単語の非定常性を考慮したツイートストリームの分類: 接尾辞配列による学習

西田　京介† 　星出　高秀† 　藤村　考†

† 日本電信電話株式会社 NTT サイバーソリューション研究所　〒 239–0847 神奈川県横須賀市光の丘 1–1
E-mail: †{nishida.kyosuke,hoshide.takashide,fujimura.ko}@lab.ntt.co.jp

あらまし　Twitter のツイートに代表される，時間経過と共にデータの性質が変化する文書ストリームに対する分類モデルを提案する．ツイートを解析して得られた文書ストリーム分類における課題，特に，データの性質が単語毎に異なるスケールで変化する問題について，提案手法は，各単語の出現確率過程の非定常性をモニタリングし，単語毎に長期的・短期的なデータに基付いて推定された 2 つの確率を切り替えることで解決する．そして，全文検索インデックスの一つである単語接尾辞配列（WSA）を利用した提案手法の実装方法について示す．WSA の利用により，提案モデルは学習と分類処理において，単語 n グラムの出現に関する時間影響を効率的に扱うことができる．性質の異なる 3 つの実データセットを用いて評価実験を行ったところ，提案手法が従来手法よりも高い分類精度を実現した．

キーワード　Twitter, 文書分類, データストリーム，コンセプトドリフト，接尾辞配列，管理図

Tweet Stream Classification with Word Non-stationarity: Learning with Suffix Arrays

Kyosuke NISHIDA†, Takashide HOSHIDE†, and Ko FUJIMURA†

† NTT Cyber Solutions Laboratories, NTT Corporation　1–1 Hikarinooka, Yokosuka-Shi, Kanagawa, 239–0847 Japan
E-mail: †{nishida.kyosuke,hoshide.takashide,fujimura.ko}@lab.ntt.co.jp

Abstract　We propose a classification model of tweet streams in Twitter, which are representative of document streams whose statistical properties will change over time. Our model solves several problems that hinder the classification of tweets; in particular, the problem that the probabilities of word occurrence change at different rates for different words. Our model switches between two probabilities based on full and recent data for each word by monitoring the word non-stationarity on the probability of word occurrence. This switching enables our model to achieve both accurate learning of stationary words and quick responses to bursty words. We then explain how to implement our model by using a word suffix array, which is a full-text search index. Using the word suffix array allows our model to handle the temporal attributes of word n-grams effectively. Experiments on three tweet data sets demonstrate that our model offers statistically significant higher classification accuracies than conventional temporally-aware classification models.

Key words　Twitter, text classification, data streams, concept drift, suffix arrays, control charts

1. はじめに

マイクロブログサービス，特に Twitter は新たなリアルタイム情報基盤として爆発的な成長を続けており，2012 年現在，Twitter では 1 日あたり 2 億件以上のツイート（自身の状況や雑感などを表す上限 140 文字の短い文書）が投稿されている．Twitter の様に，リアルタイム性が高く膨大な量のデータが生成されるメディアにおける文書ストリームの自動分類は，データマイニングにおける重要な課題であり，時間経過と共にデータの性質が変化する点が学習を非常に難しくする．

特に，ツイートストリームでは，バースト性の高い（出現頻度が急増した）単語への素早い適応と，変化の少ない単語についての正確な学習に関するトレードオフが大きな問題となる．従来の文書ストリーム分類手法では，文書単位での時間的な選択や加重により変化に適応するため，ツイートストリームの分類では高い精度を実現することができない．

本研究では，各単語の出現確率過程の非定常性をモニタリングし，単語毎に長期的・短期的なデータに基付いて推定された 2 つの確率を切り替えることで上記のトレードオフを解決するモデル P-Switch を提案する．そして，単語接尾辞配列（word

suffix array; WSA）[1] による提案手法の実装方法を示す．WSA の利用により，単語 n グラムの時間影響が効率的に扱えるようになると共に，学習・分類処理を分散・並列化できる．

論文の構成を以下に示す．**2.** 章にて文書ストリーム分類に関する従来研究を示す．**3.** 章にてツイートストリームにおける変化の解析を行い，課題を整理する．**4.** 章に提案手法を，**5.** 章に WSA による実装方法を示す．**6.** 章ではハッシュタグ付きツイートによる話題分類の評価実験を行って，提案手法が従来手法よりも高い分類精度を実現したことを示す．最後に，**7.** 章にて結論を示す．

2. 文書ストリーム分類

本章では，文書ストリーム分類の問題定義と関連研究について示した後，本研究の位置付けを示す．

2.1 コンセプトドリフト

コンセプトドリフト（concept drift）とは，学習対象の基となる，陽に与えられない統計的な性質が時間と共に変化することを指す [2], [3]．文書ストリームは時間順に整列されて連続的に与えられるものであるため，高精度な文書分類を実現するには，コンセプトドリフトの存在を考慮する必要がある．

考慮すべき変化の種別としては以下の 3 つが挙げられる．

[C1] 急激な変化（sudden shift）

[C2] 緩やかな変化（gradual drift）

[C3] 周期的な変化（recurring themes）

2.2 問 題 定 義

文書ストリーム分類の問題定義は，以下の 3 つに大別できる．

[P1] Test-Then-Train (Incremental) [4]〜[7]

分類モデルは，各時刻 $t = 1, 2, \ldots$ において，クラスラベルなしの文書 d_t を一つ受け取り，クラス $\hat{c}_t$ に分類する．分類後，d_t についての真のクラスラベル c_t が与えられる．

[P2] Test-Then-Train (Chunk) [8]〜[12]

分類モデルは，各時刻 t において，クラスラベルなしの文書集合 $D_t = \{d_{t,1}, d_{t,2}, \ldots\}$ を受け取り，$d_{t,m} \in D_t$ をクラス $\hat{c}_{t,m}$ にそれぞれ分類する．分類後，D_t についての真のクラスラベル集合 $C_t = \{c_{t,1}, c_{t,2}, \ldots\}$ が与えられる．

[P3] Train-And-Test [13]

分類モデルは，各時刻 n において，クラスラベルありの訓練文書集合 $D^*_t = \{(d^*_{t,1}, c^*_{t,1}), (d^*_{t,2}, c^*_{t,2}), \ldots\}$ と，クラスラベルなしのテスト文書集合 $D_t = \{d_{t,1}, d_{t,2}, \ldots\}$ を受け取り，$d_{t,m} \in D_t$ をクラス $\hat{c}_{t,m}$ にそれぞれ分類する．

また，**P3** の形式で全てのデータを受け取った後にバッチ学習を行う研究も取り組まれている [14]〜[17].

2.3 関 連 研 究

コンセプトドリフトの存在を考慮した文書分類のためのアプローチは，次節より示す 3 つのグループに大別できる．

2.3.1 Instance selection

instance selection は，時刻 n で与えられたテスト文書を正しく分類するために，過去に与えられた訓練文書を適切に選択して学習するアプローチである．時間窓に基づく手法が代表的であり [6]〜[9]，分類モデルは，固定/可変サイズの時間窓に含まれる最近の文書に適合するようにモデルを常時更新することで変化に対応する．

2.3.2 Instance weighting

instance weighting は，過去に与えられた訓練文書を時刻で重み付けして学習するアプローチであり，新しい文書ほど大きな重みを与えることで，環境の変化に適応しようとする手法が一般的である [6], [9], [11], [15]〜[17].

しかし，instance selection，weighting ともに変化 **C1** と **C2** のトレードオフ関係の解決が難しく，両方に適応可能な手法は未だ提案されていない．また，これらは最近の文書を重視するものであるから，変化 **C3** を効率的に扱えない．

2.3.3 Ensemble methods

Ensemble methods は，過去に構築した分類モデルを保持・統合して利用することで変化に適応するアプローチ [5], [10], [12], [13] であり，主に，変化 **C2** や **C3** に対して有効である．このアプローチでは，一定の期間ごとに，あるいは分類精度が悪化した際に，古いモデルや分類精度の低いモデルを削除して新しいモデルを追加することで，学習対象の変化に適応する．

2.4 本研究のスコープ

本研究では，最も基本的な問題定義である **P1** を扱い，変化 **C1** と **C2** への適応のトレードオフ問題の解決を目指す．我々の手法は instance weighting の一種であり，文書単位で重み付けを行う従来手法に比べて，単語単位での変化に適応可能な点が新しい． また，提案手法を ensemble methods へ拡張することによる変化 **C3** への効率的な適応は今後の課題とする．

3. データ解析: ツイートストリーム

本章では，Twitter のツイートストリームについてハッシュタグで定義される話題の変化に関する解析を行った結果と，従来手法による話題分類の精度について示し，課題を整理する．

3.1 データセット

2011 年 9 月 13 日から 26 日の期間に，Twitter 社が提供する Streaming API の statuses/filter（指定した文字列が含まれるツイートが収集可能）を利用して，表 1 に示す 3 つのデータセット: **NPB**（日本野球機構 12 球団に関するツイート），**TV**（日本のテレビジョン放送局 7 局に関するツイート），**MLB**（米メジャーリーグ 30 球団に関するツイート）を収集した．NPB と MLB では各球団，TV では各テレビ局がクラスに対応する．

表 1 ツイートデータセット（2011 年 9 月 13 日〜26 日に収集．ハッシュタグ（＝クラスラベル）を 1 つのみ含むツイートで構成．

Dataset	# Tweets	# Classes	Examples of Track Keywords (Hashtags)
NPB	314,209	12	#dragons, #hanshin, #giants, etc.
TV	249,080	7	#nhk, #etv, #ntv, #fujitv, etc.
MLB	200,521	30	#angels, #astros, #athletics, etc.

なお，これらのデータセットは，ハッシュタグが 1 つだけ付与されているツイートのみを含み，複数個のハッシュタグ，リツイート（公式・非公式 RT），メンション/リプライ（@username）が含まれるツイートは含まない．また，日本語ツイートの形態素解析には MeCab 0.98（IPA 辞書）を用い，名詞・動詞・形容

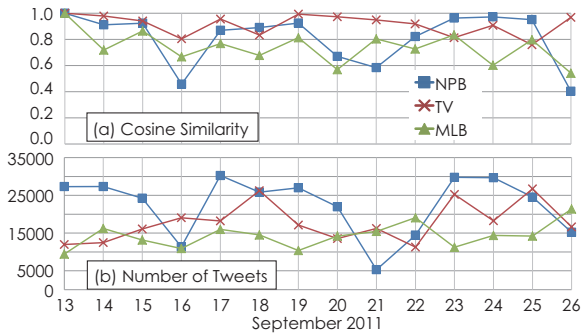

図 1　(a) 2011 年 9 月 13 日と他日におけるクラス分布のコサイン類似度の変化．(b) 各日における全ツイート数の推移．

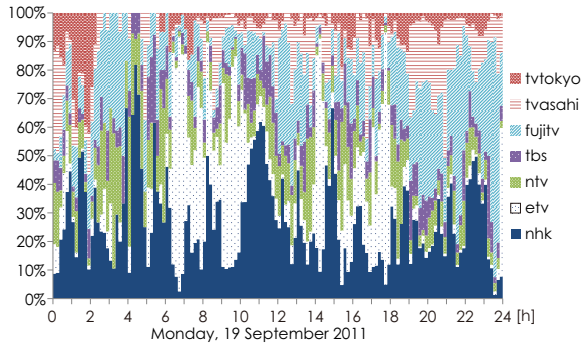

図 2　TV データセットにおける 10 分ごとのクラス分布の変化例．

詞と判定されたものを単語として抽出した．英語ツイートについては，ストップワードを除去し，句読点と非英数字を区切り文字として単語に分割した．

3.2　解析結果

3.2.1　クラス分布の時間変化

図 1 に，2011 年 9 月 13 日と他日のクラス分布の間のコサイン類似度の変化と，各日における全ツイート数の推移を示す．図 1a より，NPB と MLB データセットは TV データセットに比べてクラス分布の観点ではより変動が激しいことが分かる．一方，TV データセットのクラス分布についても，図 2 に示す様に短期的に見ると激しく振動している．

すなわち，ツイートストリームにおいてクラス分布は様々なタイムスケールで激しく常に変動している．分類モデルは緩やかな変化と急激な変化の両方に適応しなければならない．

3.2.2　単語出現確率の時間変化

図 3 に，NPB データセットにおける (a) 単語「ホームラン」が含まれるツイート数の 1 時間ごとの推移と，(b) 単語「横浜」が含まれる日別ツイート数の推移を示す．「ホームラン」は出現頻度が短期間の間に急激に上昇するバースト性の強い単語（例えば，図 3a・16 時の#chibalotte クラスの急増）であり，関連するクラスの変動が激しい．一方，単語「横浜」は比較的安定した単語で，最も関連するクラス（横浜市に本拠地を持つ #baystars）は変化していない．ただし，この単語についても，その他のクラスにおける出現頻度は日々変化している．

つまり，単語の出現に関する統計的な性質は，単語毎に異なるスケールで変化している．文書単位で選択と重み付けを行う従来の instance selection，weighting 法では，単語間の変化のスケールの違いを扱うことができない．

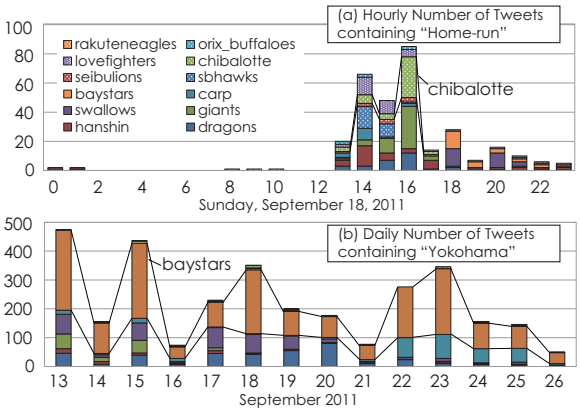

図 3　NPB データセットにおける (a) 単語「ホームラン」が含まれるツイート数の 1 時間毎の推移と，(b) 単語「横浜」が含まれる日別ツイート数の推移．

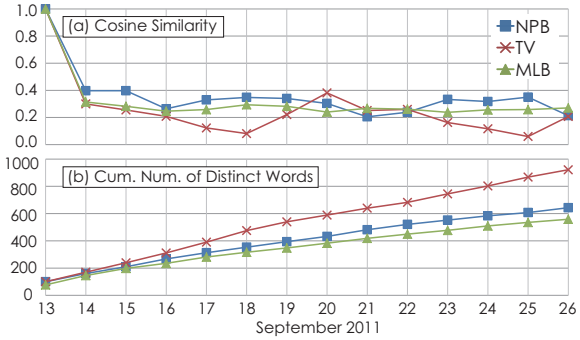

図 4　(a) 2011 年 9 月 13 日と他日におけるカイ二乗統計量上位 100 語のコサイン類似度の変化．(b) 上位 100 語の異なり単語数の累積和の推移．各クラスごとに計算し，平均を取った結果を示す．

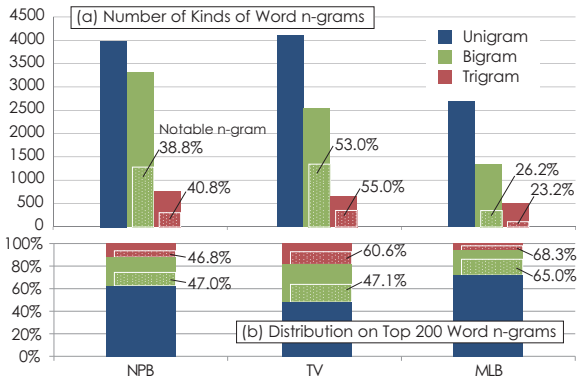

図 5　(a) 日別に 5 回以上出現した単語 n グラムの種類数と，有用な単語 n グラムの割合．有用な単語 n グラムのカイ二乗統計量 $\max_c\{\chi_c^2\}$ は，その単語 n グラムの最初の文字のそれよりも大きい．(b) $\max_c\{\chi_c^2\}$ の上位 200 語の単語 n グラムの分布．日別の結果を平均した結果を示す．

3.2.3　新しい単語の出現

新しい単語の出現について解析するため，日別の全単語に対するクラス別カイ二乗統計量 χ_c^2（計算方法は [18] 参照）を計算し，クラス毎にカイ二乗統計量の上位 100 語を選択した．図 4 に，2011 年 9 月 13 日と他日の上位 100 語の間のコサイン類似度の変化と，上位 100 語の異なり数累積和の推移について示す．

図 4 より，分類に有用な単語が日々新たに出現していることが分かる．NPB や MLB データセットではコサイン類似度のが 0 に近づいていかないことから，一部の単語は時間の経過に関わらず分類に有用であり続けている一方で，TV データセットではツイート内容が日々大きく新しいものに変化していることがわかる．また，図 4a の解析結果から，TV データセットが周期性を含んでいることが分かる．

まとめると，ツイートストリームにおいては，新しく重要な単語が常に生じるため，分類モデルはこのような単語についても正しく出現確率を推定しなければならない．

3.2.4 単語 n グラムの効果

Twitter のツイートは 140 文字に制限されているため，他メディアの文書に比べて文書に含まれる単語数が少い．そこで，単語 n グラムも分類に有用な素性として利用可能か調査した．

まず，各日において 5 回以上出現した全ての単語 n グラム ($n = 1, 2, 3$) のカイ二乗統計量 $\max_c\{\chi^2_c\}$ を計算し，それから，n グラム全体のカイ二乗統計量が，n グラムの最初の単語のそれよりも大きくなるような n グラムの存在を確認した（図 5a）．例えば，MLB データセットにおいては，単語「white sox」は,「white」よりも，クラス#whitesox を特定するために有用である．また，$\max_c\{\chi^2_c\}$ の上位 200 語に含まれる 2 グラムと 3 グラムの割合は NPB，TV，MLB データセットに対してそれぞれ 37.6%, 52%, 18.2% であった（図 5b）．

以上のことから，通常の単語を全て独立に扱う bag-of-words に比べて，単語の並び順を考慮した分類モデルにすることでツイートの分類精度を向上できる可能性がある．

3.3 従来手法の適用

Salles ら [17]，Lebanon ら [11] がそれぞれ提案した，時間情報を考慮した多項モデルナイーブベイズ分類器を問題定義 **P1** に適用する場合，以下の通り定式化できる．

$$p(c|d_t) = p_K(c|t) \prod_{w_i \in W(d_t)} p_K(w_i|c,t), \tag{1}$$

$$p_K(c|t) = \frac{\sum_{\tau=1}^{t} K_h(t-\tau)[\![c_\tau = c]\!]}{\sum_{\tau=1}^{t} K_h(t-\tau)} \tag{2}$$

$$p_K(w_i|c,t) = \frac{\sum_{\tau=1}^{t} K_h(t-\tau) f_{c,\tau}(w_i)}{\sum_{\tau=1}^{t} \sum_{w \in d_\tau} K_h(t-\tau) f_{c,\tau}(w)} \tag{3}$$

ここで，$[\![\cdot]\!]$ は，述部が真の時 1 を，偽の時 0 を返す関数である．また，$W(d_t)$ は d_t に含まれる単語集合，$f_{c,\tau}(w)$ は d_t 内での単語 w の出現頻度を表す．$K_h(t-\tau)$ はカーネル関数であり，以下に例を示す．

$$K_h(t-\tau) = [\![t-\tau < h]\!] \tag{4}$$

$$K_h(t-\tau) = (1-(t-\tau)/h) \cdot [\![t-\tau < h]\!] \tag{5}$$

式 (4) のときサイズ h の時間窓による instance selection，式 (5) のとき instance weighting（Lebanon らの実験で最も良いカーネルと報告されたもの) [11]）となる．

図 6 に，NPB，TV，MLB データセットに対して式 (1) のモデルを適用した場合のマクロ平均 F 値を示す．なお，ナイーブベイズ分類器の素性選択を，文書集合 $\{d_\tau \,|\, t-h < \tau < t\}$ を基に，カイ二乗統計量 χ^2_c [18] を利用して $W(d_t) = \{w|\max_c\{\chi^2_c(w)\} > 30.0\}$ として実施した．

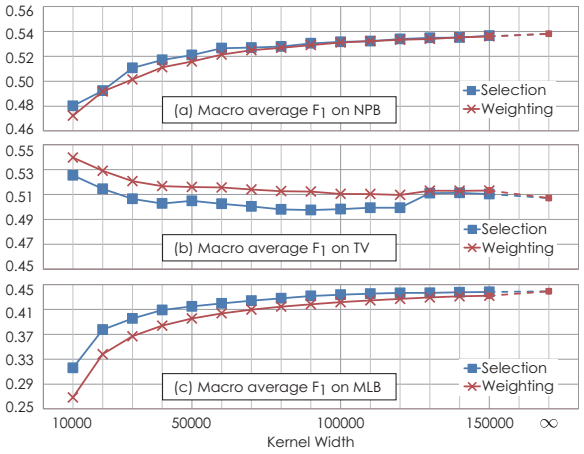

図 6　従来手法（多項モデルナイーブベイズ分類器を基にした instance selection & weighting 法; 式 (1)–(5)）のカーネル幅がマクロ平均 F 値に与える影響．

まず，TV データセットでは，時間情報を考慮することで分類精度が向上した．特に，カーネル幅を短くして最近の情報を重視した方が良い分類性能を実現したが，周期性を持つ変化の影響により，時間窓のサイズと分類精度は単調な比例関係になっていない．また，instance weighting の方が高い分類精度を実現しているのは，データに急激な変化が多く含まれるため，カーネル内のデータを等価値に扱う instance selection よりも，最近のデータに強い重みを付けた方が変化に適応し易いからである．

一方，NPB や MLB データセットでは時間情報を考慮しない場合が最も良い分類精度を実現した．これは，これらのデータセットに変化が含まれないためではなく，従来手法が一部の単語に関する急激な変化に適応できないためである．カーネル幅を短くして急激に出現確率が変化する単語に適応しようとすると，出現確率に時間変化の無い単語に関する十分な学習が行えないため，全体としては分類精度が悪化してしまう．

3.4 考　　察

以上の解析結果をまとめると，ツイートストリームにおいては，**[W1]** クラス分布の時間変化，**[W2]** 単語出現確率の時間変化，**[W3]** 新たに出現する単語，**[W4]** 各単語の時間変化のスケールの違い，**[W5]** 単語 n グラム，**[W6]** 周期性の変化，が考慮・適応すべき課題と考える．本研究では，このうち，課題 **W1** から **W5** について特に適応可能な手法を提案する．

4. 提 案 手 法

3. 章の解析結果を基に，新たな文書ストリーム分類モデル P-Switch（Word-Probability Swithching Model）を提案する．

4.1 概　　要

提案手法は，文書ストリーム $\{d_t|t = 1, 2, 3 \ldots\}$ が与えられたとき，文書 d_t を，式 (6) を最大化するクラス $\hat{c}_t$ に分類する．

$$p(c|d_t) = p(c|t) \prod_{w_i \in W(d_t)} p(w_i|c, w_{i-n+1}^{i-1}, t) \tag{6}$$

ここで，w_i は d に含まれる i 番目の単語，w_{i-n+1}^{i-1} は，d に含まれる $i-n+1$ 番目から $i-1$ 番目までの単語列，$W(d_t)$ は d_t に含まれる単語集合を表す．

式 (6) は，多項モデルナイーブベイズ分類器を基に，3 つの拡張点により時間変化への適応を可能にしたモデルと位置づけることができる．次節より，それぞれの拡張点について説明する．

4.2 クラス分布 $p(c|t)$

まず，課題 **W1** の解決のため，文書 d_t の真のクラスラベル c_t が与えられたとき，クラス分布 $p(c|t)$ を以下の様に更新する．

$$p(c|t) = (1-\gamma)p(c|t-1) + \gamma[\![c_t = c]\!] \qquad (7)$$

ここで，$[\![\cdot]\!]$ は，述部が真の時 1 を，偽の時 0 を返す関数である．式 (7) はより新しいデータを重視した指数加重移動平均 (Exponentially Weighted Moving Average: EWMA) に基づくクラス出現分布であり，γ は平滑化係数 ($0 < \gamma < 1$) である．

4.3 単語出現確率 $p(w_i|c,t)$

次に，**W2**〜**W4** の解決のため，単語出現確率の拡張を行う．

まず，我々のモデルは，時刻 t までに与えられたクラス c の文書を時間順に連結した文書 $\mathcal{D}_{c,t}$ (単語数 $|\mathcal{D}_{c,t}|$) を考える．

式 (8) に示す標準的な多項モデルナイーブベイズ分類器では，w_i のクラス c における出現確率を，最尤推定により行う．

$$p_{\mathrm{ML}}(w_i|c,t) = \frac{f_c(w_i)}{\sum_{w\in\mathcal{D}_{c,t}} f_c(w)} \qquad (8)$$

ここで，$f_c(w_i)$ はクラス c における w_i の出現頻度を表す．

さらに，提案手法では指数加重移動平均により，新しいデータを重視した単語の出現確率 $p_{\mathrm{EWMA}}(w_i|c,t)$ を推定する(注1)．

$$p_{\mathrm{EWMA}}(w_i|c,t) = \sum_{j\in J_c(w_i)} (1-\lambda)^{|\mathcal{D}_{c,t}|-j}\lambda \qquad (9)$$

$J_c(w_i) = \{j \,|\, \mathcal{D}_{c,t}[j] = w_i, 1 \leqq j \leqq |\mathcal{D}_{c,t}|\}$ は $\mathcal{D}_{c,t}$ 内の w_i の出現位置集合，λ は平滑化係数 ($0 < \lambda < 1$) を示す．

図 7 より，式 (9) によって推定される EWMA 値は，真の確率の変化を追従することが分かる．なお，p_{ML} は出現確率が安定している単語について正確に推定できるが，新しく出現した単語や，急激に出現頻度が増えた単語について，出現確率を過小推定しがちである．また，学習データ数が増えると変化への適応が鈍くなる問題もある．一方，p_{EWMA} は最近のデータに強い重みを置くため，推定確率が安定しない反面，急激な変化に適応し易い利点を持つ．

そこで，提案手法では $p_{\mathrm{EWMA}}(w_i|c,t)$ と $p_{\mathrm{ML}}(w_i|c,t)$ の値をモニタし，式 (10) のように，単語の出現確率が過去に与えられた文書集合から見て定常状態か否かによりこれら 2 つの値を切り替えて用いる．

$$p(w_i|c,t) = \begin{cases} p_{\mathrm{EWMA}} & \text{if } p_{\mathrm{EWMA}} > p_{\mathrm{ML}} + A\sigma_{c,t} \\ p_{\mathrm{ML}} & \text{otherwise} \end{cases}, \qquad (10)$$

$$\text{where} \quad \sigma_{c,t} = \sqrt{p_{\mathrm{ML}}(1-p_{\mathrm{ML}})}\sqrt{\lambda/(2-\lambda)} \qquad (11)$$

(注1)：式 (9) は，closed-form の EWMA 推定式である．

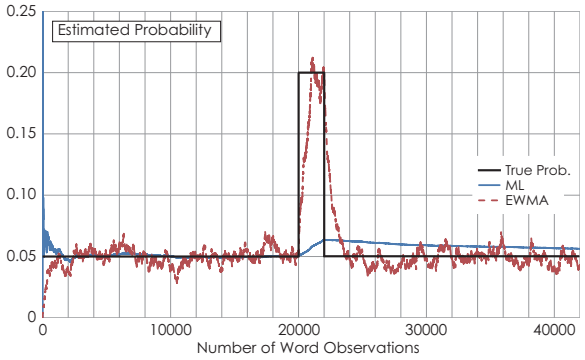

図 7　変化する真の単語出現確率に対する，最尤推定（式 (8)）と指数加重移動平均（Eq. (9), $\lambda = 0.002$）による推定シミュレーション．

式 (9)–(11) は Bernoulli EWMA 管理図 [19] による，$\mathcal{D}_{c,t}$ 内の単語列における w_i が出現するか否かの確率過程のモニタリングに相当し，式 (10) の A は管理限界係数である．

これら 2 つの確率の切り替えにより，我々のモデルは，出現確率が急激に上昇した単語への素早い適応と，出現確率が安定している単語の正確な学習の両方を同時に実現し，各単語の時間変化のスケールの違いに対応できる．

4.4 単語 n グラム出現確率 $p(w_i|c, w_{i-n+1}^{i-1}, t)$

最後に，課題 **W5** を解決すべく，提案手法ではナイーブベイズ分類器を n グラム言語モデルへ拡張する [20]．このとき，スムージングには，式 (12) に示す線形補間法の一種である Absolute Discounting 法 [21] を利用する．

$$p(w_i|c, w_{i-n+1}^{i-1}, t) = \frac{\max\{f_c(w_{i-n+1}^{i}) - \kappa, 0\}}{f_c(w_{i-n+1}^{i-1})} + \frac{\kappa \cdot r_c(w_{i-n+1}^{i-1})}{f_c(w_{i-n+1}^{i-1})} p(w_i|c, w_{i-n+2}^{i-1}, t) \qquad (12)$$

ここで，$r_c(w_{i-n+1}^{i-1})$ は，単語列 w_{i-n+1}^{i-1} の次に出現する異なり単語数である．この線形補間における最も低次の項が前節で説明した式 (10) で表される $p(w_i|c,t)$ となるため，式 (12) は時間効果を考慮したモデルとなる．

また，式 (8) のスムージングも Absolute Discounting 法により，

$$p_{\mathrm{ML}}(w_i|c,t) = \begin{cases} \dfrac{\kappa}{\sum_{w\in\mathcal{D}_{c,t}} f_c(w)} & \text{if } f_c(w_i) = 0 \\ \dfrac{f_c(w_i)}{\sum_{w\in\mathcal{D}_{c,t}} f_c(w)} & \text{otherwise} \end{cases} \qquad (13)$$

とし，式 (8) の代わりに式 (13) を用いる．

5. 実装: Word Suffix Array の利用

提案手法による分類は，入力文書 d_t に含まれる各単語 n グラム w_{i-n+1}^{i} について，各クラス c の文書を時間順に連結した文書 $\mathcal{D}_{c,t}$ における出現位置集合 $J_c(w_{i-n+1}^{i})$ を必要とする(注2)．これは全文検索のタスクであることから，各クラスの連結文書 $\mathcal{D}_{c,t}$ 毎に全文検索インデックスを構築することで学習処理を実現できる．本研究では，全文検索インデックスである接尾辞配

(注2)：$f_c(w_{i-n+1}^{i})$ と $r_c(w_{i-n+1}^{i})$ は，$J_c(w_{i-n+1}^{i})$ より取得可能である．

T = “ab#a#aa#a#ab#baa#aab#a#aa#baa#”

i	A	B	Word-aligned Suffix
1	4	2	a#aa#a#ab#baa#aab#a#aa#baa#
2	22	8	a#aa#baa#
3	9	4	a#ab#baa#aab#a#aa#baa#
4	6	3	aa#a#ab#baa#aab#a#aa#baa#
5	24	9	aa#baa#
6	18	7	aab#a#aa#baa#
7	1	1	ab#a#aa#a#ab#baa#aab#a#aa#baa#
8	11	5	ab#baa#aab#a#aa#baa#
9	27	10	baa#
10	14	6	baa#aab#a#aa#baa#

図 8　Word Suffix Array．‘#’ は単語境界．列 A は各接尾辞の開始（文字列）位置．列 B は各接尾辞の開始（単語）位置，

列（suffix array; SA）[22] のうち，単語 n グラムを効率的に扱うことが可能な単語接尾辞配列（word suffix array; WSA）[1] により提案手法を実装する．

5.1　Word Suffix Array の概要

WSA は図 8 に示す様に，入力テキスト $T[1,\ldots,l]$ が与えられたとき，各単語の開始位置のインデックス $i \in I$ に対する T の接尾辞 $S_i = T[i,\ldots,l]$ を辞書式順序に並び替えインデックス化する．WSA は，構築過程で通常の（単語境界を考慮しない）SA を用いることが特徴で，$\mathcal{D}_{c,t}$ の文字列長を l，単語数を k としたとき，構築に必要な時間と記憶容量はそれぞれ $O(l)$ と $O(k)$ となる（[23] に代表される，構築時間が文字列長に線形比例する SA 構築アルゴリズムを用いたとき）．単語 n グラムの潜在的な種類数は非常に多く $O(k^n)$ であるが，WSA では効率的に単語 n グラムを扱うことができる．また，単語の検索は二分検索で非常に高速に行うことが可能で，最も単純な実装の場合は単語の文字列長が m のとき $O(m \log k + f_c(w^i_{i-n+1}))$ の時間で検索可能である [22]．

5.2　実装詳細

WSA 構築のライブラリは，Fischer によるオリジナルの実装 [1] (注3) を利用し，内部で利用する SA 構築のライブラリには Nong らのアルゴリズム [23] を実装した sais(注4) を使用した．また，WSA の LCP（Longest Common Prefix; 接尾辞間の最大共通文字列）の構築には，Kasai らのアルゴリズム [24] を利用した．文字コードは UTF-8 を利用し，アルファベット数 $|\Sigma|$ は 256 種類とした（3 バイトの日本語 1 文字は，3 つのアルファベットで構成される）．図 9 に，WSA の構築時間に関する検証結果を示す．ツイートを連結した文字列から WSA を構築したところ，構築時間が文字列長に線形比例していることがわかる．

提案手法は，各クラス毎に WSA を構築する．WSA の逐次的な更新はサポートされていないため(注5)．新しい訓練文書 (d_t, c_t) が与えられるたびにクラス c_t の再構築を行う．学習時間を減らすためには，文書ストリームを期間別に分割し，古い文書から構築された WSA は更新せず保有し，最新期間の文書のみから WSA を再構築することが有効である．その際は，累積単語数のオフセット値を分割した WSA ごとに保存し，各

(注3)：http://ab.inf.uni-tuebingen.de/people/fischer/wordSA.tgz

(注4)：http://sites.google.com/site/yuta256/sais

(注5)：接尾辞配列については逐次更新アルゴリズムが提案されている [25]．

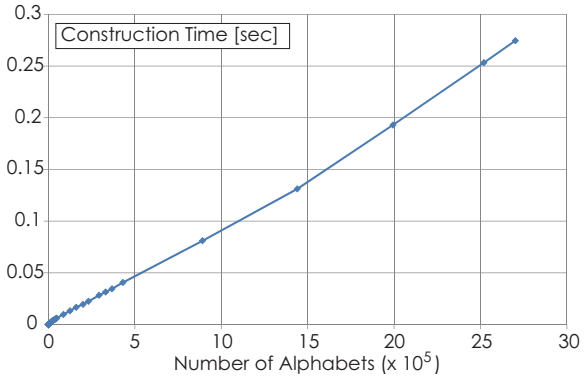

図 9　NPB データセットの#dragons クラスのツイートを連結した文字列についての Word Suffix Array の構築時間．CPU: Xeon X5680 3.33GHz, Memory: 192GB.

WSA からの w^i_{i-n+1} の検索結果にオフセット値を加えることで，$J_c(w^i_{i-n+1})$ を取得する．なお，各 WSA の学習・分類処理は並列・分散化可能である．

単語列 w（文字列長 m）の検索には，3 つのヒューリスティック: （1）検索結果のキャッシュ（2）最初のアルファベットに関する WSA の検索範囲の記憶 [22] （3）文字比較回数の削減 [22] を加えた．検索結果のキャッシュは，n グラムの単語列を検索する際，$n-1$ グラムの単語列の検索結果を利用すれば，大幅に WSA 中の検索範囲を狭めることができる．この検索の最悪のケースの計算量は $O(m \log k + f_c(w))$ となるが，Ferragina らは，2 と 3 のヒューリスティックを加えた $O(m \log k + f_c(w))$ のアルゴリズムの実際の計算量が，他に提案されている $O(m + \log k + f_c(w))$ [22] や $O(m|\Sigma| + f_c(w))$ [26] のアルゴリズムよりも少なくなることを報告している [1]．

6.　評価実験

表 1 に示すデータセットを用いて，問題定義 **P1**（2.2 節参照）における提案手法のツイート話題分類精度を評価した．

本章の実験において，式 (6) の単語集合 $W(d_t)$ は，カイ二乗統計量 χ^2_c [18] を利用して $W(d_t) = \{w | \max_c\{\chi^2_c(w)\} > 30.0\}$ として求めた．また，式 (13) のスムージングパラメータは $\kappa = 0.9$ と設定した．

6.1　各拡張点の効果

提案手法 P-Switch は，多項モデルナイーブベイズ分類器を基に，3 つの拡張点を加えて時間変化への適応を可能にしたモデルである．本節では，全データを学習・テストした後の累積分類精度とマクロ平均 F 値を用いて，各拡張点の効果を示す．

6.1.1　クラス分布 $p(c|t)$

初めに，時を考慮したクラス分布 $p(c|t)$ の効果を見るため，式 (14) に示す多項ナイーブベイズ分類器 MNB:

$$p(c|d_t) = p(c) \prod_{w_i \in W(d_t)} p_{\mathrm{ML}}(w_i|c,t) \qquad (14)$$

から，クラス分布を式 (7) へ拡張したモデル Proposal1:

$$p(c|d_t) = p(c|t) \prod_{w_i \in W(d_t)} p_{\mathrm{ML}}(w_i|c,t) \qquad (15)$$

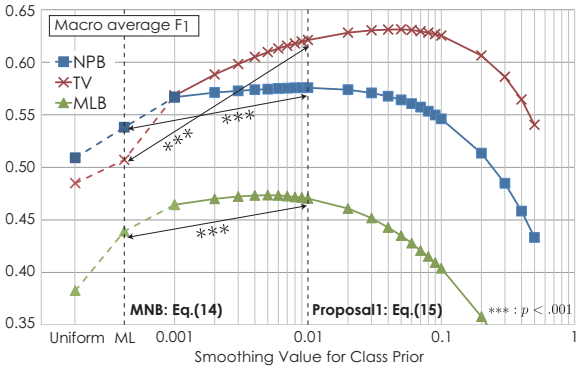

図 10　時を考慮したクラス分布 $p(c|t)$ への拡張効果．式 (7) の γ の値を 0.001 から 0.5 まで変化させた．Uniform/ML はクラス分布を一様分布/最尤推定とした場合をそれぞれ示す．

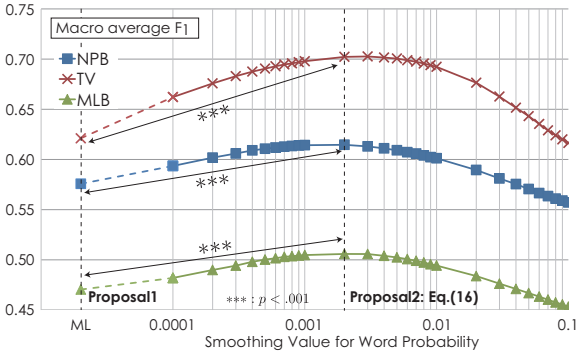

図 11　時を考慮した単語出現確率 $p(w_i|c,t)$ への拡張効果．式 (9) の λ の値を 0.0001 から 0.1 まで変化させた．ML は単語出現確率を最尤推定（式 (8)）とした場合を示す．$\gamma = 0.01$，$A = 0.5$.

について評価した．図 10 に示す様に，全てのデータセットでも，多項ナイーブベイズ分類器で時を考慮せずに $p(c)$ を最尤推定した場合（ML）や，一様分布とした場合（Uniform）に比べて，$p(c|t)$ への拡張により分類精度が有意に向上した（McNemar 検定; $p < .001$）．式 (1) の従来手法（instance selection/weighting 法）に比べて，式 (15) のモデルはクラス分布のみ最近のデータを重視したものに相当し，クラス分布と単語出現確率を同様の時間効果で扱わない点が分類に良い効果を与えている．

6.1.2　単語出現確率 $p(w_i|c,t)$

次に，時を考慮した単語出現確率 $p(w_i|c,t)$ の効果を見るため，式 (15) のモデルから，単語出現確率を式 (10) へ拡張したモデル Proposal2:

$$p(c|d_t) = p(c|t) \prod_{w_i \in W(d_t)} p(w_i|c,t) \qquad (16)$$

について評価した．なお，本実験では $\gamma = 0.01$，$A = 0.5$ と設定した．図 10 に示す様に，全てのデータセットでも，時を考慮せずに $p(w|c)$ を最尤推定した場合（ML，式 (15) のモデル）に比べて，$p(w_i|c,t)$ への拡張により分類度が有意に向上した（McNemar 検定; $p < .001$）．すなわち，短期的な傾向に基づいた $p_{\mathrm{EWMA}}(w_i|c,t)$ と，長期的な傾向に基づいた $p_{\mathrm{ML}}(w_i|c,t)$ を単語の出現状況に応じて切り替えて用いることで，各単語で異なる特性を持つ時間変化への適応が可能になった．

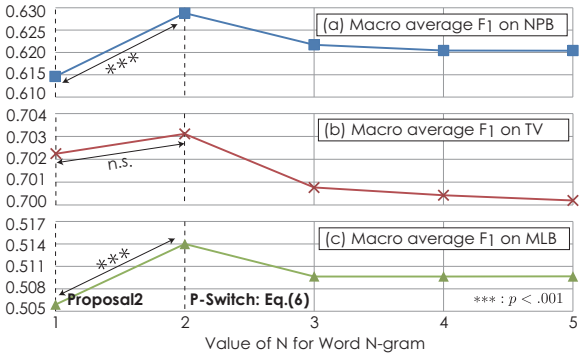

図 12　単語 n グラム出現確率 $p(w_i|c, w_{i-n+1}^{i-1}, t)$ への拡張効果．$\gamma = 0.01$，$\lambda = 0.002$，$A = 0.5$.

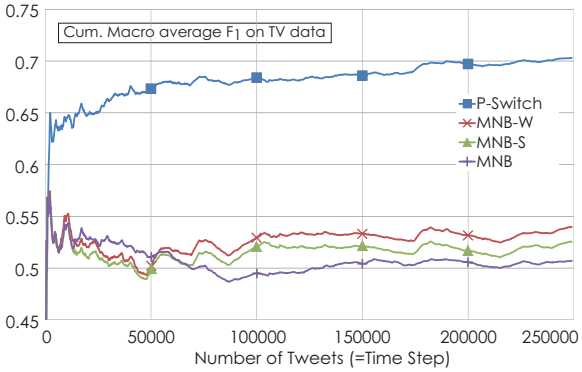

図 13　TV データセットにおける累積分類精度とマクロ平均 F 値ツイート数の推移．$\gamma = 0.01, \lambda = 0.002, A = 0.5, n = 2, h = 10000$.

6.2　単語 n グラム出現確率 $p(w_i|c, w_{i-n+1}^{i-1}, t)$

最後に，式 (16) のモデルから単語出現確率を式 (12) へ拡張した提案手法 P-Switch（式 (6)）について評価した．なお，本実験では $\gamma = 0.01$，$\lambda = 0.002$，$A = 0.5$ と設定した．図 12 に示す様に，全てのデータセットでも $n = 2$ の時に最も良い分類精度を実現した（NPB と MLB において McNemar 検定の有意差あり; $p < .001$）．n の値を大きくすることで分離精度が落ちるのは，高次の n グラムの出現頻度が少ないために確率推定の精度が下がるためである．

6.3　従来手法との比較

提案手法と従来手法の比較を行うため，多項モデルナイーブベイズ分類器（MNB: 式 (14)），Salles ら・Lebanon らによる時間情報を考慮した多項モデルナイーブベイズ分類器（MNB-W: 式 (1), (5)，MNB-S: 式 (1), (4)；詳細は 3.3 節と, [11], [16] を参照）との比較を行った．提案手法のパラメータは $\gamma = 0.01$，$\lambda = 0.002$，$A = 0.5$，$n = 2$ と設定し，MNB-W と MNB-S のカーネル幅は $h = 10000$ と設定（TV データセットにおける最良値）した．なお，図 6 に示す様に，NPB と MLB データセットにおける h の最適値は ∞（通常の MNB に相当）であった．表 2 に全手法の分類精度とマクロ平均 F 値を示す．

まず，NPB データセットにおいては，先に示した図 6 の解析結果にも示す様に，時間効果を考慮する MNB-W，MNB-S の類精度が悪化してしまう．これは，従来手法では単語単位で変化適応できないため，カーネル幅を狭くして一部の単語の急激

表 2　各データセットにおける精度とマクロ平均 F 値．太字は，同一データセット内で他手法に比べ有意に高い性能を実現したことを示す（McNemar's test; $p < .001$）．

	NPB		TV		MLB	
Methods	Accuracy [%]	MacroF$_1$ [%]	Accuracy [%]	MacroF$_1$ [%]	Accuracy [%]	MacroF$_1$ [%]
MNB	55.87	53.81	50.61	50.71	48.29	43.92
MNB-W	50.57 (−5.396)	47.24 (−6.573)	54.77 (+4.160)	53.96 (+3.256)	33.82 (−14.47)	26.85 (−17.06)
MNB-S	51.57 (−4.298)	48.02 (−5.785)	53.26 (+2.649)	52.53 (+1.822)	38.29 (−10.00)	31.60 (−12.32)
Proposal1	60.52 (+4.652)	56.61 (+2.805)	62.82 (+12.21)	62.11 (+11.40)	56.05 (+7.751)	47.03 (+3.119)
Proposal2	65.87 (+9.998)	61.46 (+7.654)	**70.77** (+20.16)	**70.22** (+19.52)	60.26 (+11.97)	50.59 (+6.671)
P-Switch	**66.90** (+11.04)	**62.88** (+9.072)	**70.80** (+20.19)	**70.31** (+19.60)	**60.86** (+12.56)	**51.40** (+7.480)

な変化に対応するよりも，十分な量のデータから学習した方が全体として分類精度が良くなるためである．

次に，TV データセットでは MNB-W，MNB-S の分類精度は MNB に比べて向上したが，逐次的な学習ツイート数の増加による分類精度の向上効果はなかった（図 13）．これはカーネル幅が短いため激しい変化には対応できるが，出現確率に時間変化の無い単語に関する学習が十分に行えないためである．

これらの従来手法の問題に対して，提案手法は全てのデータセットでも同じパラメータ値を用いて高い分類精度を実現し，かつ，学習ツイート数の増加に応じて分類精度が向上している．

7. おわりに

本論文では，Twitter のツイートに代表される，時間経過と共にデータの性質が変化する文書ストリームに対する分類モデル P-Switch（Word Probability Switching Model）を提案した．

我々は，実際のツイートストリームを解析することで文書ストリーム分類における課題を明らかにし，特に，データの性質が単語ごとに異なるスケールで変化する問題の解決に取り組んだ．提案手法では，各単語の出現確率過程をモニタリングし，長期的なデータに基付く推定と，最近のデータを重視した推定を単語毎に切り替えることで，文書単位で文書の選択や重み付けを行う従来手法に比べて有意に高い分類精度を実現した．

また，本論文では，単語接尾辞配列（word suffix array; WSA）による実装方法を示した．WSA の利用により学習・分類処理における単語 n グラムの時間影響を効率的に扱えるようになる．接尾辞配列の構築により文書分類に有効な素性集合を発見してバッチ学習に利用する研究は従来にも存在するが [27]，時間変化を考慮した文書分類の学習処理を全文検索器の構築によって実現するアプローチは本研究が初めてであり，学術的意義は非常に大きいと考える．

文　献

[1] P. Ferragina, and J. Fischer, "Suffix arrays on words," CPM, pp.328-339, 2007.

[2] G. Widmer, and M. Kubat, "Learning in the presence of concept drift and hidden contexts," Mach. Learn., vol.23, no.1, pp.69-101, 1996.

[3] A. Tsymbal, "The problem of concept drift: definitions and related work," Tech. Rep. TCD-CS-2004-15, Trinity College Dublin, 2004.

[4] S.J. Delany, P. Cunningham, and B. Smyth, "ECUE: A spam filter that uses machine learning to track concept drift," ECAI, p.627, 2006.

[5] B. Wenerstrom, and C. Giraud-Carrier, "Temporal data mining in dynamic feature spaces," ICDM, pp.1141–1145, 2006.

[6] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, and I.P. Vlahavas, "An adaptive personalized news dissemination system," J. Intell. Inf. Syst., vol.32, no.2, pp.191-212, 2009.

[7] E.S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I.P. Vlahavas, "Dealing with concept drift and class imbalance in multi-label stream classification," IJCAI, pp.1583-1588, 2011.

[8] R. Klinkenberg, and T. Joachims, "Detecting concept drift with support vector machines," ICML, pp.487-494, 2000.

[9] R. Klinkenberg, "Learning drifting concepts: example selection vs. example weighting," Intell. Data Anal., vol.8, pp.281-300, 2004.

[10] M. Scholz, and R. Klinkenberg, "Boosting classifiers for drifting concepts," Intell. Data Anal., vol.11, no.1, pp.3-28, 2007.

[11] G. Lebanon, and Y. Zhao, "Local likelihood modeling of temporal text streams," ICML, pp.552-559, 2008.

[12] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking recurring contexts using ensemble classifiers: an application to email filtering," Knowl. Inf. Syst., vol.22, pp.371-391, 2010.

[13] G. Forman, "Tackling concept drift by temporal inductive transfer," SIGIR, pp.252-259, 2006.

[14] F. Mourão, L.C. da Rocha, R.B. Araújo, T. Couto, M.A. Gonçalves, and W.M. Jr., "Understanding temporal aspects in document classification," WSDM, pp.159-170, 2008.

[15] L.C. da Rocha, F. Mourão, A.M. Pereira, M.A. Gonçalves, and W. Meira Jr., "Exploiting temporal contexts in text classification," CIKM, pp.243-252, 2008.

[16] T. Salles, L.C. da Rocha, F. Mourão, G.L. Pappa, L. Cunha, M.A. Gonçalves, and W. Meira Jr., "Automatic document classification temporally robust," JIDM, vol.1, no.2, pp.199-212, 2010.

[17] T. Salles, L.C. da Rocha, G.L. Pappa, F. Mourão, W. Meira Jr., and M.A. Gonçalves, "Temporally-aware algorithms for document classification," SIGIR, pp.307-314, 2010.

[18] Y. Yang, and J.O. Pedersen, "A comparative study on feature selection in text categorization," ICML, pp.412-420, 1997.

[19] S.E. Somerville, D.C. Montgomery, and G.C. Runger, "Filtering and smoothing methods for mixed particle count distributions," Int. Journal of Prod. Res., vol.40, no.13, pp.2991-3013, 2002.

[20] F. Peng, D. Schuurmans, and S. Wang, "Augmenting naive bayes classifiers with statistical language models," Inf. Retr., vol.7, no.3-4, pp.317-345, 2004.

[21] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependences in stochastic language modeling," Computer Speech & Language, vol.8, pp.1-38, 1994.

[22] U. Manber, and E.W. Myers, "Suffix arrays: A new method for online string searches," SIAM J. Comput., vol.22, no.5, pp.935-948, 1993.

[23] G. Nong, S. Zhang, and W.H. Chan, "Linear suffix array construction by almost pure induced-sorting," DCC, pp.193-202, 2009.

[24] T. Kasai, G. Lee, H. Arimura, S. Arikawa, and K. Park, "Linear-time longest-common-prefix computation in suffix arrays and its applications," CPM, pp.181-192, 2001.

[25] M. Salson, T. Lecroq, M. Léonard, and L. Mouchard, "Dynamic extended suffix arrays," J. Discrete Algorithms, vol.8, no.2, pp.241-257, 2010.

[26] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch, "Replacing suffix trees with enhanced suffix arrays," J. Discrete Algorithms, vol.2, no.1, pp.53-86, 2004.

[27] D. Okanohara, and J. ichi Tsujii, "Text categorization with all substring features," SDM, pp.838-846, 2009.