テキストに対するPurity尺度の適用と改良

谷口, 雄太
九州大学大学院システム情報科学府情報学専攻 : 博士後期課程

池田, 大輔
九州大学大学院システム情報科学研究院情報学部門 : 准教授

https://doi.org/10.15017/1467964

出版情報：九州大学大学院システム情報科学紀要. 19 (1), pp.1-6, 2014-01-24. Faculty of Information Science and Electrical Engineering, Kyushu University
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第 19 巻　第 1 号　平成 26 年 1 月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.19, No.1, January 2014

テキストに対する Purity 尺度の適用と改良

谷口　雄太* ・　池田　大輔**

Application and Improvement of the Purity Measure to Texts

Yuta TANIGUCHI* and Daisuke IKEDA**

(Received November 25, 2013)

Abstract: The purity measure is an unusualness measure for substrings of a given string. Although we have shown its usefulness on characterization of specific regions of genome sequences in the previous work, it has not been examined deeply how well the measure can be applied to text data, where much more symbols are used than in genome sequences. In this paper, we investigate its usefulness on texts and also show that the purity measure cannot differentiate the unusualness of substrings when many symbols are used in an input string. Therefore, we propose an improved measure called atomicity measure and show it can differentiate the unusualness of substrings better. Our experiment on alphabet sequences in texts shows both the measures distinguish word-like sequences and non-word sequences. Another experiment on word sequences (phrases), which is the case that there are a lot of symbols, shows the atomicity measure gives high values to phrases such as proper nouns and low values to idiomatic phrases that might reflect genres of texts while the purity measure is not so suggestive on phrases. We conclude that especially the atomicity measure can characterize texts well, and it will potentially be useful in text mining.

Keywords: Purity measure, Atomicity, Text data, Phrase, Characterization

1. 序　　論

計算機を用いてテキストデータを効率的に，かつある程度の意味情報を考慮に入れて処理する上で，単語のような単位は重要な位置を占めている．例えば Bag-of-Words などは代表的な例であり，単純でありながらも広いタスクにおいて，文書表現方法としての有効性が確認されている．

通常このような単位は，例えば西欧言語で書かれたテキストのように単語が空白により区切られている場合は比較的簡単に取り出すことができる．一方，日本語などのように単語間の区切りが明示されない場合，通常は形態素解析器による解析が用いられる．しかし，形態素解析器の多くは既知の単語の辞書や品詞などの付加的な情報に強く依存している．そのため，ブログなどに含まれる新しい言葉やネットスラングなどの未知語に対しては必ずしも上手くいかない．また，単語が空白により区切られている場合でも，単語という単位が必ずしも良い単位となっているとは限らず，複数の語からなるフレーズが適している場合も多い．

このような単位をどう取るかという問題に対しては，N-gram による文書表現[1] や文字列カーネル[5] などいくつかの手法がとられている．中でも岡野原ら[3] によって提案された文書分類手法は，極大部分文字列を単位として採用することで，実質的に全ての部分文字列網羅しつつも，現実的な時間・空間計算量動作する点で興味深い．膨大な数の部分文字列の中には，いつも必ず組み合わせて使われるフレーズなど，それらの部分を考慮する必要のないものが多く存在する．彼らの手法ではこういった冗長な文字列を同値関係を用いて同値類に分類し，最も長い極大部分文字列のみを考慮することにより，実際に考慮する部分文字列の数を入力長に比例する程度にまで押さえている．しかし，彼らの手法は文書分類タスクに限定されている．

より一般的な用途では，単純に文字列中の有用な部分が分かれば十分なことも少なくない．山田ら[6] は極大部分文字列を包含する文字列のクラスに対し，Purity 尺度と呼ばれる評価尺度を提案している．この尺度は部分文字列の特異さを図る尺度で，彼らはこの尺度を日本語のブログに適用した実験で，大学の名前やスパムを模して意図的に埋め込んだ文字列を検出することができたと報告している．また，ゲノム配列に対する実験で，配列の特定の領域に対し高い評価値を与えていると報告している．

我々はこれまで[4]，Purity 尺度のゲノム配列に対する有用性を明らかにした．様々なゲノム配列の部分文字列に対し Purity 尺度による評価を行い，どのような領域が高く評価されるかを網羅的に調べた．その結果，Purity 尺度は horizontally transferred genes と呼ばれる特定の遺伝子の配列の性質をうまく捉えており，これらの遺伝子の検出に Purity 尺度が有効であると分かった．

我々は Purity 尺度は部分文字列のある種の「まとまりの良さ」を捉えていると考えている．前述の horizontally trans-

平成 25 年 11 月 25 日受付
*情報学専攻博士後期課程
**情報学部門

ferred genes は，全く異なる種間での遺伝子交換によって現在の配列にもたらされたと考えられており，全く異なる文脈からもたらされた配列であるこれらの遺伝子は，組込まれた先の配列において，相対的に良くまとまっていると言え，Purity 尺度をこれを捉えているのだと考えることができる．他方テキストにおいては，単語やフレーズといった単位はまさに「まとまりの良い部分」だと言える．また，スパムのような文脈を無視して挿入された異質な部分も horizontally transferred genes のような存在に近く，Purity 尺度はこれらを外部知識の導入なしに特徴付けることができる可能性が高い．実際に山田らもこのような部分の検出を報告しているが，彼らの実験対象は日本語のブログテキストに限られており，また着目した部分も Purity の高いもの限定されており，十分な調査がなされていない．

そこで，本稿では Purity 尺度のテキストデータ処理における有用性についてより広く調査する．また，テキストデータにおいては Purity 尺度が正しく評価できない場合があることを指摘し，その改良を提案する．以下ではまず，第 2.1 節で Purity 尺度およびその改良について説明する．その後，第 3 節で実験について説明した後，最後に結論を述べる．

2. 手　　法

2.1 Purity 尺度

Purity 尺度は，与えられた文字列のある部分文字列に対し，その全体に対するある種の特異さを評価する指標である．この尺度の肝となっているのは「短かい部分文字列は長い部分文字列よりも多く出現する」という仮定である．

Purity 尺度の基本的なアイディアを説明するために，例として，文字列 T が与えられたときにその部分文字列 x の特異さを評価することを考える．x の任意の部分文字列 y について考えると，まず，y は x の一部であるので，y の T における出現頻度は T における x の出現頻度を超えることはない．さらに，仮定より，y は x より短かい文字列であるため，通常であれば x よりも多く出現すると考えられる．ここでもし，全ての y についてその T における出現頻度が x と同じであったとすると，x は特異であると考えることができる．なぜなら，x を構成する任意の部分は T においては x の一部としてしか出現しないからである．別の視点に立って言うならば，Purity 尺度が測るのは「部分文字列のまとまりの良さ」とか「部分文字列の分割不可能性」と表現できる．

実際には全ての y が x に固有ということは少ないので，どのくらいの数の y が先の仮定を逸脱しているのかを評価することになる．山田らは具体的な Purity の尺度の定義として確率，エントロピーおよび差にそれぞれ基づいた定義を提案している．彼らの説明によると，Purity 尺度の現在のアルゴリズムでは，エントロピーおよび差に基づいた計算には大きなメモリ空間が必要となり，数 MB 程度の大きさの入力文字列が限界である．自然言語テキストに使われる文字種は多く，そのため部分文字列の出現頻度は同じ長さのゲノム配列などに比べるととても少ない．Purity 尺度は部分文字列の出現頻度を利用しているため，テキストに対して適用する場合には比較的長い入力文字列が扱えることが好ましい．そこで本稿では，そのような制限のない確率に基づく定義のみを考える．以下 Purity 尺度という場合には確率に基づいた定義を指す．

ここで Purity 尺度の定義を与える．まず，$\mathbb{N}$ を正の整数の集合とする．また Σ を文字の有限集合とし，これをアルファベットと呼ぶ．0 個以上の文字からなる有限長の列の集合を Σ^* と表記し，これを文字列と呼ぶ．文字列 $x \in \Sigma^*$ の長さを $|x|$ と表記する．長さ $n \geq 1$ の文字列 $x = a_1 a_2 \ldots a_n \in \Sigma^*$，および $i \in \mathbb{N}$ に対し，$x[i] = a_i$ と定義する．また同様に，$i, j \in \mathbb{N}$ について，$i \leq j$ が満たされるとき $x[i:j] = a_i \ldots a_j$ と定義し，これを x の部分文字列と呼ぶ．文字列 $x \in \Sigma^*$ に対し，その部分文字列の集合を $sub(x)$ と表記し，$sub(x) = \left\{ \langle i, j \rangle \in \mathbb{N}^2 \mid 1 \leq i \leq j \leq |x| \right\}$ と定義する．また，2 つの文字列 $T, x \in \Sigma^*$ に対し，x の T における出現位置の集合 $pos_T(x)$ を $pos_T(x) = \{\langle i, j \rangle \in sub(T) \mid T[i:j] = x\}$ と定義し，x の T における出現回数 $freq_T(x)$ を $freq_T(x) = |pos_T(x)|$ と定義する．このとき Purity を次のように定義する．

Definition 1　入力文字列 T，およびその部分文字列 $x = T[i:j]$ が与えられたとき，T 上での x の Purity 値を以下のように定義する:

$$purity_T(x) = \frac{|\{\langle k, l \rangle \in sub(x) \mid freq_T(x[k:l]) = freq_T(x)\}|}{|sub(x)|}.$$

直感的には Purity 尺度は，入力された部分文字列 x の部分文字列 y の内，x と同じ回数出現する y の割合を計算することで，先に述べた「仮定」から x がどれだけ逸脱しているのかを量化していると言える．Suffix Tree や Suffix Array[2] などのデータ構造を用いることで，山田らが “branching string” と呼ぶ特定のクラスの部分文字列 (極大部分文字列のクラスを含む) については，線形時間で計算ができる．

2.2 Atomicity 尺度

通常の英語テキストでは 70 種類前後の文字が使われ，また単語列や日本語などを考慮するとゲノム配列などに比べ圧倒的に文字種が多いと言える．文字種が多くなると部分文字列の出現頻度は全体的に少なくなってしまう．Purity は出現頻度が同一の部分文字列の数を数えているため，出現頻度の多様性が減ることで，Purity 値が特定の値に集中してしまう問題がおきる．例えば，**表 1** は，単語列に対して Purity を適用した結果で，Purity 値順に上位 10 件を抜き出したものである．単語列の場合，文字の種類数は単語の異なり数でありかなり大きくなる．表に示した 10 件および以降の 126 件には全て同じ Purity 値が与えられており，これ

Table 1 This shows the result of the application of purity measure to word sequences (phrases). The purity measure was applied to phrases included in the "News" category of the Brown corpus dataset. Only top 10 phrases with high purity values are shown.

News	
s-purity	substring
0.666667	outcom in
0.666667	tatter remain
0.666667	essex counti
0.666667	36 year
0.666667	portion of
0.666667	lafayett squar
0.666667	rescind the
0.666667	greec and
0.666667	deadlin for
0.666667	didn t

らの単語列を互いに区別することができていない．

このような現象が起きるのは，頻度が同一のものを数えているためである．例えば入力文字列のある部分文字列 x を評価することを考える．このとき，x の部分文字列 y の総出現回数 10 回の内 9 回は x の一部として出現したとし，x の部分文字列 z の総出現回数 5 回の内 1 回のみは x の一部として出現したとする．この場合，前者 y の方がより x と強く関連していると考えられるが，Purity 尺度ではどちらの場合も Purity 値へ貢献は 0 である．

この問題を解決するために Atomicity 尺度を提案する．アイディアとしては，先の例で言うところの 8/10,1/5 といった頻度の比を関連度と見なし，評価値の計算に関連度を考慮することで，まとまりの違いをより細かいレベルで表現するというものである．具体的には次のように定義する．

Definition 2 入力文字列 T, およびその部分文字列 $x = T[i : j]$ が与えられたとき，T 上での x の Atomicity 値を以下のように定義する:

$$atomicity_T(x) = \left(\sum_{\langle k,l \rangle \in sub(x)} \frac{freq_T(x[k:l])}{freq_T(x)} \right) / |sub(x)| .$$

Atomicity 尺度を用いることで，単語列や文字種の多い言語で書かれたテキストをより適切に扱えると期待できる．

3. 実　　験

本節では 2 つのデータセットに対しそれぞれ異なる方法で Purity 尺度および Atomicity 尺度を適用し，これらの尺度のテキストデータに対する有効性を吟味する．一つは 20 Newsgroups と呼ばれるデータセットで，テキスト中のアルファベットが連続する部分 (トークン) に対し Purity 尺度を適用する．もう一つは Brown コーパスと呼ばれるデータセットで，テキスト中の単語列 (フレーズ) に対し，Purity 尺度を適用する．

3.1 20 Newsgroups

3.1.1 データセット

20 Newsgroups はニュースグループの投稿を収集したデータセットで，20 の異なるトピックからなり，それぞれ 1000 の投稿を含んでいる．ニュースグループの投稿は，ニュース記事などよりも比較的くだけた表現が多く，またプログラミングのコード片など多様なテキストが混在しており，未知語を多く含むテキストと言える．

本データセットを用いた実験の目的は，データに含まれる多様なテキスト表現に対し Purity 尺度・Atomicity 尺度がどのような値を付与するのかを調べることである．データセットに含まれる全投稿を 1 つにまとめたテキストを入力文字列とし，今回は特にアルファベットからなる部分文字列 (トークン) を尺度の評価対象とする．

まず 20 Newsgroups データセットに対し，各投稿のヘッダ部分を削除し，全ての投稿を 1 つに連結する．次により連結されたテキストからトークンを抽出する．ここでは連続する 2 文字以上のアルファベットの列をトークンとして取り出した．

3.1.2 結　果

表 2 に切り出されたトークンを Purity 尺度または Atomicity 尺度により評価した結果の一部を示す．与えられた評価値の高い順に上位 15 件，下位 15 件を示している．長いトークンは末尾を一部省略して表示している．

まず切り出されたトークンについて見てみると，'aammmaaaazzzzzziinnnnggggg" や "hahahahaha" などに代表されるように，くだけだ表現が多いことが分かる．また，コンピュータ系のトピックからは，ファイル名やソースコード中の変数名，エンコードされた添付ファイル由来の文字列など，通常の単語とは性質の異なるトークンが多く取り出されている．

次に Purity 値・Atomicity 値と合わせて，まず上位のトークンについて観察する．両尺度ともに，比較的長く一見単語のようではない比較的長い文字列に大きな値を与えている．実際には "wholesomegodfearingbiblebelievingtraditionalfamilyvalues" のように，投稿者が単語を組み合わせて作った造語なども含まれている．他方，下位のトークンについて見てみると，Purity 尺度の場合では，反復的な文字列が比較的最下位に集中しているが，Atomicity 尺度の場合ではもう広範囲に分散している．表中には示されていないより上位の部分では，両尺度ともに単語と思われる文字列が分布している．

まとめると，大まかな傾向として，Purity 値や Atomicity 値が大きい部分文字列は通常の単語とは大きく異なる文字列であり，逆に値が小さい部分文字列は単語のような文字列であると言える．これは両尺度が評価対象の部分文字列の組成，つまりそれを構成する，より小さい部分文字列の頻度を考慮しているためと考えられる．頻繁に出現する文字

Table 2 This table shows tokens and their evaluation values; purity values (left-hand side) and atomicity values (right-hand side). Fifteen tokens are shown for highest values and lowest values, respectively. For very long tokens, only their prefixes are shown.

Purity	Alphabet Sequence	Atomicity	Alphabet Sequence
0.881	brownbladerunnersugarcubeselectroni…	0.892	brownbladerunnersugarcubeselectronicblayloc…
0.844	plutoniumsurveillanceterroristciaas…	0.851	costellobeatlesspinaltapfawltytowersmuttsav…
0.842	costellobeatlesspinaltapfawltytower…	0.846	plutoniumsurveillanceterroristciaassassinationira…
0.827	prxnpueuszqeiiusmcvcrcgnwbavrxfja	0.836	prxnpueuszqeiiusmcvcrcgnwbavrxfja
0.821	wholesomegodfearingbiblebelievingtr…	0.830	pnaqevxgqaoxrviaggvpvrdlwzchbnqo
0.818	pnaqevxgqaoxrviaggvpvrdlwzchbnqo	0.827	evyynlzbboryvhfszyyhyheqqqilhek
0.817	evyynlzbboryvhfszyyhyheqqqilhek	0.825	wholesomegodfearingbiblebelievingtraditiona…
0.805	gasbpxcdhsrhpmebpjklyikuijzat	0.813	gasbpxcdhsrhpmebpjklyikuijzat
0.791	hrrrtnaiwyjmfaqxpeyrodvfdxc	0.806	abcdefghijklmnopqrstuvwxyz
0.779	iainmbanksneworderheathersbatmanpjorourke	0.805	iainmbanksneworderheathersbatmanpjorourke
0.777	gestaltpowermanagerattributesfoodspreadproduct	0.803	vxxwtpaqebkgqasgoxctzjdzmzurfm
0.776	vxxwtpaqebkgqasgoxctzjdzmzurfm	0.799	hrrrtnaiwyjmfaqxpeyrodvfdxc
0.767	moorcockpratchettdenislearydelasoulu	0.789	mqcnaitfksqaaaeeakceejwi
0.764	eocclpkstavebtdcligqhnzowc	0.788	moorcockpratchettdenislearydelasoulu
0.763	tyyobpbtlqgsurgkgdzpxwfh	0.785	gestaltpowermanagerattributesfoodspreadproduct
0.008	ttttttttttttt	0.027	compressions
0.008	hahahahahahahahahahaha	0.027	nominates
0.007	halesshavethewhalesshavethewhalesshavethewhal…	0.027	hmmmmmmmmmmmmmmmmmmmmmmmmmmm…
0.007	vethewhalesshavethewhalesshavethewhalesshavet…	0.027	regenerates
0.007	xxxxxxxxxxxxxxxx	0.026	inversions
0.006	hmmmmmmmmmmmmmmmmmmmmm…	0.026	impresses
0.006	immmmmmmmmmmmmmmmmmmmm…	0.026	deflections
0.005	jjjjjjjjjjjjjjj	0.025	exterminations
0.004	vvvvvvvvvvvvvvvvvvvv	0.025	rationals
0.003	ooooooooooooooooooooooo	0.023	emulations
0.002	esshavethewhalesshavethewhalesshave…	0.023	constantinov
0.002	shavethewhalesshavethewhalesshaveth…	0.022	separations
0.002	wmwmwmwmwmwmwmwmwmwmwmw…	0.021	groundings
0.001	vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv…	0.018	vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv…
< 0.001	vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv…	0.018	oppositions

の組み合わせは言語などに依存していると考えられ，そのような文字の組からなっている部分文字列ほど小さな評価値が与えられる．逆に，滅多に出現しない文字の組み合わせを用いている部分文字列，例えばエンコードされたデータを表現する文字列や外来語，合成された語が大きな値を与えらている．

以上の観察を定量的に確認する．具体的には，英語の単語辞書を用いて評価対象のトークンを「単語」や「単語に類似する文字列」や「それ以外」といったクラスに分類し，トークンの Purity 値・Atomicity 値の分布をクラス毎に分析する．ベースとする単語集合にはスペルチェッカ MySpell[1] の英語辞書に含まれるアルファベットのみからなる単語，および英語の Wikitionary[2] に登録されている単語を併せて用いた．以下これらの単語の集合を単に辞書と呼ぶ．

トークンの分類は以下の手順で行なった．

(1) 全トークンの内，辞書に含まれるものを全て「word1」に分類する
(2) 1 で分類されなかったトークンの内，語幹が辞書に含まれるいずれかの単語の語幹と同一であるものを「word2」に分類する
(3) 2 で分類されなかったトークンの内，辞書に含まれるいずれかの単語との編集距離が 3 以下であるものを「word3」に分類する
(4) 3 で分類されなかったトークンの内，辞書に含まれる 3 文字以上の単語を連接してできるものを「word4」に分類する
(5) 4 で分類されなかったトークンを全て「others」に分類する

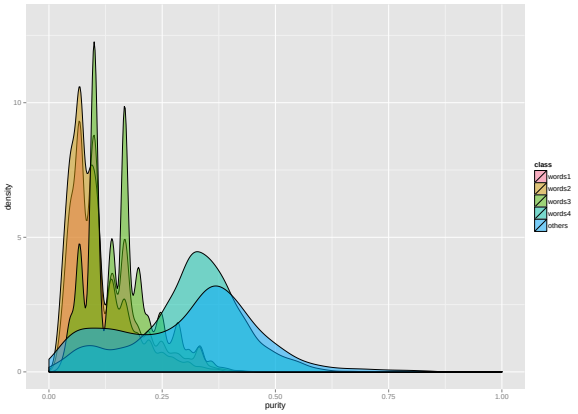

Fig. 1 Distribution of the purity values of tokens for each substring class.

ここで語幹は Snowball[3] ライブラリのアルゴリズムを用いて計算した．ここで，word1，word2 および word3 のクラスは単語もしくは単語に類似した文字列のクラスであり，word4 と others は合成された語および単語ではない文字列のクラスである．

図 1，**2** はそれぞれ，トークンの Purity 値および Atomicity 値の密度分布を，クラス毎に示したものである．密度分布はカーネル密度推定により計算した．前述したように，Purity

[1] http://code.google.com/a/apache-extras.org/p/ooo-myspell/
[2] http://en.wiktionary.org/wiki/Wiktionary:Main_Page
[3] http://snowball.tartarus.org/

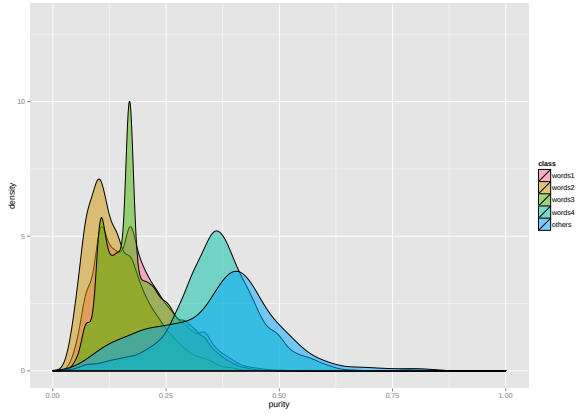

Fig. 2 Distribution of the atomicity values of tokens for each substring class.

値は特定の値に偏る性質があるため，Atomicity 値の分布ほどは滑らかではない．このような違いはあるものの，両者は大まかには同様の傾向をもっていると言える．単語に最も類似する文字列のクラス word1，word2，word3 では Purity 値・Atomicity 値はともに低い値に偏っている．また，残りの 2 つのクラスについては，低い値を与えられたトークンもあるにはあるが，全体としては比較的高めの値に偏っていて，先の観察を裏付けていると言える．このような Purity 値や Atomicity 値の偏りは，例えば機械学習などのタスクにおいて不要な語を辞書に依らず削除するのに応用できると考えられる．

3.2 Brown コーパス

3.2.1 データセット

Brown コーパスは英語のコーパスで，15 のジャンルごとにそれぞれ，様々なソースから収集されたテキストを含んでいる．ここでは本データセットを特に単語列として扱い，単語列の部分 (フレーズ) に対する Purity の評価を調べる．これはゲノム配列や通常の英文テキストと比べ圧倒的に文字数が多い場合に相当し，そのような場合における両尺度の振る舞いを調査する．Brown コーパスをデータセットとして採用するのは，文章が 20 Newsgroups とは対照的に比較的整っているため，単語の不要な多様性を排除できるためである．ここでは，単語列の部分の内，岡野原らが用いている極大部分文字列 (単語列) を解析の対象として設定する．これにより，切れ目が明示されていないフレーズに対し，頻度的に見て妥当なフレーズの境界を全て考慮することができる．

テキストは 15 のジャンル毎に 1 つに連結し，結果として 15 の入力テキストを用意し，それぞれ個別に扱った．

3.2.2 結　果

表 3 は，Brown コーパスの 15 のジャンルの内 4 ジャンルに対する結果を示したものである．Atomicity 値順で上位 10 件および下位 10 件のフレーズを示した．**表 1** が示すように Purity 尺度は多数のフレーズに同一の値を与えており，その値によってフレーズの性質が表現されていないと考え省略した．また，先の 20 Newsgroups の場合と比べてフレーズの長さ (ここではフレーズを構成する単語数) が圧倒的に短いのは，これは対象とするフレーズを出現回数が 2 回以上のものに限定しているためである．

まず，大きな Atomicity が付与されたフレーズを見てみると，どのジャンルにおいても固有名詞が多く上位を占めていることが分かる．これは文字通り，「固有」名詞がそれ特有の単語からなっているためである．他方，下位のフレーズは一般的によく使われる言い回しが多数である．これは 20 Newsgroups の場合と同様に，テキストを記述する言語で良く使われる語の並び (慣用句) がフレーズに多く含まれているからである．加えて，単語列の場合は書き手やジャンルに特有の言い回しが多く含まれていると言える．例えば “Government” や “Review” の場合が顕著である．また，“Romance” などの小説や “News” では直接話法の表現を見ることができる．

このように，Atomicity 尺度はテキストのジャンルやスタイルといった特徴を捉えていると言うことができる．そのため，文書分類や著者推定といったタスクに応用できる可能があると考えられる．

4. 結　　論

Purity 尺度をテキストデータに対し適用し，その有用性を示すとともに改良を提案した．具体的には 20 Newsgroups を用いた実験から，アルファベット列に対し Purity を評価してやることで，単語とそうでないものとに分離できることが分かった．これにより，辞書に頼らない不要語削除が期待できる．また，Brown コーパスを用いた実験では，極大部分単語列に対し Purity を評価した．Purity 値が高く評価されたものには固有名詞などが多く，逆に低く評価されたものの多くは一般的な言い回しであった．Purity 値の高い部分単語列をとり出すことで，複数の単語からなる固有名詞の抽出が期待でき，また低い部分単語列を取り出すことで，トピックや作者に応じた言い回しを考慮する文書分類に応用できると期待される．以上の結果より，Purity および Atomicity 尺度は，テキストデータ特徴付けに有用な性質を持つと結論する．

参　考　文　献

1) William B. Cavnar. Using An N-Gram-Based Document Representation With A Vector Processing Retrieval Model. In Text REtrieval Conference, 1994.
2) Dan Gusfield. Algorithms on Strings, Trees and Sequence. Cambridge University Press, New York, 1997.
3) Daisuke Okanohara and Jun’ichi Tsujii. Text categorization with all substring features. In SDM, pages 838–846, 2009.

Table 3 This table shows phrases and their atomicity values. Only the results for four genres out of fifteen genres are shown. Ten phrases are shown for highest values and lowest values, respectively.

News		**Reviews**		**Government**		**Romance**	
Atomicity	Phrase	Atomicity	Phrase	Atomicity	Phrase	Atomicity	Phrase
1.000000	puerto rico	1.000000	catfish bend	1.000000	los angel	1.000000	hong kong
1.000000	dolc vita	1.000000	wharf rat	1.000000	puerto rico	0.958333	gratt shafer
1.000000	corpus christi	1.000000	18e siecl	1.000000	du pont	0.925926	o clock
1.000000	sterl township	1.000000	sancho panza	1.000000	amici curia	0.895833	evadna mae evan
1.000000	pinar del rio	1.000000	olga moiseyeva	1.000000	conscienti objector	0.888889	walt perri
1.000000	hardwick etter	1.000000	zealous volunt	1.000000	nonresid alien	0.888889	signor raymond
1.000000	duncan phyfe	1.000000	pee wee	0.973985	region offic in atlanta ga boston ...	0.888889	v shape inlet
1.000000	scottish rite	1.000000	tea tray	0.954545	rhode island	0.863636	wet graham cracker
1.000000	notr dame	1.000000	andrea palladio	0.944444	lantern slide	0.857143	mousi chandler
1.000000	hong kong	0.944444	san francisco	0.942529	sam rayburn	0.857143	san diego
0.176417	one of a	0.192778	one of it	0.160240	in the unit state or	0.166052	he said " i
0.174724	in the presid	0.192667	it is the	0.159587	of the unit nation	0.162456	which he had been
0.174002	to the first	0.189765	in the program	0.154709	part of the unit state	0.160008	and it was not
0.173955	it will be the	0.188133	at the first	0.154322	depart of the state	0.159077	" i ll be
0.170289	to the u n	0.187400	of the "	0.146225	of the unit state or	0.157727	t have to be
0.164508	presid of the univers	0.187307	with the music	0.139310	part of the nation	0.155281	said " i m
0.163976	he said " we	0.184279	and the music	0.133474	the govern of the unit state of america in	0.147193	said " i ll
0.162041	chairman of the republican	0.182640	in the music	0.126949	of the govern of india	0.137401	he said " i m
0.155763	he said " i	0.182511	is one of the most	0.121070	of the unit state and	0.135393	and it was a
0.140840	one of the first	0.165639	one of the great	0.116950	in the unit state and	0.132477	i said " i

4) Yuta Taniguchi, Yasuhiro Yamada, Osamu Maruyama, Satoru Kuhara, and Daisuke Ikeda. The purity measure for genomic regions leads to horizontally transferred genes. Journal of Bioinformatics and Computational Biology, 11(06):1343002, 2013.
5) Choon Hui Teo and SVN Vishwanathan. Fast and space efficient string kernels using suffix arrays. In Proceedings of the 23rd international conference on Machine learning, pages 929–936. ACM, 2006.
6) Yasuhiro Yamada, Tetsuya Nakatoh, Kensuke Baba, and Daisuke Ikeda. Mining pure patterns in texts. In IIAI International Conference on Advanced Applied Informatics, pages 285–290, 9 2012.