低頻度byte列を活用した言語識別

行野顕正[†] 田中省作[††]
冨浦洋一[†††] 松本英樹[†]

本稿では，言語識別のための言語特徴として，低頻度 byte 列を活用することを提案する．一般的な言語識別では，byte 列の出現傾向の類似度を各言語・識別対象文書間で求め，その大小で言語を識別する．従来手法は，出現確率の推定値の信頼性が高い高頻度 byte 列を言語特徴に利用し，信頼性の低い低頻度 byte 列を活用してこなかった．しかし，低頻度であっても，長い byte 列は特定の言語のみに出現する傾向が強く，言語の特定に大きく貢献すると期待できる．低頻度 byte 列を利用するためには，低頻度 byte 列でも十分に識別に影響を与えることができ，かつ，出現確率推定値の変動に頑強な類似尺度が必要となる．本稿で提案する類似尺度は byte 列の積集合サイズに基づいており，これらの条件を満たしている．本稿では，提案手法の有効性を示すために，2 種類の言語集合に対して識別実験を行っている．提案手法は従来手法に比べて高い識別精度を示しており，特に，類似言語間や小規模文書の識別において顕著に優位性が認められる．

Language Identification Using Low-frequent Byte-strings

Kensei Yukino,[†] Shosaku Tanaka,[††] Yoich Tomiura[†††]
and Hideki Matsumoto[†]

This paper proposes a language identification method which uses low-frequent byte-strings as language features. A general method identifies the language of a document by choosing the language which has the most similar probability distribution of byte-strings to that of the document. Most previous methods, whose similarity measures are based on frequencies of byte-strings, never use the low-frequent byte-strings because of the fluctuation of their frequencies. However, among low-frequent byte-strings, there are a lot of effective byte-strings in language identification, which tend to appear in a particular language. The similarity measure using not only frequent byte-strings but also low-frequent ones should be robust to the fluctuation of the estimated probability and be sufficiently influenced by the low-frequent byte-strings. The similarity measure used in the proposed method is based on an intersection size of byte-strings between each language and a target document. Here are two examinations: the one is for similar languages and another is for dissimilar languages. They show that the proposed method has higher accuracy than the previous methods and has advantage in the language identification among similar languages or for short target documents.

1. はじめに

近年，言語横断検索システムなど，多言語を同時に扱うツールが開発されつつある．これらのツールは，処理対象文書の言語が既知であることを前提としている[1),2)]．しかし，現実の電子化文書では，言語が明示されていることは少ない．たとえば，インターネット上では，ほとんどの文書で言語が明示されていない．したがって，多言語処理ツールは前処理として自動言語識別を必要とする．

90 年代以降，様々な言語識別手法が提案されてきた．従来手法の多くは，識別対象文書に出現した byte 列を文書の言語特徴として扱う．そのうえで，文書での byte 列の出現傾向と各言語の学習データでの byte 列の出現傾向を比較し，最も類似した傾向を持つ言語を文書の言語として出力する．

言語特徴には，主に高頻度 byte 列が用いられてきた．高頻度 byte 列には出現確率の推定値が比較的信頼できるという利点がある．従来手法はその利点を活かし，出現確率分布の類似性に基づいた類似尺度を用いている．逆に，出現確率推定値の信頼性に乏しい低

† 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 立命館大学文学部
College of Letters, Ritsumeikan University
††† 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University

頻度 byte 列は識別に用いられてこなかった．

本稿では，低頻度 byte 列を言語特徴として活用する手法を提案する．低頻度 byte 列には長い byte 列が多い．長い byte 列は，短い byte 列に比べて特定の言語のみに出現する傾向が強い．そのため，低頻度 byte 列も言語の特定に大きく貢献すると期待できる．しかし，従来手法の類似尺度は出現確率の大きさや出現確率推定値の信頼性に強く影響を受けるため，低頻度 byte 列を利用しにくい．低頻度 byte 列を利用するためには，低頻度 byte 列でも十分に識別に影響を与えることができ，かつ，出現確率推定値の変動☆に頑強な類似尺度が必要となる．

提案手法では byte 列の積集合サイズに基づいて言語・文書間の類似度を計算する．この類似尺度は出現頻度の大小を考慮しないため，低頻度 byte 列でも十分に識別に影響を与えることができ，かつ，識別対象文書や学習データにおける出現確率推定値の変動の影響も受けにくい．

以下では，まず 2 章において従来手法について概観し，その問題点を明らかにする．3 章では，低頻度な byte 列を言語識別に活用する手法を提案する．4 章では提案手法の有効性を確認するために行った実験について述べる．

2. 従 来 手 法

2.1 言語識別の概要

主な言語識別手法は電子化文書を byte の列ととらえ，その部分 byte 列を文書の言語特徴として扱う．本稿では，長さ k の部分 byte 列を k-byte 列と呼ぶことにする．**図 1** に，ある文書から byte 列を抽出する過程を図示する．

byte 列を言語特徴とする従来手法は，識別対象文書中における byte 列の出現傾向と各言語の学習データにおける byte 列の出現傾向を比較し，最も類似した傾向を持つ言語を識別対象文書の言語として出力する．類似尺度（ないし，非類似尺度）としては，byte 列の出現確率（出現頻度）分布の類似性に基づいた類似尺度を用いている．

byte 列を言語特徴として用いる手法には，次の 3 つの利点がある．

(1) 言語特徴を小規模学習データから抽出できる．
(2) 学習時に特別な言語知識を必要としない．
(3) 言語識別と同時に符号系の識別も行える．

☆ 同一の言語における，同一の byte 列の出現確率推定値であっても，識別対象文書や学習データごとに異なる値となることを意味する．本稿では出現確率推定値の変動と略記する．

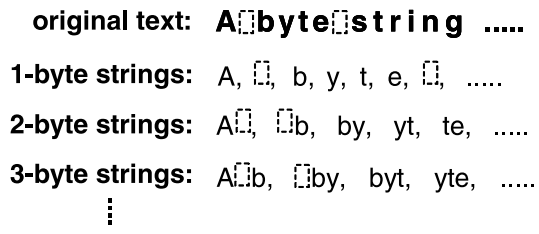


図 1 byte 列の計数
Fig. 1 Counting byte-strings.

従来手法の中には，単語を言語特徴とする手法もある[3),4)]．しかし，次の 3 つの欠点を持つため，本稿では検討しない．

(1) 高精度化には巨大な単語辞書を必要とする．
(2) 対象言語に対する言語知識が必要となる．
(3) 言語識別の前に，符号系の識別を必要とする．

2.2 byte 列を用いる従来手法の概要

byte 列を用いる従来手法は，主に高頻度 byte 列を言語特徴に用いている．高頻度 byte 列には，学習データや識別対象文書において比較的信頼性の高い出現確率の推定値を得られるという利点がある．従来手法はその利点を活かし，出現確率分布の類似性に基づいた類似尺度を用いている．

高頻度 byte 列の大半は，短い byte 列である．言語特徴が短い byte 列に偏る傾向は，従来手法に一貫してみられる．多くの手法では 1〜3-byte 列が用いられており，1〜5-byte 列と比較的長い byte 列まで使用している手法でも，実際には高頻度 byte 列に制限するため，4〜5-byte 列はほとんど使われていない．

以下に，主要な手法について概略を示す．各手法の相違点を明確にするため，いくつかの手法の識別式には等価な変換を施してある．

Cavnar らの手法[5)] では，1〜5-byte 列の出現頻度順位表の類似性に基づいて言語を識別する．識別の手順は以下のとおり．

(1) 各言語 l，各 1〜5-byte 列 x に対し，言語 l の学習データにおける出現頻度 $f_l(x)$ を計数する．byte 列を出現頻度の降順に並べ，上位 n 個の順位表 Prof(l) を構築しておく．
(2) (1) と同様，識別対象文書 d における x の出現頻度 $f_d(x)$ を計数し，Prof(d) を構築する．
(3) 次の式により，言語を識別する．

$$\hat{L} = \arg\max_{l} \sum_{x \in \mathrm{Prof}(d)} S_C(x, l),$$

$$S_C(x,l)=\begin{cases}-|\mathrm{rank}_d(x)-\mathrm{rank}_l(x)| \\ \qquad ;x\in \mathrm{Prof}(l) \\ -(n+1) \quad ;x\notin \mathrm{Prof}(l)\end{cases}.$$

ここで，$\mathrm{rank}_d(x)$ は，x の $\mathrm{Prof}(d)$ における順位を，$\mathrm{rank}_l(x)$ は，x の $\mathrm{Prof}(l)$ における順位を表す．文献 5) では言語特徴に用いる byte 列を順位が安定する byte 列に制限しており，使用する下限順位（n）を 300〜400 位とした場合が最も性能が良いと報告されている．文献 5) と同じ言語特徴を使用する手法として，Martins らの手法[6] がある．文献 6) では，類似度尺度の計算方法のみを変更し，情報理論に基づいた類似度の使用を提案している．

Dunning の手法[7] では，各言語の k-byte 列（k は一定長に固定）の出現頻度に基づいて識別対象文書の発生確率を計算し，その大小で言語を識別する☆．識別の手順は以下のとおり．

(1) 各言語 l，各 k-byte 列 x に対して言語 l の学習データにおける x の出現頻度 $f_l(x)$ を計数し，ラプラス法を用いて x の出現確率 $p_l(x)$ を求めておく．

(2) (1) と同様，識別対象文書 d における x の出現頻度 $f_d(x)$ を計数する．d 中に出現した全 byte 列の集合を $Set(d)$ とする．

(3) 次の式により，言語を識別する．

$$\hat{L}=\arg\max_l \sum_{x\in Set(d)} S_D(x,l),$$
$$S_D(x,l)=f_d(x)\log p_l(x).$$

文献 7) では，実験で用いた $k=1$〜5 のうち，$k=2$ のときが最も性能が良かったと報告されている．

Sibun らの手法[8] では，k-byte 列（k は一定長に固定）の出現確率分布間の KL-Divergence に基づいて識別を行う．この手法は，本質的に文献 7) の手法と等価であり，$p_l(x)$ の補間法のみ加算法（補正項は +0.5）に変更されている．文献 8) でも，実験で用いた $k=1$，2 のうち，$k=2$ の方が性能が良かったと報告されている．

北らの手法[9] では，k-byte 列（k は一定長に固定）の出現確率分布間の KL-Divergence を言語 → 文書，文書 → 言語の双方向で求め，その平均に基づいて識別を行う☆☆．識別の手順は以下のとおり．

(1) 文献 7) と同様の手法により，$f_l(x)$ を計数し，線形補間法を用いて $p_l(x)$ を求めておく．

(2) (1) と同様，識別対象文書 d における x の出現確率 $p_d(x)$ を求めておく．

(3) 次の式により，言語を識別する．

$$\hat{L}=\arg\max_l \sum_x S_K(x,l),$$
$$S_K(x,l)=(p_d(x)-p_l(x))\log\frac{p_l(x)}{p_d(x)}.$$

文献 9) では，$k=3$ を用いているが，その根拠は示されていない．

前田らの手法[10] では，k-byte 列（k は一定長に固定）のコサイン類似度に基づいて識別を行う．識別の手順は以下のとおり．

(1) 文献 7) などと同様の手法により，言語 l における x の出現頻度 $f_l(x)$ を求めておく．ただし，ASCII 文字集合に含まれる byte のうち記号・数字・制御文字など，言語識別に有用でない byte は前処理として文書中から取り除いておく．

(2) (1) と同様の手法により，識別対象文書 d における x の出現頻度 $f_d(x)$ を求める．

(3) 次の式により，言語を識別する．

$$\hat{L}=\arg\max_l \sum_x S_M(x,l),$$
$$S_M(x,l)=\frac{f_l(x)}{\sqrt{\sum_x f_l(x)^2}}\frac{f_d(x)}{\sqrt{\sum_x f_d(x)^2}}.$$

文献 10) では，主に主記憶容量の制約から，$k=2$ に制限されている．

2.3 従来手法の問題点

従来手法は，おおむね良好な識別精度を発揮しているものの，類似言語間における識別や小規模文書に対する識別において精度の低下が見られる．表記が類似している言語間では，高頻度 byte 列の出現確率分布の言語間差異も小さい．そのため，各言語と文書との類似度はどの言語でも似た値となり，従来手法では識別が難しくなる．小規模文書では，高頻度 byte 列であっても出現確率推定値の信頼性が下がる．そのため，出現頻度に強く影響される従来手法の類似度では正確な識別ができなくなる．

3 章で述べるように，低頻度 byte 列を活用することで，これらの状況においても高い識別精度が期待できる．しかし，従来手法において低頻度 byte 列を活

☆ 文献 7) では本来マルコフモデルが使われており，x の最後の 1 文字を削った byte 列を $x_{(-1)}$ として，$p_l(x)$ は $p_l(x)/p_l(x_{(-1)})$ と表記するのが正しい．ここでは表記の簡略化のため，$p_l(x)$ として表記する．

☆☆ 文献 9) では，文献 7) 同様，マルコフモデルが使われている．前述した理由により，$p_l(x)$，$p_d(x)$ で略記する．

用することは難しい．たとえば，文献 5) の出現頻度順位に基づいた類似度では，低頻度 byte 列は 1 の頻度差で数千〜数万の順位変動が発生するため，識別に活用できない．確率や KL-Divergence を用いている文献 7)〜9) の手法では，低頻度 byte 列は $p_l(x)$ の推定精度が悪く，補正の影響を強く受けてしまうため，識別に使いにくい．文献 10) が用いているコサイン類似度では，頻度がそのまま各 byte 列の重要度を表しており，低頻度 byte 列の重要度は低いため識別への影響力が小さい．

3. 提案手法

3.1 低頻度 byte 列の活用

本稿では，高頻度 byte 列だけでなく，低頻度 byte 列も言語特徴として利用することを提案する．低頻度 byte 列の中には相対的に長い byte 列が多い．長い byte 列は，短い byte 列よりも特定の言語のみに出現する傾向が強く，言語の特定に大きく貢献すると期待できる．**表 1** に，英語で一定割合の文書に出現する長さ k の byte 列が，それぞれ何カ国語に出現するかを一覧にして示す．表中の DF は，1,000 byte からなる 100 個の英語文書中で，何文書に出現した byte 列かを表す．k は，byte 列の長さを表す．各交点は，一定割合の英語文書に出現している k-byte 列が，欧州 10 カ国語☆の中で（英語を含めて）平均何カ国語で出現したかを表している．各言語とも，英語と同様，1,000 byte の文書 100 個内での出現有無を調べている．また，DF を求めた英語文書と，出現言語数を調べるときの英語文書は同一のものであるため，各交点の最小値は 1 である．表より，同一の DF の byte 列では，長い byte 列ほど特定の言語のみに出現する傾向が強いことが読み取れる．このような長い byte 列を活用することで，従来手法よりも多くの言語間差異をとらえることが可能になり，類似言語間や小規模文書に対しても高い識別精度が期待できる．

言語特徴に導入する byte 列として，本稿では 1〜5-byte 列を混在して用いる．多くの従来手法[7)〜10)] で用いられてきた 1〜3-byte 列に加え，長い byte 列を言語特徴に導入する．ただし，6-byte 列以上に長い byte 列は本稿では使用しない．5-byte 列までを使用した場合に比べ，予備実験において有意な精度向上が得られなかったからである．また，文献 5)，6) のように高頻度 byte 列だけに制限することはせず，低頻度な byte 列まで利用する．byte 列を計数する際には，文献 10) と同じ前処理を行い，制御文字などを文書から削除した．これにより，いっさいの修正なしに byte 列の抽出を行った場合と比べ，わずかながら精度向上が見られた．制御文字などの削除による精度向上は，比較実験を行った他の手法でも同様に観測された．

☆ アルバニア語，チェコ語，オランダ語，英語，フランス語，ドイツ語，イタリア語，ノルウェー語，ポルトガル語，トルコ語

表 1 英語文書に一定割合で出現する k-byte 列の 10 言語内での平均出現言語数
Table 1 The average of languages which have k-byte-strings.

DF(%)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
1	7.00	5.07	3.77	2.64	1.66
5	9.00	7.79	6.13	4.00	2.27
10	–	7.20	6.83	4.77	2.32
20	–	9.00	8.14	4.89	2.08
50	–	9.60	7.42	4.50	2.00
100	9.95	9.88	9.20	6.50	4.00

3.2 低頻度 byte 列使用に適した類似尺度

低頻度 byte 列を活用するためには，次の 2 つの条件を満たす類似尺度が必要である．まず，byte 列の出現確率推定値の変動に頑強で，かつ，低頻度 byte 列が十分に識別に影響を及ぼせる類似尺度でなければならない．

そこで提案手法では，以上のような要件を満たす類似尺度として，各言語 l の学習データで出現した byte 列の集合 $Set(l)$，および文書 d 中に出現した byte 列の集合 $Set(d)$ の積集合サイズを類似度尺度として用いる．言語 l と言語 d の類似度 $\mathrm{Sim}(l,d)$ は，次の式により表される．

$$\mathrm{Sim}(l,d) = |Set(l) \cap Set(d)|.$$

この類似尺度は，出現頻度の大小を考慮しないため，文書中や学習データ中における出現確率推定値の変動の影響を受けにくい．また，識別影響力がどの byte 列でも等しいため，低頻度 byte 列でも十分に識別に影響を与えられる．

積集合サイズに基づく類似尺度は，低頻度 byte 列を活用できる一方，頻度情報をすべて失っている．著者らは，このデメリットよりも，低頻度 byte 列を使用するメリットの方が大きいと考えている．

3.3 きわめて低頻度な byte 列の制限

ある言語で書かれた文書中に，他言語の固有名詞の使用や他言語の文の引用などにより，文書本来の言語とは異なる言語が混入することは自然なことである．そのため，ある byte 列がある言語の学習データのみに数回だけ出現していた場合，その byte 列が本当にその言語で発生する byte 列なのかどうかはきわめて判断しにくい．提案した類似尺度は出現確率推定値の

変動の影響を受けにくい手法ではあるが，このような byte 列の導入は過学習の原因となりうる．

ある頻度の byte 列が識別に有効に働くか，それとも過学習の原因となるかは，学習データの質と量によって決まると考えられ，一概には決めることができない．しかし，一般には，より低頻度な byte 列の方が，過学習の原因になりやすいと考えられる．

本稿では，各言語の言語特徴 $Set(l)$ のうち，きわめて低頻度の byte 列を制限できる仕組みを導入する．制限の基準として，Document Frequency（DF）を用いる．出現頻度ではなく DF を用いることで，同一の出現頻度を持つ byte 列でも，幅広い文書中に出現する byte 列の方が優先的に言語特徴に導入される．過学習の原因になる byte 列，たとえば本来他言語で出現する byte 列などは，複数の文書に出現することは少ないと考えられる．そのため，DF を用いた方が，有効な byte 列を優先的に残せる．各言語 l の言語特徴として用いる byte 列の集合 $Set_\theta(l)$ を次式により定義する．

$$Set_\theta(l) = \left\{ x \,\middle|\, \frac{df(x,l)}{|T_l|} \geq \theta \right\}.$$

ここで，$df(x,l)$ は言語 l における byte 列 x の DF を表す．T_l は言語 l の文書集合を表す．θ は閾値であり，実験的に定まる．

識別対象文書 d の言語特徴としては，d 中のすべての byte 列の集合 $Set(d)$ をそのまま用いる．

3.4 アルゴリズム

提案手法は，3.3 節で提案した言語特徴，および 3.2 節で提案した類似尺度を用いて，以下の手順で言語を識別する．

(1) 各言語 l に対し，$Set_\theta(l)$ を構築しておく．
(2) 識別対象文書 d において，出現したすべての byte 列の集合 $Set(d)$ を構築する．
(3) 次の式により，言語を識別する．

$$\hat{L} = \arg\max_l |Set_\theta(l) \cap Set(d)|.$$

本手法は，従来手法を簡略化したものともいえる．提案手法・従来手法のいずれの類似度も，文書中に出現した byte 列の識別影響力を重み付けし，その総和をとったものと考えることができる．提案手法の類似度は次のように変形することができる．

$$\hat{L} = \arg\max_l \sum_{x \in Set(d)} S_P(x,l),$$

$$S_P(x,l) = \begin{cases} 1; & x \in Set(l) \\ 0; & x \notin Set(l) \end{cases}.$$

この式より，提案手法が，従来手法における各 byte 列の識別影響力 S_α（$\alpha = C, D, K, M$）を簡略化し，最も単純な識別影響力（1 or 0）を採用していることが分かる．

4. 実　　験

4.1 類似言語間実験のためのデータ

類似言語間における提案手法の有効性を確認するため，ノルド語族を対象に識別実験を行った．ノルド語族はノルウェー語・デンマーク語・スウェーデン語からなり，最も類似した言語族の 1 つとして知られている．特に，ノルウェー語，デンマーク語では使用されるアルファベットがまったく同じで，単語の表記がきわめて類似している．そのため，従来手法[6),10)] においても識別精度が大きく低下したことが報告されている．

次の手順により，学習データを作成した．

(1) インターネット上のニュースサイト（ノルウェー語：Aften Posten[☆1]，デンマーク語：Dagbladet Arbejderen[☆2]，スウェーデン語：Dagens Nyheter[☆3]）から記事本文を抽出した．1 記事を 1 文書として扱い，各言語 100 文書をランダムに抽出した．
(2) 学習データとして，上記の各文書のランダムな位置から 1,000 byte ずつを抽出した．

また，次の手順により，テストデータを作成した．

(1) 欧州主要語[☆4]のパラレルコーパスである European Parliament Proceedings Parallel Corpus（EUROPARL）[11)] から，デンマーク語・スウェーデン語のデータを抽出した．各文書はコーパスのランダムな位置から抽出し，約 2,000 文書を抽出した．また，ノルウェー語のデータとして，ノルウェー官庁ホームページ[☆5]から記事本文を抽出した．1 記事を 1 文書として扱い，同じく約 2,000 文書を抽出した．
(2) テストデータとして，上記の各文書のランダムな位置から，50, 100, 200, 300, 400, 500 byte ずつ，それぞれ抽出した．こうして得られたデー

☆1 http://www.aftenposten.no/
☆2 http://www.arbejderen.dk/
☆3 http://www.dn.se/
☆4 フランス語，イタリア語，スペイン語，ポルトガル語，英語，オランダ語，ドイツ語，デンマーク語，スウェーデン語
☆5 http://odin.dep.no/

タを，仮想的な小規模文書として扱う．

実験データの作成では，分野やデータ源固有の表記に依存した識別が起こらないように次の点を考慮した．第 1 に，学習データ源とテストデータ源を分離した．第 2 に，学習データの記事を，幅広い分野から収集した．収集した記事中には，政治・経済・科学など多彩な分野が含まれている．第 3 に，サイト特有の形式の影響を排除するため，各記事に若干の修正を行った☆．

4.2 実験 1：閾値の設定

実験 1 では，提案手法において複数の閾値 θ の値を試し，実験的に閾値を設定した．

図 2 に，実験結果を示す．図の横軸は識別対象文書の文書サイズを表し，縦軸は識別精度を表している．この結果から，より低頻度の byte 列を用いるほど，識別精度が向上していることが分かる．

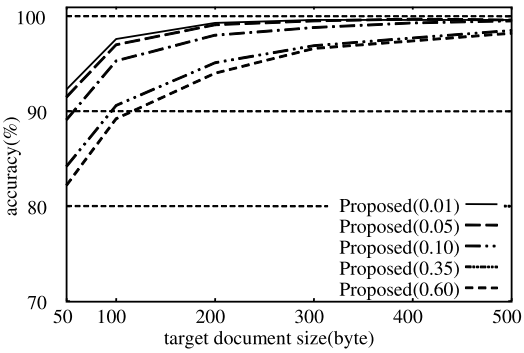
本実験では，3.3 節で言及した過学習の影響は見られなかった．しかし，予備実験において学習データ量を 10 倍に増やし，θ を下げていったところ，$\theta = 0.005$ 周辺で識別精度が飽和することが確認された．この前後において，低頻度 byte 列の利点と過学習による悪影響とが拮抗しているものと考えられる．

θ を小さくするに従って識別精度は上がるものの，識別に用いる byte 列の異なり数が指数関数的に増加する．そのため，精度，メモリ負荷の両面から見て，$\theta = 0.1$ 程度が現実的な値であると考え，この閾値を用いて，以降に述べる実験 2，実験 3 を行った．**表 2** に，デンマーク語における θ と byte 列異なり数の関係を例示する．

4.3 実験 2：類似言語間における比較

実験 2 では，提案手法（$\theta = 0.1$ を使用）と 2 つの従来手法を比較することにより，低頻度 byte 列活用の有効性を確認した．従来手法として，文献 5) の手法，および文献 10) の手法を用いた．文献 5) の手法は言語識別において最も広く参照されている手法の 1 つである．文献 10) の手法は，文献 5) の手法と直接的な性能比較が行われており，類似言語間・小規模文書に対して文献 5) よりも高い識別精度を示したことが報告されている．

各従来手法の実装では，基本的には各手法で提案されているアルゴリズムをそのままコーディングした．ただし，文献 5) の手法では，byte 列計数時の前処理を文献 10) で示されている手法に置き換えた．制御文字などの削除を行う文献 10) の前処理を用いた方が，文献 5) のオリジナル手法より若干高い精度が得られている．この変更により，各手法の前処理が同一になり，言語特徴の選択手法および類似度尺度の違いによる性能比較が明確に行える．また，文献 10) では日本語・中国語・韓国語を対象として，各言語に特有の制御文字の出現有無などを識別材料とした識別手法も提案している．しかし，対象言語集合に対する言語知識を必要とするため，多言語への応用は難しい．そのため，提案手法と直接競合する手法ではないと判断し，実装していない．



Proposed の () 内の数字は閾値 θ の値を表す．

図 2 閾値の探索
Fig. 2 Searching the threshold.

表 2 閾値と byte 列の異なり数の関係
Table 2 The relationship between θ and the number of types of byte-strings.

θ	byte 列異なり数
0.6	452
0.35	1,059
0.1	4,920
0.05	10,154
0.01	58,151

図 3 に，実験結果を示す．図 2 と同様，図の横軸は識別対象文書の文書サイズを表し，縦軸は識別精度を表している．実験結果から，提案手法が，従来手法に比べて高い識別精度を持つことが分かる．

また，図 2 および図 3 の結果を比較することで，各 byte 列の識別影響力に対する頻度による重み付けが，小規模文書に対して必ずしも効果的でないことが分かる．たとえば，提案手法（$\theta = 0.6$）と文献 5) の手法，および提案手法（$\theta = 0.35$）と文献 10) の手法は，それぞれ約 400 種類，約 1,000 種類と，ほぼ同じ異なり数の byte 列を識別に用いている．識別に用いている byte 列の種類が異なるため，直接的な比較はできないものの，提案手法の方が同程度以上の識別精度となっ

☆ (1) 各言語の文字コードを，iso-8859-01（Latin-1）に統一した．(2) 改行コード，および連続したスペースを，スペース 1 文字に変換した．(3) 各記事の先頭および末尾の 1 文を削除し，新聞名や記者名，日付などの情報を取り除いた．

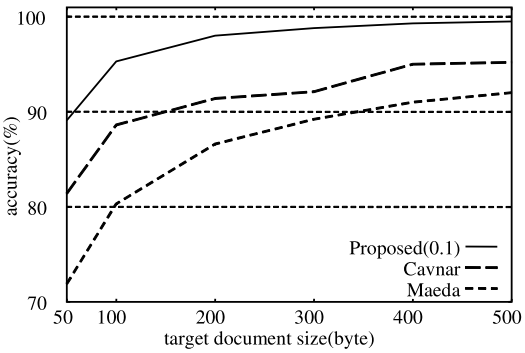


図 3 類似言語間における比較
Fig. 3 Comparing among the similar languages.

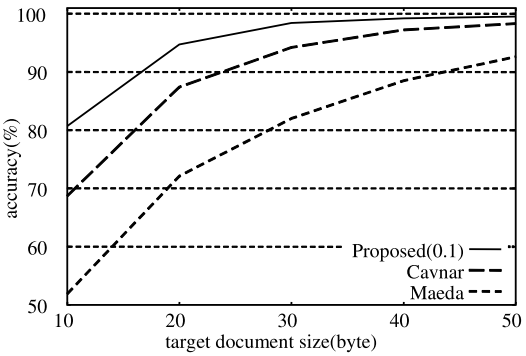


図 4 非類似言語間における識別
Fig. 4 Comparing among the dissimilar languages.

ていることが分かる．

4.4 実験 3：非類似言語間における比較

実験 3 では，非類似言語間における識別実験を行った．本実験により，提案手法が非類似言語間においても高い識別精度を維持できることが確認された．

実験データは ECI-MCI コーパス（European Corpus Initiative Multilingual Corpus I）☆から抽出した．ECI-MCI コーパスは，様々なデータ源から抽出されたサブコーパスから構成されており，欧州主要語を中心に集められている．

実験データを，次の手順で作成した．

(1) ECI-MCI コーパスのうち，比較的潤沢な量がある 10 言語☆☆に対し，各言語 100 文書をランダムに抽出し，さらに各文書のランダムな位置から，1,000 byte ずつを抽出した．

(2) 次に，学習データと重ならない文書を，各言語から約 1,000 文書ずつランダムに抽出した．テストデータとして，上記の各文書のランダムな位置から 10，20，30，40，50 byte をそれぞれ抽出した．これを，擬似的な小規模文書として用いる．識別対象文書のサイズはノルド語族に対する実験のときよりも小さい．非類似言語間では，どの手法でも，100 byte 以上のデータに対しては十分な識別精度を発揮するためである．

図 4 に，実験結果を示す．図 2 と同様，図の横軸は識別対象文書の文書サイズを表し，縦軸は識別精度を表している．これらの結果から，提案手法が，非類似言語間においても高い識別精度を発揮できることが分かる．また，欧州主要語間の識別においては，識別対象文書が 20 byte 程度あれば，精度 95%以上と十分に高い識別精度が得られることも分かる．

☆ http://www.elsnet.org/resources/eciCorpus.html
☆☆ アルバニア語，チェコ語，オランダ語，英語，フランス語，ドイツ語，イタリア語，ノルウェー語，ポルトガル語，トルコ語

5. おわりに

本稿では，低頻度 byte 列を言語特徴として言語識別に導入することを提案した．提案手法では，低頻度 byte 列を活用するために，識別対象文書中に出現した byte 列の集合と，各言語の学習データ中に出現した byte 列の集合との積集合サイズに基づいた類似度尺度を用いている．また，きわめて低頻度な byte 列における過学習の影響を防止するため，言語特徴として用いる byte 列を DF により制限している．

提案手法の有効性を確認するため，類似言語であるノルド語族や，非類似言語である欧州主要語間に対して識別実験を行った．実験の結果から，低頻度 byte 列の利用により，類似言語間や小規模文書に対しても高精度な言語識別が可能であることが示された．

今後の課題として，識別に用いる byte 列を絞り込む手法の開発を行う必要がある．筆者らは，言語識別に有用な byte 列は，全体のごく一部であると考えている．そのため，識別に有用な byte 列の選別手法を開発することで，識別精度を維持したまま，手法の省メモリ化が図れる．

謝辞 本研究は文部科学省の 21 世紀 COE プログラム「システム情報科学での社会基盤システム形成」の援助を受けて行われた．

参考文献

1) Kishida, K., hua Chen, K., Lee, S., Kuriyama, K., Kando, N., Chen, H.-H., Myaeng, S.H. and Eguchi, K.: Overview of CLIR Task at the Fourth NTCIR Workshop, *Proc. NTCIR Workshop 4*, pp.1–59 (2004).
2) Zhang, Y. and Vines, P.: Using the web for automated translation extraction in cross-

language information retrieval, *SIGIR '04: Proc. 27th annual international conference on Research and development in information retrieval*, pp.162–169, ACM Press, New York, NY, USA (2004).
3) Henrich, P.: Language identification for the automatic grapheme-to-phoneme conversion of foreign words in a german text-to-speech system, *Proc. Eurospeech 1989, European Speech Communication and Technology*, pp.220–223 (1989).
4) Giguet, E.: Multilingual Sentence Categorization according to Language, *European Chapter of the Association for Computational Linguistics SIGDAT Workshop "From Text to Tags: Issues in Multilingual Language Analysis"*, Dublin Ireland, pp.73–76 (1995).
5) Cavnar, W.B. and Trenkle, J.M.: N-Gram-Based Text Categorization, *Symposium On Document Analysis and Information Retrieval*, University of Nevada, Las Vegas., pp.161–176 (1994).
6) Martins, B. and Silva, M.J.: Language identification in web pages, *SAC '05: Proc. 2005 ACM symposium on Applied computing*, pp.764–768, ACM Press, New York, NY, USA (2005).
7) Dunning, T.: Statistical identification of language, Technical report CRL MCCS-94-273, Computing Research Laboratory, New Mexico State University (1994).
8) Sibun, P. and Reynar, J.C.: Language Identification: Examining the Issues, *5th Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, U.S.A., pp.125–135 (1996).
9) 北　研二：確率的言語モデルに基づく多言語コーパスからの言語系統樹の再構築，自然言語処理，Vol.4, No.3, pp.71–82 (1997).
10) 前田　亮，関　慶妍，吉川正俊，植村俊亮：Web文章の符号系及び使用言語の自動識別，電子情報通信学会論文誌，Vol.J84-D-II, No.1, pp.150–158 (2001).
11) Koehn, P.: A Multilingual Corpus for Evaluation of Machine Translation, *Unpublished* (2002). http://people.csail.mit.edu/u/k/koehn/public_html/

(平成 17 年 9 月 2 日受付)
(平成 18 年 1 月 6 日採録)

行野　顕正（学生会員）
2001 年九州大学工学部電気情報工学科卒業．2003 年同大学大学院システム情報科学府修士課程修了．現在，同大学院システム情報科学府博士後期課程在学中．自然言語処理の研究に従事．

田中　省作（正会員）
2000 年九州大学大学院システム情報科学研究科博士後期課程修了．九州大学情報基盤センター助手，同大学高等研究機構室員を経て，2005 年より立命館大学文学部助教授．博士（工学）．自然言語処理，言語教育への応用に関する研究に従事．言語処理学会，英語コーパス学会各会員．

冨浦　洋一（正会員）
1989 年九州大学大学院工学研究科電子工学専攻博士課程単位取得退学．同年九州大学工学部助手，1995 年同助教授，1996 年同大学大学院システム情報科学研究科助教授，2000 年同大学院システム情報科学研究院助教授，現在に至る．博士（工学）．自然言語処理，計算言語学，人工知能に関する研究に従事．言語処理学会，人工知能学会各会員．

松本　英樹
2004 年九州大学工学部電気情報工学科卒業．現在，同大学大学院システム情報科学府修士課程在学中．自然言語処理の研究に従事．電子情報通信学会学生会員．