極大部分文字列を使った twitter 言語判定

中谷 秀洋
サイボウズ・ラボ株式会社

nakatani@labs.cybozu.co.jp

1　はじめに

言語判定 (または言語判別，言語識別) はテキストが記述されている言語を推定する技術である．全文検索における言語の絞り込みや，文書分類 (スパム判定など)，本文抽出や機械翻訳など，言語モデルを用いる言語処理において言語判定は前提処理として必須であり，したがってウェブ上のデータ処理やサービスに対して有用な技術である．また昨今の国際化の流れの中で，複数の国・地域をまたがる拠点を持つ企業も少なくなく，そのためグループウェアのように閉じたネットワーク上で利用されるアプリケーションでもその必要性は増している．

言語判定の技術は早くより研究されており [4], [6]，またライブラリとしても提供されている [5], [2] が，それらの多くは文書が十分長いことを仮定しており，文書が短い場合は対象外であった．良い言語判定器は十分長い文書に対しては99%を超える精度で判定可能であるが，3〜10語の短文では90%程度の精度にとどまり，実用上十分な精度があるとは言えなかった．

一方で，twitterに代表されるマイクロブログの流行もあって，ウェブ関連サービスにて言語横断の短文を扱う機会は増している．各種業務システムにおいてもユーザにより生成された短い文章リソースは少なくない．また短文の言語判定が可能であれば，複数言語を用いて記述された文書の言語判定において，段落ないし文ごとに区切って言語判定を行わせるなど，幅広い応用も考えられる．しかし，短文を対象とした言語判定の研究はその必要性に比して少ないのが現状である．

本研究では，極大部分文字列を素性として用いることで，twitterのようなノイズの多い短文でも17言語に対して99%以上という実用的な精度で言語判定を行えることを示す．

第2章では既存の関連研究について，第3章では極大部分文字列を使った文書分類について概説する．第4章で本研究における言語判定の手法を述べ，第5章では評価のために作成したコーパスについて解説する．第6章では本研究の評価を記し，第7章では結論と課題をまとめる．

2　関連研究

言語判定に関する研究の多くは，言語判定を文書分類の問題として定式化している．文字n-gram素性により分類器を構築するのが最も一般的である [4], [5]が，分布の類似度によって判別する手法 [6]，グラフを用いる手法 [7] などがある．

これまでの研究により，十分ノイズが少なく十分長いテキストであれば100%に近い精度で言語判定を行えることが示されており [4], [9]，実用的な精度で言語判定を行うライブラリも公開されている [5], [2]．

ところがこれらの手法をtwitterのツイートに対して適用した場合，90%前後の精度に落ちてしまうことがわかっている [7]．要因の一つとして，マイクロブログ固有の表現が多いことが考えられる．同じ分類器においてもtwitterから収集したコーパスを訓練データとすることで，95%程度まで精度が向上することが示されている [10]．

もう一つの要因としては，短い文章から抽出できる素性の数が十分ではないことが考えられる．Trompら [7] は3-gramにグラフに基づく長距離の素性を加えることで，6言語のツイートの言語判定に対して95〜98%の精度を上げている．

単語辞書を用いる手法も考えられるが，巨大な辞書をメンテナンスしなければならないという難点がある．

言語判定の対応言語数は，研究論文の多くは欧文言語を中心に2〜8言語程度であり，最大でも20言語前後である．一方，ライブラリとして提供されている言語判定器の対応言語数は，Tika[1] が27，langdetect[5] が53, CLD[2] が76と全般に十分多い．

3　極大部分文字列による文書分類

岡野原ら [11] は極大部分文字列を素性に用いた文書分類を提案した．極大部分文字列とは，部分文字列間に導入されるある半順序関係における極大元を指す．

その定義は以下の通りである．

文字列 T の i 番目から j 番目までの部分文字列を T_{ij} と書く．T の相異なる部分文字列 t_1, t_2 の間に $t_1 < t_2$ の関係があるとは，$0 \leq {}^{\exists}k \leq l_2 - l_1, {}^{\forall}i$ s.t. $T_{i(i+l1)} = t_1 \Rightarrow T_{(i-k)(i-k+l2)} = t_2$ を満たすことと定義する．ただし l_n は t_n の長さとする．この順序関係における極大元を極大部分文字列と呼ぶ．

順序関係を持つ文字列同士は部分文字列内での出現頻度が常に一致することから，それらの素性 (出現の有無ないし頻度) の線形結合は順序関係により導出される同値類によってまとめられる．ここで特に同値類の代表元として極大部分文字列を選ぶと，全部分文字列を素性としたロジスティック回帰は，極大部分文字列を素性とするそれと同値であることがわかる．

訓練集合に含まれない文字列を加えた場合，一般にモデルの同値性は崩れるが，訓練集合を十分大きくとることで近似できると仮定する．

出現頻度が 2 回以上の極大部分文字列の抽出は拡張接尾辞配列を用いることで入力長に対して線形時間で行うことができる．また素性列は TRIE 木に格納することで，任意のテキストからの素性の抽出も高速に行うことができる [11].

4 短文言語判定

本研究のターゲットは，ラテン文字言語の 3 単語以上の文に対して実用的な精度で言語判定を行うこととした．実際の対象言語は表 2 に挙げる．

ラテン文字を使用する言語は最も数が多く，判別の難易度も高い．したがってラテン文字言語において高い精度で言語判定可能であれば，他の文字種の言語にも拡張可能であると期待している．複数の文字種を用いたテキストの言語判定では，文字種ごとにテキストを分割することを想定している．

4.1 モデル

言語判定は言語ラベルを付与済みの教師付きコーパスの分類問題として定式化される．本研究では，テキスト全体を連結した文字列から極大部分文字列を抽出し，それを素性とした多クラスロジスティック回帰モデルにより分類器を構築した．

パラメータの学習には確率的勾配降下法を用いた．Tsuruoka らの提案する Cumulative Penalty[8] を用いた L1 正則化によってモデルのサイズを抑える．

また後述の通り，言語ごとにデータ量の偏りがあるため，各言語ごとに訓練データのリサンプリングを行うことで，最大件数の言語にデータ件数を揃えた．

4.2 テキストの正規化

正規化のアプローチは各言語の特性に対するものと，twitter などマイクロブログの特性に対するものとの大きく 2 つに分かれる．

なお，本章では各言語の文字利用に関する考察を Unicode に基づいて解説する．Unicode のコードポイントは U+0049 などと記す．

4.2.1 トルコ語を考慮した小文字化

ラテン文字言語に対する正規化として，小文字化はモデルの縮小と訓練データの節約に効果がある．ただし多言語横断においては，大文字小文字が必ずしも一対一ではない．

I(U+0049) の小文字はほとんどの言語で i(U+0069) であるが，トルコ語やアゼルバイジャン語では ı(U+0131) である (表 1)．そのため本研究では，I(U+0049) を小文字化の対象から除外した．

表 1: 大文字-小文字対応

	大文字	小文字
トルコ語以外	I(U+0049)	i(U+0069)
トルコ語	I(U+0049)	ı(U+0131)
	İ(U+0130)	i(U+0069)

4.2.2 ルーマニア語向けの正規化

言語に本来割り当てられた文字とは異なる文字 (コードポイント) が慣習的に利用されていることもある．例えばペルシャ語では ى(U+06CC, Farsi yeh) の代わりに ي(U+064A, Arabic yeh) がペルシャ語の新聞サイトを中心に広く利用されている．これはアラビア文字圏で普及している CP1256 文字セットに ى(U+06CC) が含まれないためと考えられる．これを考慮した正規化を行うことでペルシャ語の判定精度が向上する [9].

同様に，ルーマニア語では ş(U+0219), ţ(U+021B) 等が正書法の定める文字だが，実際には ş(U+015F) や ţ(U+0163) の方が多く使用されている．これは ş(U+0219) らの文字が Unicode に採録されたのが 2001 年，実際に利用可能な PC が普及し始めたのが 2007 年と遅い時期であったため，早くより利用可能であった ş(U+015F) らでの代用が定着してしまったためと考えられる．

一方で ş(U+015F) はトルコ語やアゼルバイジャン語でも利用される文字ゆえ，本研究では ş(U+0219) らを ş(U+015F) らに置き換えるという正規化を行った．

4.2.3 ベトナム語向けの正規化

ベトナム語は 6 種類の声調を持ち，かつ声調記号が一般のテキストでも記述される．Unicode には，ベトナム語の声調記号付きアルファベットが U+1EA0 から U+1EF9 という広い範囲に割り当てられているが，それら以外にも「アルファベット＋声調記号文字」の合字として表現することも可能である．実際に twitter ではそれら 2 種類の表記がほぼ同割合で出現する．そのため，本研究では合字を 1 コードポイントに変換する正規化を行った．

4.2.4 マイクロブログ固有表現の正規化

テキスト内の@ユーザ名，ハッシュタグと URL は言語判定におけるノイズとなるので，除去した．

また，マイクロブログでは感情の表現や強調のために文字を重ねることがよく用いられている．本研究では同じ文字が 3 回以上連続した場合に 2 文字に縮めるという正規化を行った．Brody ら [3] の提案する twitter 表現の正規化を用いて生成した辞書を併用すれば性能の向上が期待できるので，今後の課題にしたい．

5 twitter コーパスの作成

言語判定器の学習・評価のために必要な 17 言語の短文コーパスセットを以下の手順で作成した (表 2)．

フランス語を例に説明する．twitter の Streaming API[1] の sample メソッド (全ツイートの 1%を取得可能) を使ってツイートを収集，Paris タイムゾーンのツイートを抽出した．全体の中ではフランス語のツイートは 1%未満であるが，Paris タイムゾーンに限ると 50〜60%となり，抽出の難易度が大きく下がる．

それらを langdetect[5] で言語判定し，以降は手作業でフランス語と判定されたデータからフランス語ではないツイートを除外し，フランス語以外と判定されたデータからフランス語のツイートを拾った．

本手順をラテン文字以外の言語に適用するに当たっては，twitter で十分多くつぶやかれている言語が少ないという問題がある．たとえばキリル文字言語ではロシア語以外の流量は非常に少なく，まとまって収集可能なのはウクライナ語程度である．アラビア文字言語についても，アラビア語以外は十分な量のコーパスを確保できる見込みが薄い．

ベンガル語などのインド系の言語も twitter 上の利用割合はごく少量である．ただしこれらは文字と言語が 1 対 1 に対応するものが多く，既存手法でも十分高い精度で判定可能なため，本研究の対象外である．

[1]https://dev.twitter.com/docs/streaming-api

表 2: データセット

言語名	訓練データ	テストデータ
チェコ語	4,583	5,329
デンマーク語	5,499	5,484
ドイツ語	44,115	9,665
英語	44,007	9,617
スペイン語	44,965	10,127
フィンランド語	4,577	4,491
フランス語	44,175	10,067
インドネシア語	44,892	10,184
イタリア語	44,197	10,160
オランダ語	44,972	9,681
ノルウェー語	7,516	8,510
ポーランド語	12,861	10,070
ポルトガル語	44,488	9,460
ルーマニア語	6,118	5,904
スウェーデン語	44,343	9,950
トルコ語	44,798	10,309
ベトナム語	10,415	10,494
計	496,521	149,502

6 評価

モデルの実装は Python で行い，ソースコードと訓練後のモデルは MIT ライセンスで公開した[2]．

極大部分文字列の抽出には，訓練データを U+0001 で結合した文字列に対して esaxx[3]を適用した．そちらのモジュールも公開準備中である．

訓練データを用いて提案手法 (LDIG) の学習を行い，テストデータについて言語判定を行った．ベースラインは CLD(76)[2] と Tika(27)[1], そして langdetect(53)[5] である (括弧内は対応言語数)．ただし CLD はインドネシア語をマレー語と判定するためそれらを同一視し，Tika はチェコ語，インドネシア語，トルコ語，ベトナム語に対応していないため 13 言語分についてのみ評価した．また langdetect は全 53 言語を判定対象とした場合 (LD53)，本研究において作成した訓練データを用いて言語プロファイルを生成した場合 (LDtw) についてそれぞれ評価した．その結果は表 3 である．本提案手法は 17 言語について平均で 99.1%の精度で言語判定を行い，比較手法のいずれよりも高い精度を示した．

Tromp らの提案する LIGA[7] は 6 言語のツイートを 95〜98%の精度で判定したと報告している．評価

[2]https://github.com/shuyo/ldig/
[3]http://code.google.com/p/esaxx/

表 3: 評価結果

言語名	LDIG	CLD	Tika	LD53	LDtw
チェコ語	99.7	95.3	—	95.5	99.8
デンマーク語	96.7	90.4	53.3	95.5	91.6
ドイツ語	99.5	96.7	59.1	87.3	94.1
英語	98.7	97.4	55.7	89.3	95.4
スペイン語	98.9	90.7	45.0	94.9	98.1
フィンランド語	99.4	99.4	94.2	98.8	99.6
フランス語	99.5	94.5	67.5	96.2	98.4
インドネシア語	99.8	92.7	—	89.6	98.9
イタリア語	99.6	93.8	87.0	96.5	98.1
オランダ語	98.3	92.8	64.4	68.6	97.0
ノルウェー語	97.1	74.3	68.5	96.3	96.3
ポーランド語	99.8	97.9	89.0	98.3	99.8
ポルトガル語	98.8	88.3	48.7	90.4	97.1
ルーマニア語	98.5	96.0	82.5	93.7	98.3
スウェーデン語	99.0	96.5	76.1	96.7	98.2
トルコ語	99.9	97.4	—	97.8	99.5
ベトナム語	99.8	98.9	—	98.9	99.4
全体	99.1	93.7	69.1	92.9	97.7

用のデータセット (9066 件) が公開されており[4]，これを用いることで [7] の結果と比較が可能である．そのデータセットについて本手法で言語判定を行った場合も 99%以上の精度で判定が可能であった (表 4)．モデルは 17 言語で学習したものをそのまま用いた．

表 4: LIGA[7] のデータセットを使った評価

言語名	件数	正解数	正解率
ドイツ語	1468	1479	99.3
英語	1488	1505	98.9
スペイン語	1531	1562	98.0
フランス語	1542	1551	99.4
イタリア語	1531	1539	99.5
オランダ語	1423	1430	99.7
全体	8985	9066	99.1

7 おわりに

極大部分文字列を素性とするロジスティック回帰により，twitter のようなノイズの多い短文でも 99%以上という実用的な精度で言語判定を行えることを示した．ただし実用にあたっては，精度を維持しつつモデルを縮小することが課題となる．

本研究では純粋に文書から抽出可能な素性のみを扱ったが，アプリケーションが持つメタデータも素性として組み込むことでさらなる精度の向上が見込める．これもあわせて今後の課題にしたい．

[4]http://www.win.tue.nl/ mpechen/projects/smm/

参考文献

[1] Apache Tika, URL: http://tika.apache.org/.

[2] chromium-compact-language-detector, URL: http://code.google.com/p/chromium-compact-language-detector/.

[3] Samuel Brody and Nicholas Diakopoulos. Cooooooooooooooollllllllllll!!!!!!!!!!!!!! Using Word Lengthening to Detect Sentiment in Microblogs. In *EMNLP*, pp. 562–570. 2011.

[4] William B Cavnar and John M Trenkle. N-Gram-Based Text Categorization. *Ann Arbor MI*, Vol. 48113, No. 2, p. 4001,

[5] Shuyo Nakatani. Language Detection Library for Java, 2010, URL: http://code.google.com/p/language-detection/.

[6] P. Sibun and J.C. Reynar. Language identification: Examining the issues. In *5th Symposium on Document Analysis and Information Retrieval*, pp. 125–135. 1996.

[7] Erik Tromp and Mykola Pechenizkiy. Graph-Based N-gram Language Identification on Short Texts. In *Belgian-Dutch Conference on Machine Learning (Benelearn)*, pp. 27–35, 2011.

[8] Y. Tsuruoka, J. Tsujii, and S. Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *ACL*, 2009.

[9] 中谷秀洋. ナイーブベイズによる言語判定. 自然言語処理勉強会@東京 (TokyoNLP), No. 2, URL: http://www.slideshare.net/shuyo/langdetect.

[10] 中谷秀洋. ∞-gram による短文言語判定. 自然言語処理勉強会@東京 (TokyoNLP), No. 8, URL: http://www.slideshare.net/shuyo/gram-10286133.

[11] 岡野原大輔, 辻井潤一. 全ての部分文字列を考慮した文書分類. 自然言語処理研究会 (SIGNL), No. 187,