

ブロック化による英書19冊の15次エントロピー計算

横 井 右 門

キーワード

エントロピー entropy 冗長度 redundancy

ブロック block

1 はじめに

エントロピーという言葉は、非常に誤解されやすい。山本夏彦は、次のように言っている。

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、コモンと言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。書いている本人が分かってない。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。

片カナ語のはんらんはいいことではないが、とめようがない。

(「豆朝日新聞」始末) 文春文庫 1995 年

エントロピーが誤解されるのは、それが片かな語であるからだけではなく、その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。この点について三十年以上も前に J. R. ピアースがつぎのように言っている。⁽¹⁾

この安直だが誤りを招きやすい観念は専門家の間にさえたくさんの混乱をひき起こした。本当はコミュニケーション理論は電気通信の分野のある問題を解くために生まれたのである。そのエントロピーは、統計力学のエントロピーとの数学的類似のためにエントロピーと名づけられたのである。このエントロピーが主に関係する問題は、統計力学が取り組む問題とはまったく異なるものである。

この点に留意すれば、つぎのような「エントロピー」の使い方は情報理論に関係ないことが理解できる。

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。何事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。

井波律子 酒池肉林 講談社 1993 年

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力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。

牧野 昇 日本社会を襲う四つの危機 Voice 1998. 12

本稿はあくまでも情報理論のエントロピーについて報告するものである。

2 バーナードの計算

南氏によれば、G.A.Barnard は 1955 年 Statistical Calculation of Word Entropies for Four European Languages, IRE Trans — IT — 1 においてつぎのようにヨーロッパ系各国語についてエントロピーを計算している。⁽²⁾ F1, F2, F3, F8 はそれぞれ 1 次, 2 次, 3 次, 8 次エントロピーのことである。

	英語	仏語	独語	スペイン語
F1	4.124	3.98	4.10	4.015
F2	3.56			
F3	3.3			
F8	2.35			

この計算結果は、よく引用されている。まず 1956 年ソ連のヤグロムが底が 2 の対数ではなく、底が 10 の常用対数を使って発表している⁽³⁾。なお、不可解であるが 2 次, 3 次のエントロピー 3.56, 3.3 をシャノンが計算したことにしている⁽⁴⁾。

南氏だけでなく、藤田氏も次のように引用している。

英語、フランス語、ドイツ語などのアルファベットの最大エントロピーは 4.7 [bit] であるが、単純な確率過程とみても、おのこの 4.14, 3.98, 4.10 [bit] で 10 ~ 20 % の冗長度である。もしマルコフ過程とみるともっと冗長度は大きくなる⁽⁵⁾。

堀氏も 8 次エントロピーの 2.35 という値を最高記録として紹介している。1956 年のヤグロムはともかく、約 40 年経過してもバーナードの計算が引用されるのは、エントロピーの計算が容易でないからであろう。堀氏は、次のように言っている。

何しろ、前にみたように、文字のつながり方の制約は、三つや四つどころでなく、もっともっと長い文字の連鎖まで及ぶことが明らかなのだから。たぶん、少なくとも 14 か 15 の相つづく文字の列のすべての可能なものを数え上げ、それらの頻度分布をしらべてエントロピーを計算しなければ、実際の値に近い値は求まらないだろう。しかしそれは大変な仕事で、まだ誰もやっていない。ある人が英語の中に出てくる相つづく八つの文字のすべての列の出現頻度をしらべてエントロピーを計算したのが今までの最高記録である。その値は 2.35 であったが、実際はこれよりもっともっと小さいだろう。あとは大ざっぱに見当をつけるよりほかないのだが、まず 1.3 ぐらいだろうというのが、今のところの定説になっている。

3 ブロック化と非ブロック化

たとえば次の文

My name is Aram.

を

[My] [n] [am] [e] [is] [A] [ra] [m.]

と数え上げるのがブロック化であり、

[My][y][n][na][am][me][e][i][is][s][A][Ar][ra][am][m.]
と数え上げるのが非ブロック化である。「非ブロック化」は、筆者の造語である。生硬であることは否定しない。情報理論は一点から一点へ正確に早く情報を伝達するための理論であるからブロック化だけで考えてよいが、堀氏の言うように「文字の列のすべての可能なもの」を数え上げようとする上例でわかるように [y] [na] 等が数えられなくなってしまう。非ブロック化によるエントロピーについてはすでに報告済みであるが⁽⁷⁾、エントロピーの「実際の値」が 1.3 よりも小さく、8 次エントロピーも 2.35 よりも小さいであろうことが推測できる結果が出ている。(付録参照)

本稿はブロック化による計算と非ブロック化による計算に差があるかについて定量的に報告する。

4 データ

次の 19 冊の英書をスキャナーを使わずに、手作業で入力し、テキスト・データを作成した。

約 2000 時間を費やしている。全体で約 9.5 メガバイトである。著者名の下に略号、書名、発行所、発行年の順で列記する。

Tom Clancy

- C1 The Hunt for Red October : Harper Collins, 1993
- C2 Op-Center : Berkley Novel, 1995

Frederick Forsyth

- F1 The Day of the Jackal : Bantam Books, 1995
- F2 The Dogs of War : Bantam Books, 1995
- F3 The Odessa File : Bantam Books, 1973
- 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, 1977

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

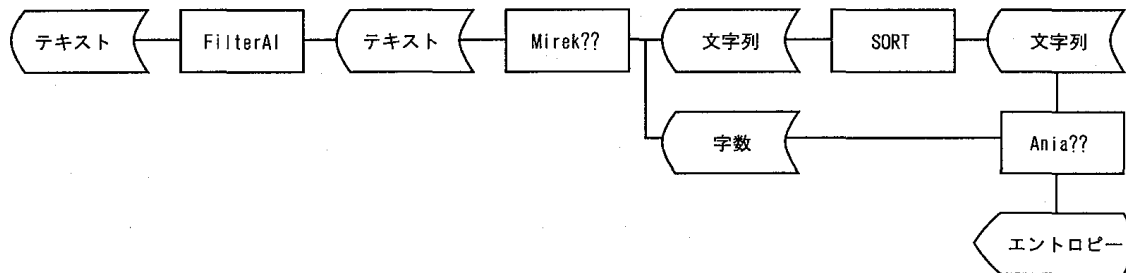
GA Sophie's World : Berkley Books, 1996

H. G. Wells

WE A Short History of the World : Collins, 1953

5 計算システム

文字列の長さ(エントロピー次数)ごとに次のようなプロセス・フローチャートになる。付録のプログラム・リストは 2 次エントロピー計算のためのものである。



5-1 プログラム FilterAl. c (付録参照)

テキスト・ファイルを読み, 英字とスペースを選ぶ。改行キーはスペースに置き換える。

5-2 プログラム Mirek??. c (付録参照)

ブロック化した ?? 連字の文字列レコードを出力する。?? には次数に対応して 01 から 15 までの種類がある。SORT 後のエントロピー計算の確率のために, SORT の対象にならないファイルの字数 1 レコードからなるファイルを出力する。

5-3 SORT コマンド

MS - DOS の整列コマンドを使う。

5-4 プログラム Ania ? ? . c (付録参照)

各次数のエントロピーを計算する。

6 計算結果

ブロック化によるエントロピー計算

	C 1	C 2	F 1	F 2	F 3
字数	970,567	537,312	771,683	810,586	596,877
平均語長	4.414206	4.312595	4.435900	4.413581	4.325885
標準偏差	2.392379	2.369556	2.414696	2.368385	2.333826
最大語長	27	35	22	24	35
H 0 1	4.080464	4.058987	4.063707	4.068114	4.063637
H 0 2	3.742419	3.710313	3.694234	3.703772	3.705273
H 0 3	3.484230	3.373429	3.357674	3.370493	3.364551
H 0 4	3.075395	3.039718	3.033887	3.042296	3.033457
H 0 5	2.770276	2.721004	2.728988	2.738629	2.715958
H 0 6	2.485898	2.420116	2.441744	2.453659	2.421134
H 0 7	2.228443	2.153669	2.184509	2.196421	2.159056
H 0 8	1.996424	1.917237	1.957680	1.967087	1.927200
H 0 9	1.794395	1.713981	1.758993	1.766522	1.726126
H 1 0	1.619661	1.541856	1.588377	1.595806	1.555371
H 1 1	1.470086	1.396485	1.441962	1.448589	1.410054
H 1 2	1.342822	1.273381	1.316689	1.322604	1.286709
H 1 3	1.233584	1.168478	1.209573	1.214928	1.180881
H 1 4	1.139302	1.078123	1.116803	1.121894	1.089975
H 1 5	1.057564	1.000449	1.036435	1.041319	1.011366

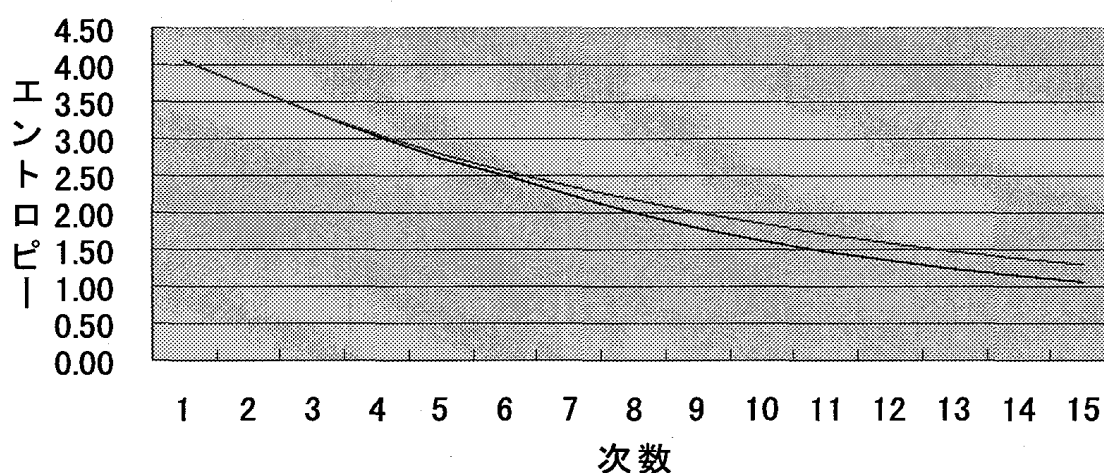
	F4	K1	M1	M2	P1
字数	64,003	371,522	644,252	511,075	404,989
平均語長	4.270953	4.757866	4.623770	4.414708	3.989992
標準偏差	2.401242	2.633331	2.590444	2.438119	2.105130
最大語長	35	27	27	31	24
H 0 1	4.077524	4.105486	4.090534	4.071662	4.028679
H 0 2	3.711067	3.752289	3.725483	3.713669	3.674967
H 0 3	3.316109	3.427694	3.381199	3.367188	3.313550
H 0 4	2.878737	3.086142	3.043566	3.024513	2.968768
H 0 5	2.459713	2.738057	2.729305	2.706558	2.647110
H 0 6	2.106632	2.408260	2.435980	2.408381	2.348010
H 0 7	1.818638	2.119895	2.172138	2.143418	2.085436
H 0 8	1.586561	1.874202	1.938185	1.907501	1.854946
H 0 9	1.410991	1.667889	1.736971	1.705514	1.659304
H 1 0	1.249493	1.495801	1.562835	1.534622	1.494658
H 1 1	1.125436	1.352811	1.418123	1.389680	1.358350
H 1 2	1.022268	1.232509	1.293968	1.266487	1.235835
H 1 3	0.934820	1.129970	1.187740	1.162664	1.134431
H 1 4	0.860979	1.042483	1.097099	1.073083	1.047205
H 1 5	0.799345	0.966610	1.018299	0.995253	0.971277

ブロック化によるエントロピー計算

	P2	P3	P4	Q1	Q2
字数	287,287	298,069	304,275	547,589	638,157
平均語長	3.957715	3.981495	3.972066	4.348522	4.312332
標準偏差	2.079894	2.183470	2.177281	2.381582	2.298410
最大語長	29	34	34	27	27
H 0 1	4.036607	4.053653	4.054947	4.063091	4.073119
H 0 2	3.681290	3.701183	3.701292	3.704909	3.714484
H 0 3	3.314106	3.340410	3.337266	3.356160	3.369981
H 0 4	2.953241	2.986432	2.985530	3.019879	3.036310
H 0 5	2.619666	2.648623	2.646158	2.706220	2.722496
H 0 6	2.312506	2.336198	2.338658	2.410149	2.430749
H 0 7	2.044254	2.061834	2.067259	2.147911	2.168368
H 0 8	1.811827	1.826532	1.832315	1.916677	1.936918
H 0 9	1.616247	1.626367	1.632117	1.715384	1.736283
H 1 0	1.451723	1.460402	1.464483	1.545764	1.564638
H 1 1	1.313912	1.320635	1.324688	1.400021	1.417955
H 1 2	1.197084	1.203565	1.206561	1.276917	1.294051
H 1 3	1.097861	1.103480	1.106159	1.171605	1.187877
H 1 4	1.012699	1.017823	1.020361	1.081189	1.096672
H 1 5	0.938954	0.943969	0.945989	1.003030	1.017629
	Q3	Q4	GA	WE	
字数	622,674	468,054	964,734	712,486	
平均語長	4.231167	4.152141	4.370858	4.822602	
標準偏差	2.209136	2.234328	2.431823	2.702675	
最大語長	20	20	31	27	
H 0 1	4.060667	4.059951	4.054536	4.084961	
H 0 2	3.699681	3.697690	3.700056	3.711221	
H 0 3	3.351394	3.343415	3.359420	3.376215	
H 0 4	3.014119	2.996766	3.308310	3.031720	
H 0 5	2.697816	2.674967	2.730551	2.719066	
H 0 6	2.410193	2.379017	2.452290	2.426639	
H 0 7	2.152292	2.116822	2.203634	2.169604	
H 0 8	1.923975	1.885608	1.979098	1.944580	
H 0 9	1.727152	1.688270	1.782295	1.746793	
H 1 0	1.557361	1.519972	1.611534	1.576796	
H 1 1	1.413374	1.378014	1.465353	1.432002	
H 1 2	1.289908	1.256671	1.339137	1.308271	
H 1 3	1.184743	1.153418	1.231332	1.201653	
H 1 4	1.093757	1.064728	1.137609	1.109820	
H 1 5	1.015051	0.989083	1.056242	1.031445	

7 ブロック化と非ブロック化の比較

F1 の The Day of the Jackal を例にして説明する。1 次エントロピーでは、当然ながら双方のエントロピーは等しい。しかし、15 次エントロピーでは、非ブロック化のほうが 1.286567 であり、ブロック化のほうが 1.036435 である。その差は実に 0.239 もある。グラフで分かるように、上の曲線が非ブロック化によるエントロピーであり、下の曲線がブロック化によるものである。



8 結論

すでに報告した非ブロック化によるエントロピー計算でも、8 次の段階で、どの英書もエントロピーの値はバーナードの計算した値より低くなっているし、15 次エントロピーは 1.05 と 1.31 の間である。本稿のブロック化によるエントロピー計算では、15 次の段階で 1.1 と 0.9 の間に収束している。堀氏の予測をはるかに上回った事になる。

9 今後の計画

語のエントロピー，それも 3 個の連続する単語列のエントロピーを計算するつもりである。

1 語の長さを 4.5 バイトとし，2 個のスペースを挿入すれば，3 単語で 15 連字に対応するからである。

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) ヤグロム (井関・西田訳) : 情報理論入門, みすず書房 (1958) p121
- (4) 同上 p124
- (5) 藤田広一 : 情報理論, 昭晃堂 (1996) p24
- (6) 堀 淳一 : エントロピーとは何か : 講談社 (1994) p194
- (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 (1999) p149

```

0001:  /* FilterAl.c 英字とスペースを選ぶ */
0002:  #include <stdio.h>
0003:  #include <string.h>
0004:  #include <time.h>
0005:  #include <math.h>
0006:  void main(void)
0007:  {
0008:      int i = 0, j = 0, cnt = 0, chara_cnt = 0, dummycnt = 0;
0009:      long int nowtime, start, finish;
0010:      char chara, infile[50], outfile[50];
0011:      FILE * fp0, * fp1;
0012:
0013:      printf("Input-file name: ");
0014:      scanf("%s", infile);
0015:      printf("%s\n", infile);
0016:
0017:      printf("Output-file name: ");
0018:      scanf("%s", outfile);
0019:      printf("%s\n", outfile);
0020:
0021:      fp0 = fopen(infile, "r");
0022:      fp1 = fopen(outfile, "w");
0023:
0024:      time(&start);
0025:      if(fp0 != NULL)
0026:      {
0027:          chara = getc(fp0);
0028:          while(chara != EOF)
0029:          {
0030:              if(65 <= chara && chara <= 90)
0031:              {
0032:                  chara = chara + 32;
0033:              }
0034:              if(chara == 0x0a)
0035:              {
0036:                  chara = 32;
0037:              }
0038:              if((97 <= chara && chara <= 122) || (chara == 32))
0039:              {
0040:                  fputc(chara, fp1);
0041:                  chara_cnt = chara_cnt + 1;
0042:              }
0043:              chara = getc(fp0);
0044:          }
0045:      }
0046:      time(&finish);
0047:      time(&nowtime);
0048:      fclose(fp0);
0049:      fclose(fp1);
0050:      printf("(FilterAl.c) %s infile: %s outfile: %s",
0051:              ctime(&nowtime), infile, outfile);
0052:      printf("\nTime: %4.0f %i bytes\n",
0053:              difftime(finish, start), chara_cnt);
0054:  }

```

ブロック化による英書 19 冊の 15 次エントロピー計算

```

0001:  /* Mirek02.c 英語 2 連字集計 step 0 */
0002:  #include <stdio.h>
0003:  #include <string.h>
0004:  #include <time.h>
0005:  #include <math.h>
0006:
0007:  void main(void)
0008:  {
0009:      int i = 0, j = 0, cnt = 0, stringcnt = 0, dummycnt = 0;
0010:      long int nowtime, start, finish;
0011:      char chara, infile[50], work0[50], outfile[50],
0012:           diskette[50];
0013:      FILE * fp0, * fp1, * fp2, * fp3;
0014:
0015:      printf("Input-file name: ");
0016:      scanf("%s", infile);
0017:      printf("%s\n", infile);
0018:
0019:      printf("Output-file name: ");
0020:      scanf("%s", outfile);
0021:      printf("%s\n", outfile);
0022:
0023:      printf("Diskette file name: ");
0024:      scanf("%s", diskette);
0025:      printf("%s\n", diskette);
0026:
0027:      fp0 = fopen(infile, "r");
0028:      fp1 = fopen(outfile, "w");
0029:      fp2 = fopen(diskette, "w");
0030:      fp3 = fopen("filesize.txt", "w");
0031:
0032:      work0[0] = 32;
0033:
0034:      time(&start);
0035:      if(fp0 != NULL)
0036:      {
0037:          chara = getc(fp0);
0038:          while(chara != EOF)
0039:          {
0040:              fputc(chara, fp1);
0041:              chara = getc(fp0);
0042:              fputc(chara, fp1);
0043:              /* fputc('¥0', fp1); */
0044:              fputc('¥n', fp1);
0045:              chara = getc(fp0);
0046:              stringcnt = stringcnt + 1;
0047:          }
0048:          fprintf(fp3, "%d¥n", stringcnt);
0049:      }
0050:      time(&finish);
0051:      time(&nowtime);
0052:      fclose(fp0);
0053:      fclose(fp1);
0054:      printf("¥n(Mirek02.c) %s infile: %s outfile: %s",
0055:             ctime(&nowtime), infile, outfile);
0056:      fprintf(fp2, "(Mirek02.c) %s infile: %s outfile: %s",
0057:             ctime(&nowtime), infile, outfile);
0058:      printf("¥n¥4.Of seconds Wordcnt = %i strings ¥n",
0059:             difftime(finish, start), stringcnt);
0060:      fprintf(fp2, "¥4.Of seconds Wordcnt = %i strings¥n",
0061:             difftime(finish, start), stringcnt);
0062:      fclose(fp2);
0063:      fclose(fp3);
0064:  }

```

```

0001: /* Ania02.c Calculation of 2nd order entropy */
0002: #include <stdio.h>
0003: #include <string.h>
0004: #include <time.h>
0005: #include <math.h>
0006:
0007: void main(void)
0008: {
0009:     double entropy = 0, proba = 0;
0010:     int totalcnt, stringcnt = 0, uniquecnt = 0;
0011:     long int nowtime, start, finish;
0012:
0013:     char infile[50], outfile[50], work0[50],
0014:         trace[50], diskette[50];
0015:     const char * Ania;
0016:     FILE * fp0, * fp1, * fp2, * fp3;
0017:
0018:     printf("Input-file name: ");
0019:     scanf("%s", infile);
0020:     printf("%s\n", infile);
0021:
0022:     printf("Output-file name: ");
0023:     scanf("%s", outfile);
0024:     printf("%s\n", outfile);
0025:
0026:     printf("Diskette-file name: ");
0027:     scanf("%s", diskette);
0028:     printf("%s\n", diskette);
0029:
0030:
0031:     fp0 = fopen(infile, "r");
0032:     fp1 = fopen(outfile, "w");
0033:     fp2 = fopen(diskette, "w");
0034:     fp3 = fopen("filesize.txt", "r");
0035:
0036:     time(&start);
0037:     if(fp0 == NULL)
0038:     {
0039:     }
0040:     else
0041:     {
0042:         fscanf(fp3, "%d", &totalcnt);
0043:         printf("%7d\n", totalcnt);
0044:         Ania = fgets(trace, 50, fp0);
0045:         strcpy(work0, trace);
0046:         while(Ania != NULL)
0047:         {
0048:             if(strcmp(trace, work0) == 0)
0049:             {
0050:                 stringcnt = stringcnt + 1;
0051:             }
0052:             else
0053:             {
0054:                 fprintf(fp1, "%7d %s", stringcnt, trace);
0055:                 proba = (float)stringcnt / (float)totalcnt;
0056:                 entropy = entropy + proba * log(proba) / log(2);
0057:                 strcpy(trace, work0);
0058:                 stringcnt = 1;
0059:                 uniquecnt = uniquecnt + 1;
0060:             }
0061:             Ania = fgets(work0, 50, fp0);
0062:         }
0063:         printf("%7d %s", stringcnt, trace);
0064:         fprintf(fp1, "%7d %s", stringcnt, trace);
0065:         proba = (float)stringcnt / (float)totalcnt;
0066:         entropy = entropy + proba * log(proba) / log(2);
0067:     }
0068:     fclose(fp0);
0069:     fclose(fp1);
0070:     entropy = entropy / 2.0;
0071:     fp2 = fopen(diskette, "w");
0072:     time(&finish);

```

ブロック化による英書 19 冊の 15 次エントロピー計算

```
0073:     time(&nowtime);
0074:     printf("\n(Ania02.c) %s", ctime(&nowtime));
0075:     fprintf(fp2, "(Ania02.c) %s\n", ctime(&nowtime));
0076:     printf("%4.0f seconds  filename: %s 2nd order entropy = %8.6f\n",
0077:           difftime(finish, start), infile, entropy);
0078:     fprintf(fp2, "%4.0f seconds  filename: %s 2nd order entropy = %8.6f\n",
0079:           difftime(finish, start), infile, entropy);
0080:     printf("total: %d strings, unique: %d strings\n",
0081:           totalcnt, uniquecnt);
0082:     fprintf(fp2, "total: %d strings, unique: %d strings\n",
0083:           totalcnt, uniquecnt);
0084:     fclose(fp2);
0085: }
```