〈経営研究　第16巻　第1号　平成14年9月〉

英書19冊の語単位エントロピー計算

横　井　右　門

キーワード

著作権　copyright　　跼天蹐地（きょくてんせきち）
平均情報量（エントロピー）　entropy　　情報理論　information theory
文字列　character string　　語　word　　語彙　vocabulary

ばじめに

「エントロピー」という言葉は，熱力学でも使われ，またその定義式が同一であるため現在，なお誤解が多い言葉である。本稿が報告するのはあくまでも，熱力学に関する「エントロピー」ではなく，情報理論の「エントロピー」である。また実際に計算されたことが余りない尺度である。

1.「エントロピー」の使用例の紹介

2000年8月23日付け日本経済新聞に同志社大学の室田　武教授が「もっとエントロピーに目を」という論文を発表している。その冒頭部分を引用する。

「日本は年増加が3億トンの肥満国

日本の経済や社会の展望をエントロピー法則とのかかわりを通じて考えてみたい。最初に言えるのは，エントロピーは増大する一方の物理量だという点である。粗い言い方だが，これを社会・経済的な文脈に置き換えると，この世の中では廃熱や廃物が増える一方である，ということだ。

古代エジプトのファラオをはじめ，不老不死の人間がいないのと同様に，生物学的な意味での生命は遅かれ早かれ劣化するし，生物に有用な物質やエネルギーはやがて劣化して役に立たなくなってしまう。エントロピーの法則は「熱力学第2法則」の別名で，このように，すべての普遍的な真理として認める法則である。

このために，エントロピーを語るのは憂鬱だという心理が生まれやすい。なるほど，廃熱と言えば，ダイオキシンなどごみ問題が頭に浮かぶ。そういう厄介ごとは，なるべくなら考えたくないところである。

しかし,エントロピーが増える一方だというのは否定しがたい法則である。とすれば，避けずに正面から向き合う方が，むしろ賢明かもしれない。物理学上の意味合いはさておいて，人間の経済的な話にしてしまえば，エントロピーとは，エネルギーや物質の，いわば「品質」のよしあしの指標である。質のいい，有用なもののエントロピーは小さく，劣悪になればそのエントロピーは大きい。」

室田教授の例は，熱力学におけるエントロピーについて書いてあることが明白である。

以下，「エントロピー」の近年収集した例をまとめて紹介する。

石原慎太郎：国家なる幻想，文芸春秋97年2月号

「エントロピーは人間たちにこの世の有限について教えたといわれるが，ほとんどの人間にはその実感も乏しく人々は未だに地球や宇宙の広大さの迷妄から覚めることはない。」

西部　邁：改革のエントロピー，正論97年2月号

「エントロピーの増大こそがエントロピーの減少なのだといわれても私にはまったくの珍粉漢である。」

早坂　暁：花へんろ通信，週刊新潮97年1月2日号

「そして同一均質社会ほど，嫉妬エネルギーが増大するのだ。いわばエントロピー増大だ。

M. ミッチェル・ワールドロップ：複雑系，新潮社，1996，P404

「この法則（熱力学第二法則）が働いているかぎり，宇宙には，原子が無秩序化し，すべてのものが冷え，崩壊し，終局を迎えるという避けがたい流れがある。この分子スケールの無秩序さの増大，つまり物理学者のいうエントロピーの増大を，どうしたら逆転させることができるか，と彼らは問うのだが，結局，人類が消滅し，すべての星が冷たくなったそのずっとあとに，コンピュータはこの大偉業を成し遂げる方法を見つける――そして『光あれ！』とのたまい，まったく新しい低エントロピーの宇宙を生じさせる。

アシモフのこの小説を読んだときファーマーは十四歳だったが，当時でも，これがある深遠な問題へ向う道を指し示しているような気がした。もしエントロピーがつねに増大しているなら，そしてもし原子スケールの不規則さや無秩序が避けがたいものなら，なぜ宇宙はいまもって恒星や惑星や雲や樹木を生み出すことができるのだろうか，とファーマーは自問した。（中略）『ぼくは生命や組織化が，ちょうどエントロピーの増大が避けがたいのと同じ程度に避けがたいものだという考え方に立っている』とファーマーはいう。」

牧野　昇：日本社会を襲う四つの危機，Voice1997年12月号，PHP研究所

「自然法則にはかなわないわけで地球を汚染する方向は簡単であるが，こ

れを凝縮させるのは難しいのである。

コーヒーに砂糖を入れると溶解して拡がる。これはきわめて自然の流れであるが，逆に溶けた砂糖をコーヒーから抽出して戻すには大変なカネとエネルギーが必要だ。またコメ粒を握ってばら撒くのは一瞬で済むが，これをもとのように集めるのは大変な苦労が必要だ。かくのごとく，地球は分散化し，役に立たない資源が増えていき，ついには死滅していく。これがエントロピー法則に基づく地球の運命である。」

井波律子：酒池肉林，講談社，1993 年

「いまや前人未踏の高みにのぼった始皇帝の得意や思うべし。なにごとにつけてもスケールの大きい始皇帝のことだから，その奢侈もまたはなはだしいかぎりだった。ただ始皇帝の場合は，エントロピーを暴発させ，長夜の宴でただらに富を蕩尽した殷の紂王とは異なり，その奢侈にも一種のダイナミズムが認められる。」

以上の「エントロピー」の使用例は，ほとんどが熱力学のエントロピーを意識しているように想像できる。中には「エントロピー」を使う必要がないと思われるのもある。ここに山本夏彦が（「豆朝日新聞」始末）文春文庫 1995 年において指摘した文を引用しておく。

「ユニオン・ショップとクローズド・ショップの違いも事典では分からない。両方読んでもどこが違うか定かでない。オーディナリ・ピープルとコモン・ピープルの違いも分からない。共に並，普通，尋常以下無数の字句が並んでいるが，その大半は同じだから酷似した言葉だと思うと，イギリス人はオーディナリと言われれば，喜ぶが，コモンといわれるとあなどられたと思う。コモンは一段下なのだそうだ。

かくて字引はそれを知らない人はもとより，すこし知るものにも役に立たない。そのことを字引の編者は知らなすぎる。エントロピーの如きはどの事

典を見ても分からない。書いている当人が分かってない。分からないことを分かったふりして書くのはさぞつらかろうと思うのにつらくない。」

しかし，少し長くなるが，つぎに情報理論のエントロピーと熱力学のエントロピーとの違いを意識して書かれた例を紹介しておく。

西部　邁：電子国家という無残な未来像，「エントロピーのおそろしさ」，Voice　2000年12月号

「情報理論の根底にあるのは（C・E・シャノンの出した）エントロピーの概念であり，それは『無秩序の度合』を表す。だから情報の存在意義は『エントロピーの減少』にあるといってさしつかえない。

——中略——

個人や個別組織のミクロ次元におけるエントロピーの減少が，社会のマクロ次元ではエントロピーを増大させている。このいわば『合成の誤謬』とでもよぶべき状況が生じているのは，情報技術にかんする社会的なルール（法律と徳律）が準備されていないからである。はっきりしているのは，そのルールはITによっては提供されないということだ。そのことに触れずにIT革命を呼号するのは，国家の本質はルールの体系にこそあり，ということを押さえていないからだ。それもそのはず，IT革命は反国家思想によって煽り立てられているのである。

エントロピーが最大となるのは『あらゆる事柄が同じ確率で起こる』とき，いいかえれば一寸先は闇の状態の場合である。だからエントロピーという言葉は，日常語においては，『類似性の度合』という意味を持つ。今の社会はその意味でもエントロピーを増大させている。個性の尊重だの独創性の発揮だのを誰しもがこぞって礼賛している。価値の多様化だの欲望の差異化だのも誰しもが好む標語である。そしてその結果はといえば，皆が類似の情報機器を装備し，そしてそれら似たもの同士が（水の分子のブラウン運動の

ように）衝突するにつれ，『何が起こっても不思議ではない』といったような無秩序の支配である。――したがって，母親が娘を保険金殺人に処そうが，息子が叱られるのを嫌って母親を惨殺しようが，エントロピー増大の結果であって，驚くには当たらないのである。

また熱力学にあっては，『利用不可能なエネルギー量』のことをエントロピーとよぶ。産業や生活の廃棄物がこの意味でのエントロピーを高めていることは論を俟たない。リサイクル可能なのは廃棄物のおそらく一，二割であり，それゆえエントロピーは（廃棄物のストックの累積とともに）増大する。私の問いたいのは，情報機器の氾濫ははたしてこの意味でもエントロピー増大から無縁でおれるかということである。」

エントロピーという言葉が誤用されるのは，それがカタカナ語であり，かつ定義式が熱力学でも，情報理論でも同じ形をしているせいもある。西部氏のように正確に使用されるようになってくるまでに，かなり長い時間が経過したが，少しずつ誤解が消滅しつつある傾向を示している。実は四十年も前に，この誤解についてJ. R. Pierceがつぎのように喝破している。

Thus, in physics, entropy is associated with the possibility of converting thermal energy into mechanical energy. If the entropy does not change during a process, the process is reversible. If the entropy increases, the available energy decreases. Statistical mechanics interprets an increase of entropy as a decrease in order or, if we wish, as a decrease in our knowledge.

The applications and details of entropy in physics are of course much broader than the examples I have given can illustrate, but I believe that I have indicated its nature and something of its importance. Let us now consider the quite different purpose and use of the entropy of communication theory.

In communication theory we consider a message source, such as a writer or a speaker, which may produce on a given occasion any one of many possible

messages. The amount of information conveyed by the message increases as the amount of uncertainty as to what message actually will be produced becomes greater. A message which is one out of ten possible messages conveys a smaller amount of information than a message which is one out of a million possible messages. The entropy of communication theory is a measure of this uncertainty and the uncertainty, or entropy, is taken as the measure of the amount of information conveyed by a message from a source. The more we know about what message the source will produce, the less uncertainty, the less the entropy, and the less the information.

We see that the ideas which gave rise to the entropy of physics and the entropy of communication theory are quite different. One can be fully useful without any reference at all to the other. Nonetheless, both the entropy of statistical mechanics and that of communication theory can be described in terms of uncertainty, in similar mathematical terms. Can some significant and useful relation be established between the two different entropies and, indeed, between physics and the mathematical theory of communication? (1)

2. バーナードのエントロピー計算例

本稿の目的は，最近の英文について語単位エントロピーの値を実際に計算し，データとして提供することにある。「自然言語は自然現象である」といわれている (14)。挑戦するのに恰好のテーマである。当然情報理論のエントロピーであり，熱力学のエントロピーの計算ではない。通信工学の世界ではなく，自然言語の世界において，実際に計算されているエントロピーの例が少ない。そのせいもあってか，1955 年に G. A. Barnard が Statistical Calculation of Word Entropies for Four European languages, IRE Trans-IT-1 において発表した値が最近でも南氏によって，つぎのように引用されているからである。(2)

	英語	仏語	独語	スペイン語
1次	4.124	3.98	4.10	4.015
2次	3.56			
3次	3.3			
8次	2.35			

この計算値は，他にも藤田氏，堀田氏が引用している。[3] [4]

なおバーナードは Word Entropy としているが，計算してみると文字列のエントロピーであることが推定できる。[7]

3. ヤグロムのエントロピー計算と引用

ソ連のヤグロム兄弟は，1957年「情報理論入門」[5] においてつぎのように述べている。

「また，他の国語についても，同じように計算できる。たとえば，ラテン文字のアルファベットには，全部で26個の文字がある。したがって，ラテン文字のアルファベットで書かれた文章の文字に含まれる最大の情報 ε_0 は，10進単位で $\log 26 \fallingdotseq 1.415$ である。さらに，色々な国語で，それぞれの文字が現われる相対頻度を計算すると，一つの文字に含まれる平均情報量 ε_1 は，英文では1.242（10進）単位，仏文では1.200，独文では1.233になる。」この文脈からは，誰が計算したのか不明確であるが，この1.242，1.200，1.233は2進単位に換算すると，それぞれ4.126, 3.986, 4.096になる。バーナードの計算の4.124，3.98，4.10に極めて近い値である。計算尺で計算したときの視読誤差の範囲である。

またヤグロムは次のように記述している。[6]

「シャノンは，二つ及び三つの文字のいろいろな組合せの相対頻度の表を作って，英語での ε_2 と ε_3 を計算し，10進単位で $\varepsilon_2 = 1.076$，$\varepsilon_3 = 0.993$

であることがわかった（前に述べたように，英語では，個個の文字を独立と考えて，それぞれの文字の相対頻度だけを使って計算すれば，一つの文字に含まれる情報は $\varepsilon 1 = 1.242$ 単位であった)」

この 1.076, 0.993 を 2 進単位に換算すると，それぞれ 3.574 と 3.299 になる。バーナードは 2 次エントロピーを 3.56，3.3 と計算しているが，極めて近い値である。

またヤグロムは次のように続けている。(6)

「さらに，英文に現われるいろいろな単語の頻度に関する統計資料を使って，シャノンは近似計算で，$\varepsilon 8$（つまり，すぐ前に在る 7 コの文字がわかったものとして，文章の文字一つあたりに分配される情報）がほぼ，0.6 単位になることをしめした。」バーナードの 8 次エントロピーは二進で 2.35 であるが，これを 10 進単位に換算すると 0.707 になるが小数点以下第 1 位で誤植があり 7 が 6 に変ったとすると「0.6 単位」と印刷されたと解釈することも不可能ではない。この訳書は実に誤字誤植が多い。例えば，114 ページではつぎの記述がある。

「1 / log2 = 2.3219...」

これは 3.3219 でなければならない。121 ページに次の記述がある。

「まず，上にのべたことから，たとえばロシア文字のアルファベットでは，110 枚の紙に文字 ё を，87 枚の紙には文字 о を，‥‥，2 枚の紙には ф を書いた 1000 枚の紙片をよくまぜて入れてあるツボから，一枚の紙片を取り出して，文章のそれぞれの文字をきめるとすれば，一つの文字に含まれる情報は 1.342（10 進単位）に等しい」。これはその直前のページの記述から「110 枚の紙に文字 о を，87 枚の紙に文字 ё を」でなければならない。

4. 本稿の目的

ヤグロムがバーナードの計算をシャノンの計算としかつ対数の底を 2 ではなく，10 として発表したり，南氏，藤田氏，堀田氏が最近にいたるまで引

用しているのはエントロピーの計算が厄介な問題を抱えているからである。バーナードが計算に使用した計算手段が何であるかは判らないが，手動計算機または有効数字が最大4ケタであることから計算尺であることが想像できる。

さらに本質的に厄介なことはデータの入力である。どうしても原文を目視して，手作業で入力せざるを得ない。スキャナーによる入力も考えられるが，正読率99パーセントのスキャナーシステムがあったとしてもペーパーバックの1ページは約2000字であるから1ページあたり20字の誤字が出来することになる。1ページ20字の誤字がある文書の校正作業は非常に苦しい作業である。それなら，水準の高いタイピストが手作業で正確に入力したほうが効率的であるといえる。これを裏付ける事実がある。

東京大学の月尾氏，坂村氏が次のように言及している。(11)

「月尾　それからゲームも同様です。ただ坂村さんのいわれるように，日本は文化を積極的に発信していこうという意欲は弱い。インターネットで非常にアクセスの多いサイトにアメリカの議会図書館が運営する『アメリカン・メモリー』というものがあります。これは議会図書館に建国以来蓄積された膨大な資料の中から，アメリカ文化を代表するものを数百万点選択してデジタル情報にし，世界中どこからでもアクセスできるようにしようという試みです。日本も遅ればせながら昨年やっと国会図書館がはじめましたが，十年後れています。

坂村　『アメリカン・メモリー』は徹底していますね。

北京大学に行ったとき，五百台ぐらいのコンピュータを並べてパチパチ入力をしているから，なにをしているんですかと尋ねたら，アメリカン・メモリーの下請けなんだと（笑）。この入力はトリプル・モジュラー・リダンダンシーという入力方式で，三人一組で同じものを打ち込んで，もし違う入力があったときは多数決をとって二人が同じものを正しいとして記録するやり方ですから人手がかかる。アメリカでやると人件費が高いから中国でやって

いるんですね。」

もし，完成度の高いスキャナー・システムが実用可能ならばこのような入力方法を採用しないはずである。

コンピュータの進歩により計算手段についてはバーナードが苦労したであろう問題は現在存在しない。しかし，入力データをどのように作成するかと言う問題は相変わらず，存在する。それが実際に計算されたエントロピーが少なく，1955年のバーナードの計算結果が今なお引用されている原因であろう。そういう意味で，前川守氏が実際に計算した業績は高く評価されなければならない。[12] しかし発表された数値のけた数は少ない。これは，データ量が小さいからであろう。やはりデータ入力が大問題であることを示唆している。

adventure novelsを主とするペーパーバックから打ち込んだデータを数年かけて1050万バイト用意した。文字列についての15次までのエントロピーはすでに計算ずみである。[7] [8] 本稿は文字列ではなく語単位のエントロピーをこのデータによって計算し，計算結果を提供しようとするものである。

5. 著作権の問題

東京工業大学の田中穂積氏たちは次のように著作権問題に触れている。「日本はコーパスの整備という観点で，欧州に大きく立ち遅れている。著作権問題も大らかな欧州の事情に比べると，意識し過ぎて『跼天蹐地（きょくてんせきち)』ともいうべき状況に陥っているようにもみえる」[14]

例えば，関西大学の西本英樹氏他による「ドキュメント解析のための感性表現抽出システム」[15] という自然言語処理分野で高く評価すべき労作があるが，使用しているデータは夏目漱石の「虞美人草」である。著作権法第51条第2項は著作権の保護期間は著作者の死後五十年を経過するまでの間，存続すると規定している。夏目漱石は1916年に逝去している。したがって

著作権法上の問題は起きない。しかし「虞美人草」は明治40年1907年に発表されている。英語においてさえH.G. Wellsは約50年前todayではなくto-dayを使っている。現在英文にto-dayが使用されることはまず想像し難い。西本氏他による研究が現在この瞬間にわれわれが使用している日本語をデータとして採用することができたなら，さらに高く評価されるはずである。

著作権法第2条第1項15号は複製について次のように規定している。

「複製　印刷，写真，複写，録音，録画その他の方法により有形的に再製することをいい，次に掲げるものについては，それぞれ次に掲げる行為を含むものとする」

この「その他の方法」が複製という全体集合に属する補集合であるならば，ペンで書籍を書き写すことも著作権法上の「複製」になってしまう。また「印刷，写真，複写，録音，録画」という順序で規定した意義がなくなる。

著作権法第32条第1項は引用について次のように規定している。

「公表された著作物は，引用して利用することができる。この場合において，その引用は公正な慣行に合致するものであり，かつ，報道，批評，研究その他の引用の目的上正当な範囲内で行なわれるものでなければならない。

また名城大学の北川善太郎教授は，私的複製が許容されると述べておられる。(13)

以上の理由から今回のデータ作成は，原文を目で見て，キーボードから手作業で入力したものである。著作権法上の問題にはあたらない。しかし作成したテキスト・データが英語圏に流出したら倫理的に管理責任は問われるであろう。そこでデータは利用するとき以外は暗号化し，かつオフライン状態のコンピュータのハード・ディスクに保管してある。

6. 語単位エントロピー計算の問題点

すでに計算した文字列についての非ブロック化およびブロック化の15次までのエントロピーは，英語のアルファベット26文字およびスペースの計

27文字について計算したものである。[7] [8] この計算は英語の大文字はすべて小文字に直し，英字26文字以外はすべてワード・セパレータとしておこなったものである。大文字と小文字の処理は非常に難しい。THEとTheとtheの取り扱いですら難しい。作者によってあるいは出版社によって，大文字と小文字の使い方に微妙な違いがある。各chapterの始まりが，先頭の一文字だけが大文字のとき，先頭の1語が大文字だけで書かれているとき，先頭の4語が全部大文字で書かれているとき，あるいは先頭の一行がすべて大文字で書かれているとき等，多岐にわたっている。Chapterの先頭のTHEとTheとを別の単語とするか，同じ単語とするか。あるいは文の先頭のTheと文中のtheとは同じ単語なのか別の単語なのか，プログラムで処理するにはあまりにも難しい。

ハイフネーションについても同じような困難さがある。その2音節以上の単語にハイフンが含まれているとき，辞書に載っている段階でハイフンがあるのか，行のお終いだったから，ハイフンがあるのか，これもプログラムで判断するのは難しい。したがって本稿ではすべて英字は小文字として扱う。シャノンの計算と同じである。またすでに計算した文字列のエントロピーと比較するためにも現段階ではやむを得ぬ選択である。しかし，ここから別の問題が発生する。

英語では長さ1バイトの単語は，aとIである。しかし，英字以外の文字はすべてワード・セパレータとして扱うと，don'tはdonとtという単語に別れてしまい存在しない単語が出来てしまう。Frederick Forsythの短編Shepherdを例にして説明する。一番長い単語は，1229行目に出てくるone-twenty-one-point-five-megacycleという35バイトの単語である。これがone, twenty, one, point, five, megacycleという6個の単語になってしまう。Shepherdの平均単語長はAscii96文字について計算すると4.2710である。これを大文字を小文字にし，コンマ，ピリオド，ハイフン，クォーテーション・マーク等をすべてワード・セパレータとして，計算すると4.1917となる。情報理論では「アルファベット」という言葉を我々の普段使うのとは少し違

う意味で使う。

符号化により情報源符号に割り当てられた各系列を符号語 codeword と呼び，符号語すべての集合を符号 code と呼ぶ。また符号語に用いられる記号の集合を符号アルファベットという。[9] 1 ビットのとき符号アルファベットは {0, 1} が符号アルファベットである。小文字とスペースの 27 文字のとき符号アルファベットは {a, b, c, ⋯, y, z, } である。我々がワードプロセッサで漢字第二水準まで使った場合，約 7000 の符号アルファベットを使うことになる。英語テキストデータについてそのテキストの語彙は，符号アルファベットとして考える。フレデリック・フォーサイスのシェパードは 11,275 語からなり，語彙数は 2264 語である。そこでこのテキストの場合符号アルファベットは 2264 個であるとする。各符号アルファベットの出現確率がわかれば，確率の底を 2 とする対数値と確率の積の総和を求めると，1 単語あたりのエントロピーが求められる。

ブロック化の場合も非ブロック化の場合も，高次エントロピーを計算するときの単位はビット／バイトである。従って語単位エントロピーを計算するときにデータとして使用するテキストごとの平均単語長を計算しなければならない。その平均単語長で 1 単語あたりのエントロピーを割れば単位ビット／バイトのエントロピーが計算できる。

7. テキスト・データにした書籍

つぎの 19 冊である。著者名の下に略号，書名，発行所，発行年の順で列挙する。

Tom Clancy

C1　The Hunt for Red October: Harper Collins, 1993

C2　Op-Center: Berkley Novel, 1995

Frederick Forsyth

F1 The Day of Jackal: Bantam Books, 1995

F2 The Dogs of War: Bantam Books, 1995

F3 The Odessa File: Bantam Books, 1993

F4 The Shepherd: Corgi Books, 1990

Akira Kohchi

K1 Why I survived A-Bomb: Institute for Historical Review, 1989

Anne McCaffrey

M1 The Crystal Singer: Corgi Books, 1991

M2 Crystal Line: Del Rey Books, 1992

Robert B. Parker

P1 A Catskill Eagle: Dell, 1985

P2 Pastime: Berkley Novel, 1992

P3 God Save the Child: Penguin Books, 1997

P4 The Godwulf Manuscript: Dell, 1987

A. J. Quinnell

Q1 Man on Fire: Orion, 1994

Q2 In the Name of the Father: Signet Novel, 1987

Q3 The Blue Ring: Orion, 1994

Q4 Message from Hell: Orion, 1996

Jostein Gaarder

Ga Sophie's World: Berkley Books, 1996

H. G. Wells

We　A Short History of the World: Collins, 1953

8. 語単位エントロピー計算システム

IBM のドキュメント記述方式である HIPO（Hierarchy of Input Plus Output）の pseudo coding（擬似コード）を使用して計算システムを説明する。これは前川守氏の「文章を科学する」[12] でも採用されている手法である。

第 1 ステップ

テキスト・データを一文字ずつ読み，アルファベット以外の文字が現われたら，1 文字列すなわち 1 単語を 1 レコードとして出力ファイルを出力する。このとき 1 単語を 1 符号アルファベットと考える。そのとき文字数と総単語数のファイルも同時に出力する。（付録 1 参照）

```
void main（void）
{
    ファイル等データ定義
    ファイル・オープン等初期処理
    最初の一文字を読む
  while（データが存在する）
  {
      文字が大文字なら小文字にする
      if（文字が小文字なら）
      {
          work[i] = 文字
          i の増分
      }
      else
      {
```

```
            wordcnt の増分
            当該単語の出力
        }
        次の 1 文字を読む
    }
    ファイル・クローズを含む後処理
    wordcnt と文字数からなる小ファイルを出力する
}
```

第 2 ステップ

第 1 ステップの出力ファイルを昇順で整列 sort する。

第 3 ステップ

同一単語数を集計してゆき，それを第 1 ステップの総単語数で割り，出現確率を求め，その確率の対数値を求め，確率と対数値の積を累計していく。同一単語の種類の数が語彙 vocabulary ということになる。最後のレコードを処理し終わったとき期待値であるエントロピー（平均情報量）が計算できている。それを第 1 ステップの出力である文字数と単語数から求めた平均単語長で割ることによって，1 バイトあたりの語単位エントロピーが計算できる。エントロピーの計算には対数関数を使うので，<math.h> を include しなければならない。(付録 2 参照)

やはり擬似コードで説明する。

```
#include <math.h>
void main（void）
{
    データ定義
    ファイル・オープン等初期処理
    小ファイルから平均単語長を求める。
```

```
先頭の単語を読む
while（データがある）
{
    入力単語数の増分
    頻度の増分
    if（単語が変った）
    {
        出現確率の計算
        エントロピーの計算
        頻度をゼロにする
    }
    次の単語を読む
}
語単位エントロピーを計算する
1バイトあたりエントロピーを計算する
  後処理
}
```

9. データおよび計算結果および結論

計算結果はつぎのとおりである。1バイトあたり平均情報量が本稿の目的とする語単位エントロピーである。

英書19冊の語単位エントロピー計算

書籍略号	総文字数	英字数	単語数	平均単語長	語単位 平均情報量	1バイトあたり 平均情報量	語彙
C1	970,567	731,374	167,367	4.396882	9.797138	2.241969	11,793
C2	537,312	398,168	095,319	4.177216	9.647379	2.309524	9,336
F1	771,683	587,605	135,980	4.321260	9.642291	2.231361	11,879
F2	810,586	615,299	142,541	4.316646	9.661153	2.238117	11,732
F3	596,877	447,488	105,486	4.242156	9.531138	2.246768	9,524
F4	64,003	47,966	11,443	4.191733	8.962947	2.138244	2,264
K1	371,522	286,110	62,132	4.604874	10.092015	2.191594	9,659
M1	644,252	491,656	108,946	4.512841	9.830372	2.178311	11,226
M2	511,075	380,936	88,407	4.308889	9.659572	2.241778	9,621
P1	404,989	288,775	73,622	3.922401	9.171490	2.338234	6,883
P2	287,287	205,248	52,612	3.901163	9.075912	2.326463	5,492
P3	298,069	218,471	55,756	3.918341	9.192564	2.346035	5,912
P4	304,275	223,102	56,959	3.916888	9.164284	2.339685	6,029
Q1	547,589	408,693	96,531	4.233800	9.497183	2.243182	9,108
Q2	638,157	478,384	112,748	4.242949	9.558496	2.252796	10,045
Q3	622,674	466,224	111,175	4.193604	9.358046	2.231504	8,283
Q4	468,054	350,077	84,985	4.119280	9.261587	2.248351	6,943
GA	964,734	728,213	168,716	4.286571	9.545811	2.226911	11,540
WE	712,476	560,950	117,686	4.766497	9.650530	2.024659	11,922

この計算結果から判断できることは，1バイト当たりの平均情報量すなわちエントロピーは，2.024から2.34の範囲である。これは文字列エントロピーの第7次から第8次エントロピーに相当する。[7] 平均単語長は3.9から4.6の範囲であるが，単なる文字列と違って単語の場合は意味のある文字列であり，それゆえにエントロピーは大きく減少するという前川守氏の見解[12]を新しく19個の計算結果の提供によって裏付けていることになる。

10. 今後の計画

本稿は語単位の第1次エントロピーを計算した。当然語単位第2次エント

ロピー，語単位第3次エントロピーを計算する計画である。また日本語はホワイト・ハイフネーションであり，ワード・セパレータが存在しない。日本語のエントロピーと英語のエントロピーを比較するときはスペース等も符号アルファベットと見なし，英語は語単位ではなく文字列エントロピーを使わなければならない。

参考文献

(1) J. R. Pierce : Symbols, Signals and Noise, Harper & Row (1961) , p23
(2) 南　敏： 情報理論，産業図書（1995），p55
(3) 藤田広一：情報理論，昭晃堂（1996），p24
(4) 堀　淳一：エントロピーとは何か：講談社（1994），p194
(5) ヤグロム（井関，西田訳）：情報理論入門，みすず書房（1958），p121
(6) ヤグロム（井関，西田訳）：情報理論入門，みすず書房（1958），p124
(7) 横井右門：英書19冊の15次エントロピーの計算：経営研究第13巻第2号（1995）
(8) 横井右門：ブロック化による英書19冊の15次エントロピー計算：経営研究第13巻第3号（2000）
(9) 今井英樹：情報理論：昭晃堂（2000），p11
(10) H.G.Wells: A Short History of the World, Collins（1953）
(11) 月尾嘉男，西垣通，坂村健：東大IT三羽烏の「緊急提言」，諸君！，文藝春秋社，(2001年10月号)，p185
(12) 前川守：1000万人のコンピュータ科学3　文学編　文章を科学する，岩波書店，(1995) p85～87
(13) 北川善太郎：電子著作権管理システムとコピーマート，情報処理，Vol 38, No 8 (1997)
(14) 田中穂積，亀井真一郎，森口稔，加藤安彦：大きなコーパスを共有しよう，情報処理 Vol.41 No.7，情報処理学会，(2000) p776
(15) 西本英樹，泉原健史，吉田宣章，桑原尚史，堀井康史，山内昭：ドキュメント解析のための感性表現抽出システム，Cyber　Ecology，オフィス・オートメーション学会，p76

付録 1

```
001:/* Wort01.c 語単位レコード出力 */
002:#include <stdio.h>
003:void main(void)
004:{
005:    int i = 0, ccnt = 0, wordcnt = 0;
006:    float ave;
007:    char chara, work[100], fname1[50], fname2[50];
008:    FILE * fp1, * fp2, * fp3;
009:    printf("file1:");
010:    scanf("%s", fname1);
011:    printf("file2:");
012:    scanf("%s", fname2);
013:    fp1 = fopen(fname1, "r");
014:    fp2 = fopen(fname2, "w");
015:    printf("(Wort01.c) \n");
016:    chara = getc(fp1);
017:    while(chara != EOF)
018:    {
019:       if(65 <= chara && chara <= 90)
020:       {
021:          chara = chara + 32;
022:       }
023:       if(97 <= chara && chara <= 122)
024:       {
025:          work[i] = chara;
026:          i = i + 1;
027:       }
028:       else
029:       {
030:          if(i == 0)
031:          {
032:          }
033:          else
034:          {
035:             ccnt = ccnt + i;
036:             wordcnt = wordcnt + 1;
037:             work[i] = '\0';
038:             i = 0;
039:             fputs(work, fp2);
040:             fputc('\n', fp2);
041:          }
042:       }
043:       chara = getc(fp1);
044:    }
045:    ccnt = ccnt + i;
046:    wordcnt = wordcnt + 1;
047:    work[i] = '\0';
048:    fputs(work, fp2);
049:    fputc('\n', fp2);
050:    ave = (float)ccnt / (float)wordcnt;
051:    printf("%d words  %7d characters  ave: %8.6f\n", wordcnt, ccnt, ave);
052:    fclose(fp1);
053:    fclose(fp2);
054:    fp3 = fopen("statis.txt", "w");
055:    fprintf(fp3, "%7d %7d\n", wordcnt, ccnt);
056:}
```

付録 2

```
001:/* Wort03.c  語単位エントロピー計算 */
002:#include<stdio.h>
003:#include <string.h>
004:#include <math.h>
005:void main(void)
006:{
007:   char work[100], trace[100], * killa,
008:        fname1[50];
009:   int n, in_cnt = 0, out_cnt = 0, frequency = 0, words, characters;
010:   double proba, average, entropy = 0, selfinfo, entropybyte;
011:   FILE * fp1, * fp3;
012:   printf("in file 1: ");
013:   scanf("%s", fname1);
014:   fp1 = fopen(fname1, "r");
015:   fp3 = fopen("statis.txt", "r");
016:   fscanf(fp3, "%7d %7d", &words, &characters);
017:   average = (float)characters / (float)words;
018:   printf("(Wort03) ");
019:   printf("%7d words, %d characters average = %8.6f\n",
020:           words, characters, average);
021:   killa = fgets(work, 100, fp1);
022:   strcpy(trace, work);
023:   while(killa != NULL)
024:   {
025:      in_cnt = in_cnt + 1;
026:      frequency = frequency + 1;
027:      n = strcmp(work, trace);
028:      if (n > 0)
029:      {
030:         proba = (float)frequency / (float)words;
031:         selfinfo = log(proba) / log(2.0);
032:         entropy = entropy + proba * selfinfo;
033:         out_cnt = out_cnt + 1;
034:         frequency = 0;
035:         strcpy(trace, work);
036:
037:      }
038:      killa = fgets(work, 100, fp1);
039:   }
040:   frequency = frequency + 1;
041:   proba = (float)frequency / (float)words;
042:   selfinfo = log(proba) / log(2.0);
043:   entropy = entropy + proba * selfinfo;
044:   entropybyte = entropy / average;
045:   printf("%8.6f %8.6f %8.6f\n", entropy, average, entropybyte);
046:   out_cnt = out_cnt + 1;
047:
048:   printf("in %d records, out %d records\n", in_cnt, out_cnt);
049:   fclose(fp1);
050:   fclose(fp3);
051:}
052:
```