替えや一番目の **foreach** の計算量は係数が非常に小さいと考えられるので、考慮に入れていない。形態素数とクラス数の関係については考察を行なっておらず、次節で述べる実験の結果を図5.3に掲げるにとどめる。計算時間は形態素数とクラス数の関係を表わす曲線と横軸に囲まれた部分の面積に比例することになるが、このグラフを見ると実際にはかなり線形に近いことがわかる[3]。

[3]正確には対象となる形態素に依存するが、近似的に形態素数の関数とみなすことができる。

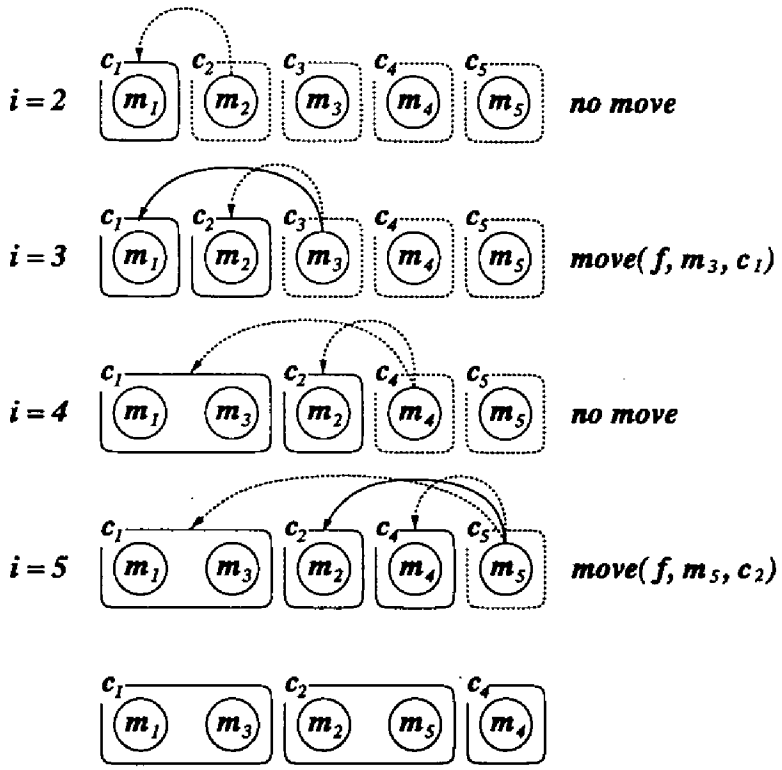

図 5.2: クラスタリングの概念図

頻度の高い形態素から移動を試みることとしているのは、頻度の高い形態素の移動のほうが平均クロスエントロピーに与える影響が大きいと考えられるので、早い段階での移動が後の移動によって影響されにくく、収束がより速くなると考えたためである。

上述のアルゴリズムによって得られたクラス分類からさらに探索を進めてより良いクラス分類が得られるかを試みることができる。このアルゴリズムとして、さらに形態素の移動を試みること[9]やクラスの併合を試みること[22]が考えられる。我々は、これらのアルゴリズムを小さなコーパスに対する予備実験で適用してみたが、必要となる計算時間が膨大である割にはテストコーパスに対するクロスエントロピーの改善が小さかった。よって、次節では、上述のアルゴリズムによる実験結果につ

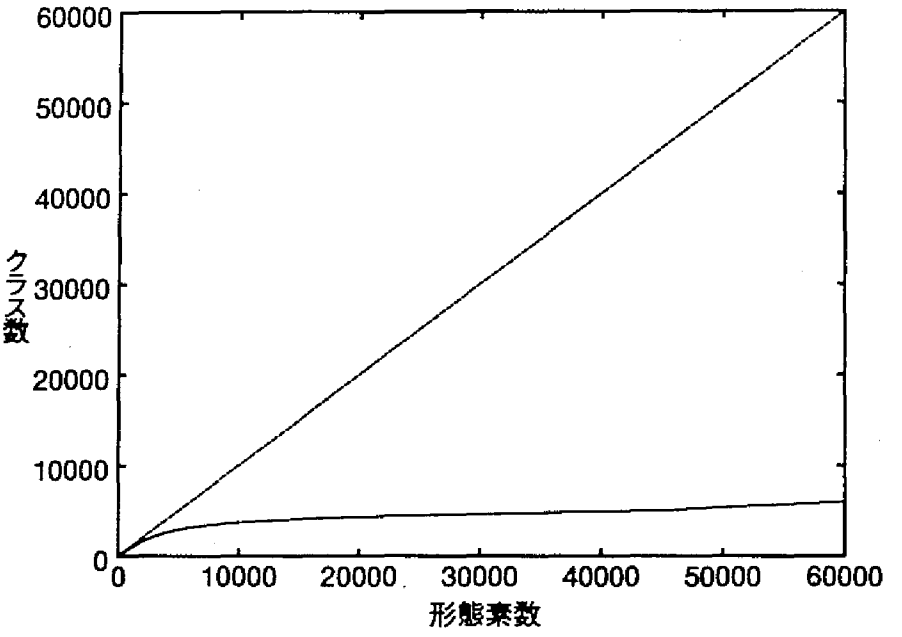

図 5.3: 形態素数とクラス数の関係

いて述べる。

5.4 評価

我々は、前節で説明したクラスタリング方法を評価するために、同じ学習コーパスから推定された形態素 2-gram モデルとクラス 2-gram モデルを、クロスエントロピーで比較した。この節では、この実験の条件と結果を提示し、考察を行なう。

5.4.1 実験の条件

実験に用いたコーパスは、分割方法も含めて第4章と全く同じである。前節で述べたように、クラス関数の推定ではこの9個の学習コーパスのうちの8つから n-gram モデルを推定し、残りの1つのコーパスに対してクロスエントロピーを求めるということを9通り行なって得られる平均ク

ロスエントロピーを評価規準とする。それぞれのコーパスに含まれる文と形態素の数は表4.1の通りである。既知形態素は、2個以上の学習コーパスに現れる59,956個の形態素とした。形態素2-gramモデルは、これらに対応する状態の他に、各品詞の未知語に対応する状態(15個)と文区切りに対応する状態を持つ。同様に、クラスモデルは、既知形態素をクラスタリングすることで得られるクラスに対応する状態と、各品詞の未知語に対応する状態と文区切りに対応する状態を持つ。

形態素2-gramモデルとクラス2-gramモデルを比較するために、これらを同じ学習コーパスから構成し、同じテストコーパスに対してクロスエントロピーを計算した。また、予め与えられた品詞をクラス分類とした場合の品詞2-gramモデルを構成し、同じテストコーパスに対してクロスエントロピーを計算した。それぞれの言語モデルの構成の手順は以下の通りである。

- 形態素2-gramモデル
  1. 削除補間により式(4.1)の補間係数を推定
  2. すべての学習コーパスを対象に形態素2-gramの頻度と形態素1-gramの頻度を計数
- 品詞2-gramモデル
  1. 削除補間により式(5.1)の補間係数を推定
  2. すべての学習コーパスを対象に品詞2-gramの頻度と品詞1-gramの頻度を計数
- クラス2-gramモデル
  1. 削除補間により式(4.1)の補間係数を推定
  2. 前節で述べた方法でクラス関数を推定
  3. 削除補間により式(5.1)の補間係数を再推定

表5.1: 形態素クラスタリングによる形態素2-gramモデルの改善の結果

言語モデル	状態数	クラス数	クロスエントロピー
単語2-gramモデル	59,972	59,956	4.6053 = 4.1674 + 0.4379
品詞2-gramモデル	31	15	5.6051 = 5.1672 + 0.4379
クラス2-gramモデル	5,990	5,974	4.5648 = 4.1269 + 0.4379

クラス数に15品詞の未知語に対応する記号と文区切り記号の合計16を加算すると状態数になる。

4. すべての学習コーパスを対象にクラス2-gramの頻度とクラス1-gramの頻度を計数

各モデルに含まれる未知語モデルは、第4章で説明した文字2-gramモデルである。既知形態素集合が共通なので、この部分のクロスエントロピーへの寄与は一定である(表4.3参照)。

5.4.2 結果と考察

表5.1は各モデルのテストコーパスのクロスエントロピーである。クラス2-gramモデルは形態素2-gramモデルや品詞2-gramモデルよりも低いクロスエントロピーとなっている。このことから、提案手法によって推定されたクラスによるクラス2-gramモデルが、形態素2-gramモデルや品詞2-gramよりも、予測力という点で良い言語モデルであると結論できる。Brownらの先行研究[22]やNeyらの先行研究[9]では、得られたクラスn-gramモデルの状態数は当然ながら減少しているが、テストコーパスのクロスエントロピー[4]が上昇し予測力という点で良い言語モデルとなっていない。この差異は目的関数にあると考えられる。

クラス推定のためのコーパスを単語n-gramモデルの推定用のコーパス

[4]これらの先行研究ではパープレキシティーを用いて評価しているが、第2章で述べた通り、クロスエントロピーと本質的には同じである。また、表記のみを区別した単語を予測単位としている点も異なっているが本質的には同じである。

とは別に用意するというアイデアは、Kneserら[24]によってすでに提案されている。しかし、この論文で報告されている実験に用いられている方法は、計算量を減らすことを目的に確率的言語モデルに特別な制限を設けており、結果として元となるアイデアの近似となっている。この文献では、ドイツ語と英語に対して以下の4つの結果を報告している。

1. 単語2-gramモデル
2. 人間が与えた品詞を用いた品詞2-gramモデル
3. 提案手法のクラスタリング結果を用いたクラス2-gramモデル
4. 従来手法[5]のクラスタリング結果を用いたクラス2-gramモデル

ドイツ語に対する実験結果では人間が与えた品詞を用いた場合が最も良い結果を与えている。もし、この文献で提案されている手法が本当に有効であるなら、人間が与えた品詞n-gramを初期状態として、提案したクラスタリングを再度行えば、さらに予測力が高いモデルが得られると思われるが、文献にはそのような考察あるいは実験については触れられていない。また、英語に対する実験結果では提案手法の結果と従来手法の結果はほとんど同じとなっている。これは元となるアイデアが必ずしも有効に働いていない結果であると考えられる。さらに、この文献で報告されている実験結果はこのように不安定なので、この手法を日本語など他の言語に適用した場合どのような結果が得られるかが予測できないという問題もある。しかしながら、我々の提案する基準では、予測力という点でより良い言語モデルとなることが、少なくとも日本語に対して、実験的に傍証されたと言える[6]。

得られたクラスモデルの状態数は、形態素モデルの状態数の約10.0%である。このことは、記憶容量という点でも、クラス2-gramモデルが形

[5]単語n-gramモデルの推定用のコーパスがクラス推定のためのコーパスと同じである方法

[6]英語コーパス(Penn Treebank[25]のWSJ)に対して行なった実験でも予測力の改善を確認している。

態素2-gramモデルよりも優れていることを示す。実験に用いた2-gramモデルの状態遷移表の大きさは、配列による単純な実装を仮定すると、状態数の2乗に比例するので記憶容量は0.998%に縮小する。また、非零要素の数を計数した結果、この数は形態素2-gramモデルでは724,870であり、クラス2-gramモデルでは245,283であった。よって、ハッシュやリンクリストなどを用いて非零要素だけを記憶する場合の記憶容量の縮小率は33.8%程度と推定される。現在実用となっている音声認識などでは、形態素3-gramを用いるのが一般的である。この場合、記憶容量の差はさらに拡大する。なお、品詞モデルは、記憶容量という点では非常に良いが、予測力が低過ぎて実用的であるとはいえない。

付録A.1は得られたクラスタの例である。多くのクラスタが、**クラスタ1**のように我々の言語直観に照らし合わせて、納得できるクラスタであった。このクラスタの「キロ/名詞」で示されるように、品詞の異なる形態素が同一のクラスとみなされている場合も観測された。一方、**クラスタ2**のように我々の言語直観に合致しないクラスタもあった。これは、我々が行なった形態素クラスタリングは、クラス2-gramモデルの改善という観点からのクラスタリングであることと、得られた形態素の分類が準最適解であることを考えると特に不自然ではないであろう。

5.5 結論

この章では、クラスn-gramモデルを仮定して、形態素の最適なクラス分類を求める方法について述べた。この方法は、クラス推定のためのコーパスを形態素n-gramモデルの推定用のコーパスとは別に用意するというアイデアに基づいている。この方法を実装し、形態素クラスタリングを行った。この結果得られたクラス分類を用いたクラス2-gramモデルと、同一の学習コーパスから推定した形態素2-gramモデルを比較した。その結果、クラス2-gramモデルが形態素2-gramモデルよりも、少なくとも日本語においては、予測力と記憶容量の両方の点で良い言語モデルとなることが分かった。この結果は、クラスn-gramモデルを用いたとしても同じであろう。

第 6 章

形態素解析

日本語に対する形態素解析とは、日本語の文(文字列)を入力とし、これを表記と品詞の直積として定義される形態素に分割する処理である。この章では、これを実現する手法の一つとしての確率的形態素解析とその基礎となる確率的言語モデルにおける最尤解の探索方法について述べる。

6.1 形態素解析の定義

日本語の形態素解析は、日本語のアルファベット$\mathcal{X}$のクリーネ閉包に属する文字列$\boldsymbol{x} \in \mathcal{X}^*$を入力として、これを表記$\mathcal{W} = \mathcal{X}^*$と品詞$\mathcal{T}$の直積として定義される形態素$\mathcal{M} = \{(w, t) | w \in \mathcal{W} \wedge t \in \mathcal{T}\}$の列$\boldsymbol{m} \in \mathcal{M}^*$に分解して出力することである。このとき、出力される形態素列の表記の連接は、入力のアルファベット列に等しくなければならない。つまり、入力のアルファベット列(長さl)を$\boldsymbol{x} = x_1 x_2 \cdots x_l$とし、出力の形態素列(要素数$h$)を$\boldsymbol{m} = m_1 m_2 \cdots m_h$とすると以下の式が成り立つ必要がある。ただし、$w(m)$は形態素$m$の表記を表し、$\boldsymbol{w}(\boldsymbol{m})$は形態素の連接$\boldsymbol{m}$の表記の連接を表わすものとする。

$$\boldsymbol{w}(\boldsymbol{m}) = w(m_1)w(m_2)\cdots w(m_h) = x_1 x_2 \cdots x_l = \boldsymbol{x} \tag{6.1}$$

一般に、これを満たす解は一意ではない。形態素解析の問題は、可能な解の中から人間の判断(正解)に最も近いと推測される形態素列(単語分

割と品詞割り当て)を選択し出力することである。この選択の基準としては、文法家が自身の言語直観を頼りにした規則に基づく方法と大量の正解例(形態素解析済みコーパス)からの推定を基準にする方法がある。以下では、後者の一つである確率的形態素解析について説明する。

6.2 確率的形態素解析

確率的形態素解析器は、品詞という概念を内包する確率的言語モデルを基にして、与えられた文字列$\boldsymbol{x}$に対する確率最大の形態素列$\hat{\boldsymbol{m}}$を計算し出力する。これは、以下の式で表される。

$$\begin{aligned}
\hat{\boldsymbol{m}} &= \operatorname*{argmax}_{w(\boldsymbol{m})=\boldsymbol{x}} P(\boldsymbol{m}|\boldsymbol{x}) \\
&= \operatorname*{argmax}_{w(\boldsymbol{m})=\boldsymbol{x}} P(\boldsymbol{m}|\boldsymbol{x})P(\boldsymbol{x}) \quad (\because P(\boldsymbol{x})\text{は}\boldsymbol{m}\text{によらない}) \\
&= \operatorname*{argmax}_{w(\boldsymbol{m})=\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{m})P(\boldsymbol{m}) \quad (\because \text{ベイズの公式}) \\
&= \operatorname*{argmax}_{w(\boldsymbol{m})=\boldsymbol{x}} P(\boldsymbol{m}) \quad (\because P(\boldsymbol{x}|\boldsymbol{m}) = 1)
\end{aligned}$$

この式の最後の$P(\boldsymbol{m})$が品詞という概念を内包する確率的言語モデルである。このようなモデルとして、第4章で説明した形態素n-gramモデルや第5章で説明したクラスn-gramモデルを用いることができる。クラスとして品詞を用いることも可能である。図6.1はこれらの関係の概念図である。クラスn-gramモデルを用いた場合、最尤解の探索にはクラスが与えられた時の形態素の条件付確率($P(m_i|c_i)$)の計算が余計に必要であるという点に注意しなければならない。

6.3 解探索のアルゴリズム

形態素n-gramモデルによる形態素解析器は、入力として文字列$\boldsymbol{x}$を受けとり、品詞という概念を内包する確率的言語モデルを用いて計算される確率が最大の形態素列$\hat{\boldsymbol{m}}$を式(6.1)で表わされる条件の下で計算し出力する。解の探索には動的計画法[26]を用いることができ、入力の文字数

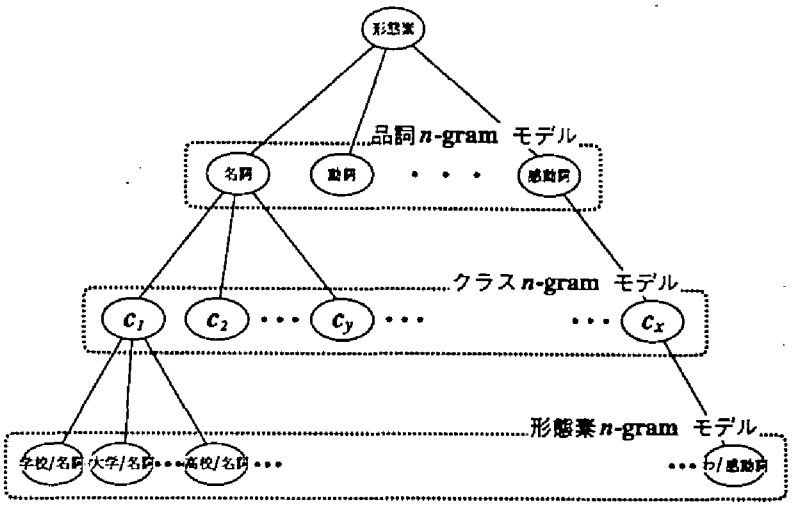

図 6.1: 各 n-gram モデルの概念図

n に対して計算時間が $O(n)$ となるアルゴリズムが提案されている [27, 28]。

これらの文献を参考にして、確率が最大の解 (最尤解) を効率良く計算する方法を実装した。このアルゴリズムは、入力である文字列を一文字ずつ読み、解探索の中間状態を記憶しておく表を更新していく。したがって、既知形態素を記憶している辞書は各時点で終る形態素を列挙できるように設計しておく必要がある。これは、複数文字列のテキスト照合にオートマトンを用いる Aho と Corasick の方法 [29] を形態素解析の辞書検索に応用した方法 [30, 31] で実装した。また、未知語モデルは、文頭から各時点での位置までの文字列の全ての接尾辞の生成確率を返せるように設計しておく必要がある。これは、文頭から各位置までの文字列の生成確率を記憶しておく表を用意し、入力の文字列を一文字ずつ読みながら表の値を更新するという方法で実装した。解探索の中間状態を記憶しておく表は図6.2のようになっている。この表の横方向は入力文中の位置に対応し、縦方向は状態に対応する。表の各要素はトレリスのノードに対応しており、表中の位置で示された文字と状態で終る形態素の中で最大の確率を与える形態素と、探索のために便宜的に与えられたポイン

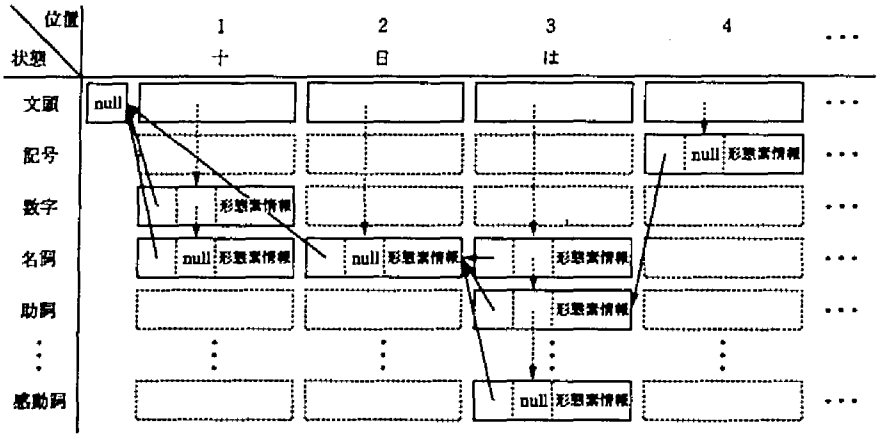

図 6.2: 最尤解の探索に用いる表

ターを2つ持っている。1つは、そのノードへの最尤の経路における一つ前のノードを指しており、表を埋めた後の最尤の経路の探索に用いられる。もう1つは、その位置で終る他のクラスのノードを指しており、このポインターをたどることで、この位置での接続確率のチェック対象を存在するクラスだけに限定することが可能になる。図6.2の例の場合、入力文の1文字目で終る形態素の中で最尤経路の探索の対象となるのは、表の1文字目の欄を上から順にポインターをたどることで、「十/数字」と「十/名詞」だけであると分かる。入力の文字列を全て読み終ると、文末に対応する状態への遷移を同様に行なう。その後、文末に対応する状態から「最大の確率を与える形態素へのポインター」を逆にたどることで最尤解となる形態素列が得られる。

6.4 評価

日本語の確率的形態素解析の先行研究として、品詞3-gramモデルを用いた結果が報告されている[32]。我々の実験の結果、予測力という点では、品詞3-gramモデルよりも形態素2-gramモデルのほうが優れていることが分かっている。したがって、形態素解析の精度も形態素2-gramモ

デルのほうが良くなると考えられる。この節では、形態素2-gramモデルを用いた形態素解析の結果を提示することに加えて、以下の点を明らかにするための実験を行った。

1. 外部辞書(第4章)による解析精度の向上

2. 形態素クラスタリング(第5章)による解析精度の向上

以下では、まず形態素解析精度の評価基準について述べ、実験の条件を明確にし、上述の実験の結果を提示し評価する。また、文法の専門家による形態素解析器との解析精度の比較を行なった結果について述べる。なお以下では、「クラス n-gram モデル」などの言語モデルを表す表現を、文脈から明らかな場合には、その言語モデルに基づく形態素解析器を表すためにも用いる。

6.4.1 評価基準

我々が用いた評価基準は、先行研究[32]と同じ再現率と適合率である。これらは、次のように定義される。EDRコーパスに含まれる形態素数を N_{EDR}、解析結果に含まれる形態素数を N_{SYS}、分割と品詞の両方が一致した形態素数を N_{COR} とすると、再現率は N_{COR}/N_{EDR} と定義され、適合率は N_{COR}/N_{SYS} と定義される。例として、コーパスの内容と解析結果が以下のような場合を考える。

コーパス

外交/名詞 政策/名詞 で/助動詞/ は/助詞 な/形容詞 い/語尾

解析結果

外交政策/名詞 で/助詞 は/助詞 な/形容詞 い/語尾

この例において、分割と品詞の両方が一致した形態素は「は/助詞」と「な/形容詞」と「い/語尾」であるので、$N_{COR}=3$ となる。また、コーパスには6つの形態素が含まれ、解析結果には5つの形態素が含まれているので、$N_{EDR}=6, N_{SYS}=5$ である。よって、再現率は $N_{COR}/N_{EDR}=3/6$ となり、適合率は $N_{COR}/N_{SYS}=3/5$ となる。

表 6.1: 品詞毎の形態素数とクラス数

品詞	助詞	名詞	語尾	動詞	記号
形態素の数	108	44453	95	8352	80
クラスの数	59	3680	74	1260	30
平均要素数	1.83	12.08	1.28	6.63	2.67

助動詞	接尾語	数字	副詞	形容動詞	形容詞
110	789	1473	1411	2035	572
69	262	90	193	199	89
1.59	3.01	16.37	7.31	10.23	6.43

連体詞	接続詞	接頭語	感動詞	合計
128	148	170	32	59956
36	31	71	13	6156
3.56	4.77	2.39	2.46	9.74

$$\text{平均要素数} = \frac{\text{形態素の数}}{\text{クラスの数}}$$

6.4.2 実験の条件

実験に用いたコーパスやクラスタリングの基準は、第5章と同じである。ただし、形態素クラスタリングのアルゴリズムに、同一の品詞だけを併合するという条件を付けた。これにより、品詞とクラスと形態素の関係が、図6.1のような木構造となり、クラスから品詞が曖昧性なく分かるので、形態素解析のアルゴリズムが簡便になる。この変更の結果、クラスタリングによって得られるクラス 2-gram モデルのクロスエントロピーは、第5章で提示した結果よりわずかに高かった。

品詞毎の形態素数とクラスタリングの結果得られたクラスの数を表6.1に掲げた。平均要素数は、形態素数をクラス数で割った値である。この値は、内容語において高く、機能語において低いことが観測される。このことから、品詞 n-gram モデルにおいては機能語を一般化し過ぎてお

り、形態素n-gramモデルにおいては内容語を特殊化し過ぎているということが分かる。

6.4.3 外部辞書と形態素クラスタリングによる精度向上

図6.3 (p.69)は、形態素クラスタリングの結果を用いたクラス2-gramモデルの、外部辞書を持つ場合と持たない場合の、クロスエントロピーと形態素解析の精度である。このグラフから次のようなことが分かる。まず、学習コーパスの大きさと解析精度の関係であるが、解析精度は、コーパスの大きさに対して単調に増加している。しかし、コーパスがある程度大きくなるとこの増加量は小さくなっている。このことは、さらなる精度向上を達成するためには、学習コーパスを増やすという単純な方法は、コーパスの作成コストを考えると、得策ではないということを意味する。次に、外部辞書を付加することによる解析精度の向上であるが、クロスエントロピーの減少から予測される通り、外部辞書を付加することにより解析精度が向上した。グラフから分かるように、学習コーパスの大きさが小さい方が、外部辞書を付加することによる効果が大きい。この理由は、学習コーパスが大きくなると、外部辞書の元となる辞書などに記述されている形態素の大部分が学習コーパスに含まれることになり、テストコーパスに含まれる未知形態素の割合が減少することであると考えられる。この議論から、確率的形態素解析器を用いて学習コーパスと異なる分野の文を解析する場合には、未知形態素となるであろうその分野特有の用語(表記と品詞)を収集しておき、これを外部辞書として付加することでかなりの精度の向上が望めると考えられる。分野特有の用語の収集方法としては、その分野の専門用語辞書などを直接用いることや、その分野の大量の文例から文字n-gram統計を用いて抽出し品詞を推定すること[33]などが考えられる。

表6.2は、外部辞書を備えない場合と備えた場合の、形態素2-gramモデルとクラス2-gramモデルによるクロスエントロピーと形態素解析の精度である。また、先行研究との比較のため、外部辞書を備えていない場合の品詞3-gramモデルによるクロスエントロピーも表中に記載して

表 6.2: 各言語モデルによるクロスエントロピーと形態素解析の精度

モデル	クロスエントロピー	再現率	適合率
形態素 2-gram	4.6053	93.23%	89.36%
形態素 2-gram+ 外部辞書	4.5437	93.37%	89.75%
クラス 2-gram	4.5654	93.32%	89.78%
クラス 2-gram+ 外部辞書	4.5039	93.41%	90.12%
品詞 3-gram	5.8643	–	–

いる。この結果から、外部辞書の有無に関わらず、我々が提案する方法によって得られる単語のクラス分類を用いることで、形態素解析の精度が再現率と適合率の双方で向上していることが分かる。これは、クロスエントロピーの減少から予測される通りの結果である。このように、確率モデルを用いた言語の解析では、クロスエントロピーが減少するようにモデルを改善することで、自然に形態素解析などの解析精度が向上することが見込まれる。ただし、このクロスエントロピーと解析精度の関係は、単調であることが解析的に導出できるような確固たる関係ではないことに注意しなければならない。クロスエントロピーと解析精度の関係が逆になっている例(上述の関係の反例)として、表6.2の中の「形態素 2-gram+ 外部辞書」と「クラス 2-gram」のエントロピーと適合率が挙げられる。

永田[32]は、品詞 3-gram モデルを用いた形態素解析について述べている。この文献での評価基準は、我々が用いたものと全く同じというわけではなく、単語分割のみや読みも含めた再現率と適合率を報告している。このような評価の一つとして 72,000 文で学習した品詞 3-gram モデルの単語分割の精度として 90.6% の再現率と 91.7% の適合率を報告している。このモデルとの比較を可能にするために、約 47,000 文の学習コーパスで学習した「クラス 2-gram+ 外部辞書」の単語分割の精度を計算した。この結果、再現率は 94.8% であり、適合率は 94.9% であり、学習コーパス

が少し小さいにもかかわらず品詞3-gramモデルの結果を双方で上回っている。解析精度に関しては全ての条件が同じというわけではないので単純な比較は適切ではないが、この結果は、本手法の優位性を実験的に示すと考えられる。また、クロスエントロピー(表6.2参照)の差は十分有意であると考えられるので、この点からも本手法の形態素解析の精度という点での優位性が十分予測される。しかし、より長い文脈から次の品詞を予測しているという品詞3-gramモデルの良い点も無視できない。この点を採り入れて、形態素3-gramモデルに対して形態素クラスタリングを実行し、その結果を用いてクラス3-gramモデルを構築すれば、クロスエントロピーがさらに下がり、形態素解析の精度も上がると考えられる。ただし、実用とするためには、遷移表や解探索のための表が大きくなることによる記憶域の増大と可能な組合せの増加による解探索に必要な時間が増加するという問題にも注意を払う必要がある。

6.4.4 文法の専門家による形態素解析器との比較

我々は、上述の実験に加えて、文法の専門家による形態素解析器と確率的形態素解析器を解析精度という点で比較するという実験を行なった。この際に最大の問題となるのは評価基準である。確率的形態素解析器の解析精度の比較は容易に行なえる。つまり、上述の実験のように、同一の学習コーパスと同一のテストコーパスを用いた解析結果の再現率と適合率を比較すればよい。これは英文における単語の品詞推定[34, 16, 35, 36, 37, 18]の精度の比較にも用いられる標準的な方法である(英語では単語区切りに曖昧性がないので再現率と適合率は同じ値になる)。しかし、文法の専門家による形態素解析器の解析精度の比較は一般に容易ではない。これは、それぞれの文法の専門家によって形態素の定義(品詞体系や単語区切り)に違いがあり、正解となるべき形態素解析結果を共有できないことに起因する。その結果、形態素解析器の評価としては、あるいくつかの文の解析結果を文法の専門家も含めた形態素解析器の製作者が観察することで計算される値が用いられる。また、テストは最後に一回だけ行なわれるのではなく、テストの結果を見て形態素解析器を修正するという

表6.3: 京都大学テキストコーパスの大きさ

コーパス	文数	形態素数	文字数
学習コーパス	8,584	206,812	366,599
テストコーパス	921	22,484	39,826

こともあり、完全なオープンテストになっていないこともある。このようなテストの結果得られる精度は、客観性に欠けるので、おおよその目安としてのみ意味があり、複数の形態素解析器の比較に用いることはできない。この問題は、文法の専門家による形態素解析器と確率的形態素解析器の解析精度の比較を行なう際にも現れる。

上述の問題を解決する方法として、同じ文法基準(品詞体系や単語区切)を持つ形態素解析済みコーパスと文法の専門家による形態素解析器を用いることが考えられる。これが、本研究で我々が選択した解決方法である。具体的には、京都大学で開発された文法の専門家による形態素解析器JUMAN[38]とその解析結果を人手で修正したコーパス[39]を用いた。つまり、コーパスを学習コーパスとテストコーパスに分割し(表6.3)、学習コーパスから構成した確率的形態素解析器(外部辞書を備えたクラス2-gramモデル)とJUMANを用いてテストコーパスを解析した結果を、テストコーパスにあらかじめ付与されている正解と比較して、それぞれの再現率と適合率を計算した。なお、外部辞書の形態素集合は、学習コーパスには出現するが既知形態素とならなかった形態素集合である。表6.4はこの結果である。この表から、テストコーパスにおいては、確率的形態素解析器の誤りが文法の専門家による形態素解析器の誤りに対して25%程度少ないことが分かる。この実験で使用した解析済みコーパスがJUMANの出力の訂正の結果であることや、コーパスの訂正の過程で訂正結果を参考にしてJUMANを改良していることを考えると学習コーパスでの比較が適切かも知れない。この場合は、確率的形態素解析器の解析精度は表6.4に示されるように圧倒的に良い。未知語モデルを文字ク

表 6.4: 文法家による形態素解析器と確率的形態素解析器の精度比較

形態素解析解析器	学習コーパス		テストコーパス	
	再現率	適合率	再現率	適合率
JUMAN3.2	94.29%	93.67%	94.51%	94.02%
クラス 2-gram+外部辞書	98.42%	98.48%	95.84%	95.67%

ラスタリングしたクラス n-gram モデルとすることや、外部辞書の源として JUMAN の辞書や別のコーパスを JUMAN で解析した結果から得られる学習コーパスに現れない高頻度の形態素を用いることで、確率的形態素解析器の精度はさらに向上すると考えられる。

本実験で比較の対象とした文法の専門家による形態素解析器は、初版の完成から 10 年弱の期間を経ており、この間に莫大な人的資源が投入され様々な改良が施されている。一方、我々の確率的形態素解析器がパラメータ推定に用いた学習コーパスは 8,584 文であり、これを作成する費用はそれほど高くはない。これは、確率的形態素解析器が、文法の専門家による形態素解析器に対して優位である点の一つである。現状での学習コーパスの大きさは $10^{5.56}$ 文字と比較的小規模であり、図 6.3 の EDR コーパスにおける学習コーパスの大きさと解析精度の関係から、コーパスを増量し確率的言語モデルを再学習するということを繰り返すことで、この品詞体系でのより高精度の形態素解析器が容易に実現できると予測される。これと並行して確率的言語モデルの改善を行なうことも重要である。以下に、より良い確率的形態素解析器を実現するための指針をまとめる。

- 解析済みコーパスの保守と増量

  コーパスの修正

  人手による修正を受けた解析済みコーパスにも誤りはあり、さらなる修正が必要である。確率的形態素解析器の出力との比

較は、これらの誤りを指摘する上で有効であろう。

コーパスの増量

すでに指摘したように、学習コーパスは多ければ多いほど良い。新たな文に正解を付加するときには、人手による修正を受けたコーパスを全て用いて、最も良い言語モデルを学習し、その結果得られる確率的形態素解析器による解析結果を修正することで、人手による修正のコストを最小限に抑える必要がある。

品詞体系の変更

形態素解析器の出力を用いた研究や開発の過程で、品詞体系の変更が要求されることがある。例えば、京都大学テキストコーパス[39]では「みんな/名詞」と「みんな/副詞」を区別していない。このような区別が必要になれば、まず解析済みコーパスの一部をこの区別を加えて修正し、これと残りのコーパスで問題となる形態素が出現しないコーパスから形態素解析器を学習し、問題となる形態素が出現する文を曖昧な部分以外を固定して解析し直すことで、人手による修正のコストを最小限に抑えることができる。

- 確率的言語モデルの改良

確率的言語モデルの改善方法は、本論文で提案した形態素クラスタリング以外にも提案されいてる。これらは、未知語モデルにも適用できる。

可変記憶長マルコフモデル

形態素n-gramモデルでの形態素予測は固定長の先行事象を条件部にもつが、この長さを先行する形態素列に応じて変化させる[40,41]。

キャッシュモデル

直前のいくつかの単語の分布(キャッシュ)を用いてn-gramモデルのパラメータを動的に変化させる[42]。

複数のモデルの補間

複数のクラスn-gramモデルを補間したモデル[43]や、複数の分野から推定された確率的言語モデルを補間したモデル[12]を用いる。

これらの改良をうまく組み合わせることで言語モデルの予測力が向上し、結果としてより高い精度の形態素解析器が実現できる。

- 解探索のアルゴリズムやデータ構造の改良
  これによる解析速度や記憶容量の改良は、解析精度の向上にはつながらないが、実用化する上で重要である。解探索のアルゴリズムやデータ構造は、モデルのクラスに依存する点に注意しなければならない。

これらの改善は独立に行なえるので、組織的な取り組みが可能になる。このように、高い精度を実現するための方法論が確立していることが確率的手法の最大の利点であろう。

6.5 結論

この章では、確率的形態素解析について説明し、形態素クラスタリングと外部辞書の付加による精度向上について述べた。形態素クラスタリングとしては、形態素n-gramモデルをクロスエントロピーを基準としてクラスn-gramモデルに改良する方法を提案した。形態素2-gramモデルとクラス2-gramモデルを実装し実験を行なった結果、形態素解析の精度の向上が観測された。また、未知語モデルに外部辞書を付加する方法を提案した。同様の実験を行なった結果、形態素解析の精度の向上が観測された。これは、学習コーパスとは異なる性質を持つ分野の形態素解析器や解析済みコーパスを作成するのに特に有効であろう。両方の改良を行なったモデルによる形態素解析実験の結果の精度は、先行研究として報告されている品詞3-gramモデルの精度を上回った。これは、我々のモデルが形態素解析の精度という点で優れていることを示す結果である。

これらの実験に加えて、人間の言語直感に基づく形態素解析器との精度比較の実験を行なった。この結果、確率的形態素解析器の誤りは文法家による形態素解析器の誤りに対して25%程度少なかった。形態素解析における確率的な手法は、人間の言語直感に基づく形態素解析器と比較して、現時点で精度がより高いという長所に加えて、今後のさらなる改良にも組織的取り組みが可能であるという点で有利である。

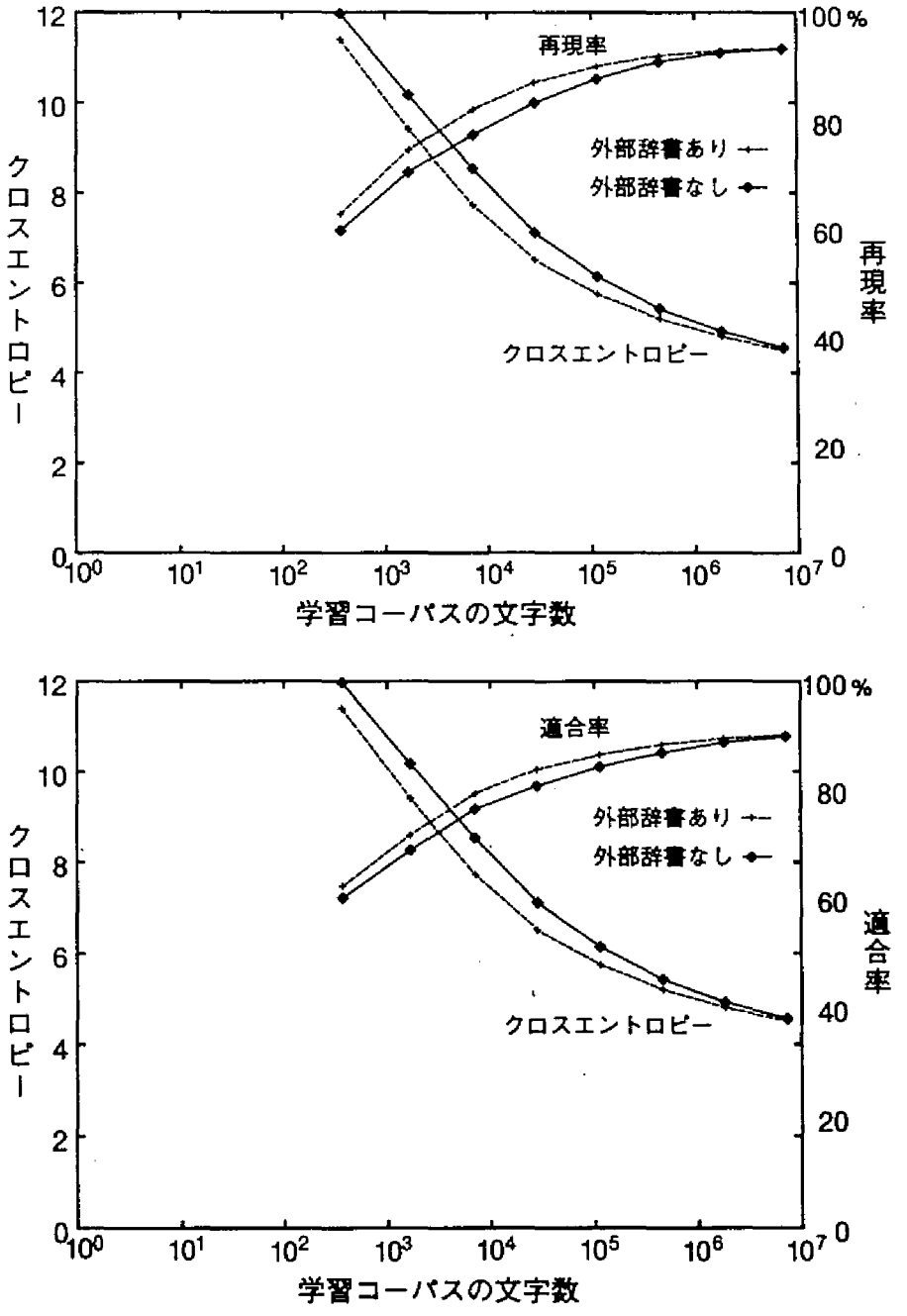

図 6.3: 学習コーパスの大きさと形態素解析精度の関係

第 7 章

文節を単位としたモデル

前章までの言語モデルは、直前の連続する形態素列や文字列に基づいて、次の形態素や文字を予測するモデルである。この章では、言語学が提案する文節という単位を予測単位とし、係り受けという構造を内包するモデルを提案する。形式的には、前章までの言語モデルが確率正規文法に属するのに対して、この章で説明するモデルは確率文脈自由文法に属する。この文法の終端記号は文節の属性であり、これは内容語の主辞と機能語の主辞の直積として表される。属性からの文節の形態素列の予測には形態素n-gramモデルを、未知語モデルには文字n-gramを用いる。

7.1 文節モデル

日本語の文は、1個以上の内容語と0個以上の機能語と句読点からなる文節と呼ばれる単位の連接とみなすことができる。これは、内容語の集合を$Cont$、機能語の集合を$Func$、句読点の集合を$Sign$とすると以下の式で定義される。

$$Bnst = Cont^{+}Func^{*} \cup Cont^{+}Func^{*}Sign$$

ここで、$+$と$*$はそれぞれ正閉包とクリーネ閉包を表す。この文節を予測単位とするモデルを構成することができる。このようなモデルとして、第一に考えられるのは、文節n-gramモデルであろう。しかし、係り受けとして知られる複数の文節間の関係は、必ずしも連続した文節間のみ

ではない。この関係をモデル化するためには、確率正規言語に属する n-gram モデルでは不十分である。離れた要素間の関係を記述するために、さまざまな文法が提案されている。この章では、これらの文法の一つである確率文脈自由文法 (SCFG) [5,6] を用いて文節間の係り受けをモデル化する。

確率文脈自由文法でまず問題となるのは、終端記号と非終端記号であろう。終端記号として、文節をそのまま用いることが考えられるが、この数は非常に大きく、データスパースネスの問題を引き起こす。そこで、クラス n-gram モデルの考え方を応用して、文節を何らかのグループに分類し、これを終端記号とすることが考えられる。この分類には、以下で定義される属性を用いることにする。

$$attrib(b) = \langle last(cont(b)),\ last(func(b)),\ last(sign(b)) \rangle \qquad (7.1)$$

関数 $cont$, $func$, $sign$ は、それぞれ文節を引数として、その内容語、付属語、句読点を返す。また、関数 $last(\boldsymbol{m})$ は形態素列 $\boldsymbol{m}$ の最後の形態素の品詞を返す。空形態素列の場合は NULL を返す。文節の属性が与えられると、文節の具体的な形態素列は、内容語列と機能語列を独立に第4章で説明した形態素 n-gram モデルによって生成される。

7.2 係り受けのモデル

係り受けとして知られる文節間の関係を記述するために、一般的に認められている複数の係り受け関係の非交差を仮定し、文節の属性を終端記号とする確率文脈自由文法を導入する。日本語の係り受けの性質として、文において前に位置する文節が、後に位置する文節に係ることが分かっている。さらに、係り受け関係をすでに何が係っているかに依存しない二項関係であると仮定する。したがって、これを文脈自由文法の生成規則として表すと $B \Rightarrow AB$ となる。ここで、A は係り文節を表す非終端記号であり、B は受け文節を表す非終端記号である。ここで、非終端記号を終端記号と同じように文節の属性とすることもできるが、付加的な情報との直積とすることで、係り受けの性質を反映するように特殊化

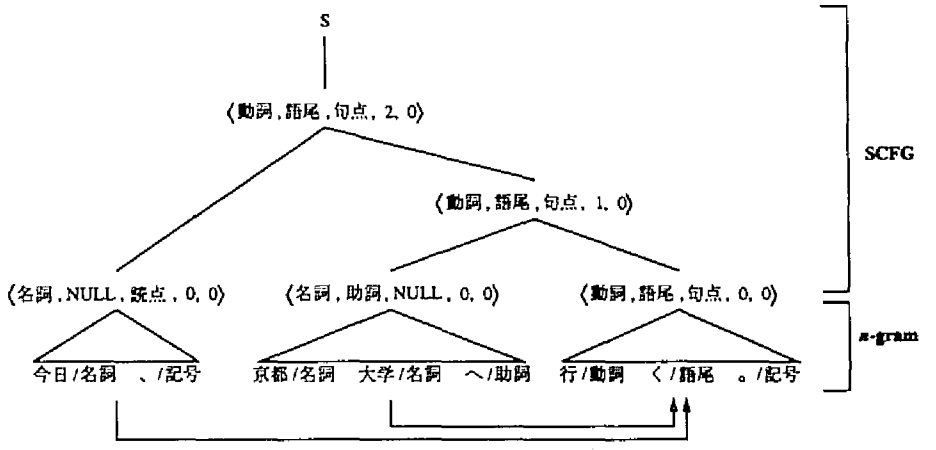

図 7.1: 文節単位の係り受けモデル

することもできる。文の位置という意味で近い文節間の係り受けは、遠い文節間の係り受けよりも高い頻度で生じることが分かっているので、この性質をモデルに組み込むために、いくつの文節を受けているかを付加的な情報として加えることとした。また、読点を含む文節は、それに先行する文節の大半を受けることが多いことが分かっている。この性質をモデルに組み込むために読点を含む文節を受けた数も付加的な情報として加えることとした。データスパースネスの問題に対処するために、これらの数には上限を設けた。受けた文節の数と受けた読点を含む文節の数をそれぞれ d, v とすると、終端記号の集合 T と非終端記号の集合 V は以下のように表される (図7.1参照)。

$$\begin{aligned} T &= attrib(b) \times \{0\} \times \{0\} \\ V &= attrib(b) \times \{1, 2, \cdots, d_{max}\} \times \{0, 1, \cdots, v_{max}\} \end{aligned}$$

ここで、終端記号には係る文節がないという点に注意しなければならない。この結果、生成規則は以下のような形式になる。ただし、開始記号 S からの生成は例外である。また、a は文節の属性を表すとする。

$$S \Rightarrow \langle a, d, v \rangle \tag{7.2}$$

$$\langle a_1, d_1, v_1 \rangle \Rightarrow \langle a_2, d_2, v_2 \rangle \langle a_3, d_3, v_3 \rangle \tag{7.3}$$

$$
\begin{aligned}
a_1 &= a_3 \\
d_1 &= \min(d_3+1,\ d_{max}) \\
v_1 &= \begin{cases} \min(v_3+1,\ v_{max}) & \text{if } cont(a_2) = \text{用言} \\ & \quad \wedge\ sign(a_2) = \text{読点} \\ v_3 & \text{if otherwise} \end{cases}
\end{aligned}
$$

ある文の属性列は、開始記号にこれらの生成規則を何回か適用して生成される。各生成規則には確率が付与されており、属性列の生成確率はこれらの積となる。この生成確率は、図7.1の例では、以下のように計算される。ただし、d_{max} および v_{max} は十分大きいとする。

$$
\begin{aligned}
&P(\langle \text{名詞,NULL,読点,0,0}\rangle\langle \text{名詞,助詞,NULL,0,0}\rangle\langle \text{動詞,語尾,句点,0,0}\rangle) \\
&= P(S \Rightarrow \langle \text{動詞,語尾,句点,2,0}\rangle) \\
&\quad \times P(\langle \text{動詞,語尾,句点,2,0}\rangle \Rightarrow \langle \text{名詞,NULL,読点,0,0}\rangle\langle \text{動詞,語尾,句点,1,0}\rangle) \\
&\quad \times P(\langle \text{動詞,語尾,句点,1,0}\rangle \Rightarrow \langle \text{名詞,助詞,NULL,0,0}\rangle\langle \text{動詞,語尾,句点,0,0}\rangle)
\end{aligned}
$$

生成規則の確率値は、係り受けが付与されたコーパスからその頻度を計数し、以下の式を用いて最尤推定することで得られる。

$$
\begin{aligned}
&P(S \Rightarrow \langle a_1,\ d_1,\ v_1\rangle) \\
&\quad \underset{MLE}{=} \frac{N(S \Rightarrow \langle a_1,\ d_1,\ v_1\rangle)}{N(S)} \\
&P(\langle a_1,\ d_1,\ v_1\rangle \Rightarrow \langle a_2,\ d_2,\ v_2\rangle\langle a_3,\ d_3,\ v_3\rangle) \\
&\quad \underset{MLE}{=} \frac{N(\langle a_1,\ d_1,\ v_1\rangle \Rightarrow \langle a_2,\ d_2,\ v_2\rangle\langle a_3,\ d_3,\ v_3\rangle)}{N(\langle a_1,\ d_1,\ v_1\rangle)}
\end{aligned}
$$

本研究では用いていないが、係り受けが付与されていないコーパスからのパラメータ推定の方法として、Inside-Outside アルゴリズム[44]と呼ばれる方法がある。

7.3 低頻度事象への対処

確率文脈自由文法にも、補間を導入することができる。文法 G_1 と文法 G_2 による生成規則の確率をそれぞれ P_1, P_2 とすると、これらを補間し

た確率Pは以下の式で与えられる。ただし、$A \in V$かつ$\alpha \in (V \cup T)^*$である。

$$\begin{aligned} P(A \Rightarrow \alpha) &= \lambda_1 P_1(A \Rightarrow \alpha) + \lambda_2 P_2(A \Rightarrow \alpha) \\ \text{ただし } 0 \leq \lambda_j \leq 1 \ (j = 1,2) &\text{ かつ } \lambda_1 + \lambda_2 = 1 \end{aligned} \tag{7.4}$$

文法G_1として文法G_2よりも凡化レベルの高い文法を選択すれば、文法G_2の低頻度事象の問題に対処していることになる。補間係数の値は、形態素n-gramモデルや文字n-gramモデルの場合と同じように、Held-out法や削除補間法にによって求めることができる。

7.4 形態素クラスタリング

これまでに説明したモデルは、文節の属性として内容語や附属語の品詞を用いていたが、これを形態素やそのクラスに変更することで、予測精度が向上すると考えられる。すでに説明したモデルは品詞というクラスの特別な例に基づいているので、文節の属性として内容語や附属語のクラスを用いるモデルへの変更の必要はなく、単に式(7.1)の関数$last$を形態素列$\boldsymbol{m}$の最後の形態素のクラスを返すように変更すればよい。

7.4.1 目的関数

形態素クラスタリングの目的は、形態素n-gramの場合と本質的には同じであり、クロスエントロピーという観点でより良い言語モデルを構成することである。したがって、クラス分類の目的関数は、以下の式のように削除補間を応用することで得られる平均クロスエントロピーである。形態素n-gramの場合と異なるのは、モデルMだけである。

$$\overline{H} = \frac{1}{m} \sum_{i=1}^{m} H(L_i, M_i) \tag{7.5}$$

ここで、M_iはi番目以外の$m-1$の部分コーパスから推定された係り受けモデル(補間係数の推定も含む)であり、L_iはi番目の部分コーパスを表す。

7.4.2 アルゴリズム

アルゴリズムは、形態素 n-gram の場合と異なり、トップダウンである。つまり、初期状態では同一の品詞に属する形態素は一つのクラスとなっており、繰り返し部分では各形態素の分離を試みる。形態素 n-gram の場合と同様に、計算量という観点から最適解を選択するということは不可能なので、貪欲アルゴリズムを用いることにした。このアルゴリズムは以下の通りである (図7.2参照)。なお、$\overline{H}$ は式 (7.5) で与えられる平均クロスエントロピーである。

$\mathcal{M}$ を頻度の降順に並べ $m_1, m_2, \ldots, m_n$ とする
$c_1 := \{m_1, m_2, \ldots, m_n\}$
$C = \{c_1\}$
foreach i $(1, 2, \cdots, n)$
  $f(m_i) := c_1$
foreach i $(1, 2, \cdots, n)$
  $c := \mathbf{argmin}_{c \in C \cup c_{new}} \overline{H}(move(f, m_i, c))$
  if $(\overline{H}(move(f, m_i, c)) < \overline{H}(f))$ **then**
    $f := move(f, m_i, c)$
    if $(c = c_{new})$ **then**
      $C := C \cup \{c_{new}\}$

計算量は、二番目の **foreach** での繰り返しの回数は形態素数 $|M|$ に比例し、**argmin** での繰り返しの回数はクラス数 $|C|$ に比例するので、全体で $O(|M| \cdot |C|)$ である。クラス数 $|C|$ は、全ての形態素が独立したクラスに分けられる場合に最大 $(|C| = |M|)$ となり、全ての形態素が同一のクラスとなる場合に最小 $(|C| = 1)$ となる。従って、初期化における全体の計算量は、最良の場合が $O(|M|)$ であり、最悪の場合が $O(|M|^2)$ である。ただし、形態素の並べ替えや一番目の **foreach** の計算量は係数が非常に小さいと考えられるので、考慮に入れていない。

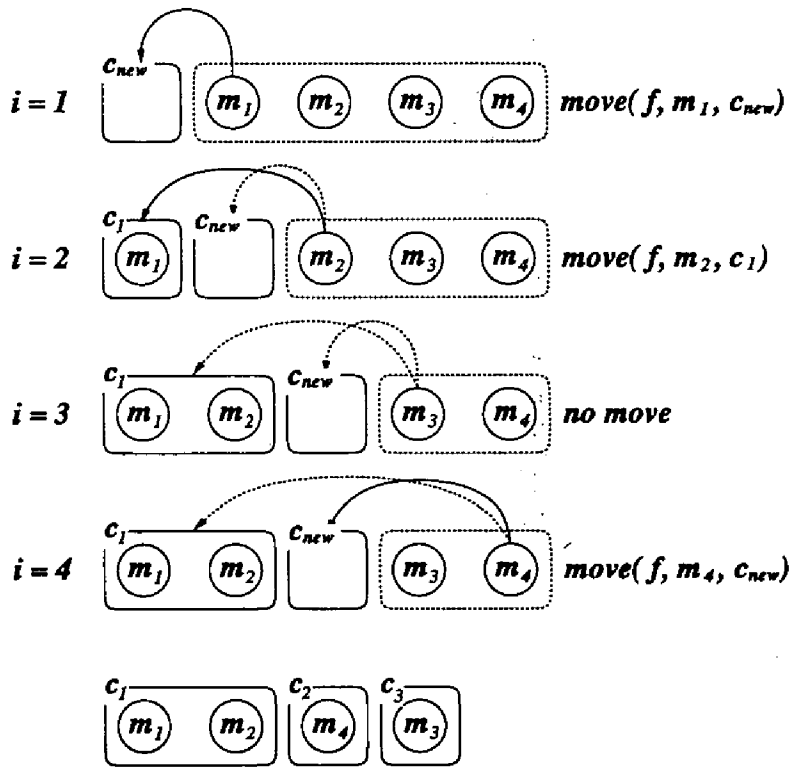

図 7.2: クラスタリングの概念図

7.5 評価

以上で説明した係り受けモデルを構成し、クロスエントロピーを計算した。この節では、この結果を提示し、それに対する考察を述べる。

7.5.1 実験の条件

実験に用いたコーパスは、文字 n-gram モデルや形態素 n-gram モデルの実験と異なり、少し小さくなっている。これは、あらかじめコーパスに付加された構造を係り受けに変換できない文が存在したからである。表7.1は各コーパスの大きさと1文あたりの平均文節数である。アルファベット数は、文字 n-gram や形態素 n-gram モデルの実験と同じ6,879とし

表 7.1: 実験に用いたコーパス(係り受けモデル)

用途	文数	文節数	文節数 / 文数
学習	174,524	1,610,832	9.23
評価	19,397	178,415	9.20

ている。モデルの説明では可変であった d_{max} と v_{max} は共に1とした。これらを平均クロスエントロピーを基準として学習することも可能であるが、以下の実験では固定である。

テストコーパスの係り受けは、コーパスにあらかじめ付加されたものを用いた。したがって、テストコーパスに含まれる文字列の出現確率は、その文字列のすべての生成方法による確率を合計した値ではなく、コーパスに示された生成方法のみによる値である(コーパス近似)。なお、係り受けが明示されていない文に対しては、動的計画法を用いたアルゴリズムにより、出現確率が最大となる係り受けとその確率値(Viterbi近似)を、文に含まれる文字数の3乗に比例する時間で求めることができる。これは、第8章で述べる構文解析である。

品詞係り受けモデルとクラス係り受けモデルを比較するために、これらを同じ学習コーパスから構成し、同じテストコーパスに対してクロスエントロピーを計算した。それぞれの言語モデルの構成の手順は以下の通りである。なお補間に用いた確率文脈自由文法は、生成規則の確率が等確率分布であること以外は係り受けモデルの文法と同じである。

- 品詞係り受けモデル

  1. 削除補間により式(7.4)の補間係数を推定
  2. すべての学習コーパスを対象に生成規則の頻度を計数

- クラス係り受けモデル

  1. 削除補間により式(7.4)の補間係数を推定

表 7.2: 形態素クラスタリングによる係り受けモデルの改善の結果

言語モデル	終端記号数	クロスエントロピー
品詞係り受けモデル	576	5.3536
クラス係り受けモデル	10,752	4.9944

2. 前節で述べた方法でクラス関数を推定
3. 削除補間により式(7.4)の補間係数を再推定
4. すべての学習コーパスを対象に生成規則の頻度を計数

各モデルに含まれる文節モデルと未知語モデルは、第4章で説明した形態素2-gramモデルと文字2-gramモデルである。各文節の主辞の形態素の予測までが係り受けモデルに含まれるとすれば、この部分のクロスエントロピーへの寄与は一定である。

7.5.2 評価実験

表7.2は各モデルのテストコーパスのクロスエントロピーである。クラス係り受けモデルは、品詞係り受けモデルよりも低いクロスエントロピーとなっている。このことから、提案手法による形態素クラスタリングは係り受けモデルにも有効であり、これにより推定されたクラスによるクラス係り受けモデルが、品詞係り受けモデルよりも、予測力という点で良い言語モデルであることが分かる。なお、形態素n-gramモデルを仮定してクラスタリングした結果得られるクラスを用いた場合のクロスエントロピーは6.3358であった。この結果は、形態素のクラスは目的に応じて大きく異なることを意味する。

付録A.2は得られたクラスタの例である。形態素2-gramモデルにおけるクラスタリング結果と同様、多くのクラスタが**クラスタ1**のように我々の言語直観に照らし合わせて、納得できるクラスタであった。一方、**クラスタ2**のように我々の言語直観に合致しないクラスタもあった。これは、我々が行なった形態素クラスタリングは、クラス係り受けモデルの

表 7.3: クロスエントロピーの内訳

モデルの部分	クロスエントロピー
文節の主辞の予測	3.6652
形態素予測	0.8700
未知語の文字予測	0.4592
合計	4.9944

改善という観点からのクラスタリングであることと、得られた形態素の分類が準最適解であることを考えると特に不自然ではないであろう。

クロスエントロピーの分枝性[21]から、代入によって多段になっている確率的モデルによるクロスエントロピーは、各部分の寄与に分解されるので、独立に計算することができる。係り受けモデルは、文節の主辞以外の形態素列を予測する部分と、未知語の文字列を予測する部分に分解できる。表7.3は、このようにして計算したクロスエントロピーの各部分の内訳である。この結果を見ると、テストコーパスに対するクロスエントロピーには、文節の主辞を予測する部分がかなり大きく寄与していることが分かる。よって、短期的により良い言語モデルを構成するためには、この部分を改良することが近道であると考えられる。形態素クラスタリングによるクラス係り受けモデルの構成は、この部分を有意に改善している。長期的には、文節の主辞以外の形態素列を予測する部分や未知語モデルを改善することが望ましい。これには、各生成規則の確率を、先行する生成規則の条件付き確率とする[45]ことや、複数の生成規則を統合する[46]ことが考えられる。

7.6 結論

この章では、まず、文節の属性を利用した係り受けモデルを提案した。このモデルにより、未知語から係り受けまでを一貫してモデル化しているので、係り受けまでの言語現象を考慮にいれた音声認識や読み推定な

どを同時に行うことができる。次に、モデルの改善方法として、このモデルに対して準最適なクラス分類を求めるアルゴリズムについて述べた。このアルゴリズムは、クラス推定のためのコーパスを係り受けモデルの推定用のコーパスとは別に用意するというアイデアに基づいている。実験の結果、クラスタリングによる予測力の向上が観測された。これは、上述の応用を行った場合の精度向上につながる。

第 8 章

構文解析

日本語に対する構文解析とは、日本語の文(文字列)を入力とし、これを文節に分割すると同時に文節間の係り受け関係を決定する処理である。この章では、これを実現する手法の一つとしての確率的構文解析とその基礎となる係り受けモデルにおける最尤解の探索方法について述べる。

8.1 確率的構文解析

日本語の構文解析は、日本語のアルファベット $\mathcal{X}$ のクリーネ閉包に属する文字列 $x \in \mathcal{X}^*$ を入力として、これを文節に分割し、それらの分割間の係り受け関係を出力することと定義できる。このとき、出力される文節列の表記の連接は、入力のアルファベット列に等しくなければならない。一般に、これを満たす解は一意ではない。構文解析の問題は、可能な解の中から人間の判断(正解)に最も近いと推測される構文を選択し出力することである。この選択の基準としては、文法家が自身の言語直観を頼りにした規則に基づく方法と大量の正解例(構文解析済みコーパス)からの推定を規準にする方法がある。以下では、後者の一つである確率的構文解析について説明する。

確率的構文解析器は、係り受けという概念を内包する確率的言語モデルを基にして、与えられた文字列 x に対する確率最大の構文木(図7.1参照)を計算し出力する。これは、以下の式で表される。ただし、$w(T)$ は

構文木 T の文節列の表記の連接を表す。

$$\begin{aligned}\hat{T} &= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\operatorname{argmax}} P(T|\boldsymbol{x}) \\ &= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\operatorname{argmax}} P(T|\boldsymbol{x})P(\boldsymbol{x}) \quad (\because P(\boldsymbol{x})はTによらない) \\ &= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\operatorname{argmax}} P(\boldsymbol{x}|T)P(T) \quad (\because ベイズの公式) \\ &= \underset{\boldsymbol{w}(T)=\boldsymbol{x}}{\operatorname{argmax}} P(T) \quad (\because P(\boldsymbol{x}|T)=1)\end{aligned}$$

この式の最後の $P(T)$ が係り受けという概念を内包する確率的言語モデルである。このようなモデルとして、第7章で説明した品詞係り受けモデルやクラス係り受けモデルを用いることができる。

8.2 解探索のアルゴリズム

構文解析に用いる確率的言語モデルは、式(7.2)の開始記号からの導出を除いて、式(7.3)のような Chomsky 標準形[47, 48]に制限されている。したがって、解探索のアルゴリズムには動的計画法[26]の一種である CYK 法[47]を、確率文脈自由文法に拡張したアルゴリズム[49]を用いることができる。ただし、例外である開始記号からの導出確率を最後に掛ける必要がある。CYK 法による文脈自由文法の構文解析の計算量は、入力の記号数を n として $O(n^3)$ である。確率を扱う拡張は、CYK 表に非終端記号とともにそこから部分文字列が生成される確率を記憶しておくことで実装されるので、計算量には影響しない。したがって、確率的構文解析の計算量は $O(n^3)$ である。文全体の生成確率を計算するためには、各終端記号の生成確率の初期値として、文節の属性から文節が生成される確率を与えておけばよい。

8.3 評価

この節では、係り受けモデルを用いた構文解析の結果を提示することに加えて、形態素クラスタリング(第7章)による解析精度の向上について述べる。

表 8.1: 評価基準の説明のための例

文節番号	係り先(正解)	係り先(結果)	文節
1	2	4	今日/名詞 と/助詞
2	4	4	明日/名詞 、/記号
3	4	4	京都/名詞 大学/名詞 へ/助詞
4	-	-	行/動詞 く/語尾 。/記号

8.3.1 評価基準

我々が用いた評価基準は、文節単位の係り受けの正解率である。ただし、最後の文節は係り先を持たず、その直前の文節は必ず最後の文節に係るので、これらを評価の対象としていない。例として、コーパスの内容と解析結果が表8.1のような場合を考える。この例では、4つの文節がある。このうち、係り先があらかじめコーパスに与えられた正解と一致した文節は、文節2と文節3である。文節3は、評価の対象ではないので、2つのうち1つが正解であったことになる。よって、この例の正解率は1/2となる。

8.3.2 実験の条件

実験に用いたコーパスは、第7章と同じである。第7章で提案したモデルを用いれば、未知語処理や形態素解析なども含めた、文字列からの構文解析一括して行うことができる。しかし、この節で述べる実験では、文節列を入力としている。形態素クラスタリングの結果の品詞毎の傾向は、モデルとして形態素 n-gram モデルを用いた場合と類似していた。つまり、各クラスの平均要素数は、内容語において高く、機能語において低いことが観測される。このことから、品詞係り受けモデルにおいては、とくに機能語を一般化し過ぎていることが分かる。

8.3.3 構文解析の精度の評価

表8.2: 形態素クラスタリングによる係り受けモデルの改善の結果

言語モデル	クロスエントロピー	解析精度
品詞係り受けモデル	5.3536	68.77%
クラス係り受けモデル	4.9944	81.96%
無条件に次の文節を選択	-	53.10%

表8.2は、品詞係り受けモデルとクラス係り受けモデルによるクロスエントロピーと構文解析の精度である。この結果から、形態素クラスタリングは、構文解析の精度を向上させることが分かる。これは、形態素解析の場合と同様に、クロスエントロピーの減少から予測される通りの結果である。

藤尾ら[50]は、分類語彙表[51]のクラスレベルでの係り受けと形態素レベルでの係り受けの統計結果を補間した統計的モデルによる構文解析を提案し、EDRコーパスを用いた実験の結果として80.48%の解析精度を報告している。学習コーパスとテストコーパスが全く同じというわけではないので直接的な比較はできないが、我々のモデルによる解析精度はこれを有意に上回っている。これに加えて、本手法のモデルの利点は、未知語処理(表記からの品詞の推定)や形態素解析を同時に実行できることである。これは、係り受けの優先規則を人手で与える構文解析手法に対する優位性でもある。確率的方法のより根本的な利点は、確率的言語モデルは応用に依存しないので、第2章で例示した、音声認識をはじめとする認識系や読み付与などの解析系などのあらゆる応用に用いることができ、形態素クラスタリングなどの確率的言語モデルの予測力の改善は、これらの応用全ての精度を向上させることである。予測力の改善には、各部分への独立した取り組みが可能であるので、組織的な取り組みが可能な点も利点である。

8.4 結論

この章では、係り受けモデルを用いた構文解析を提案し、形態素クラスタリングよる予測力と構文解析の精度向上について述べた。形態素クラスタリングとしては、形態素n-gramの場合と同じように、クロスエントロピーを基準として行った。これにより、構文解析の精度の向上が観測された。形態素クラスタリングの結果得られたクラスに基づいたクラス係り受けモデルによる構文解析の精度は、先行研究として報告されている構文解析の精度を上回った。これは、我々のモデルが構文解析の精度という点で優れていることを示す結果である。

第 9 章

むすび

本論文では、確率的言語モデルのコーパスからの推定方法を述べ、日本語のエントロピーの上限を推定した。確率的言語モデルはまた、自然言語の認識系や解析系の共通の部分であり、応用上重要な位置を占める。これを改善することは、これらの応用の精度を同時に向上させることにつながるので非常に有用である。本論文では、この要求に答えるために、確率的言語モデルの改善方法を提案した。改善方法の中心的アイデアは、削除補間法を応用することで定義できる学習コーパス内での平均クロスエントロピーを下げることを目標として、モデルの可変部分のパラメータを決定することである。音声認識や読み付与の応用で一般的に用いられている確率的言語モデルである形態素 n-gram モデルの改善方法として、形態素クラスタリングとその結果を用いたクラス n-gram モデルを提案した。これを実装し実験した結果、予測力の有意な向上が観測された。応用の一例としての形態素解析の精度も同時に向上した。

形態素 n-gram モデルでは、連続しない要素間の関係を記述することができないので、このモデルクラスでの改善には上限がある。今後、より複雑な言語現象に対応し、高精度な音声認識や読み付与を実現するためには、構文情報を用いることが必要だと考えられる。このような観点から、本論文では、係り受けを確率的文脈自由文法でモデル化し、構文を考慮に入れた確率的言語モデルを提案した。さらに、この改善方法として形態素クラスタリングとその結果を用いた係り受けのモデルを提案した。これを実装し実験した結果、予測力の有意な向上とともに、応用の

一例としての構文解析の精度向上も観測された。

係り受けのモデルは、文節の属性を終端記号とする確率文脈自由文法であるが、文節の実際の形態素列を記述するモデルとして形態素n-gramモデルを、未知語の文字列を記述するモデルとして文字n-gramモデルを内包している。今後、さらに複雑な言語現象をモデル化し、確率的手法による認識系や解析系の精度向上が望まれる。これには、世界知識を確率的なネットワークとして保持しておくことや、照応などの現象に対応するために、文脈に応じてこれを動的に変更することなどが考えられる。このような確率的モデルは、文のモデルとして係り受けのモデルをその一部とするであろう。また、モデルに可変の部分を設け、この部分の具体的な値を推定するという方法でモデルの改善が図られるであろう。本論文は、このようなより複雑な確率的言語モデルの基礎となる。

参考文献

1) C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, 1948.

2) C. E. Shannon. Prediction and Entropy of Printed English. *Bell System Technical Journal*, Vol. 30, pp. 50–64, 1951.

3) Noam Chomsky. Three Models for the Description of Language. *IRE Transaction on Infomation Theory*, Vol. 2, No. 3, pp. 113–124, 1956.

4) John E. Hopcroft and Jeffrey D. Ulman. *Introduction to Automata Theory, Language and Computation.* Addison-Wesley Publishing, 1979.

5) King Sun Fu. *Syntactic Methods in Pattern Recognition*, Vol. 12 of *Matehmatics in Science and Engineering.* ACCADEMIC PRESS, 1974.

6) C. S. Wetherell. Probabilistic Languages: A Review and Some Open Questions. *ACM Computing Surveys*, Vol. 12, No. 4, pp. 361–379, 1980.

7) 韓太舜, 小林欣吾. 情報と符号化の数理. 岩波書店, 1994.

8) Lalit R. Bahl, Frederick Jelineck, and Robert L. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 2, pp. 179–190, 1983.

9) Hermann Ney, Ute Essen, and Reinhard Kneser. On Structuring Probabilistic Dependences in Stochastic Language Modeling. *Computer Speech and Language*, Vol. 8, pp. 1–38, 1994.

10) Mark Nelson. データ圧縮ハンドブック. トッパン, 1994.

11) 西村雅史, 伊東伸泰, 山崎一孝, 荻野紫穂. 単語を認識単位とした日本語ディクテーションシステム. 情報処理学会研究報告, 第SP15巻, pp. 27–34, 1997.

12) Satoshi Sekine. The Domain Dependence of Parsing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 96–102, 1997.

13) F. Jelinek. Self-Organized Language Modeling for Speech Recognition. Technical report, IBM T. J. Watson Research Center, 1985.

14) L. E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Process. *Inequalities*, Vol. 3, pp. 1–8, 1972.

15) 日本電子化辞書研究所. EDR 電子化辞書仕様説明書, 1993.

16) Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A Practical Part-of-Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp. 133–140, 1992.

17) David Elworthy. Does Baum-Welch Re-estimation Help Taggers? In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pp. 53–58, 1994.

18) Bernard Merialdo. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, Vol. 20, No. 2, pp. 155–171, 1994.

19) 竹内孔一, 松本裕治. HMM による日本語形態素解析システムのパラメータ学習. 情報処理学会研究報告, 1995.

20) 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, 第96-NL-116巻, 1996.

21) 堀部安一. 情報エントロピー論. 森北出版, 第2版, 1997.

22) Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479, 1992.

23) Fredelick Jelinek, Robert L. Mercer, and Salim Roukos. Principles of Lexical Language Modeling for Speech Recognition. In *Advances in Speech Signal Processing*, chapter 21, pp. 651–699. Dekker, 1991.

24) R. Kneser and H. Ney. Improved Clustering Techniques for Class-Based Statistical Language Modelling. In *Eurospeech*, pp. 21–23, 1993.

25) Mitchell P. Marcus and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

26) Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

27) H. Ney. The Use of One-Stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 32, No. 2, pp. 263–271, 1984.

28) Masaaki Nagata. A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A* N-Best Search Algorithm. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 201–207, 1994.

29) Alfred V. Aho. Algorithms for Finding Patterns in Strings. In *Handbook of Theoretical Computer Science*, Vol. A: Algorithms and Complexity, pp. 273–278. Elseveir Science Publishers, 1990.

30) Hiroshi Maruyama. Backtracking-Free Dictionary Access Method for Japanese Morphological Analysis. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 208–213, 1994.

31) 森信介. DFA による形態素解析の高速辞書検索. EDR 電子化辞書利用シンポジウム, 1997.

32) 永田昌明. EDR コーパスを用いた確率的日本語形態素解析. EDR 電子化辞書利用シンポジウム, pp. 49–56, 1995.

33) 森信介, 長尾眞. n グラム統計によるコーパスからの未知語抽出. 情報処理学会研究報告, 1995.

34) Steven J. DeRose. Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, Vol. 14, No. 1, pp. 31–39, 1988.

35) Evangelos Dermatas and George Kokkinakis. Automatic Stochastic Tagging of Natural Language Texts. *Computational Linguistics*, Vol. 21, No. 2, pp. 137–163, 1995.

36) Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. Equations for Part-of-Speech Tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 784–789, 1993.

37) Carl G. de Marcken. Parsing the LOB corpus. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 243–251, 1990.

38) 松本裕治, 黒橋禎夫, 山地治, 妙木裕, 長尾真. 日本語形態素解析システム JUMAN 使用説明書 version 3.2. 京都大学工学部長尾研究室, 1997.

39) 黒橋禎夫, 長尾眞. 京都大学テキストコーパス・プロジェクト. In *Proceedings of the Third Annual Meeting of the Association for Natural Language Processing*, pp. 115–118, 1997.

40) Hinrich Schütze and Yoram Singer. Part of Speech Tagging Using a Variable Memory Markov Model. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 181–187, 1994.

41) 春野雅彦, 松本裕治. 文脈木を利用した形態素解析. 情報処理学会研究報告, 1996.

42) Roland Kuhn and Renato de Mori. A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6, pp. 570–583, 1990.

43) John G. McMahon and Francis J. Smith. Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies. *Computational Linguistics*, Vol. 22, No. 2, pp. 217–247, 1996.

44) John D. Lafferty. A Derivation of the Inside-Outside Algorithm from the EM Algorithm. Technical report, IBM T. J. Watson Research Center, 1993.

45) Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer, and Salim Roukos. Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 31–37, 1993.

46) Satoshi Sekine and Ralph Grishman. A Corpus-based Probabilistic Grammar with Only Two Non-terminals. In *The Forth International Workshop on Parsing Technologies*, pp. 216–223, 1995.

47) John E. Hopcroft and Jeffrey D. Ulman. オートマトン 言語理論 計算論 I. サイエンス社, 1984.

48) John E. Hopcroft and Jeffrey D. Ulman. オートマトン 言語理論 計算論 II. サイエンス社, 1984.

49) 北研二, 中村哲, 永田昌明. 音声言語処理. 森北出版, 1996.

50) 藤尾正和, 松本裕治. 統計的手法を用いた係り受け解析. 情報処理学会研究報告, 第96-NL-117巻, 1997.

51) 国立国語研究所. 分類語彙表, 秀英出版, 1993.

発表リスト

研究会

1. 長尾 眞, 森 信介. 大規模日本語テキストのnグラム統計の作り方と語句の自動抽出. 情報処理学会研究報告, pp. 1–8, 1993.

2. 森 信介, 長尾 眞. nグラム統計によるコーパスからの未知語抽出. 情報処理学会研究報告, 1995.

3. 森 信介, 長尾 真. 統計によるタグ付きコーパスからの統語規則の獲得. 情報処理学会研究報告, 1995.

4. 森 信介, 長尾 眞. 形態素bi-gramと品詞bi-gramの重ね合わせによる形態素解析. 情報処理学会研究報告, 1996.

5. 森 信介, 長尾 眞. 語彙化マルコフモデルによる英語品詞タグ付け. 電子情報通信学会技術研究会報告, pp. 31–38, 1995.

6. 森 信介. DFAによる形態素解析の高速化. 情報処理学会研究報告, 1996.

7. 森 信介. クラスbigram言語モデルの補間. 情報処理学会研究報告 96-NL-118, 1997.

8. 森 信介. DFAによる形態素解析の高速辞書検索. EDR電子化辞書利用シンポジウム, 1997.

9. 森 信介, 長尾 真. 係り受けを用いた確率的言語モデル. 情報処理学会研究報告 96-NL-122, 1997.

10. 森 信介, 山地 治, 長尾 真. 予測単位の変更による単語n-gramモデルの改善. 情報処理学会研究報告 1997 (掲載予定).

国際学会

1. Makoto Nagao and Shinsuke 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. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 611–615, 1994.

2. Shinsuke Mori and Makoto Nagao. Parsing Without Grammar. In *The Forth International Workshop on Parsing Technologies*, pp. 174–185, 1995.

3. Shinsuke Mori and Makoto Nagao. Word Extraction from Corpora and Its Part-of-Speech Estimation Using Distributional Analysis. In *Proceedings of the 16th International Conference on Computational Linguistics*, 1996.

論文

1. 森 信介, 長尾 眞. タグ付きコーパスからの統語規則の獲得. 情報処理学会論文誌, Vol. 37, No. 9, 1996.

2. 森 信介, 山地 治. 日本語の情報量の上限の推定. 情報処理学会論文誌, Vol. 38, No. 11, 1997.

3. 森 信介, 西村 雅史, 伊東 伸泰. クラスに基づく言語モデルのための単語クラスタリング. 情報処理学会論文誌, Vol. 38, No. 11, 1997.

4. 森 信介, 長尾 眞. 形態素クラスタリングによる形態素解析精度の向上. 自然言語処理 Vol. 5, No. 2, 1998 (掲載予定).

付録 A

得られたクラスタの例

ここでは、第5章と第7章で述べたクラスタリングの結果得られたクラスの例を示す。それぞれのクラスタリング実験では、クラスタリングの基準が異なる。したがって、得られるクラスの性質は異なると考えられる。

A.1 クラス n-gram モデルを基準とした実験結果

クラスタ 1

[人 / 接尾語 件 / 接尾語 カ所 / 接尾語 平方メートル / 接尾語 点 / 接尾語 隻 / 接尾語 頭 / 接尾語 匹 / 接尾語 戸 / 接尾語 基 / 接尾語 世帯 / 名詞 校 / 接尾語 ヘクタール / 接尾語 羽 / 接尾語 棟 / 接尾語 元 / 接尾語 票 / 接尾語 世帯 / 接尾語 キロ / 名詞 リットル / 接尾語 席 / 接尾語 桁 / 接尾語 巻 / 接尾語 編 / 接尾語 ｋｍ / 接尾語 曲 / 接尾語 ｍｍ / 接尾語 マルク / 接尾語 Ｋ / 接尾語 品目 / 接尾語 床 / 接尾語 ミクロン / 接尾語 ヵ所 / 接尾語 つがい / 名詞 ズロチ / 接尾語 首 / 接尾語 筆 / 接尾語 ＮＺドル / 接尾語 ＫＨｚ / 接尾語]

クラスタ 2

[の / 助詞 や / 助詞 および / 接続詞 及び / 接続詞 ないし / 接続詞 ならびに / 接続詞 もしくは / 接続詞 イコール / 接続詞

たる / 助動詞 らしい / 接尾語 カタロニア / 名詞 や / 接尾語
質 / 接尾語 はじめ / 接尾語 テラビア / 名詞 中心 / 接尾語]

A.2 クラス係り受けモデルを基準とした実験結果

クラスタ 1

[る / 語尾 た / 語尾 した / 語尾 える / 語尾 ます / 語尾]

クラスタ 2

[ば / 助詞 ても / 助詞 たり / 助詞 ながら / 助詞 だり / 助詞
ども / 助詞 たら / 助詞 ところで / 助詞 きゃ / 助詞 なら /
助詞 ては / 助詞 ったら / 助詞 やいなや / 助詞]