学習データ中のノイズによる言語モデルへの影響

執筆者：徳永 崇

指導教官：新納 浩幸

平成１３年３月２日

目次

Chapter 1

序論

1.1 はじめに

自然言語を機械的に処理する際、処理の基準となる言語モデルが必要となる。高い精度をもつ自然言語処理システムを構築するためには、優れた言語モデルが必要であり、言語モデルの精度向上が望まれている。本論文では、言語モデルの精度向上を考える際に、注意を要すると思われる点を1つ取り上げ、考察する。

ここでいう言語モデルとは、N-gram モデルをはじめとする確率的言語モデルのことである。言語モデルの生成には多量のテキストを含むデータが必要となり、言語モデルの生成の際に利用されるこのようなデータのことを特に学習データという。日本語の言語モデルの生成には大量の日本語学習データが必要となり、英語の言語モデルの生成には大量の英語学習データが必要となる。一般に、学習データ量が多いほど、精度の面で良好な言語モデルが得られる。しかし学習データの質の面から考えると、言語モデルの生成に有効に作用すると思われるデータ、つまり言語モデルの精度向上に結びつくデータが存在するほかに、言語モデル生成の際に言語モデルの精度向上には結び付かないデータが存在すると考えられる。後者のようなデータをここではノイズと呼び、本論文において注目すべきデータであると考える。

利用する学習データが異なれば、その学習データ中でノイズとして扱われるデータの姿も異なると考えられる。本論文では新聞記事を学習データとして用いている。新聞記

事中におけるノイズの具体例としては、図や写真など明らかに言語モデルの生成に役立たないと考えられるデータのほか、表やグラフなど部分的に単語列として成り立っているが文章として成立するには至っていないデータなどが挙げられる。これらのデータをノイズとして扱い、学習データ中から除去することは比較的容易であり、これまでも言語モデルの生成には不要なものとして扱われてきた。そこで、本論文では文章中の単語列からノイズに該当する単語列を選定し除去することを考える。テキストデータの中にも言語モデルに影響を与える単語列が存在することを考え、学習データからそれらを除去することにより、より一層の言語モデルの精度向上を試みる。

本論文では、言語モデルを生成する際に、学習データ量が増加しても総データ量に対するノイズの割合が大きければ、生成される言語モデルの精度の向上は望めず、場合によっては精度が低下することを確認し、その結果を通して学習データに含まれるノイズと、そのノイズが与える言語モデルへの影響について考える。

1.2 本論文の構成

本論文は最初に、茶筅を用いた形態素解析による単語分割について(第 2 章)、確率的言語モデルである N-gram モデルについて(第 3 章)、言語モデルと学習データの関係について(第 4 章)、言語モデルの精度と学習データ中のノイズに関する実験(第 5 章)及び考察(第 6 章)、実験結果に対するまとめ(第 7 章)へと進む。

また、巻末にはプログラムのソースリストを添付した。

Chapter 2

形態素解析の利用について

2.1 日本語の形態素解析と単語分割

与えられた文を単語に分割し、各単語に品詞や語形変化等の情報を与える処理を形態素解析(morphological analysis)と呼ぶ。英語の場合には、単語が空白により区切られているために、単語を容易に同定することができるが、日本語、中国語等においては、単語間に空白を入れるという習慣がないため、単語の同定が容易ではない。従って、日本語の形態素解析では単語分割をおこなう必要がある。

本論文における言語モデルの生成は、学習データを用いて言語モデルを生成する際に、学習データが単語分割されている必要がある。そこで日本語形態素解析ソフト「茶筌」の単語分割機能を利用して学習データを単語分割している。

2.2 茶筌

2.1.1 茶筌とは

茶筌システムは、広く自然言語処理研究に資するため無償のソフトウェアであり、計算機による日本語の解析の研究を目指す多くの研究者に共通して使える形態素解析ツールを提供するために開発された。本システムの原形は、京都大学長尾研究室および奈

良先端技術大学院大学松本研究室において開発された日本語形態素解析システムJUMAN である。JUMAN は、京都大学および奈良先端技術大学院大学のスタッフおよび多くの学生の協力を得て作成されたものである。

2.2.1 茶筌の利用例

表 2.1 に茶筌の利用例を示す。なお、茶筌のバージョンは 2.0 である（version2.0)。形態素解析の実行は一行ずつ行われる。コスト最小（それぞれの形態素の区切りで最小コストとの差が許容されるコスト幅以内）の解を求め、結果を出力する。また、文字コードとしては日本語 EUC あるいは JIS（ISO－2022－JP）で実行可能である。出力時の文字コードは日本語 EUC である。

[例文]　卒業研究に着手するためには次の条件が必要です。

表 2.1：茶筌の出力結果

＜見出し＞	＜読み＞	＜基本形＞	＜品詞＞
卒業	ソツギョウ	卒業	名詞-サ変接続
研究	ケンキュウ	研究	名詞-サ変接続
に	ニ	に	助詞-格助詞-一般
着手	チャクシュ	着手	名詞-サ変接続
する	スル	する	動詞-自立　サ変・スル　基本形
ため	タメ	ため	名詞-非自立-副詞可能
に	ニ	に	助詞-格助詞-一般
は	ハ	は	助詞-係助詞
次	ツギ	次	名詞-一般
の	ノ	の	助詞-連体化
条件	ジョウケン	条件	名詞-一般

茶筌を利用すると上記のような出力結果が得られる。＜見出し＞部分より、文章を単語分割したデータを容易に得ることができる。

Chapter 3

N-gram モデル

3.1 確率的言語モデル

3.1.1 確率的言語モデルについて

言語の数学的モデルとしは形式文法やオートマンが伝統的に用いられているが、これらは基本的に言語を記号列の集合としてとらえている。別の見方をすれば、言語に属する記号列に対しては 1 を、それ以外の記号列に対しては 0 を与えるような 2 値関数により言語を規定している。

一方、単に記号列が言語に属するか否かだけではなく、記号列の生起する確率(記号列の起こりやすさ)も考慮した確率的な言語を考えることができる。記号の有限集合Σに対し、Σ上の確率的言語（probabilistic language）を 2 項組（L, P）により定義する。ここで、LはΣ上の言語であり、PはΣ^*から [0,1] への実数値関数（確立関数）である。また、関数Pは次の条件を満たす。

(1) $x \in \Sigma^*$に対し、$x \notin L \Rightarrow P(x) = 0$

(2) $x \in \Sigma^*$に対し、$x \in L \Rightarrow 0 \leq P(x) \leq 1$

(3) $\Sigma_{x \in L} P(x) = 1$

本論文では、自然言語のモデルに確率的言語を用いる。これにより、文あるいは単語列、文字列などに対して、それらが起こる確率を考えることができるようになるが、これらの確率を与えるモデルのことを確率的言語モデル（probabilistic language model）あるいはと呼ぶ。

確率的言語モデルを用いることによる利点としては、以下のようなものをあげることができる。

(1) 自然言語には、統語的曖昧性や意味的曖昧性など、さまざまなレベルの曖昧性が存在する。確率的な情報を用いることにより、曖昧さのある複数の候補（仮説）の中から、最も確からしいものを選んだり、候補の間の順位付けを行うことができる。また、確率値の小さい候補を枝刈りすることにより、処理の効率化を行うことができる。

(2) 確率的言語モデルの大きな特徴として、言語モデルが自然言語をどれだけ正確に近似しているかという性能評価尺度を情報理論に基づき定義できる点がある。これにより、複数の異なった言語モデルの中から、どの言語モデルが優れているかを調べることができる。

(3) 確率理論や情報理論に基づき、言語モデルのパラメータを自動的に推定することができる。大量の言語データからパラメータを推定することにより、高精度で（accurate)、適用範囲が広く（broad-coverage)、頑健な（robust）モデルを構築することができる。

言語モデル理論の中心的な課題は、自然言語の確率・統計的性質を十分に反映した確率的言語モデルを、与えられた言語データから作成することである。しかし、自然言語は複雑であり、しかも多様性、個別性に富んでいるために、すべてにわたって万能の言語モデルを作ることはほぼ不可能である。言語モデルが使われている目的やあるいはどのような言語現象をモデル化するのかにより、異なった言語モデルを考える必要がある。この際、言語モデルを実際に用いるためには、次のようなアルゴリズムを明らかにしておく必要がある。

(1) 与えられた単語列が言語モデルから生成される確率を計算するアルゴリズム。

(2) 与えられた言語データから言語モデルのパラメータを推定するアルゴリズム。

(3) 場合によっては、単語列を生成した言語モデルの内部状態が問題となることがある。このような場合には、与えられた単語列を生成した言語モデルの最適な内部状態を求めるアルゴリズムが必要となる。たとえば、文脈自由文法において、与えられた単語列に対する最も確からしい導出を求める問題がこの場合に相当する。

3.1.2 代表的な言語モデル

言語モデルのうち、特に重要で代表的であると考えられるものは、N-gram モデル（N-gram model）、隠れマルコフモデル（hidden Markov model; HMM）、確率文脈自由文法（probabilistic context-free grammar）の 3 つである。

N-gram モデルは、情報理論の創始者であるシャノン（Shannon, C. E.）が自然言語情報源の統計的モデルとして用いたことで有名である。現在、各種の応用で最も広範に用いられている言語モデルである。N-gram モデルは、単語の生起を（N−1）重マルコフ過程で近似したモデルであり、モデルのパラメータ推定は最尤推定法に基づいて行われる。具体的には、単語の N 個組と（N−1）個組の相対頻度から求めることができる。

隠れマルコフモデルは、確率・統計的な音声言語処理を広める契機となったモデルであると考えられる。1970 年代に IBM の音声グループが音声認識に隠れマルコフモデルを適用し大きな成功を収めたことに端を発し、現在では隠れマルコフモデルは音響モデルの主流となっている。自然言語処理においても、確率的形態素解析を初めとするさまざまなところに適用されている。

確率文脈自由文法は、自然言語処理で広く用いられている文脈自由文法の生成規則に、その規則の適用される確率を付与したモデルである。

3.2 N-gram モデル

3.2.1 概要

N-gram モデルは、確率・統計的自然言語処理の分野で最も広範に使われている言語モデルである。非常に単純なモデルであり多くの欠点がある反面、非常に強力なモデルでもある。また、N-gram モデルを改良することにより、より強力なモデルを作成しようという試みも種々行われている。このため、N-gram モデルには、さまざまな変種モデルとでもいうべきモデルが存在する。

まず、シャノン・ゲーム（Shannon game）と呼ばれるものを紹介しよう。これは、自然言語の文が途中まで与えられたときに、次にどのような単語がくるかを予測するという問題である。例として、次の文の下線の部分の単語を予測することを考えてみよう。

Nana eats an ___

この例の場合、直前の単語は“an”であるので、下線の部分は母音で始まる名詞と考えられる。さらに2つ前の単語をみると“eats”であるので、下線の部分には食べものがくるらしいと分かる。母音で始まる名詞の食べものとなるとかなり範囲が限定され、下線の部分は“apple”や“orange”ではないかと予測することができる。この例が示すように、単語の予測には直前にある数個の単語の情報が有効であると考えられる。

上の例では、「名詞」や「食べもの」といった概念を用いて単語の予測を行ったが、これを機械的に行うにはどうすればいいのだろうか。最も簡単な方法は、大量の自然言語のテキスト（学習データ）から、各単語が“an”や“eats an”の後に現れる条件付き確率を求め、確率が大きい単語を予測するという方法である。条件付き確率は、学習データ中に出現する単語列の相対頻度から求めることができる。単語列Wの出現頻度を$C(W)$で表すことにすると、たとえば“apple”が“an”あるいは“eats an”の後に現れる確率は次のようになる。

$$P(apple|an) = \frac{C(an\ \ apple)}{C(an)} \tag{3.1}$$

$$P(apple \mid eats\ \ an) = \frac{C(eats\ \ an\ \ apple)}{C(eats\ \ an)} \tag{3.2}$$

式（3.1）のように、条件付き確率を求める際に、直前の単語のみを用いるモデルをバイグラム（bigram）と呼ぶ。また、式（3.2）のように、直前の2つの単語を用いるモデルをトライグラム（trigram）と呼ぶ。

一般に、N-gram モデルとは、単語の生起が直前の（N−1）単語にのみ依存すると考えた言語モデルである。言い換えると、ある時点でどのような単語が選ばれるかは、直前にある（N−1）個の単語の影響のみを受け、N 単語より前にある単語の影響はまったく受けないと考える。しかし、人が言葉を話したり書いたりする際に、どのような単語を選ぶかは、対話の流れや文の構文や意味、あるいは人それぞれの好みなどのさまざまな影響を受けており、直前の限られた範囲にある単語の影響だけを受けているとは考えがたい。一見、N-gram モデルの仮定は大きく間違っているようにみえるが、この大胆な仮定が現実的にきわめて有効であることを、多くの確率・統計的自然言語処理システムが実証している。

3.2.2 最尤推定法

言語モデルにおける重要な課題は、与えられた学習データから言語モデルを推定することである。一般には、言語現象の背後に確率モデル（確率分布）を想定し、このモデルを特徴付けるパラメータ（母数）を学習データから決定するということが行われる。ここで説明する最尤推定法は、いくつかの望ましい性質を備えた推定法であり、多くの言語モデル推定の際の基礎となっている。

観測値 $X_1,\cdots,X_N$ が未知パラメータ θ を含む確率分布 $P_\theta(X)$ から抽出された標本だとする。ここで、各 X_i は独立に抽出された標本だとすると、全観測値の同時確率分布は次のように計算することができる。

$$\begin{aligned} P_\theta(X_1,\cdots,X_N) &= P_\theta(X_1)\cdots P_\theta(X_N) \\ &= \prod_{i=1}^{N} P_\theta(x_i) \end{aligned} \tag{3.3}$$

各X_iに具体的な観測データ$x_1,\cdots,x_N$が与えられると、上式は未知パラメータθの関数になる。これを尤度（likelihood）あるいは尤度関数（likelihood function）と呼び、$L(\theta)$と表す。

$$L(\theta)=\prod_{i=1}^{N}P_\theta(x_i) \tag{3.4}$$

最尤推定法とは、尤度$L(\theta)$を最大にするようなパラメータ$\hat{\theta}$を推定する方法である。

多くの確率モデルでは、尤度を直接最大化するよりも、尤度の対数

$$\log L(\theta)=\sum_{i=1}^{N}\log P_\theta(x_i) \tag{3.5}$$

を最大化する方が計算が簡単になる。尤度の対数は、特に対数尤度（log likelihood）と呼ばれている。対数尤度を最大化するに$\hat{\theta}$は、次の式を解くことにより得られる。

$$\frac{\partial}{\partial\theta}\log L(\theta)=0 \tag{3.6}$$

これを尤度方程式（likelihood equation）と呼ぶ。なお、最尤推定法は、一致性を持ち、かつ漸近的に正規な推定量を与えることが知られている。

3.2.3 N-gram モデル

確率的言語モデルの基本的な役割は、与えられた単語列$w_1^n=w_1\cdots w_n$に対しその生起確率$P(w_1^n)$を計算することである。確率論の乗法定理を用いると、$P(w_1^n)$は次のように書き換えることができる。

$$P(w_1^n)=P(w_1)P(w_2\mid w_1)\cdots P(w_n\mid w_1^{n-1})$$

$$= \prod_{i=1}^{n} P(w_i \mid w_1^{i-1}) \tag{3.7}$$

ここで、条件付き確率$P(w_i \mid w_1^{i-1})$に対し、条件部ある単語列w_1^{i-1}のことを単語の履歴（history）と呼ぶ。

式（3.7）より、条件付き確率$P(w_i \mid w_1^{i-1})$を求めることができれば、単語列全体の生成確率を計算することができる。しかし、すべての単語の組み合わせに対し$P(w_i \mid w_1^{i-1})$を求めることは現実的に不可能である。仮に3万語の語彙で20単語の文まで考えるとしても、$30{,}000^{20}$個もの値を推定しなければならない。現実的なモデルとするためには、単語の履歴w_1^{i-1}を同値類に分類しモデルのパラメータ数を削減する必要がある。

一般に、ある時点で生起する事象の確率が、その直前のN個の時点で生起した事象だけの影響を受けるとき、これをN重マルコフ過程（N-th order Markov process）と呼ぶ。N-gramモデル（N-gram model）は、単語の生起を（N−1）重マルコフ過程で近似したモデルである。すなわち、N-gramモデルでは、ある時点での単語の生起は直前の（N−1）単語のみに依存すると考えている。したがって、N-gramモデルにおいては、

$$P(w_1^n \mid w_1^{n-1}) = P(w_n \mid w_{n-N+1}^{n-1}) \tag{3.8}$$

となる。なお、N=1,2,3の場合を、それぞれユニグラム(unigram)、バイグラム(bigram)、トライグラム（trigram）と呼ぶ。ユニグラムは、単語が直前の語に影響されずに独立に生起するということであり、これは単語の生起確率と等しい。また、すべての単語が等確率で生起すると考えたモデルのことを特別にゼログラム（zerogram）と呼ぶ。

ここで、バイグラムによる単語列w_1^nの生成確率を考えると、式（3.7）より次のようになる。

$$P(w_1^n) = \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \tag{3.9}$$

$i=1$の場合には$P(w_1 \mid w_0)$となるが、ここでw_0は文頭を表す特別な記号であると考えればよい。また、すべての単語列に対する確率値の合計が 1 となるためには、単語列の終りにも文末を表す特別な記号が付いていると考える必要がある。N-gram の場合には、文頭に（N−1）個の特別な記号があると考えると都合がよい。

次に、N-gram の確率の推定について説明する。いま、単語列w_1^nが学習データ中に出現する回数を$C(w_1^n)$で表すことにする。N-gram の確率は、学習データ中に出現する単語の N 個組と（N−1）個組の相対頻度から、次のように推定することができる（最尤推定）。

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n})}{C(w_{n-N+1}^{n-1})} \tag{3.10}$$

なお、最尤推定で求められた確率値であることを強調する場合には、$P_{ML}(w_n \mid w_{n-N+1}^{n-1})$のように書く。N の値が大きいほど、学習データから信頼性の高い N-gram の値を推定するのが難しくなるため、通常はバイグラムやトライグラムがよく用いられる。

3.3 ゼロ頻度問題とスムージング

3.3.1 ゼロ頻度問題

N-gram の確率値を単純に相対頻度により推定すると、学習データ中に出現しない単語組の確率値を 0 にしてしまうという大きな欠点がある。また、たとえ学習データ中に出現しても、出現頻度の小さな単語列に対しては、統計的に信頼性のある確率値を推定するのが難しい。

確率・統計的自然言語処理では、多くの場合、単語（あるいは単語列）の生起は二項分布に従っているという単純な仮定をおいている。このため、単語wが総単語数 N 単語の学習データ中にr回出現したときの出現確率は、最尤推定により$P(w)=r/N$として求められる。

もし単語wが学習データ中に出現しなければ、wの出現確率は 0 と推定されてしまう。出現確率が 0 ということは、単語wは決して出現しないということを意味するが、

しかしこれには大きな問題がある。単語wは、たまたま、この学習データ中に出現しなかっただけであり、別の学習データ中には出現する可能性も十分にありえる。また、確率・統計的自然言語処理では、複数の単語列の確率の積から文の確率を算出する。この場合、どれか 1 つの単語列の確率が 0 だと、文全体の確率も 0 になるという問題がでてくる。このような問題は、ゼロ頻度問題（zero frequency problem）と呼ばれている。

ゼロ頻度問題に対処するために、単純に単語の出現回数を使うのではなく、出現回数を補正した値を使うということが行われる。このように出現回数を補正することを一般にディスカウンティング（discounting）と呼ぶ。なお、出現回数rをr^*に補正した際の比$d_r = r^*/r$をディスカウント係数（discount coefficient）と呼ぶ。

3.3.2 スムージング

ゼロ頻度問題など、N-gram における確率値の推定に関する問題に対処するために、通常はスムージング（smoothing）あるいは平滑化と呼ばれる手法を用いる。なお、N-gram の確率値は、基本的に単語列の相対頻度から算出するため、多くのスムージング手法では頻度のディスカウンティング手法をその基礎として用いている。

代表的なディスカウンティング法である、グッド・チューリング推定法（Good-Turing estimation）では、出現回数rの補正値として、以下で定義されるr^*を用いる。

$$r^* = (r+1)\frac{N_{r+1}}{N_r} \tag{3.11}$$

N_rは、いままでと同様に、学習データ中にr回出現したn単語列の総数である。

後述するように、上のような補正値を用いた場合には、未知単語列（学習データ中に出現しないn単語列）の確率の総和がN_1/Nとなる。したがって、グッド・チューリング推定法による単語列w_1^nの出現確率は以下で与えられる。

$$P(w_1^n)=\begin{cases}\dfrac{r^*}{N} & C(w_1\cdots w_n)>0\\ \dfrac{N_1}{N_0 N} & C(w_1\cdots w_n)=0\end{cases} \tag{3.12}$$

学習データ中に1回以上出現した(すなわち$C(w_1^n)>0$)すべてのn単語列に対し、グッド・チューリング推定法による単語列の出現確率の和を求めると次のようになる。

$$\begin{aligned}\sum_{w_1^n:C(w_1^n)>0} P(w_1^n) &= \sum_{r\geq 1}\frac{N_r r^*}{N} = \sum_{r\geq 1}\frac{(r+1)N_{r+1}}{N}\\ &= \sum_{r\geq 1}\frac{rN_r}{N} - \frac{N_1}{N}\end{aligned} \tag{3.13}$$

ここで、次の式が成り立つことを注意する。

$$N=\sum_{r\geq 1} rN_r \tag{3.14}$$

したがって、

$$\sum_{w_1^n:C(w_1^n)>0} P(w_1^n) = 1-\frac{N_1}{N} \tag{3.15}$$

上式は、グッド・チューリング推定法を用いた場合には、学習データ中に出現するすべてのn単語列に対する確率値の合計が1にならない(1にN_1/Nだけ足りない)ことを示している。つまり、未知単語列に対する確率値の合計をN_1/Nと推定していることになる。すなわち、

$$\sum_{w_1^n:C(w_1^n)=0} P(w_1^n) = \frac{N_1}{N} \tag{3.16}$$

$N_r = 0$の場合には、式（3.11）による出現回数の補正を行うことが不可能となる。また、$N_r \neq 0$であっても、rが大きい場合にはN_rの値が不安定になる（統計的な信頼性が失われる）。この問題に対処する最も簡単な方法は、rが小さい場合のみ式(3.11)による補正を行い、rが大きい場合には学習データから得られた出現回数をそのまま使うというものである。

他の方法として、N_rに対する補正値を使うものがある。まず、0 でないN_rだけを順番に並べる。

$$N_{r1}, N_{r2}, \cdots, N_{ri}, \cdots \qquad (r_1 < r_2 < \cdots < r_i < \cdots \text{かつ} \ N_r \neq 0)$$

ここで、N_{ri}に対する補正値$S(N_{ri})$を以下のように定義する。

$$S(N_{ri}) = \frac{2N_{ri}}{r_{i+1} - r_{i-1}} \tag{3.17}$$

r_iが小さいときは、ほとんどの場合、隣との間隔は 1 である$(r_{i+1} - r_i = 1)$。したがって、$r_{i+1} - r_{i-1} = 2$となり$S(N_r)$はN_rに等しい。しかし、r_iが大きくなるにつれ、$S(N_{ri})$はN_{ri}より小さくなる。次に、$\log N_r$と$\log r$が線形の関係にあるという経験則（ジップの法則）を利用し、回帰直線

$$\log S(N_r) = -m \cdot \log r + b \quad (m, b \text{は定数}) \tag{3.18}$$

により、さらに$S(N_r)$の補正を行う。

また、rが小さいときにはN_rの値をそのまま使い、rが大きい場合のみ$S(N_r)$を使う方法は、単純グッド・チューリング推定法（simple Good-Turing estimation）と呼ばれている。

また、チャーチよろびゲイルにより提案された、改良グッド・チューリング推定法（enhanced Good-Turing estimation）という手法もあるが、この手法は、元来、バイ

グラムに対して考案されたものなので、トライグラム以上に適用するには注意すべき点がある。

グッド・チューリング推定法を基礎にしたスムージング手法に、バックオフ・スムージング（back-off smoothing）がある。学習データ中に出現しないN-gramの値を低次の（N−1）gramの値から推定する。以下では、バックオフ・スムージングを用いた単語バイグラムの推定について説明する。

まず、学習データ中に単語列w_1w_2が出現する場合には、グッド・チューリングの推定値を用いて、条件付き確率$P(w_2 \mid w_1)$を求める。

$$P(w_2 \mid w_1) = \frac{C^*(w_1w_2)}{C(w_1)} = \frac{C^*(w_1w_2)}{C(w_1w_2)} \frac{C(w_1w_2)}{C(w_1)}$$
$$= d_{C(w_1w_2)} \frac{C(w_1w_2)}{C(w_1)} \tag{3.19}$$

上式において、$C^*(\cdot)$はグッド・チューリング推定値を意味している。また、$d_{C(w_1w_2)}$はディスカウント係数である。

次に、学習データ中に単語列w_1w_2が出現しない場合について考える。グッド・チューリング推定法では、学習データ中に出現したすべての単語バイグラムに対し、式（3.19）を用いて推定した確率値の合計は 1 にはならない。1 からの不足分を$C(w_1w_2)=0$となるような単語列に対して分配することを考える。ここで、次のような関数βを定義する。

$$\beta(w_1) = 1 - \sum_{w_2:C(w_1w_2)>0} P(w_2 \mid w_1) \tag{3.20}$$

右辺の$P(w_2 \mid w_1)$は、式（3.19）により求められたものである。関数$\beta(w_1)$は、単語w_1に対し、$C(w_1w_2)=0$となるようなすべての単語w_2に対する条件付き確率の和を与えている。$\beta(w_1)$を単語w_2の出現確率$P(w_2)$に従って、$C(w_1w_2)=0$となるような単語列に分配することにより、$P(w_2 \mid w_1)$の値を推定する。このような分配を行う

関数をαとすると、$C(w_1w_2)=0$の場合には、

$$P(w_2 \mid w_1) = \alpha P(w_2) \tag{3.21}$$

となる。αは、単語w_1に依存する関数であり次のように定義される。

$$\begin{aligned}\alpha &= \alpha(w_1) \\ &= \frac{\beta(w_1)}{\sum_{w_2:C(w_1w_2)=0} P(w_2)} \\ &= \frac{1-\sum_{w_2:C(w_1w_2)>0} P(w_2 \mid w_1)}{1-\sum_{w_2:C(w_1w_2)>0} P(w_2)}\end{aligned} \tag{3.22}$$

また、$C(w_1)=0$の場合には、以下のように定義する。

$$P(w_2 \mid w_1) = P(w_2) \tag{3.23}$$

一般のN-gramに対するバックオフ・スムージングも、上記と同様の手続きにより、再起的に定義することができる。

$$\begin{aligned}&P(w_n \mid w_{n-N+1}^{n-1}) \\ &= \begin{cases} d_{C(w_{n-N+1}^{n-1})} P_{ML}(w_n \mid w_{n-N+1}^{n-1}) & C(w_{n-N+1}^{n-1}) > 0 \\ \alpha(w_{n-N+1}^{n-1}) P(w_n \mid w_{n-N+2}^{n-1}) & C(w_{n-N+1}^{n-1}) = 0 \end{cases}\end{aligned} \tag{3.24}$$

実際にバックオフ・スムージングを用いる場合、出現回数の大きい単語列に対しては単純に相対頻度（最尤推定値）を用い、出現回数の小さな単語列に対してだけグッド・チューリングの推定値を用いるという方法がとられることがある。すなわち、適当な定数kに対し、単語列の出現回数rが$r>k$であるときは$d_r=1$（すなわち$r^*=r$）と

し、$r \leq k$であるときのみグッド・チューリングの推定値を用いる。この方法を特別にカッツ・スムージング（Katz smoothing）と呼ぶ。

グッド・チューリング推定法では、学習データ中に出現しない未知単語列に配分される頻度は、学習データ中に1回のみ出現した単語列の出現回数に等しい。すなわち、N_1である。一方、カッツ・スムージングでは、単語列の出現回数rが$1 \leq r \leq k$の範囲にあるときだけ$d_r r$に補正される。したがって、カッツ・スムージングにおいて、未知単語列に対する配分をN_1とするためには、d_rが次の等式を満たす必要がある。

$$\sum_{r=1}^{k}(r - d_r r)N_r = N_1 \tag{3.25}$$

さらに、d_rによる補正が、グッド・チューリング推定法における補正と比例するようにするために次の条件を課す。

$$1 - d_r = \mu\left(1 - \frac{r^*}{r}\right) \quad (\mu \text{は定数}) \tag{3.26}$$

上記の2つの条件を満たすd_rを求めると、$1 \leq r \leq k$に対しては、d_rは次のように与えられる。

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}} \tag{3.27}$$

なお、カッツは、$k = 5$程度が適当な値であると報告している。またカッツは、$d_1 = 0$としても実用上問題ないと報告している。$d_1 = 0$とすることは、出現回数が1回のみのデータは捨ててもよいことを意味しているので、これにより言語モデルの記憶空間を大きく節約することができる。

3.4 パープレキシティ

3.4.1 言語のエントロピー

言語は、単語列を生成する情報源であると考えることができる。いま、言語Lにおける単語列$w_1^n = w_1 \cdots w_n$の生成確率を$P(w_1^n)$とすれば、言語Lのエントロピーは次式により計算することができる。

$$H_0(L) = -\sum_{w_1^n} P(w_1^n) \log P(w_1^n) \tag{3.28}$$

また、1単語当たりのエントロピーは次のようになる。

$$H(L) = -\sum_{w_1^n} \frac{1}{n} P(w_1^n) \log P(w_1^n) \tag{3.29}$$

3.4.2 パープレキシティ

式(3.29)の$H(L)$は言語から生成される単語を特定するために必要な情報量(ビット)であり、各単語の後に平均して$2^{H(L)}$個の単語が後続可能であることを示している。すなわち、

$$PP = 2^{H(L)} \tag{3.30}$$

は情報理論的な意味での単語の平均分岐数を表しており、パープレキシティ(perplexity)と呼ばれている。言語のパープレキシティが大きい(単語の平均分岐数が多い)ほど、単語を特定するのが難しく、言語として複雑であることになる。

3.5 言語モデルの評価

3.5.1 言語モデルの評価

自然言語は複雑であり、自然言語の真のモデルを知ることは不可能である。我々が取り扱う言語モデルは、自然言語の近似モデルである。

言語のモデルの評価に当たっては、言語モデルが自然言語をどれだけ正確に近似しているかということが問題となる。もちろん、言語モデルを特定の応用システムに組み込んだときの性能より、間接的に言語モデルを評価することも可能である。しかし、応用システムの性能はさまざまな要因が絡んでくるために、純粋に言語モデル自身の性能を評価することは困難なことが多い。

確率的言語モデルの大きな特徴は、客観的な評価尺度を情報理論に基づき定義できるという点にある。一般的に使われている評価尺度は、クロス・エントロピー（cross-entropy）やテストセット・パープレキシティ（test-set parplexity）である。

実際の言語モデルの評価実験では、評価用のテキスト集合を用いてクロス・エントロピーやテストセット・パープレキシティを算出する。この際に、学習データを評価用データとして用いるだけでは評価として不十分である。学習データ以外のデータに対しても評価を行う必要がある。通常は、与えられた言語データの一部を評価用データとして残しておき、それ以外のデータを学習データとして用いるということにが行われる。

3.5.2 テストセット・パープレキシティ

実際に言語モデルの評価を行う際には、評価のためのテキスト集合を定め、そのテキスト集合に対するパープレキシティを求めるということが行われる。これを、テストセット・パープレキシティ（test-set perplexity）と呼ぶ。

テストセット・パープレキシティは、評価用のテキスト集合（単語列）w_1^N に対する1単語当たりのエントロピーを求め、式（3.30）から算出することができる。言語にエルゴード性（情報源の統計的性質がその情報源から生成される十分に長い系列に完全に現れているということ）が仮定できる場合には、

$$H = -\frac{1}{N}\log P_M(w_1 \cdots w_N) \tag{3.31}$$

から、

$$PP = 2^{H(L)} = P_M(w_1 \cdots w_N)^{-\frac{1}{N}} \tag{3.32}$$

と計算することができる。

Chapter 4

言語モデルと学習データ

4.1 言語モデルと学習データ

4.1.1 言語モデルの学習データ

言語モデルの推定には、大規模な言語データベースが必要不可欠である。これらの言語データベースをコーパス（corpus, pl. corpora）と呼ぶ。コーパスには、データの収集法あるいは利用目的などにより、さまざまな形態が存在する。電子化されたテキストを単に集めてきたものから、品詞や構文情報などの各種の言語情報を付与したものなどいろいろである。

コーパスは、付与されている情報の違いや多様性により、次のような形態に分類することができる。

(1) 生コーパス（raw corpus）

単なる電子化テキストの集積であり、テキストに何も加工がなされていないものをを指す。

(2) タグ付きコーパス（tagged corpus）

コーパス中の各テキストが単語単位に分割されており、各単語に品詞あるいは品

詞コード（タグ, tag）が付けられているものを指す。

(3) 解析済み（あるいは分析済み）コーパス（analyzed corpus）

品詞情報に加え、構文構造や係り受け構造等のさらに高度な言語情報が付与されているコーパスである。

本論文では、学習データとして新聞記事（'94 年の毎日新聞）を用いている。また、そのデータ量は記事 1 年分である。

4.1.2 CMU-Cambridge Statistical Language Modeling Toolkit

本論文ではテキストデータから N-gram に基づく言語モデルを生成することが出来る CMU-Cambridge Statistical Language Modeling Toolkit を用いる。この Toolkit は Carnegie Mellon University の Roni Rosenfeld により初版が作られ、現在では Cambridge University の Philip Clarkson も参加して改訂版が広く配布されている。フリーのソフトであり、比較的高速に言語モデルを生成できるため、この Toolkit を利用することとした。N-gram モデルにおけるゼロ頻度問題には、グッド・チューリング推定法を基礎にした、バックオフ・スムージングにより対処している。

また、この Toolkit は、英語の文章のように単語と単語の間に空白が存在するテキストデータの利用を前提としているので、日本語の文章のように、単語と単語の間に空白が存在しないテキストデータには直接利用できない。そこで、本論文では形態素解析ソフトの茶筌により単語分割されたテキストデータを用いる。

利用した学習データに対する、1-gram (ユニグラム)、2-gram (バイグラム)、3-gram (トライグラム)、それぞれの生起確率の分布表が連なった形で生成される言語モデルは、相当量の大きさを持つデータとなる。本論文のように新聞記事 1 年分を学習データとして利用すると、生成される言語モデルは 200 メガバイトを超える大きさとなる。

利用した学習データ中に 1 度も出現しない 3 単語列（3-gram）の生起確率を求める必要がある場合には、2-gram と 1-gram 及び、スムージングで対処された未知語の生起確率により近似値を求めることができる。また、このソフトにはテストセット・パープレキシティを行うソフトも付属しており、学習データとは異なるテキストデータを用意することで、パープレキシティを測定できる。

4.1.3 言語モデルの精度と学習データ量

N-gram モデルにおいて、N-gram の値が大きいモデルほど、よりよいモデルになりそうに思えるが、これには大きな問題がある。1つは、N を大きくするに従い、急激にモデルのパラメータ（単語の条件付き確率）数が増えるという点である。パラメータ数は指数オーダーで増えていくので、N が大きくなるとパラメータを正確に推定することが不可能になる。さらに指摘しておきたいのは、単純に N の値を大きくしても、N-gram モデルの精度は現実的には向上しないという点である。表 4.1 は、スイッチボード・コーパス（Switchboard corpus）における N-gram モデルの次数とモデルの精度を示している。

表 4.1 : N-gram モデルの次数とモデルの精度

N	パープレキシティ	単語誤り率
3	92.0	49.73%
4	91.2	49.17%
5	91.3	−

単語誤り率とあるのは、N-gram モデルを音声認識に適用したときの単語の認識誤り率（word error rate）である。このコーパスの語彙数は 22,643 語であり、これは N=0 の場合（単語が等確率で生起すると考えた場合）のパープレキシティが 22,643 であることを意味している。N=3 でパープレキシティは 92.0 まで減少しているが、N を 3 以上にしても精度の向上はごくわずかである。以上で述べたようにパラメータ推定と精度の観点から、通常は N=2（バイグラム）あるいは N=3（トライグラム）を用いる場合が多い。

N-gram モデルの確率値も、学習データ中の単語列の相対頻度から求めることができる。この際、大量の学習データを用いるほど、より精度の高い言語モデルを作ることができる。表 4.2 は、3-gram（トライグラム）での学習データ量とモデルの精度の関係を調べるために、IBM の音声認識グループが行った実験結果を示している。

表 4.2：トライグラムの学習データ量とモデルの精度

学習データの単語数	パープレキシティ
100,000	121.3
200,000	113.3
500,000	100.1
1,000,000	85.6

表が示すように、学習データ量が増えるに従いパープレキシティは単調に減少している。しかし、どれだけ大量の学習データを用いても、すべての単語組に対する確率値を求めることはできない。すなわち、学習データ中には出現しないが、別のデータには出現するという単語組も非常に多く存在する。やはり IBM の音声認識グループが、レーザーに関する特許文書を用いてトライグラムのカバー率に関する調査を行っている。この調査では、トライグラムの学習データとして 150 万単語、テスト・データとして 30 万単語のテキストを用いた。その結果、テスト・データ中の 23%のトライグラムは学習データ中に含まれなかったことを報告している。

4.2 学習データ中のノイズ

4.2.1 ノイズの定義

前述したように、言語モデルを生成する際に、学習データ量が多いほど良好な言語モデルが得られる。しかし学習データの質の面から考えると、言語モデルの生成に有効に作用すると思われるデータと、モデル生成の際にモデルの精度向上には結び付かないデータが存在する。後者のようなデータをここではノイズと呼ぶ。

学習データ中のノイズの具体例を図 4.1 及び図 4.2 に示す。図 4.1 及び図 4.2 に示した例は、本論文における学習データである新聞記事中のノイズの例であり、部分的に単語列として成り立っているが文章として成立するには至っていないテキストデータである。

例 1 ----------

建設（ゼネコン、建設、橋りょうなど、）	約３億３０００万円
開発、不動産・住宅	約７４００万円
銀行・証券	約５３００万円
コンピューター・通信	約３４００万円
大手電気設備	約３２００万円
自動車関連	約２６００万円
長野県人オーナー会社	約７６００万円

図 4.1：新聞記事中のノイズの例 1

例 2 ----------

□来年１月１日の首相はだれか？－－野党議員の回答

	村山富市	河野洋平	海部俊樹	羽田孜	無回答
□新進党□					
《旧新生》					
小沢辰男	●				
加藤六月			●		
羽田孜				●	
渡部恒三				●	
□無所属□					
楢崎弥之助				●	

図 4.2：新聞記事中のノイズの例 2

4.2.2 ノイズの影響

学習データ中のノイズが、学習データから生成される言語モデルに対して、どのような影響を与えているのかを考える。

本論文において学習データは新聞記事である。新聞記事中における図や写真などは、明らかに言語モデルの生成に役立たないといえる。これらのデータは、必ずしも言語モデルの生成及び、生成された言語モデルの精度に悪影響を与えているとは言い切れないが、これらのデータが学習データに加わることによるデータ量の増加による効果は、テキストデータ（単語列）のみによるデータ量の増加の効果とは異なるものになるといえる。この点において、これらのデータは学習データ中のノイズであるといえる。

学習データの増加に伴うパープレキシティの減少、つまり言語モデルの精度向上は、言語モデルの生成に対して有効な学習データの増加によってのみ実現される。学習データ中におけるノイズの割合を低下させ、ノイズが与える言語モデルへの影響を低減させることを考える。

4.2.3 ノイズの選定と除去

本論文では、テキストデータ中のカッコ“()”を含む文章に注目し、このカッコとその中に位置する単語列がノイズに該当するのではないかという予測を立て、学習データ中からカッコを含む部分をカッコごと除去することとした。

学習データからのカッコの除去は簡単な C のプログラムを書き、そのプログラムを用いて行った。

ノイズに該当すると思われるカッコを、学習データ中から除去することで、学習データのデータ量は僅かながら減少する。本論文では、新聞記事を学習データとして使用しているため、語句や用語に対する補足や説明のために、ある程度の量のカッコが、テキストデータに含まれていると推測される。学習データ中から一定量のテキストデータを除去することができれば、学習データ量の低減にも結び付けることができる。さらに、除去したテキストデータがノイズに該当するデータであるとすると、学習データからノイズを除去したことに伴う言語モデルの精度向上も期待できると推測される。除去の例を図 4.3 に示す。

なお今年の新聞休刊日は２月１１日(金)、３月２１日(月)、４月１０日(日)、５月５日(木)、６月１２日(日)、７月１０日(日)、８月１４日(日)、９月２３日(金)、１０月１０日(月)、１１月１３日(日)、１２月１１日(日)の予定です。
第３５回毎日芸術賞(１９９３年度)の受賞者が次の通り決まりました。(敬称略、５０音順)

↓

なお今年の新聞休刊日は２月１１日、３月２１日、４月１０日、５月５日、６月１２日、７月１０日、８月１４日、９月２３日、１０月１０日、１１月１３日、１２月１１日の予定です。
第３５回毎日芸術賞の受賞者が次の通り決まりました。

図 4.3：ノイズの除去例

本論文では、言語モデルに N-gram モデルを用いている。前述したように N-gram モデルにおいては単語の並び(単語列)が重要な要素となる。新聞記事においてカッコ“()”は、直前の単語及び前方の単語に対して、補足や説明のために用いられている場合が多い存在であると推測した。実際に新聞を読み進める際、これらの補足や説明は有用なものではあるが、N-gram モデルの生成においては、単語と単語の連続性を阻害している可能性があると推測される。したがって、これらのカッコ “()” とその中に含まれる単語列を学習データ中から除去することにより、学習データ中の単語と単語の連続性をより強めることを試みる。

4.3 言語モデルの評価方法

4.3.1 パープレキシティの比較

学習データ中からノイズに該当すると思われる部分を選定し、選定したノイズ部分を学習データ中から除去する。ノイズを含む学習データと、ノイズを除去した学習データから Toolkit を用いてそれぞれ言語モデルを生成する。それぞれの言語モデルを用いてパープレキシティを測定し、結果を比較する。図 4.4 に評価の流れを示す。

また、テストセット・パープレキシティを行うには、学習データとは異なるテキストデータが必要となる。本論文では、'95 年の毎日新聞の記事 1000 文を使用している。

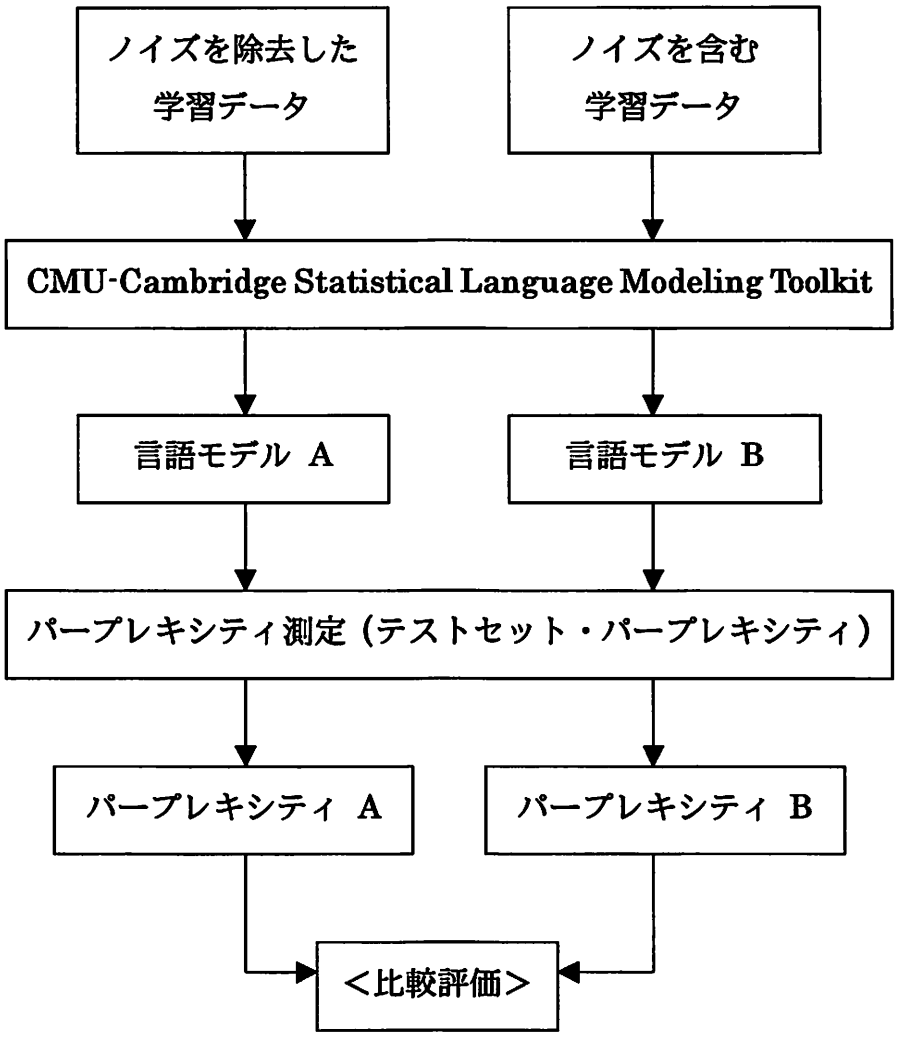

図 4.4：評価の流れ

4.3.2 並べ替え問題

言語モデルの評価としては、ある課題を言語モデルを用いて解き、その精度から評価することも可能である。ここでは単語の並べ替え問題により言語モデルの評価を行う。

まず文章を単語分割し、その単語列をランダムに並べ替える。次に言語モデルから得られる単語列の生起確率を利用して、並び替えられた単語列から元の文章を推定する。言語モデルの精度が高いほど、並べ替え問題の正解率は高くなる。正解率の差から言語モデルの精度を比較評価する。

また、言語モデルの生起確率を用いた単語の並べ替え問題は、割り当て問題と等価である。そこで、当研究室の高橋 篤史 氏のビタビアルゴリズムツールを用いて並べ替え問題を解いた。

さらに、この問題を解くには言語モデルから生起確率を求める処理が必要となる。Toolkit から作成される言語モデルは、実際には単語列とその生起確率を記した巨大なファイルである。そのため単語の生起確率を求めるためには、ファイルの中から特定の単語列を探す検索モジュールが必要となる。ここでは高速文字列検索システム SUFARY を利用した。図 4.5 に並べ替え問題の例を示す。

[例文] 私は茨大生です。

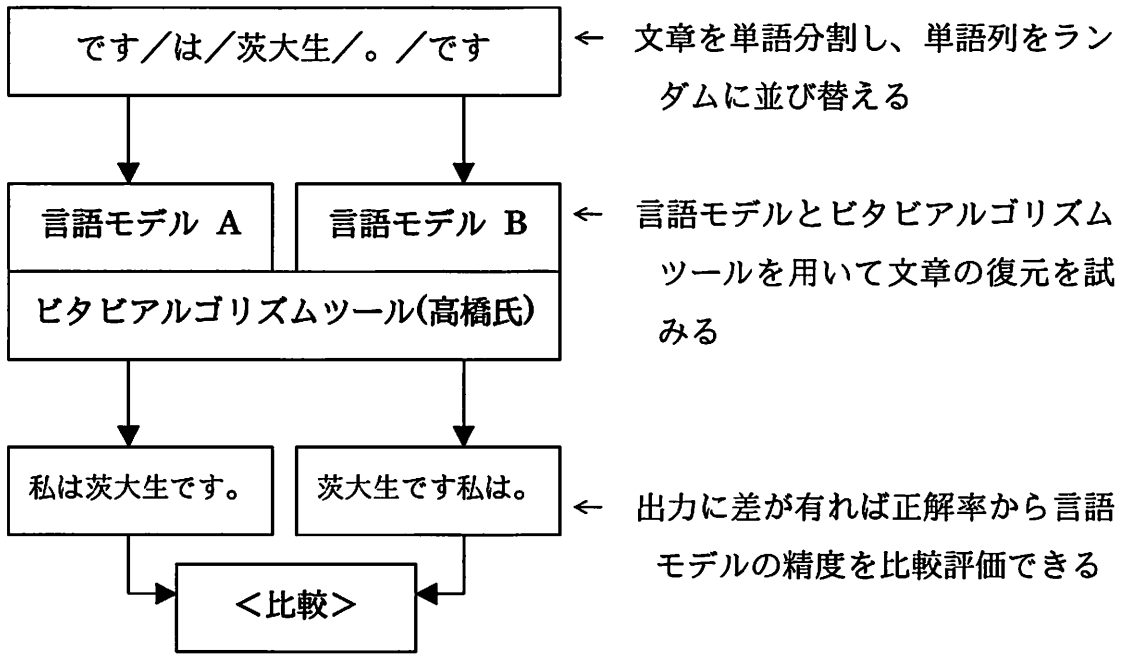

図 4.5：並べ替え問題の例

Chapter 5

実験

5.1 実験の手順

1. 毎日新聞の新聞記事（'94 年度版）1 年分から、図や表、グラフなどを取り除く

2. 1 の新聞記事から形態素解析を用いて単語分割を行い学習データ A を作成

3. 学習データ A を容量が同じになるように 10 分割し学習データ A1～A10 を作成

4. 学習データ A1～A10 をノイズ除去プログラムに通し学習データ B1～B10 を作成

5. A，B それぞれの学習データから、Toolkit を用いて、それぞれ言語モデル MA1～MA10，言語モデル MB1～MB10 を生成

6. パープレキシティの測定（テストセット・パープレキシティ）を行い、生成された各言語モデルの精度を比較

7. 並べ替え問題を通じて、正解率から各言語モデルの精度を比較

5.2 実験準備

5.2.1 学習データ A 作成

学習データの増加に伴うパープレキシティの減少、つまり言語モデルの精度向上を確認するために、容量が異なる 10 種類の学習データを作成する。

図や表、グラフなどを取り除いた'94 年の毎日新聞 1 年分のテキストデータ（学習データ A と呼称）の大きさは、126MB 弱（125640505 byte）である。なお、このデータは茶筌（version 2.0）により単語分割されている。このデータを容量が同じ大きさになるよう 10 分割し、それらを組み合わせることにより、10 種類の容量の異なる学習データを作成する。表 5.1 に各学習データの容量を示す。

表 5.1：学習データ A の容量

学習データ	容量(MB)
A1	14.1
A2	27.2
A3	40.3
A4	53.4
A5	66.6
A6	77.7
A7	90.4
A8	101.8
A9	113.7
A10	125.6

5.2.2 学習データ B 作成

学習データ A1～A10 を、ノイズ除去プログラムに通し、ノイズが除去された学習データ B1～B10 を作成する。ノイズ除去の具体例は前述したとおり。表 5.2 に各学習データの容量を示す。

表 5.2：学習データ B の容量

学習データ	容量(MB)
B1	13.4
B2	25.8
B3	38.2
B4	50.6
B5	63.1
B6	73.6
B7	85.7
B8	96.4
B9	107.6
B10	119.0

学習データ A1 と学習データ B1 の容量差が、約 0.7MB となり、学習データ A10 と学習データ B10 の容量差が、約 6.6MB となった。新聞記事 1 年分の学習データにおいて、ノイズ除去により、カッコ“()”とカッコの中に位置するテキストデータ（単語列）が、およそ 6.6MB 分除去された。これにより、学習データ全体の約 5%のテキストデータが除去されたことになる。

5.2.3 パープレキシティの測定

各学習データから Toolkit を用いて各言語モデルを生成した後、各言語モデルの精度を測定するために、Toolkit に付属のソフトを用いて、テストセット・パープレキシティ（パープレキシティの測定）を行った。

テストセット・パープレキシティを行うには、学習データとは異なるテキストデータを用意する必要がある。評価用のテキストデータには'95 年の毎日新聞記事 1000 文を利用している。表 5.3 に学習データ A のパープレキシティの値を、表 5.4 に学習データ B のパープレキシティの値を示す。

表 5.3：学習データ A のテストセット・パープレキシティ

学習データ	容量(MB)	Computation based on words	Number of 1-grams	Number of 2-grams	Number of 3-grams	Entropy(bits)	Perplexity
A1	14.1	864503	20001	496610	1536348	7.94	245.53
A2	27.2	864472	20001	717920	2589229	7.78	219.31
A3	40.3	864748	20001	881240	3455939	7.71	209.06
A4	53.4	865024	20001	1017042	4241645	7.66	201.77
A5	66.6	865173	20001	1138049	4992626	7.60	194.33
A6	77.7	865355	20001	1239794	5636550	7.58	191.26
A7	90.4	865618	20001	1348108	6355533	7.54	186.44
A8	101.8	865692	20001	1433863	6943960	7.51	182.63
A9	113.7	866155	20001	1523232	7563870	7.48	178.44
A10	125.6	867138	20001	1607549	8164659	7.44	173.77

表 5.4：学習データ B のテストセット・パープレキシティ

学習データ	容量(MB)	Computation based on words	Number of 1-grams	Number of 2-grams	Number of 3-grams	Entropy(bits)	Perplexity
B1	13.4	864819	20001	479849	1476409	7.88	235.22
B2	25.8	864730	20001	695814	2494173	7.72	211.03
B3	38.2	865345	20001	855013	3330308	7.66	201.74
B4	50.6	865427	20001	987377	4089269	7.60	194.09
B5	63.1	865750	20001	1106624	4814797	7.55	187.89
B6	73.6	865810	20001	1205995	5436518	7.53	184.56
B7	85.7	866073	20001	1312846	6132853	7.49	179.80
B8	96.4	866139	20001	1397230	6703057	7.46	176.41
B9	107.6	866572	20001	1484915	7300876	7.43	172.15
B10	119.0	867576	20001	1568224	7883590	7.39	167.67

5.2.4 並べ替え問題と例文

並べ替え問題を通して評価する言語モデルは、ノイズを含む学習データ A10 から生成した言語モデル MA10 と、ノイズを除去した学習データ B10 から生成した言語モデル MB10 である。

並べ替え問題に用いた例文を以下に示す。これらは、単語数が 10〜30 単語の例文、20 文である。

・数人の日本人の行方も分からないままだ。

・これらの人々の無事を祈らずにはいられない。

・阪神大震災の被災地、神戸では被災者への募金活動も始まった。

・阪神大震災の教訓や難民支援の経験が生かされているのだろう。

・海外の人道危機に対する民間の支援態勢は充実してきた。

・被災者には食料や医薬品、毛布などの生活必需品や簡易住宅が必要だ。

・住民に伝染病が広がるのも防がねばならない。

・公共建築の耐震性の弱さが被害を広げたと考えざるをえない。

・耐震建築の普及や、防災体制の整備を手伝う。

・そのためには、専門家の派遣や NGO を通じた長期的な取り組みが必要になろう。

・慢性 C 型肝炎ウイルス感染が原因で、がんになることも少なくない。

・半年間のインターフェロン治療で体内からウイルスがいなくなる患者は三、四割といわれる。

・標準期間を超えてインターフェロン治療に保険が認められるには、ウイルスが一時検出されなくなるなどの条件がある。

・時計メーカーが、目覚し時計の機能開発に知恵を絞っている。

・担当者は「消費者が求める機能を探って商品を開発し、目覚し時計を復権させたい」と話している。

・地下鉄の騒音よりも大きな音で、だれでも無理やり起こす目覚し時計「スーパーライデン」を売り出す。

・ただ、スポーツクラブはプールやスタジオ、マシンジムなど設備の面だけでは差別化は難しい。

・一方、スポーツクラブも一定以上の店舗網を敷くことができるかが生き残り戦略のポイントとなる。

・インストラクターなどの教育システムやプログラムの開発は、小さな店舗網のクラブでは難しい。

・株式市場は、昨年秋以降、景況感の悪化を急速に織り込んで下げ足を速めた。

5.3 実験結果

5.3.1 パープレキシティの比較

学習データとして、94 年の毎日新聞 1 年分のテキストデータを利用した。95 年の毎日新聞のテキストデータ 1000 文をテストセットとしてテストセット・パープレキシティを求めた。結果を図 5.1 に示す。図の横軸はテキストデータのファイルの大きさ、縦軸はパープレキシティを示しており、上方の折れ線がノイズ（カッコ）を含む学習データ A から生成した言語モデル MA のパープレキシティの推移であり、下方の折れ線がノイズを除去した学習データ B から生成した言語モデル MB のパープレキシティの推移である。

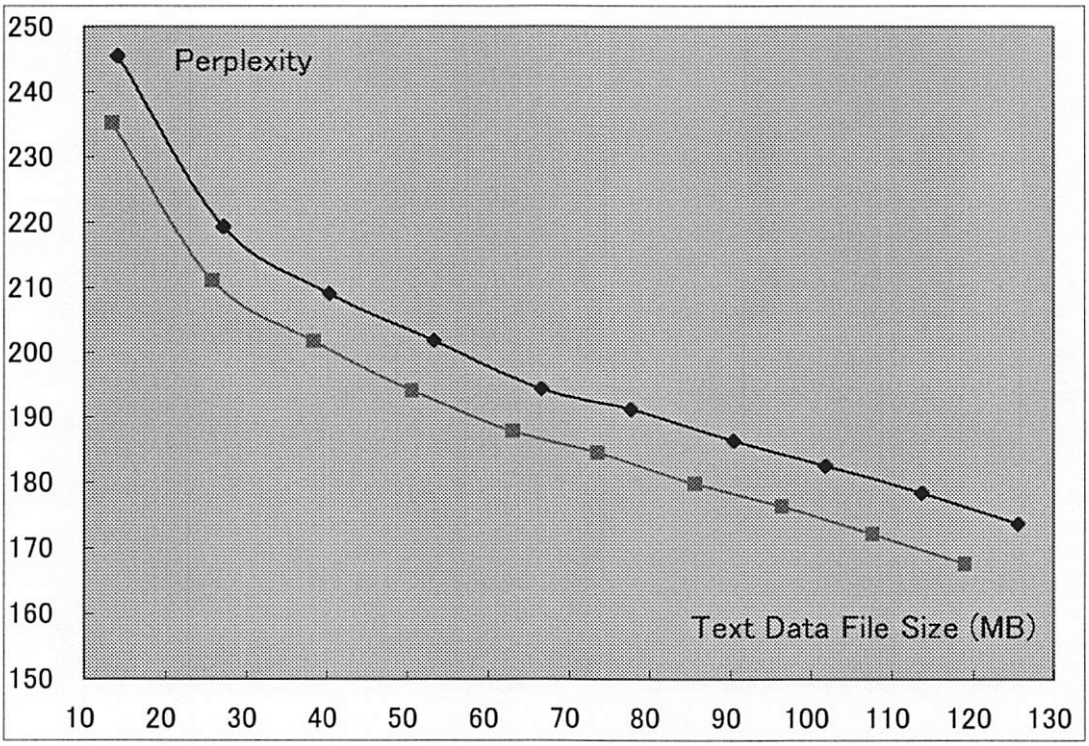

図 5.1：パープレキシティの比較

図 5.1 から、学習データの増加に伴うパープレキシティの減少、つまり言語モデルの精度向上が確認できる。また、ノイズを含まない学習データ B から生成した言語モデル MB の方が、ノイズを含む学習データ A から生成した言語モデル MA よりパープレキシティが低くなる、つまり言語モデルの精度が高くなることが確認できた。同時に、学習データ中におけるカッコ“()”が、ノイズに該当することも確認できた。

5.3.2 並べ替え問題

言語モデルの生起確率を用いた単語の並べ替え問題は、割り当て問題と等価である。そこで、当研究室の高橋氏のビタビアルゴリズムツールを用いて並べ替え問題を解いた。単語数が 20 単語前後の例文を 20 文用意して実験を行ったが、2 つの言語モデル、MA, MB を通して、完全に復元された文章は一文も無く、正解率は共に 0%だった。結果、並べ替え問題では、今回作成した 2 つの言語モデルの精度を評価するのは困難であるといえる。

以下に実験結果を示す。なお、結果に示された文は単語分割されたままの状態である。

＜＜言語モデル MA10 を用いた結果＞＞

●分から ない まま の 人 の 数 行方 も 日本人 だ 。

○これら の 人々 に は い られ ない の 無事 を 祈ら ず 。

●被災 地 で 始まっ た の は 、 被災 者 へ の 募金 活動 も 震災
神戸 阪神 大 。

○生かさ れ て いる の が 阪神 大 の 難民 支援 の 経験 や 震災 教訓
だろ う 。

●充実 し て き た の は 海外 民間 の 危機 に対する 人道 支援 態勢 。

○食料 毛布 など 生活 必需 品 は 必要 だ が 、 被災 者 の 住宅 や
医薬品 や 簡易 に 。

●伝染病 が 広がる ば なら ない の 住民 に も 防が ね 。

○ざる を え ない と の 考え を 広げ た の が 建築 耐震 公共 性
被害 弱 さ 。

●体制 の 整備 や 防災 の 普及 を 、 耐震 建築 手伝う 。

○長期 的 に は 、 その ため に 必要 な の が 専門 家
派遣 や NGO を 通じ た 取り組み なろ う 。

●C 型 慢性 肝炎 ウイルス がん に 感染 が 原因 で ない こと も 、
少なく なる 。

○患者 の 体内 で は なく なる と いわ れる ウイルス が 、三 割 から 四
年間 い インターフェロン 治療 半 。

●認め られる の が 検出 さ れ 超え て は なる が 、 ある 期間 など を
条件 に インターフェロン に 標準 治療 が なく 一時 ウイルス 保険 。

○機能 開発 、 知恵 目覚し 時計 絞っ て を に メーカー 時計 が
の いる 。

●たい 」 と 話し て いる が 、 消費 者 は 「 目覚し を し 探っ て
商 品 を 開発 担当 者 を 求める 機能 さ せ 時計 復権 。

○無理やり 起こす の で 、 だれ で も 「 より 大きな 音 」 も ライデン
目覚し 時計 を 売り出す 地下鉄 スーパー 騒音 。

●プール など の 面 で は 、 スポーツ クラブ は 、 スタジオ 設備 や
差別 化 は 難しい だけ ただ マシン ジム 。

○と なる こと が できる の か 一方 、 一定 以上 の 店舗 網 が 生き残り
戦略 を も スポーツ クラブ 敷く ポイント 。

●プログラム は 、 教育 インストラクター など の は 難しい の で
システム の 開発 や 小さな 店舗 網 クラブ 。

○株式 市場 織り込ん で は 、 昨年 秋 以降 、 下げ 足 を 急速 に
悪化 を 速め 景況 た の 感 。

<<言語モデル MB10 を用いた結果>>

●分から ない まま の 人 の 数 行方 も 日本人 だ 。

○これら の 人々 に は い られ ない の 無事 を 祈ら ず 。

●被災 地 で 始まっ た の は 、 被災 者 へ の 募金 活動 も 震災
神戸 阪神 大 。

○生かさ れ て いる の が 阪神 大 の 難民 支援 の 経験 や 震災 教訓
だろ う 。

●充実 し て き た の は 海外 民間 の 危機 に対する 人道 支援 態勢 。

○食料 毛布 など 生活 必需 品 は 必要 だ が 、 被災 者 の 住宅 や
医薬品 や 簡易 に 。

●伝染病 が 広がる ば なら ない の 住民 に も 防が ね 。

○ざる を え ない と の 考え を 広げ た の が 建築 耐震 公共 性
被害 弱 さ 。

●体制 の 整備 や 、 手伝う 耐震 建築 の 普及 を 防災 。

○長期 的 に は 、 その ため に 必要 な の が 専門 家
派遣 や NGO なろ う を 通じ た 取り組み 。

●C 型 慢性 肝炎 ウイルス がん に 感染 が 原因 で ない こと も 、
少なく なる 。

○患者 の 体内 で は なく なる と いわ れる ウイルス が 、三 割 から 四
年間 い インターフェロン 治療 半 。

●認め られる の が 検出 さ れ 超え て は なる が 、 ある 期間 など を
条件 に インターフェロン に 標準 治療 が なく 一時 ウイルス 保険 。

○機能 開発 、 知恵 目覚し 時計 絞っ て を に メーカー 時計 が
の いる 。

●たい 」 と 話し て いる が 、 消費 者 は 「 目覚し を し 探っ て
商品 を 開発 担当 者 を 求める 機能 さ せ 時計 復権 。

○無理やり 起こす の で 、 だれ で も 「 より 大きな 音 」 も ライデン
目覚し 時計 を 売り出す スーパー 地下鉄 騒音 。

●プール など の 面 で は 、 スポーツ クラブ は 、 スタジオ 設備 や
差別 化 は 難しい だけ ただ マシン ジム 。

○と なる こと が できる の か 一方 、 一定 以上 の 店舗 網 が 生き残り
戦略 を も スポーツ クラブ 敷く ポイント 。

●プログラム は 、 教育 インストラクター など の は 難しい の で
システム の 開発 や 店舗 網 小さな クラブ 。

○株式 市場 織り込ん で は 、 昨年 秋 以降 、 下げ 足 を 急速 に
悪化 を 速め 景況 た の 感 。

Chapter 6

考察

本論文の基本的な目的である、言語モデルの精度に対する学習データ中のノイズの影響は、実験結果により確認できたと思われる。学習データ中からノイズに該当する単語列を選定し除去することによる、言語モデルの精度向上は、パープレキシティの減少により確認できた。しかし、今回確認できた言語モデルの精度向上は、これらの言語モデルを用いたアプリケーションの動作に差異を与えるような大きなものではなく、僅かなものに留まっている。並べ替え問題を用いた比較評価に失敗したのもその一例である。

学習データ中には、本論文においてノイズとして扱ったカッコ“()”の他にも、様々な姿をした数多くのノイズが存在すると考えられるが、本論文の例からも推測できるように、そのどれもが劇的な言語モデルの精度向上を導くものではないと考えられる。また、利用する学習データの違いにより、ノイズとして最も明確に現れてくる単語列も異なってくると考えられる。

Chapter 7

おわりに

カッコ“()”を含む単語列をノイズとして扱い、学習データ中から除去することで、言語モデルの精度向上を導くことができた。

学習データからノイズを除去することにより、5%程度の学習データ容量の削減と3%程度のパープレキシティの低減が見られた。すなわち、学習データから生成することができる言語モデルにおいて、カッコとカッコの中に含まれる単語列が、生成される言語モデルの精度向上に対して有効に作用することはなく、逆に悪影響を及ぼすものであると考えられ、それらを除去することにより、言語モデルの精度向上を望むべきであるといえる。

謝辞

本研究の遂行及び論文の作成において多大な御助言及び御指導を賜りました新納 浩幸 教官(茨城大学工学部システム工学科)に深い感謝の意を表します。

また、本研究で利用した学習データ及び評価用データは、毎日新聞CD-ROM '94版及び毎日新聞CD-ROM '95版から得ています。利用を許可して頂いた毎日新聞社に深く感謝致します。

さらに、御指導を頂きましたシステム工学科計算機応用学講座の教官の方々、本研究を進めるにあたり御助言、御協力を頂きました、荒井 亮一 氏(茨城大学大学院理工学研究科システム工学専攻)、同研究室の高橋 篤史 氏(茨城大学工学部システム工学科)、星 秀明 氏(茨城大学工学部システム工学科)に深く感謝致します。

最後に、池谷 昌紀 氏(茨城大学大学院理工学研究科システム工学専攻)に深く敬意を表すと共に、心より感謝申し上げます。

Bibliography

[1] 北 研二："確率的言語モデル"，東京大学出版会（1999）.

[2] CMU-Cambridge Statistical Language Modeling Toolkit
http://svr-www.eng.cam.ac.uk/~prc14/toolkit

[3] ChaSen
http://cl.aist-nara.ac.jp/lab/nlt/chasen

[4] SUFARY
http://cl.aist-nara.ac.jp/lab/nlt/ss

[5] 高橋 篤史："ビタビアルゴリズムの汎用ツールの作成"，H12 年度 茨城大学工学部システム工学科 卒業論文（2001）.

[6] 池谷 昌紀："文字ベースの HMM による複合語単語分割の誤り修正"，H10 年度 茨城大学工学部システム工学科 卒業論文（1999）.

付録 A　プログラムソースリスト

```
**********************************************
                ノイズ除去プログラム
**********************************************

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

main(int argc,char *argv[])
{
   FILE *f1;
   char buf[10000],buf1[]="iÊ",buf2[]="iË";
   int r,i,j,l,m,n;
   int len;
   if(argc!=2){
      printf("argc=%d¥n",argc);
      printf("° ú¿ô¤Î¿ô¤¬° ã¤¦¥n¥a");
      exit(1);
   }

   if((f1=fopen(argv[1],"r"))==NULL){
      printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
      exit(1);
   }
```

```
  while(fgets(buf,10000,f1)!=NULL){
    len = strlen(buf);
    for(i=0;i<len-1;i++){
      if((buf[i]==buf1[0])&&(buf[i+1]==buf1[1])){
        for(l=i+2;l<len-1;l++){
          if((buf[l]==buf2[0])&&(buf[l+1]==buf2[1])){
            i=l+3;
            goto skip1;
          }
        }
      }
    skip1:
      putchar(buf[i]);
    }
    printf("¥n");
  }

  if((r=fclose(f1))==-1) exit(1);
}
```

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

N-gram 表から 1-gram 表を取り出すプログラム

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

main(int argc,char *argv[])
{
  FILE *f1;
  char buf1[256],buf2[]="1-grams:¥n",buf3[]="¥n";
  int r,i=0,j=0,k=0,m=0;
```

```
if(argc!=2){
   printf("argc=%d¥n",argc);
   printf("° ú¿ô¤Î¿ô¤¬° ã¤¡¥n¥a");
   exit(1);
}

if((f1=fopen(argv[1],"r"))==NULL){
   printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
   exit(1);
}

while(fgets(buf1,256,f1)!=NULL){
   //      printf("here(%d)¥n" j);
   if(strstr(buf1,buf2)!=NULL){
      if(k!=0){
         for(;;){
            //          printf("here(1)¥n");
            fgets(buf1,256,f1);
            if(strcmp(buf1,buf3)==0){
               exit(0);
            }
            i=8;
            while(buf1[i]!='¥n'){
               putchar(buf1[i]);
               i++;
            }
            printf("¥t");
            for(m=0;m<7;m++){
               putchar(buf1[m]);
            }
            printf("¥n");
            i=0;
         }
      }
```

```
        else if(k==0){
          k++;
        }
      }
      //      j++;
    }

    if((r=fclose(f1))==-1) exit(1);
}
```

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

N-gram 表から 2-gram 表を取り出すプログラム

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

main(int argc,char *argv[])
{
    FILE *f1;
    char buf1[256],buf2[]="2-grams:¥n",buf3[]="¥n",buf4[]="¥t";
    int r,i=0,j=0,k=0,m=0,n=0,x=0,w=0;

    if(argc!=2){
        printf("argc=%d¥n",argc);
        printf("° ú¿ð□Î¿ð□¬° ã□¦¥n¥a");
        exit(1);
    }

    if((f1=fopen(argv[1],"r"))==NULL){
        printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
        exit(1);
    }
```

```
while(fgets(buf1,256,f1)!=NULL){
  //      printf("here(%d)¥n",j);
  if(strstr(buf1,buf2)!=NULL){
    if(k!=0){
      for(;;){
        //          printf("here(1)¥n");
        fgets(buf1,256,f1);
        if(strcmp(buf1,buf3)==0){
          exit(0);
        }
        i=8;
        while(buf1[i]!='¥n'){
          if(buf1[i]==' '){
            if(x<2){
              if(x==0){
                printf("ii");
              }
              else if(x==1){
                printf("¥t");
              }
              x++;
              w=1;
            }
            i++;
            continue;
          }
//          if(buf1[i]==' '&&w==1){
//            break;
//          }
          putchar(buf1[i]);
          i++;
        }
        w=0;
        x=0;
```

```
            printf("¥t");
            for(m=0;m<7;m++){
              putchar(buf1[m]);
            }
            printf("¥n");
            i=0;
          }
        }
        else if(k==0){
          k++;
        }
      }
      //      j++;
    }

    if((r=fclose(f1))==-1) exit(1);
}
```

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

N-gram 表から 3-gram 表を取り出すプログラム

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

main(int argc,char *argv[])
{
    FILE *f1;
    char buf1[256],buf2[]="3-grams:¥n",buf3[]="¥n";
    int r,i=0,j=0,k=0,m=0,n=0;

    if(argc!=2){
      printf("argc=%d¥n",argc);
```

```
    printf("° ú¿ð¤Î¿ð¤¬° ã¤¦¥n¥a");
    exit(1);
}

if((f1=fopen(argv[1],"r"))==NULL){
    printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
    exit(1);
}


while(fgets(buf1,256,f1)!=NULL){
    //      printf("here(%d)¥n",j);
    if(strstr(buf1,buf2)!=NULL){
        if(k!=0){
            for(;;){
                //          printf("here(1)¥n");
                fgets(buf1,256,f1);
                if(strcmp(buf1,buf3)==0){
                    exit(0);
                }
                i=8;
                while(buf1[i]!='¥n'){
                    if(buf1[i]==' '){
                        if(n<2){
                            printf("ii");
                            n++;
                        }
                        i++;
                        continue;
                    }
                    putchar(buf1[i]);
                    i++;
                }
                n=0;
                printf("¥t");
```

```
            for(m=0;m<7;m++){
               putchar(buf1[m]);
            }
            printf("¥n");
            i=0;
          }
        }
        else if(k==0){
          k++;
        }
      }
      //      j++;
    }

    if((r=fclose(f1))==-1) exit(1);
}
```

****

並び替えプログラム

****

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define tate 5000
#define yoko 100


int yomikomi(char *buf11,char buf41[][yoko]){
    int len,k,i,m=0,n=0;

    len = strlen(buf11);
    for(k=0;k<len-1;k++){
```

```
        if(buf11[k]==' '){
            buf41[m][n]='¥0';
            m++;
            n=0;
        }
        else{
            buf41[m][n]=buf11[k];
            n++;
        }
    }
    buf41[m][n]='¥0';
    //for(i=0;i<m;i++){
    //printf("%s¥n",buf41[i]);
    //}
    m=m+1;
    return(m);
}

void ransuu(int m1,int *ra){

    //int m=0,s,n,w,rann[5000];

    int m=0,s,n,w,*rann;
    rann=(int *)malloc(sizeof(int)*m1);

    for(n=0;n<m1;n++){
        rann[n]=0;
    }
    w=m1;
    while(w!=0){
        w=0;
        s=rand()%m1;
        if(rann[s]==1){
            w=1;
            continue;
```

```
        }
        ra[m]=s;
        rann[s]=1;
        m++;
        for(n=0;n<m1;n++){
          if(rann[n]==0){
            w++;
          }
        }
      }
      free(rann);
}

/*
char out(int m2,char buf7[][yoko]){
  int i;
  for(i=0;i<m2;i++){
    printf("%s¥n",buf7[i]);
  }
  }*/

main(int argc,char *argv[])
{
  FILE *f1;
  char buf1[10000],buf3[]="¥t";
  char buf4[tate][yoko];/*,buf6[tate][yoko];*/
  int r,i,j,k,l,m=0,n=0,s=0;
  int len,ra[tate];
  if(argc!=2){
    printf("argc=%d¥n",argc);
    printf("°  ú¿ô¤Î¿ô¤¬°  ã¤¬¥n¥a");
    exit(1);
  }

  if((f1=fopen(argv[1],"r"))==NULL){
```

```
        printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
        exit(1);
    }

    while(fgets(buf1,10000,f1)!=NULL){

        m=yomikomi(buf1,buf4);

        ransuu(m,ra);

//      printf(" %c%c",-95,-95);

        for(i=0;i<m;i++){
            printf("%s",buf4[ra[i]]);
//              printf("%s ",buf4[ra[i]]);
            if(i!=m-1){
                printf(" ");
            }
        }

        printf("¥n");
//      printf(" <s>¥n");

        //      narabi(m,buf4,buf6);

        //out(m,buf6);

    }

    if((r=fclose(f1))==-1) exit(1);
}
```

```
**********************************************************
          並び替えられた文から組み合わせ表を作成するプログラム
**********************************************************

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
//#include"sufury"

#define tate 5000
#define yoko 100


int yomikomi(char *buf11,char buf41[][yoko]){
   int len,k,i,m=0,n=0;

   len = strlen(buf11);
   for(k=0;k<len-1;k++){
      if(buf11[k]==32){
         buf41[m][n]='¥0';
         m++;
         n=0;
      }
      else{
         buf41[m][n]=buf11[k];
         n++;
      }
      buf41[m][n]='¥0';
   }
   //for(i=0;i<m;i++){
   //printf("%s¥n",buf41[i]);
   //}

   return(m);
}
```

```
//void ransuu(int m1,int *ra){

// //int m=0,s,n,w,rann[5000];

//   int m=0,s,n,w,*rann;
//   rann=(int *)malloc(sizeof(int)*m1);

//   for(n=0;n<m1;n++){
//     rann[n]=0;
//   }
//   w=m1;
//   while(w!=0){
//     w=0;
//     s=rand()%m1;
//     if(rann[s]==1){
//       w=1;
//       continue;
//     }
//     ra[m]=s;
//     rann[s]=1;
//     m++;
//     for(n=0;n<m1;n++){
//       if(rann[n]==0){
//       w++;
//       }
//     }
//   }
//   free(rann);
//}

/*
char out(int m2,char buf7[][yoko]){
  int i;
  for(i=0;i<m2;i++){
```

```
    printf("%s¥n",buf7[i]);
  }
  }*/

main(int argc,char *argv[])
{
  FILE *f1;
  char buf1[10000],buf3[]="¥t";
  char buf4[tate][yoko];/*,buf6[tate][yoko];*/
  char buf5[tate][yoko];
  int r,i,i1,i2,i3,j,k,l,m=0,n=0,s=0,v=0,w=0;
  int len,ra[tate];

  if(argc!=2){
    printf("argc=%d¥n",argc);
    printf("° śæō¤æō¬° ć¤¦¥n¥a");
    exit(1);
  }

  if((f1=fopen(argv[1],"r"))==NULL){
    printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
    exit(1);
  }

  while(fgets(buf1,10000,f1)!=NULL){

    m=yomikomi(buf1,buf4);

//    ransuu(m,ra);

//    printf(" %c%c",-95,-95);

//    for(i=0;i<m+1;i++){
//      printf("%s¥n",buf4[i]);
//    }
```

```
    v=1;

    for(i1=0;i1<m+1;i1++){
      for(i2=i1+1;i2<m+1;i2++){
        for(i3=i2+1;i3<m+1;i3++){
//          printf("%d¥t",v);    //1
            printf("%s ",buf4[i1]);
            printf("%s ",buf4[i2]);
            printf("%s¥n",buf4[i3]);
            v++;
//          printf("%d¥t",v);    //2
            printf("%s ",buf4[i1]);
            printf("%s ",buf4[i3]);
            printf("%s¥n",buf4[i2]);
            v++;
//            printf("%d¥t",v);    //3
            printf("%s ",buf4[i2]);
            printf("%s ",buf4[i1]);
            printf("%s¥n",buf4[i3]);
            v++;
//          printf("%d¥t",v);    //4
            printf("%s ",buf4[i2]);
            printf("%s ",buf4[i3]);
            printf("%s¥n",buf4[i1]);
            v++;
//            printf("%d¥t",v);    //5
            printf("%s ",buf4[i3]);
            printf("%s ",buf4[i1]);
            printf("%s¥n",buf4[i2]);
            v++;
//            printf("%d¥t",v);    //6
            printf("%s ",buf4[i3]);
            printf("%s ",buf4[i2]);
            printf("%s¥n",buf4[i1]);
            v++;
```

```
        }
      }
    }

//      printf(" ¥n");

      //      narabi(m,buf4,buf6);

      //out(m,buf6);

    }

  if((r=fclose(f1))==-1) exit(1);
}
```

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

組み合わせ表からコスト表を作成するプログラム

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "sufary.h"

main(int argc,char *argv[])
{
  FILE *f1;

  char *s;
  SUFARY *ary3,*ary2,*ary1;
```

```
long i,pos;
float bo1,bo2,bo3,bo12;

char buf1[10000],buf2[10000],buf3[]="¥t",buf11[10000],buf22[10000];
char buf33[10000],buf222[10000];
char buf7[10000],buf8[10000],buf9[10],buf10[10];
char buf44[10000],buf55[10000];
int r,i1,i2,i3,j,k,l,m=0,n=0,v=0,w=0,x=0,num=1;
int len,len1,len2/*,ra[tate]*/;

if(argc!=5){
   printf("argc=%d¥n",argc);
   printf("°  ú¿ð□Î¿ð□¬°  ã□¦¥n¥a");
   exit(1);
}

if((f1=fopen(argv[1],"r"))==NULL){
   printf("File ¥"%s¥" cannot open!! ¥n¥a",argv[1]);
   exit(1);
}

if((ary3=sa_openfiles(argv[4],NULL))==NULL){
   fprintf(stderr,"Error¥n");
}
if((ary2=sa_openfiles(argv[3],NULL))==NULL){
   fprintf(stderr,"Error¥n");
}
if((ary1=sa_openfiles(argv[2],NULL))==NULL){
   fprintf(stderr,"Error¥n");
}

while(fgets(buf1,10000,f1)!=NULL){
   len = strlen(buf1);
   for(v=0;v<len-1;v++){
      buf2[v]=buf1[v];
```

```
        buf222[v]=buf1[v];
      }
      buf2[v]='¥0';
      buf222[v]=' ';
      buf222[v+1]='¥t';
      buf222[v+2]='¥0';

    sa_reset(ary3);
    sa_reset(ary2);
    sa_reset(ary1);

    if(sa_sel(ary3,buf222)==CONT){
      for(i=sa_left(ary3);i<=sa_right(ary3);i++){
        pos=sa_aryidx2txtidx(ary3,i);
        s=sa_getline(ary3,pos);
//          printf("%d ",num++);
        printf("%s¥n",s);          /*<case1> out*/
//          printf("%s <case1>¥n",s);     //<case1> out
        free(s);
        break;
      }
//      printf("%s¥n",buf2);
//      printf("¥n");
    }
    else if(sa_sel(ary3,buf222)==FAIL){
 //     printf("here1¥n");
      w=0;
      v=0;
      strcpy(buf11,buf2);
      strcpy(buf22,buf11);
//      printf("here11¥n");
      for(;;){
        if(buf11[v]==' '){
          w++;
        }
```

```
            if(w==2){
              break;
            }
            v++;
          }
          buf11[v]='¥0';
//          printf("%d¥n",v);
//          printf("%s¥n",buf22);
//          printf("<buf11>%s¥n",buf11);
//          printf("here12¥n");
          w=0;
          v=0;
          x=0;
        //        printf("%d¥n",strlen(buf22));
          while(buf22[v]!=' '){
            v++;
          }
          v++;
//          printf("%d<v>¥n",v);
          while(buf22[v]!='¥0'){
            buf33[x]=buf22[v];
            x++;
            v++;
          }
//          x++;
          buf33[x]='¥0';
//          printf("¥n");
//          printf("%d<v>¥n",v);
//          printf("%d<x>¥n",x);
//          printf("<buf33>%s¥n",buf33);
          x=0;
//          printf("%s <non>¥n",buf11);
//          printf("here2¥n");
//          goto skip1;
          sa_reset(ary2);
```

```
        if(sa_sel(ary2,buf11)==CONT){
//          printf("here21¥n");
          for(i=sa_left(ary2);i<=sa_right(ary2);i++){
            pos=sa_aryidx2txtidx(ary2,i);
            s=sa_getline(ary2,pos);
            strcpy(buf7,s);
            len1 = strlen(buf7);
            for(v=len1-15;v<len1-8;v++){
              buf9[x]=buf7[v];
              x++;
            }
            buf9[x]='¥0';
//          printf("<buf9>%s¥n",buf9);
            bo1=atof(buf9);
//          printf("<bo1>%f¥n",bo1);
//          printf("%s <s>¥n",s);
            free(s);
            break;
          }
          x=0;
//          printf("here3¥n");
          sa_reset(ary2);
          if(sa_sel(ary2,buf33)==CONT){
//          printf("here4¥n");
            for(i=sa_left(ary2);i<=sa_right(ary2);i++){
              pos=sa_aryidx2txtidx(ary2,i);
              s=sa_getline(ary2,pos);
              strcpy(buf8,s);
              len2 = strlen(buf8);
              for(v=len2-7;v<len2;v++){
                buf10[x]=buf8[v];
                x++;
              }
              buf10[x]=' ';
//            printf("<bif10>%s¥n",buf10);
```

```
            bo2=atof(buf10);
//          printf("<bo2>%f¥n",bo2);
            free(s);
            break;
          }
//        printf("%d ",num++);
          printf("%s ¥t%7.4f¥n",buf22,bo1+bo2);    /*<case2> out*/
//        printf("%s ¥t%7.4f <case2>¥n",buf22,bo1+bo2);  //<case2> out
          goto skip2;
        }
        else if(sa_sel(ary2,buf33)==FAIL){
//        printf("%s <base2>¥n",buf22);
//        printf("<buf11>%s¥n",buf7);
//        printf("<buf33>%s<non>¥n",buf33);
          v=0;
          x=0;
          while(buf33[v]!=' '){     /*buf55=buf33(2-gram)□Î･å□ƒ□ÎÊý(1-gram)*/
            v++;
          }
          v++;
          while(buf33[v]!='¥t'){
            buf55[x]=buf33[v];
            v++;
            x++;
          }
          buf55[x]='¥0';
          w=0;
          v=0;
          strcpy(buf44,buf33);
          for(;;){
            if(buf44[v]==' '){     /*buf44=buf33(2-gram)□ÎÁ°  □ÎÊý(1-gram)*/
              w++;
            }
            if(w==1){
              break;
```

```
        }
        v++;
      }
      buf44[v]='¥0';
//    printf("<buf44>%s¥n",buf44);
//    printf("<buf55>%s¥n",buf55);
      sa_reset(ary1);
      if(sa_sel(ary1,buf44)==CONT){
        x=0;
        for(i=sa_left(ary1);i<=sa_right(ary1);i++){
          pos=sa_aryidx2txtidx(ary1,i);
          s=sa_getline(ary1,pos);
          strcpy(buf7,s);
          len1 = strlen(buf7);
          for(v=len1-15;v<len1-8;v++){
            buf9[x]=buf7[v];
            x++;
          }
          buf9[x]='¥0';
          bo2=atof(buf9);
          free(s);
          break;
        }
        sa_reset(ary1);
        x=0;
        if(sa_sel(ary1,buf55)==CONT){
          for(i=sa_left(ary1);i<=sa_right(ary1);i++){
            pos=sa_aryidx2txtidx(ary1,i);
            s=sa_getline(ary1,pos);
            strcpy(buf8,s);
            len2 = strlen(buf8);
            for(v=len2-7;v<len2;v++){
              buf10[x]=buf8[v];
              x++;
            }
```

```
                buf10[x]=' ';
                bo3=atof(buf10);
                free(s);
                break;
              }
//            printf("<bo2>%f¥n",bo2);
//            printf("<bo3>%f¥n",bo3);
//            printf("%d ",num++);
              printf("%s ¥t%7.4f¥n",buf22,bo1+bo2+bo3);   /*<case3> out*/
//            printf("%s ¥t%7.4f <case3>¥n",buf22,bo1+bo2+bo3);   //<case3> out
              goto skip2;
            }
            else if(sa_sel(ary1,buf55)==FAIL){
//            printf("%s <base6>¥n",buf22);
//            printf("<buf11>%s¥n",buf11);
//            printf("<buf33>%s<non>¥n",buf33);
//            printf("<ubf44>%s¥n",buf44);
//            printf("<buf55>%s<non>¥n",buf55);

              sa_reset(ary1);
              if(sa_sel(ary1,"<UNK>")==CONT){
                for(i=sa_left(ary1);i<=sa_right(ary1);i++){
                  pos=sa_aryidx2txtidx(ary1,i);
                  s=sa_getline(ary1,pos);
                  strcpy(buf8,s);
                  len2 = strlen(buf8);
                  for(v=len2-7;v<len2;v++){
                    buf10[x]=buf8[v];
                    x++;
                  }
                  buf10[x]=' ';
                  bo3=atof(buf10);
                  free(s);
                  break;
                }
```

```
                }
                else if(sa_sel(ary1,"<UNK>")==FAIL){
                  printf("<UNK><non>¥n");
                  goto skip2;
                }
//              printf("<bo2>%f¥n",bo2);
//              printf("<bo3>%f¥n",bo3);
//                printf("%d ",num++);
                printf("%s ¥t%7.4f¥n",buf22,bo1+bo2+bo3); /*<case6> out*/
//                printf("%s ¥t%7.4f <case6>¥n",buf22,bo1+bo2+bo3); //<case6> out
                goto skip2;
              }
            }
            else if(sa_sel(ary1,buf44)==FAIL){
//            strcpy(buf55,buf33);
              printf("%s <base7>¥n",buf22);
              printf("<buf11>%s¥n",buf11);
              printf("<buf33>%s<non>¥n",buf33);
              printf("<buf44>%s<non>¥n",buf44);

              sa_reset(ary1);

              goto skip2;
//            bo1=0.0;
            }
          }
        }
        else if(sa_sel(ary2,buf11)==FAIL){
//        printf("%s <base1>¥n",buf22);
//        printf("<buf11>%s<non>¥n",buf11);
          sa_reset(ary2);
          if(sa_sel(ary2,buf33)==CONT){
            x=0;
            for(i=sa_left(ary2);i<=sa_right(ary2);i++){
              pos=sa_aryidx2txtidx(ary2,i);
```

```
            s=sa_getline(ary2,pos);
            strcpy(buf8,s);
            len2 = strlen(buf8);
            for(v=len2-7;v<len2;v++){
              buf10[x]=buf8[v];
              x++;
            }
            buf10[x]='¥0';
//          printf("%d ",num++);
            printf("%s ¥t%s¥n",buf22,buf10);      /*<case4> out*/
//          printf("%s ¥t%s <case4>¥n",buf22,buf10);      //<case4> out
//          printf("<buf33>%s¥n",s);
            free(s);
            goto skip2;
//          break;
          }
        }
        else if(sa_sel(ary2,buf33)==FAIL){
//        printf("%s<base3>¥n",buf22);
//        printf("<buf33>%s<non>¥n",buf33);
          v=0;
          x=0;
          while(buf33[v]!=' '){      /*buf55=buf33(2-gram)□Î･å□ſ□ÎÊý(1-gram)*/
            v++;
          }
          v++;
          while(buf33[v]!='¥t'){
            buf55[x]=buf33[v];
            v++;
            x++;
          }
          buf55[x]='¥0';
          w=0;
          v=0;
          strcpy(buf44,buf33);
```

```
        for(;;){
          if(buf44[v]==' '){      /*buf44=buf33(2-gram)¤ÎÁ° ¤Î(1-gram)*/
            w++;
          }
          if(w==1){
            break;
          }
          v++;
        }
        buf44[v]='¥0';
//      printf("<buf44>%s¥n",buf44);
//      printf("<buf55>%s¥n",buf55);
        sa_reset(ary1);
        if(sa_sel(ary1,buf44)==CONT){
          x=0;
          for(i=sa_left(ary1);i<=sa_right(ary1);i++){
            pos=sa_aryidx2txtidx(ary1,i);
            s=sa_getline(ary1,pos);
            strcpy(buf7,s);
            len1 = strlen(buf7);
            for(v=len1-15;v<len1-8;v++){
              buf9[x]=buf7[v];
              x++;
            }
            buf9[x]='¥0';
            bo2=atof(buf9);
            free(s);
            break;
          }
          sa_reset(ary1);
          x=0;
          if(sa_sel(ary1,buf55)==CONT){
            for(i=sa_left(ary1);i<=sa_right(ary1);i++){
              pos=sa_aryidx2txtidx(ary1,i);
              s=sa_getline(ary1,pos);
```

```
                strcpy(buf8,s);
                len2 = strlen(buf8);
                for(v=len2-7;v<len2;v++){
                  buf10[x]=buf8[v];
                  x++;
                }
                buf10[x]=' ';
                bo3=atof(buf10);
                free(s);
                break;
              }
//              printf("<bo2>%f¥n",bo2);
//              printf("<bo3>%f¥n",bo3);
//                printf("%d ",num++);
              printf("%s ¥t%7.4f¥n",buf22,bo2+bo3); /*<case5> out*/
//                printf("%s ¥t%7.4f <case5>¥n",buf22,bo2+bo3); //<case5> out
              goto skip2;
            }
            else if(sa_sel(ary1,buf55)==FAIL){
//              printf("%s<base8>¥n",buf22);
//              printf("<buf11>%s¥n",buf11);
//              printf("<buf33>%s<non>¥n",buf33);
//              printf("<buf44>%s¥n",buf44);
//              printf("<buf55>%s<non>¥n",buf55);

              sa_reset(ary1);
              if(sa_sel(ary1,"<UNK>")==CONT){
                for(i=sa_left(ary1);i<=sa_right(ary1);i++){
                  pos=sa_aryidx2txtidx(ary1,i);
                  s=sa_getline(ary1,pos);
                  strcpy(buf8,s);
                  len2 = strlen(buf8);
                  for(v=len2-7;v<len2;v++){
                    buf10[x]=buf8[v];
                    x++;
```

```
                }
                buf10[x]=' ';
                bo3=atof(buf10);
                free(s);
                break;
              }
            }
            else if(sa_sel(ary1,"<UNK>")==FAIL){
              printf("<UNK><non>¥n");
              goto skip2;
            }
//          printf("<bo2>%f¥n",bo2);
//          printf("<bo3>%f¥n",bo3);
//            printf("%d ",num++);
            printf("%s ¥t%7.4f¥n",buf22,bo2+bo3); /*<case8> out*/
//            printf("%s ¥t%7.4f <case8>¥n",buf22,bo2+bo3); //<case8> out
            goto skip2;
          }
        }
        else if(sa_sel(ary1,buf44)==FAIL){
//        strcpy(buf55,buf33);
//        printf("%s<base9>¥n",buf22);
//        printf("<buf11>%s<non>¥n",buf11);
//        printf("<buf33>%s<non>¥n",buf33);
//        printf("<buf44>%s<non>¥n",buf44);

          sa_reset(ary1);
          if(sa_sel(ary1,"<UNK>")==CONT){
            x=0;
            for(i=sa_left(ary1);i<=sa_right(ary1);i++){
              pos=sa_aryidx2txtidx(ary1,i);
              s=sa_getline(ary1,pos);
              strcpy(buf7,s);
              len1 = strlen(buf7);
              for(v=len1-15;v<len1-8;v++){
```

```
                buf9[x]=buf7[v];
                x++;
            }
            buf9[x]='¥0';
            bo2=atof(buf9);
            free(s);
            break;
        }
        sa_reset(ary1);
        x=0;
        if(sa_sel(ary1,buf55)==CONT){
            for(i=sa_left(ary1);i<=sa_right(ary1);i++){
                pos=sa_aryidx2txtidx(ary1,i);
                s=sa_getline(ary1,pos);
                strcpy(buf8,s);
                len2 = strlen(buf8);
                for(v=len2-7;v<len2;v++){
                    buf10[x]=buf8[v];
                    x++;
                }
                buf10[x]=' ';
                bo3=atof(buf10);
                free(s);
                break;
            }
//          printf("<bo2>%f¥n",bo2);
//          printf("<bo3>%f¥n",bo3);
//          printf("%d ",num++);
            printf("%s ¥t%7.4f¥n",buf22,bo2+bo3); /*<case9> out*/
//          printf("%s ¥t%7.4f <case9>¥n",buf22,bo2+bo3); //<case9> out
            goto skip2;
        }
        else if(sa_sel(ary1,buf55)==FAIL){
//          printf("%s <base10>¥n",buf22);
//          printf("<buf11>%s<non>¥n",buf11);
```

```
//                printf("<buf33>%s<non>¥n",buf33);
//                printf("<ubf44>%s<non>¥n",buf44);
//                printf("<buf55>%s<non>¥n",buf55);

                  sa_reset(ary1);

                  if(sa_sel(ary1,"<UNK>")==CONT){
                    for(i=sa_left(ary1);i<=sa_right(ary1);i++){
                    pos=sa_aryidx2txtidx(ary1,i);
                    s=sa_getline(ary1,pos);
                    strcpy(buf8,s);
                    len2 = strlen(buf8);
                    for(v=len2-7;v<len2;v++){
                      buf10[x]=buf8[v];
                      x++;
                    }
                    buf10[x]=' ';
                    bo3=atof(buf10);
                    free(s);
                    break;
                     break;
                    }
                  }
                  else if(sa_sel(ary1,"<UNK>")==FAIL){
                    printf("<UNK><non>¥n");
                    goto skip2;
                  }
//                printf("<bo2>%f¥n",bo2);
//                printf("<bo3>%f¥n",bo3);
//                printf("%d ",num++);
                  printf("%s ¥t%7.4f¥n",buf22,bo2+bo3);   /*<case10> out*/
//                printf("%s ¥t%7.4f <case10>¥n",buf22,bo2+bo3);   //<case10> out
                  goto skip2;
                }
              }
```

```
            else if(sa_sel(ary1,"<UNK>")==FAIL){
              printf("<UNK><non>¥n");
              goto skip2;
            }
//          goto skip2;
          }

        }
        printf("here10¥n");
        printf("<buf11>%s<non>¥n",buf11);
        w=0;
        v=0;
        strcpy(buf11,buf2);
        for(;;){
          if(buf11[v]==' '){
            w++;
          }
          if(w==1){
            break;
          }
          v++;
        }
        buf11[v]='¥0';
        printf("<buf11>%s¥n",buf11);
        sa_reset(ary1);
        if(sa_sel(ary1,buf11)==CONT){
          for(i=sa_left(ary1);i<=sa_right(ary1);i++){
            pos=sa_aryidx2txtidx(ary1,i);
            s=sa_getline(ary1,pos);
            printf("<buf11>%s¥n",s);
            free(s);
            break;
          }
        }
        else if(sa_sel(ary1,buf11)==FAIL){
```

```
                printf("<buf11>%s<non>¥n",buf11);
            }
        }
    skip2:
    }
//      printf("1111 ¥n");

    //      narabi(m,buf4,buf6);

    //out(m,buf6);

    skip1:
  }
  sa_closefiles(ary3);
  sa_closefiles(ary2);
  sa_closefiles(ary1);
  if((r=fclose(f1))==-1) exit(1);
}
```