Program Integrated Information の統計的法則

加藤 直孝[1]　　有澤 誠[2]

(株)日本アイ・ビー・エム ナショナル・ランゲージ・サポート[1]

慶應義塾大学 環境情報学部 大学院 政策・メディア研究科[2]

katosan@jp.ibm.com[1]　　arith@sfc.keio.ac.jp[2]

1. はじめに

言語の統計的法則で有名なものに Zipf の法則がある．中でも，単語の使用頻度に順位を振り，その順位と使用頻度の関係を示す法則が有名である[1]．Zipf の法則は，自然界におけるベキ法則(Power Law)の例であることが明らかになっている[2]．さらに，Li らはこの Zipf の法則を遺伝子解析等に応用している[3]．

ベキ法則(スケール法則あるいはパレートの法則ともよばれる)は自然界の複雑なシステムを理解するために重要な法則となっている[4]．しかし，なぜベキ法則が色々なところに現れるのか，その理由はまだわかっていない[5]．自然言語処理分野からも更なる研究が必要である．そこで，我々は，テキストファイル上の PII(Program Integrated Information)文字列の統計的な特徴を調べた．その結果，Zipf の法則では述べていない新たな統計的法則が PII に存在することを発見した．本稿はこの発見した 2 つの法則を紹介する．

2. 概要

コンピュータに関する翻訳には2種類ある．マニュアルの翻訳とプログラムに統合した文字列の翻訳である．IBM ではこの文字列を Program Integrated Information (PII)とよぶ．マニュアル翻訳は通常の本の翻訳と共通する点が多い．一方，PII の語句や文章は通常の本やマニュアルの語句や文章とは異なる．そのため，PII の翻訳も通常の本の翻訳とは異なった特徴をもつ[6]．ただし，PII の原典である英語について，統計的にも通常の英語とは異なった PII としての特徴をもつかどうかはこれまで議論がなされていない．

プログラムに統合した文字列を翻訳するためには，文字列はプログラム本体から分離し，外部化したテキストリソースファイル上に置く．外部化した文字列(string)は，キー(key)とそれに対応する原語（通常は英語だから，以下では英語とする）からなる．本稿では，この文字列のことを PII 文字列とよぶ．多くの場合，このテキストリソースファイルは平テキスト形式である．PII の例は次のようである．ここで等号の左辺がキーで，右辺が PII 文字列である．

keyProfileLocation = Copy profile to

PII ファイルは，上記のようなキーとそれに対する PII 文字列の定義の繰り返しで，平テキストファイルを構成している．通常の文章であれば，文が一つの単位になっている．ところが PII 文字列は，キーに対応した文字列が一つの単位になっている．上記の例のように必ずしも文を構成しない．

そこで，我々は一つひとつの PII 文字列に関する単語数および PII 文字列(Type)の繰り返し数を調べた．その結果，PII 文字列の単語数や繰り返し出現する PII 文字列にも新たな統計的法則が存在することを発見した．なお，本稿における PII 文字列は，実際に GUI 上に出現する文字列をさすのではなく，テキストリソースファイル上の PII 文字列をさす．次章以後で，実験の概要，結果と法則，および考察を述べる．

3. 実験の概要

3.1 実験した PII

CATIA Version5 Release15(Dassault Systems 社の CAD/CAM アプリケーションソフトウェア)の PII と Microsoft 社の 122 個のアプリケーションの PII を調べた．調べた PII はいずれも原典となる英語である．前者については，第一筆者が PII とマニュアルの日本語翻訳を担当している．内容はメカニカル設計，シミュレーション，配線，配管，ヒューマンエルゴノミクスなど多岐にわたる．PII テキストファイル数は約 8500，キーの数は 17 万以上存在する．いっぽう後者は Microsoft Glossary と呼ばれ，Microsoft が Web 上に公開している PII である[7]．公開している Microsoft の PII にはキーはなく，PII 文字列のみを公開している．しかし，内容から Glossary にリストしているものは一つひとつのキーに対する文字列であることがわかる．一つの CSV(Comma Separated Value)のテキストファイルが一つのアプリケーションの PII となっている．PII の対は各国別に整理している．日英対に関しては，Microsoft Power Point, Windows XP SP2, Outlook Express など，122 個のアプリケーションの PII を公開している．本稿の実験で使用した Microsoft の PII は 2005 年 9 月に Web からダウンロードしたものである．

3.2 実験項目

上記 CATIA と Microsoft の PII に対して次のようにして 2 つのデータを取得した．

(1) アプリケーションごとに，一つひとつの PII 文字列にいくつの単語(Token)が存在するかを調べた．なお，

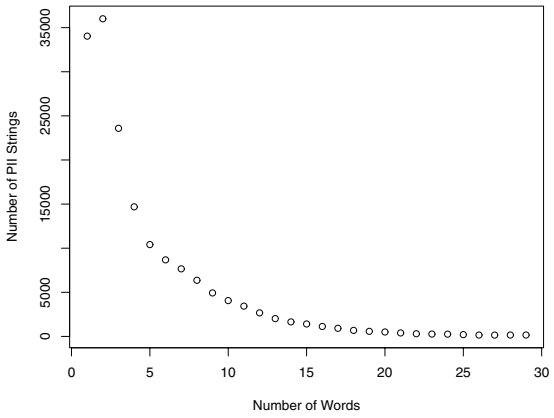

図 1　PII の単語数の分布

Microsoft の場合は 122 個のファイル全体における，PII 文字列に対する単語数も調べた．

(2) アプリケーションごとに，PII 文字列(Type)が PII ファイル内に何回出現するかを調べた．なお，CATIA の場合は，PII テキストファイルが複数あるから，PII ファイル全体で出現回数を調べた．

データを取得するためのアプリケーションプログラムの作成には Perl を用いた．単語の取得には正規表現の文字クラス「¥w」を用いている．

4. 実験結果

4.1 CATIA の PII に関する実験結果

4.1.1 PII 文字列(Token)と単語数

CATIA Version5 Release15 の英語の PII のファイル数等を表1に示す．図1に PII の単語数の分布を示す．

表 1　CATIA Version5 Release15 英語の PII

PII のファイル数	8483 個
PII ファイル全体の PII 文字列数（Token）	172058 個
PII 文字列数（Type）	94129 個
平均単語数	4.98 単語
最大単語数	961 単語
単語数順 PII 文字列の中央値の単語数	2.5 単語

図 1 は横軸(X 軸)に単語数，縦軸(Y 軸)にその単語数をもつ PII 文字列の数を示す．30 単語以上は概ね右にフラットに続くから省略している．29 単語以下で全体のキーの 99.2%をカバーしている．PII 文字列の中で最も多い単語数は 2 単語である．大半の PII 文字列は数単語で，PII 文字列全体の 7 割は 5 単語以下に集中している．

Y 軸を 10 を底とする対数目盛りに変更すると，図 2 のような回帰直線が引ける．回帰式は次のようになる．

$$\log Y = 4.52 - 0.085X \quad (Y = 10^{4.52-0.085X})$$

相関係数は 0.98(R^2)である．

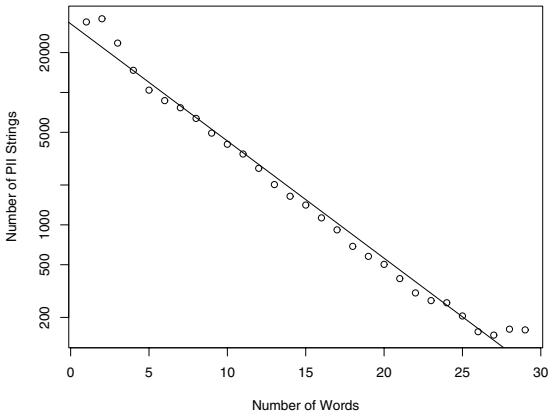

図 2　PII の単語数の分布（縦軸対数）

4.1.2 PII 文字列の繰り返し出現数

翻訳では，同じ原語であっても意味によって翻訳が変わる．そのため，PII 文字列は意味が異なれば，キーを分ける必要がある．そのため，同じ PII 文字列が繰り返し PII ファイル上に出現する．図 3 は，CATIA で同一 PII 文字列(Type)が PII テキストファイル全体で何回出現するかを示している．横軸(X 軸)に出現回数，縦軸(Y 軸)に横軸の値の出現回数を持った異なる文字列の種類の数を示す．縦軸の数に横軸の数をかけると，その出現回数をもった PII の総数になる．縦軸は対数目盛りである．たとえば文字列「Name」は 333 回出現する．すなわち「Name」という文字列を持ったキーは 333 個存在する．逆に 333 回出現する文字列は「Name」のみで(333,1)の座標位置になっている．「Align Left」という文字列は 3 回出現している．同様に 3 回出現している文字列は全体で 3882 種類あり，座標位置(3,3882)に対応している．約 17 万 PII 文字列中一つの文字列に一つのキーだけが存在するものは約 7 万 PII 文字列で，残り約 10 万 PII 文字列には複数のキーが存在し，PII ファイル上に繰り返し出現する．

図 3 の両軸を対数目盛りにし，出現回数を 49 回までに限定したグラフを図 4 に示す．出現回数 49 回までの PII 文字列は，全体の PII 文字列の 92.6%をカバーしている．図中の直線は回帰直線である．回帰式は次のようになる．

$$\log Y = 4.83 - 2.56\log X \quad (Y = 10^{4.83}/X^{2.56})$$

相関係数(R^2)は 0.96 である．

以上のように，全体の多くをカバーしかつ基本となる区間で，CATIA の PII 文字列は単語数および繰り返し出現数に関して一定の関係式が成立している．

4.2 Microsoft の PII に関する実験結果

Microsoft が Web 上で公開している 122 個の PII にはさまざまな規模の PII が存在する．PII のキーの数は 11 個のものから，WindowsXPSP2 の 167766 個のまで多様である．

4.2.1 PII 文字列(Token)と単語数

Microsoft の 122 個の PII 全体を概観するために，図 5 に，各アプリケーション(OS も含む)の PII のキーの数と PII の文

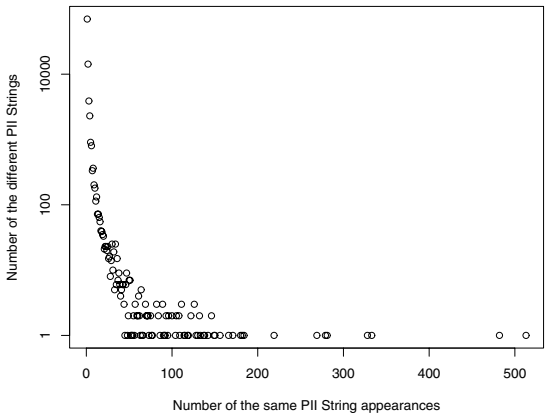

図 3　同じ PII 文字列の出現頻度分布（縦軸対数）

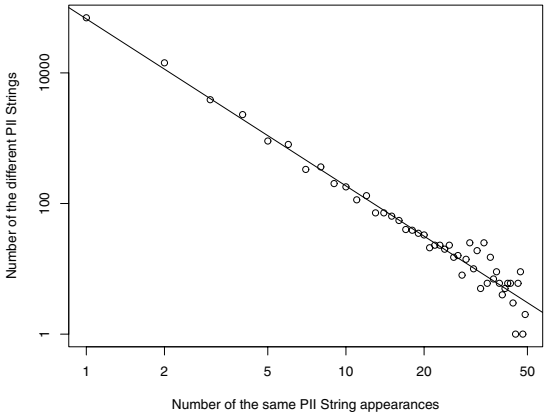

図 4　同じ PII 文字列の出現頻度分布（両軸対数）

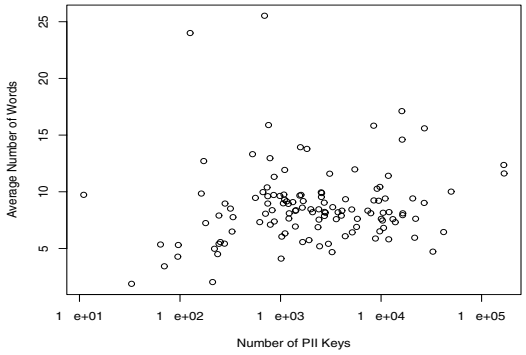

図 5　平均単語数の分布

字列の平均単語数を取ったグラフを示す．横軸は各アプリケーションの PII のキーの数である．縦軸はそれぞれの単語数である．平均単語数は，ばらついているけれども，8 単語付近のデータが多いことがわかる．

PII 文字列(Token)と単語数について，Microsoft の 122 個のアプリケーションすべてを調べた．本稿では，図 6 に 4 個のアプリケーションの結果を示す．図 6 は横軸(X 軸)に単語数，縦軸(Y 軸)に PII 文字列の数を示す．縦軸は 10 を底とする対数目盛りである．4 つのアプリケーションは PII の個数を基準に選でいる．それぞれ，約 1000 個，約 1 万個，約 5 万個，15 万個近くのものを選んだ．どのアプリケーションも 49 単語以下の部分を見ると，ほぼ直線状の分布を示す．図中の直線は回帰直線で 49 単語以下のデータのみに基づいた回帰直線である．50 単語以上は省略している．表 2 に，それぞれのアプリケーション名と PII の総数，回帰式，決定係数を示す．「All Program Mix」とあるのは Microsoft122 個全てのアプリケーションの PII 文字列のデータである．

表 2　PII の単語数の分布の回帰式

アプリケーション（Total PII）	回帰式	R^2
VirtualPCforMac（1082）	logY=1.96-0.040X	0.902
Word2003SP1 (11924)	logY=3.13-0.0604X	0.960
VisualStudio.NET (49778)	logY=3.72-0.0485X	0.980
WindowsXPSP2 (167766)	logY=4.16-0.0440X	0.985
All Programs Mix (980320)	logY=4.94-0.0442X	0.980

4.2.2 PII 文字列の繰り返し出現数

本節では，Microsoft の PII 文字列の繰り返し出現回数の結果について述べる．

図 7 は，Microsoft のそれぞれのアプリケーションで同一 PII 文字列(Type)が PII テキストファイル全体で何回出現するかを示している．横軸(X 軸)に出現回数，縦軸(Y 軸)に横軸の値の出現回数をもった異なる文字列の種類の数を示す．両軸とも対数目盛りである．縦軸の数に横軸の数をかけると，その出現回数を持った PII の総数になる．選んだアプリケーションは図 6 で取り上げたものを選んだ．回帰直線は全体の点について引いていない．プロットしている個数の半分の個数を求め，少ない出現回数の点から，求めた個数までの点で回帰直線を引いている．これは，出現回数が多い部分で文字列の種類が1で水平の直線になる部分を排除するためである．点の数が少ないときを考慮して，個数は半分にまで減らしている．表3に，それぞれのアプリケーション名と PII の総数，回帰式，決定係数を示す．他のアプリケーションもこの 4 個のアプリケーションと同様に回帰直線が引ける．正確には，PII のキーの数が少ないアプリケーションでは，PII 文字列の繰り返しが起きないものがある．そこで，繰り返し回数の回帰式は，PII のキーの数が 1000 以上のアプリケーション 84 個について調べた．84 個すべてに同様の回帰直線が引ける．以上のように Microsoft の PII 文字列も単語数および繰り返し出現数に関して一定の関係式が成立している．

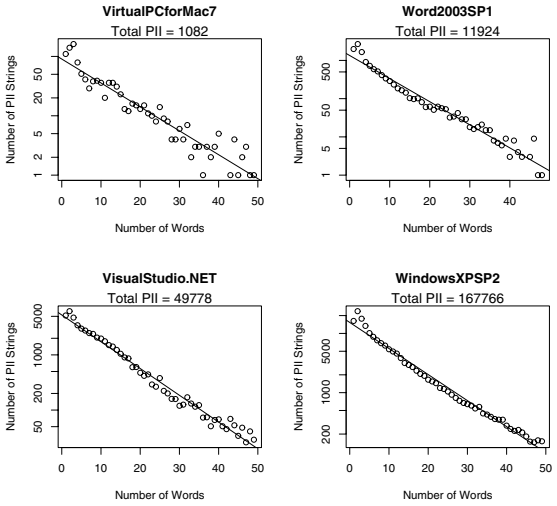

図 6　PII の単語数の分布（縦軸対数）

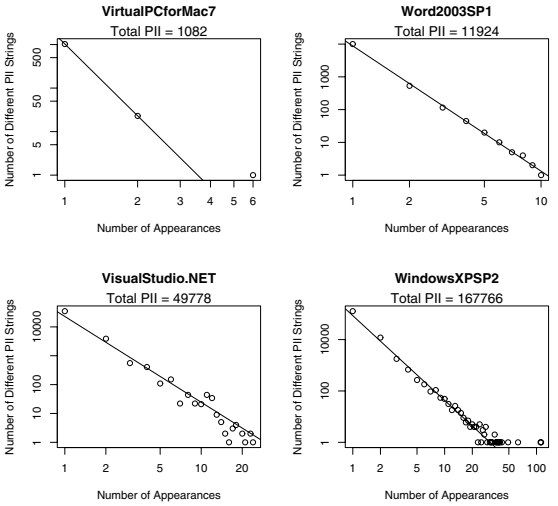

図 7　同じ PII 文字列の出現頻度分布（両軸対数）

表 3 Microsoft の繰り返し出現数の分布の回帰式

アプリケーション(PII Total)	回帰式	R^2
VirtualPCforMac (1082)	logY=3.01−5.49logX	1.00
Word2003SP1 (11924)	logY=3.94−3.82logX	0.998
VisualStudio.NET (49778)	logY=4.36−2.97logX	0.933
WindowsXPSP2 (167766)	logY=4.93−3.30logX	0.991

5. 考察

単語数のグラフ(図 2 と図 6)，および，繰り返し出現回数のグラフ(図 4 と図 7)からわかるように，CATIA と Microsoft の PII は共通の法則にしたがっている．PII 文字列と単語数の間には，「PII 文字列単語数の法則（PII の第一法則）」とよべる関係が存在している．また，PII 文字列の繰り返し出現回数とその回数を持った PII 文字列の種類の間には，「PII 文字列繰り返しの法則（PII の第二法則）」とよべる関係が存在している．「PII 文字列単語数の法則」は変数の片方を対数にすることで直線に回帰できる．一方，「PII 文字列繰り返しの法則」では変数の両方を対数にすることで，直線に回帰できる．なぜそのような違いがあるかは，今後の研究課題である．PII の第二法則はベキ法則である．

CATIA と WindowsXP(SP2)は，ほぼ同じ規模の PII をもち，分布のパターンも良く似ている．PII 文字列の単語数では PII のキーがある一定以上の数になると，CATIA や WindowsXPSP2 や Microsoft の 122 個を加えた全体に見られるようなパターンが基本のパターンになっている．たとえば 2 単語にピークがある．PII 文字列の繰り返し数でも，同様に PII の総数が多い CATIA や WindowsXPSP2 が基本パターンであると予想できる．

本稿には紹介しなかった Microsoft のアプリケーションでも，単語数に関してはすべて，繰り返し数に関しても PII の総数が小さく繰り返しが起きないもの以外はすべて法則に従っている．また，法則に従いながらも，それぞれのアプリケーションはそれぞれに個性（特徴）をもっている．たとえば，繰り返し出現回数のグラフでは，ほぼすべてが回帰線上にあるものと，さらに，異なり PII 文字列 1 の値に水平線が引けるものがある．

6. おわりに

本稿提出直前に，次のことがわかった．PII の単語数の分布は，単語数が多い部分(約 10 単語以上)において，CATIA も Microsoft も，両対数グラフ上で直線に回帰できる．すなわち，単語数の分布でもベキ法則に従っている．今後の研究課題にするとともに，稿を改め紹介したい．

謝辞

本研究では那須川哲哉氏からアドバイスをうけた．ここに深く感謝の意を表する．

参考文献

[1]Christopher D. Manning, Hinrich Schutze, “Foundations of Statistical Natural Language Processing,” The MIT Press, 1999

[2]Wentian Li, “Random texts exhibit Zipf’s-law-like word frequency distribution,” IEEE Transactions on Information Theory 38:1842-1845, 1992

[3]Wentian Li and Yaning Yang, “Zipf's Law in Importance of Genes for Cancer Classification Using Microarray Data,” Journal of Theoretical Biology, 219(4):539-551, 2002

[4]Per Bak, how nature works, Copernicus, 1996

[5]ジョン・アレン・パウロス, “天才数学者, 株にハマる”, ダイヤモンド社, 2004

[6]加藤直孝，有澤誠，“Conceptualization of Program Integrated Information”，情報処理学会研究報告，2005-NL-166，2005

[7]Microsoft Glossary ftp site:
ftp://ftp.microsoft.com/developr/msdn/newup/Glossary/, 2005