# テキスト中のイベントの生起時間帯判定

野呂太一[†] 乾 孝司[††]
高村大也[†††] 奥村 学[†††]

本論文では，ブログテキスト中に記述されたイベントが，実世界において朝，昼，夕，夜のどの時間帯で生起したかを自動判定するアルゴリズムを提案する．生起時間帯を判定する直接的な情報として，「午後 3 時に〜した」等の明示的な時間表現が考えられるが，ブログテキストでは，明示的な時間表現が現れにくい．そこで，本研究では，明示的な時間表現の代わりに，イベントの生起時間帯を連想させる語（「出勤」，「花火」等）の情報を利用する．イベントの生起時間帯を連想させる語集合を人手のみで収集することは表現の多さから現実的にほぼ不可能な作業である．そのため，提案手法では，ブートストラップ的に，イベントの生起時間帯の学習と並行して同時に，イベントの生起時間帯を連想させる語を自動獲得する．

# Time Period Identification of Events in Text

Taichi Noro,[†] Takashi Inui,[††] Hiroya Takamura[†††]
and Manabu Okumura[†††]

We propose a machine learning-based method for identifying when each event in weblog texts occurs: morning, daytime, evening, or night. Earlier study analyzed only explicit temporal expressions for events and mapped them on time-line in newswire texts. However, other texts such as weblogs contain few explicit temporal expressions. We therefore use various implicit temporal expressions extracted automatically. Specifically, we adopt naive bayes classifiers backed up with the EM algorithm, and support vector machines.

## 1. はじめに

近年，Web の発達とともに電子化されたテキストの量は増加を続けている．この膨大な電子化テキストの中には，多くの企業や人々にとって有益な情報が多く含まれており，これらの情報を自動で収集，整理する技術が，情報検索や情報抽出等の研究分野で活発に検討されている．

このようなテキスト群の中には，ある出来事，イベントについて記述されたものも少なくなく，このようなテキスト群をイベントの生起時間という観点から整理する研究がある[4),8)]．たとえば，関連のある一連のイベントの内容を，逐次，ニュース記事として発表する場合，ニュース記事の発表順は，実世界でまさにイベントが生起した時間帯とは必ずしも同じ順序ではない．このような状況においてニュース記事を読むことで事実の理解を深める際は，実世界でのイベントの生起順にニュース記事の内容を自動的に並べ替える技術が役立つと考えられる．

イベントの生起時間に従ってテキスト群を整理する従来研究の多くは上記のような応用を見据えており，必然的に，ニュース記事を対象としてきた．ニュース記事では，「3 日午後 3 時ごろ」のように，言語表現として明示的に生起時間に関する情報を与えるという特徴があり，従来研究では，このような明示的な時間表現を捕捉することが技術の基本的な柱となっている．

しかし，イベントを記述しているテキストのすべてがニュース記事のように明示的な時間表現を有しているわけではない．たとえば Web 日記，ブログ等（以下まとめてブログと呼ぶ）も，ニュース記事同様にイベントを記したテキストといえる．ただし，ブログは，ニュース記事とは違って個人が日々の生活内でのイベントを記述することが多く，かつ，新聞社等が発表するニュース記事のように真実を分かりやすく報道するための報知的な性格を持つ媒体ではない等の理由から，

† 富士通株式会社
Fujitsu Limited
†† 東京工業大学統合研究院
Integrated Research Institute, Tokyo Institute of Technology
††† 東京工業大学精密工学研究所
Precision and Intelligence Laboratory, Tokyo Institute of Technology

イベントの生起時間を明示的に示すことは稀である．

ブログは，上記のような性質を持っており，既存手法では時間情報の解析が困難である．しかし，近年急速に利用者を増やしており，企業における Web マーケティングや，消費者の購買活動時の情報収集等の情報源として現在注目を集めている．たとえば，ブログテキスト内に記述されたイベントの，実世界における生起時間を自動判定することができれば，「人々は，朝，会社（学校）に行く前に何をしているのか？」や「昼休みの時間に購入されやすいお菓子は何か？」といった消費者動向分析等，さまざまな応用の実現に結びつくと期待できる．

以上のような背景をふまえ，本論文では，ブログテキストを対象とし，ブログテキスト中に記述されたイベントの生起時間帯を判定する新しい手法を提案する．本論文では特に，1 日（24 時間）内の活動に注目し，ブログテキスト中の各文に対して，その文に記述されたイベントの生起時間帯を，朝，昼，夕，夜の粒度で判定する手法を提案する．もちろん，生起時間帯をどのような粒度で判定するかは想定する応用に依存するが，先に述べたような消費者動向分析では，朝，昼，夕，夜という比較的粗い粒度の情報でも十分に価値があると考えられる．

先に述べたように，ブログテキスト中には明示的な時間表現があまり現れないため，既存手法を用いてブログテキストに記述されたイベントの生起時間帯を判定することは困難である．この問題に対応するために，本研究では，ある時間帯に偏って生起するイベントや，そのイベントにおける構成要素を表す語に着目する．たとえば，一般常識的に，「通勤」は朝のイベントであり，「花火」は夜のイベントである．また，「トースト」は朝食イベントに登場しやすい構成要素である．このような，ある時間帯に偏って現れ，そのイベントの生起時間帯が連想されやすい（例：「通勤」– 朝，「花火」– 夜）語の情報を活用する．以下，本論文では，このような語を**時間帯連想語**と呼ぶ．

ただし，時間帯連想語を人手であらかじめ収集，把握しておくことは，表現の多さから現実的にほぼ不可能である．そのため，時間帯連想語の獲得とイベントの生起時間帯判定を，機械学習手法を用いて自動的に並行して進める手法を提案する．提案手法は，少量のブログテキスト（内の各文）に人手で時間帯情報を付与した教師ありデータと，時間帯情報を付与していない大量の教師なしデータの情報を両方用いる半教師あり学習アルゴリズムによって実現されている．

本論文の構成は以下のとおりである．まず，2 章で関連研究について述べる．3 章では，本研究で利用するブログデータについて説明する．その後，4 章で提案手法について述べ，5 章で評価実験の結果を述べる．最後に 6 章で本論文をまとめる．

## 2. 関 連 研 究

Setzer ら[8] や Mani ら[4] はニュース記事中の時間情報を解析するための取り組みとして，イベント，および時間情報表現へのアノテーションを研究目的としている．これによりイベントの生起時間の決定を可能にするとともに，複数イベント間の相対的な生起順序関係に着目し，イベントの整列を行う．小倉ら[14] は，ニュース記事を対象とし，1 文章中のイベントの時系列化を目指した．特に，複数イベント間の前後関係を求める時間推論に焦点を置いた．以上 3 つの先行研究は，ニュース記事を対象としたもので，明示的な時間情報がある程度含まれることが前提となっており，本研究とは方向性が異なるものである．

本研究と類似した目的を持つものに，土屋ら[12] の研究がある．これは，あらかじめ用意した時間判断知識のデータベースをもとに，未知語（時間判断データベースに存在しないもの）から連想される時間を導き出すものである．辞書の見出し語と説明文の関係を利用し，既知語と未知語の関連度を計算して，未知語から連想される時間情報を取得している．これに対し，本研究では，人々の行動のデータ（ブログ）から時間情報を取得する．本研究では，土屋らのような辞書にあたる外的知識資源は必要としない．また，テキスト中のイベントを抽出する課題と関連したものでは，倉島ら[10] の研究がある．これはブログから個人が書いた街での体験を抽出するもので，正規表現パターンを用意して対象となる文を獲得している．しかし，これは解析対象を地名・ランドマークの出現する文に絞り，文末の単語を，サ変名詞と行為を意味する動詞に絞って抽出している．一方，本研究では，イベントの種類を限定せず，抽出する対象はより広いものである．

## 3. ブログデータ

本研究ではブログを解析対象とする．本章では，本研究で利用するブログデータについて説明する．このデータは，以降で説明する実験において，訓練および評価用データとして使用する．

本研究で利用するブログデータは南野ら[13] が収集したブログエントリ集合の一部であり，各ブログエントリを 1 文ごとに自動的に切り出したものを使用している．ブログデータは文末に句点が記述されないこ

a. 羽田に行きました.
b. ひどく痛むので近くの整形外科に.
c. ようやく正月だなぁと感じた.
d. スイカ割りをしたけど割れなかった.

**図 1** 〈event=1〉を付与した文の例
Fig. 1 Samples labeled with ⟨event=1⟩ tag.

a. 生姜紅茶を，一日一杯は飲んでます.《習慣》
b. ほんとにかわっちゃったの？《台詞》
c. ご無沙汰しております.《挨拶》
d. 可能性がないなら連れて帰りたい.《主張》
e. おいしいー！《感想》
f. 20 名さまにプレゼント！《広告の文句》

**図 2** 〈event=0〉を付与した文の例
Fig. 2 Samples labeled with ⟨event=0⟩ tag.

とも多く，誤りなく完全に文分割を実現することは難しい．本研究では，句点や HTML タグの情報を用いた簡単な規則に基づいてブログエントリを文に分割した．自動分割の後，文集合から無作為に文を抽出し，人手で文分割精度を検証したところ，およそ 95% であった．

### 3.1 タ グ

本研究を進めるうえで，ブログエントリ内の各文に〈event〉，〈time slot〉の 2 種類のタグを付与した．以下，それぞれについて説明する．

#### 3.1.1 〈event〉タグ

〈event〉タグは，文にイベントが記述されているか否かに関する情報を 0/1 の 2 値で保持する．文にイベントが記述されているときは〈event=1〉を付与し，イベントが記述されていないときは〈event=0〉を付与した．

〈event=1〉を付与した文の例を**図 1** に示す．(b) は，文末で「行った」が省略されている．また，(c) のように心情が変化したこともイベントとした．(d) は，「割れる」というイベントは起きなかったが，「割れない」というイベントが生起したと考え，イベントとして扱う．

続いて〈event=0〉である文の例を**図 2** に示す．イベントを表していない文には，何らかの説明をしていたり，主張や感想を述べていたりするもの等が多く含まれる．

#### 3.1.2 〈time slot〉タグ

〈time slot〉タグは，イベントが生起した時間帯を “朝”，“昼”，“夕”，“夜”，および時間帯に関する記述がなく時間帯が不明な “無” の 5 値の情報を保持する．ただし，〈event=0〉の文には時間帯生起情報を保持させる動機付けが存在しないことから，〈time slot〉タグは〈event=1〉の文にのみ付与した．以下，説明の便宜上，〈time slot=朝〉を単に〈朝〉で示すことがある．〈昼〉，〈夕〉，〈夜〉，〈無〉も同様である．また，タグに関して言及する場合と同様に単にイベントの時間帯に関して言及する場合も三角括弧表現（〈朝〉等）を用いる．また，〈朝〉，〈昼〉，〈夕〉，〈夜〉の 4 種類の時間帯に注目する際に〈朝〉〜〈夜〉という表記を用いることがある．

タグ付与の際に，目安として，各時間帯に以下の定義を与えた．

**朝：**04:00〜10:59，早朝から午前中，朝食
**昼：**11:00〜15:59，昼から夕方の前，昼食
**夕：**16:00〜17:59，夕方から日没前
**夜：**18:00〜03:59，日没後から夜明け，夕食

各時間帯は，対象文内の情報のみから判断可能なものと，文内情報のみでは判断ができないが，対象文の前後の文脈情報を見ることによって判断可能になるものがある．前者の例を (1) に示す．

(1) a. 朝から自転車で郵便局へ行く．〈朝〉
b. 昼休みに眠気覚ましのガムを買った．〈昼〉
c. 今日は 16 時過ぎに帰路につく．〈夕〉
d. 夕ご飯の後で友達と花火をした．〈夜〉

続いて，後者の例を下の (2) に示す． (2-a2) が対象文の前後の文脈情報を見ることによって判断可能になる文である．ここで， (2) の 2 つの文はブログエントリ内で連続して現れている．この場合，まず (2-a1) は文内の情報から〈朝〉と判断できる．次に， (2-a2) は，その文単体では生起時間帯の判断が困難であるが，(2-a1) に後続して現れており，実世界でも連続したイベントであると高い確率で推測できることから〈朝〉だと判断する．

(2) a1 朝 9:00 に自転車で郵便局へ行く．〈朝〉
a2 郵便局の帰りに某ショップへ．〈朝〉

次の (3) のように，1 文で複数のイベントを記述している文も存在する．このような場合は，文の末尾に記述されているイベントのみに注目して時間帯を判断した． (3) の場合は「帰る」というイベントに対して判断を行い，〈夕〉が付与される．ただし，「朝から晩まで働いた」のように，イベント自体が複数の時間帯にまたがっている場合は〈無〉とした．

(3) 今日は朝学校に行って，昼には弁当を食べ，夕方帰った．〈夕〉

### 3.2 集 計

267 人の異なる書き手のブログエントリに上記のタグを付与した．タグづけを行ったブログエントリ数は

**表 1** 〈event〉タグの内訳
Table 1 Number of sentences labeled with ⟨event⟩ tag.

| event | 数 |
|---|---|
| 1 | 14,220 |
| 0 | 56,555 |
| 計 | 70,775 |

**表 2** 〈time slot〉タグの内訳
Table 2 Number of sentences labeled with ⟨time slot⟩ tag.

| time slot | 数 |
|---|---|
| 朝 | 711 |
| 昼 | 599 |
| 夕 | 207 |
| 夜 | 1,035 |
| 無 | 11,668 |
| 計 | 14,220 |

7,413 であり，総文数は 70,775 である．

各タグの内訳を**表 1**，**表 2** に示す．表 1 から，⟨event=1⟩ の文と ⟨event=0⟩ の文の数は偏っており，⟨event=1⟩ の文の割合が少なく ⟨event=0⟩ の文が多いことが分かる．また，表 2 から，⟨time slot⟩ タグについても〈無〉がほかに比べてかなり多いことが分かる．この時間帯の偏りは，時間帯判定を実現する際に無視できない事柄である．この偏りについては 4.4 節で再び言及し，4.5 節で偏りへの対処法を述べる．

## 4. 提 案 手 法

本章では，ブログテキストを解析対象とし，そのテキスト内に記述されたイベントの生起時間帯を判定する手法について述べる．提案手法は，「朝食」や「トースト」，「通勤」といった，ある時間帯に偏って生起するイベントやそのイベントにおける構成要素となり，イベントの生起時間帯が連想されやすい語（時間帯連想語）の情報に基づいて生起時間帯を判定する．

仮に「朝食」という単語が，それを含む文を〈朝〉と判定する強い手掛かり，つまり時間帯連想語であることが分かっているとする．これによって，たとえば「朝食にトーストを食べた」という文が〈朝〉であることが分かり，さらにこの文から，「トースト」が〈朝〉の連想語である可能性があることが分かる．このような考え方を繰り返すことにより，ブートストラップ的に時間帯連想語が獲得でき，同時に文を正しく分類できるようになると考えられる．

この考えを実現するために，イベントの生起時間帯のタグが付けられたブログデータを種として，時間帯のタグが付いていない大量のブログデータをあわせて利用する，半教師付き学習が適用できる．そこで，教師付き学習手法のナイーブベイズ分類器（Naive Bayes classifiers）[5] を Expectation Maximization（以下，EM）アルゴリズム[1] で補強する半教師付き学習法を適用する．分類アルゴリズムとしてナイーブベイズ分類器を用いたのは，ナイーブベイズ分類器を EM アルゴリズムと組み合わせることにより文書分類で高い性能を発揮することが Nigam ら[6] によって示されているからである．

### 4.1 ナイーブベイズ分類器による時間帯判定

まずナイーブベイズ分類器の一種である多項モデルについて説明する．多項モデルでは，カテゴリ $c$ が与えられたときに事例 $x$ が生起する確率は，

$$P(x|c,\theta) = P(|x|)|x|! \prod_w \frac{P(w|c)^{N(w,x)}}{N(w,x)} \quad (1)$$

となる．ナイーブベイズ分類器を文の時間帯分類に適用する場合は，各文が事例 $x$ に相当し，$c \in \{$朝，昼，夕，夜，無$\}$ となる．$P(|x|)$ は，長さ（単語数）が $|x|$ の文が生起する確率であり，$N(w,x)$ は文 $x$ 中での素性（単語）$w$ の出現頻度である．多項モデルでは，文の生起は全語彙の中から単語を 1 つ選び出す試行の繰返しとしてモデル化されている．

### 4.2 ナイーブベイズ分類器と EM アルゴリズムの組合せ

EM アルゴリズムはいくつかの変数（隠れ変数と呼ばれている）が観測できない状況で，モデルを最尤推定もしくは事後確率最大化推定する手法である．Nigam らはナイーブベイズ分類器と EM アルゴリズムを組み合わせることを提案している．ナイーブベイズ・モデルの式において，関係ない要素を無視すると，次の式を得る：

$$P(x|c,\theta) \propto \prod_w P(w|c)^{N(w,x)}, \quad (2)$$

$$P(x|\theta) \propto \sum_c P(c) \prod_w P(w|c)^{N(w,x)}. \quad (3)$$

以降，モデルのパラメータ群をまとめて $\theta$ と表す．

$c$ を隠れ変数とし，ディリクレ分布をパラメータの事前分布とすると，対数尤度の隠れ変数に関する期待値（$Q$ 関数）は次のように定義できる：

$$Q(\theta|\bar{\theta}) = \log P(\theta) + \sum_{x \in D} \sum_c P(c|x,\bar{\theta}) \times \log \left( P(c) \prod_w P(w|c)^{N(w,x)} \right) \quad (4)$$

ここで，

$$P(\theta) \propto \prod_c P(c)^{\alpha-1} \prod_w P(w|c)^{\alpha-1}$$

であり，また，$\alpha$ はハイパーパラメータ，$D$ はモデルの推定に用いられる事例の集合である．

この $Q$ 関数より，次の EM 計算式が得られる：

E-ステップ：

$$P(c|x,\bar{\theta}) = \frac{P(c|\bar{\theta})P(x|c,\bar{\theta})}{\sum_c P(c|\bar{\theta})P(x|c,\bar{\theta})}, \tag{5}$$

M-ステップ：

$$P(c) = \frac{(\alpha-1) + \sum_{x\in D} P(c|x,\bar{\theta})}{(\alpha-1)|C| + |D|}, \tag{6}$$

$$P(w|c) = \frac{(\alpha-1) + \sum_{x\in D} P(c|x,\bar{\theta})N(w,x)}{(\alpha-1)|W| + \sum_w \sum_{x\in D} P(c|x,\bar{\theta})N(w,x)}. \tag{7}$$

ここで $|C|$ はカテゴリ数，$|W|$ は素性の種類数を表す．ラベル付き事例については式 (5) は使用されない．その代わりに，$c$ が事例 $x$ のカテゴリならば $P(c|x,\bar{\theta})$ は 1 とし，そうでなければ 0 とする．

EM アルゴリズムの変種に tempered EM[2] がある．この変種では，モデルの複雑さを調整することができるという利点を持ち，本研究では，先に説明したオリジナルの EM の代わりに，tempered EM を採用する．tempered EM は，E-ステップで式 (5) の代わりに次式を使用することで実現できる[6]：

$$P(c|x,\bar{\theta}) = \frac{\{P(c|\bar{\theta})P(x|c,\bar{\theta})\}^\beta}{\sum_c \{P(c|\bar{\theta})P(x|c,\bar{\theta})\}^\beta}. \tag{8}$$

ここで，$\beta$ はモデルの複雑さを決めるハイパーパラメータで正の値をとる．この値が小さいほど，計算途中の隠れ変数の事後確率値を信用しないことになる．

ラベルなしデータに対してラベルありデータが極端に少ないと，学習を繰り返していくうちにラベルなしデータの影響が強くなりすぎて，結果が悪くなってしまうことがある．そのため，新たなハイパーパラメータ $\lambda$（$0 \le \lambda \le 1$）を用いて，ラベルなしデータの影響が小さくなるように式 (4) の右辺の第 2 項を次式と入れ換える[6]：

$$\sum_{x\in D^l}\sum_c P(c|x,\bar{\theta}) \log\left(P(c)\prod_w P(w|c)^{N(w,c)}\right) + \lambda \sum_{x\in D^u}\sum_c P(c|x,\bar{\theta}) \cdot \log\left(P(c)\prod_w P(w|c)^{N(w,c)}\right).$$

ここで，$D^l$ はラベルありデータ，$D^u$ はラベルなしデータである．この式が示すように，$\lambda$ の値が小さいほどラベルなしデータの影響が小さくなる．

この新たな $Q$ 関数を用いて導出したアルゴリズムを使用した．また，$Q$ 関数の値の変化が十分に小さくなることを終了条件とした．

### 4.3 時間帯連想語の取得

前節で述べた EM アルゴリズムから推定された確率値を用いることによって，$P(c|w)$ が，

$$P(c|w) = \frac{P(c)P(w|c)}{\sum_c P(c)P(w|c)} \tag{9}$$

として求まる．$P(c|w)$ は，単語 $w$ が与えられたときに $w$ に関連するイベントが時間帯 $c$ で生起する確率を表している．つまり，$P(c|w)$ の値が高い $w$ を選択，収集することで，陽に時間帯連想語が取得できる．

### 4.4 問 題 点

以上が時間帯判定の基本的な考え方であるが，ここで，3.2 節で述べた〈time slot=無〉の文に起因する 2 つの問題点を確認する．続く 4.5 節では，問題点をふまえて時間帯判定法を改良する．

問題点の 1 つ目は，たいていの場合，〈time slot=無〉となる文には時間帯を連想させる表現がまったく存在しないという性質的な特徴である．他の値（〈朝〉〜〈夜〉）が付与された文には，解析の焦点となる時間情報がたいてい含まれているが，〈無〉にはそのような情報（単語）が含まれていない可能性が高い．これにより，〈無〉が付与された文では，他の文と比べて単語の分布傾向の特徴が著しく異なり，前節までに述べた手法を単純に適用するだけでは判定精度の低下を招く恐れがある．2 つ目は，〈time slot=無〉となる文がほかに比べて非常に多いという量的な特徴である．表 2 から分かるように，〈無〉は他と比べて 10 倍以上多い．この偏りが生起確率の推定に悪影響を及ぼすことが予想できる．

### 4.5 2 段階からなる時間帯判定法

前述の問題点を考慮し，2 つの分類器を段階的に適用する時間帯判定手法（以下，手法 A）を提案する．**図 3** にその概要を示す．

1 段階目の分類器（以下，時間情報なしフィルタ）は，2 値分類器によって〈time slot=無〉の文とそれ以外の文を分類する．時間情報なしフィルタによって，〈無〉となる文を削除し，残りの〈朝〉〜〈夜〉の生起時間帯を持つイベント文を，2 段階目の分類器（以下，時間帯 4 値分類器）で分類する．時間帯 4 値分類器の学習には，前述したナイーブベイズ分類器と EM アルゴリズムを組み合わせたものを使用する．

### 4.6 比 較 手 法

比較対象として，〈time slot=無〉に起因する問題

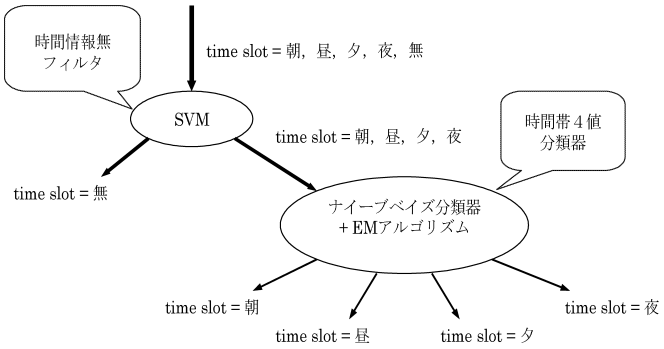


**図 3** 2 段階からなる時間帯判定法
Fig. 3 The 2-step identification method.

点を考慮しない手法（以下，手法 B）を試す．手法 B は，時間情報なしフィルタを使用せずに，〈time slot〉の 5 つの値（〈朝〉〜〈夜〉および〈無〉）を同時に 1 度で分類する分類器（以下，時間帯 5 値分類器）を作成する手法である．時間帯 5 値分類器の学習には，時間帯 4 値分類器と同様に，ナイーブベイズ分類器と EM アルゴリズムを組み合わせたものを使用する．

また，従来研究に対応させた，明示的な時間表現のみを利用する手法として，明示的な時間表現から構成される正規表現との照合に基づく時間帯分類手法を試した．この手法では，たとえば，以下の 4 つのいずれかの正規表現と照合する文を〈朝〉と判定する．

正規表現 1
`[(午前)(午前の)(朝)(朝の)(am)(AM)(am の)(AM の)][456789(10)]` 時
正規表現 2
`[(04)(05)(06)(07)(08)(09)]` 時
正規表現 3
`[(04)(05)(06)(07)(08)(09)]:[0-9]{2,2}`
正規表現 4
`[456789(10)][(am)(AM)]`

## 5. 実　　験

本章では，提案手法の有効性を評価実験によって示す．

本研究の目的は，ブログテキスト中に記述されたイベントの生起時間帯を判定することであるが，現実のブログテキストでは，表 1 に示したように，過半数以上の文は〈event=0〉となるイベント生起と無関係な内容である．そこで，以下の実験では，SVM に基づく素朴なイベント抽出器によってブログテキスト内からイベントが記述された文を抽出し，イベントが記述されていると判定された文を対象にして，そのイベントの生起時間帯を判定した．以下ではまず，イベント抽出の概要を述べ（5.1 節），その後，提案手法の実験結果を示す（5.2 節）．

### 5.1 イベント抽出

Support Vector Machines（SVM）[9)]☆を用いて，ブログテキスト内の各文について，イベントが記述されているか否かを判定する．

実験データには，3 章で述べたデータを利用し，〈event=1〉の文を正例，〈event=0〉の文を負例とした．精度評価には，10 分割の交差検定によって得られた結果を用いた．SVM のソフトマージンパラメータの値は次の手続きで導いた．実験での 10 分割交差検定は，実験データすべてを 9 対 1 に分割し，その 9 割を訓練データに，1 割を評価データとして利用するものである．パラメータ推定では，ここで得られた訓練データに対してさらに 10 分割交差検定を施すことで得た．つまり，訓練データをさらに 9 対 1 に分割し，それぞれをパラメータ推定用訓練データとパラメータ推定用評価データとすることで予測正解率を算出する．そして，予測正解率（$F$ 値）が最高となった際のパラメータ値を最終的に利用した．

イベント抽出に使用した素性情報には，内容素性と文末素性の 2 種類がある．内容素性は文内の単語（名詞，動詞，助詞–格助詞），および挨拶表現（“ありがとうございます”，“お世話になりました” 等の定型表現 6 種類）の存在の有無である．単語の品詞情報はすべて ChaSen ☆☆によって得た（これ以降も同様）．また，文末素性を**表 3** に示す．3.1.1 項で示した図 1 と図 2 の比較から分かるように，イベントか否かの判定には時制等のモダリティ情報が有効に働くと考えられたため文末素性を考慮した．なお，表 3 の文末表現タイプは “過去”，“現在”，“断定”，“推量”，“理由”，

☆ TinySVM software package：http://www.chasen.org/~taku/software/を利用した．
☆☆ http://chasen.naist.jp/hiki/ChaSen から入手可能．

**表 3** イベント抽出に使用した文末素性
Table 3 End-of-sentence features for the event extraction.

| | |
|---|---|
| 最終文節内の情報 | |
| | 品詞が “助詞–格助詞” となる助詞（9 種類） |
| | 名詞か記号のみで構成されているか |
| | 動詞の有無 |
| | 末尾の記号 |
| | 末尾が副詞か |
| | 末尾が “名詞–サ変接続” か |
| | 文末表現タイプ（19 種類）[11] |
| | 文末 $n$ 形態素 |
| 最終文節に係る文節内の情報 | |
| | 品詞が “助詞–格助詞” となる助詞（9 種類） |
| | 品詞が “助詞–係助詞” となる助詞（9 種類） |
| | 末尾が副詞か |
| | 末尾が “助詞–連体化” か |

**表 4** イベント抽出の結果
Table 4 Results of the event extraction.

| | | |
|---|---|---|
| 正解率 | 0.869 | (0.791) |
| 精度 | 0.720 | (0.479) |
| 再現率 | 0.579 | (0.268) |
| $F$ 値 | 0.639 | (0.341) |

“要望”，“叙述”，“伝聞”，“状態” 等 19 種で，値は横山らの手法[11] によって得た．文末形態素数はいくつかを試したが，以下では，最も精度が高かった $n = 2$ での結果を紹介する．

分類精度を**表 4** に示す．表 4 の左列は，上述の素性を使用した場合の結果である．右列の括弧内は，一般にテキスト分類で使用される素性として，品詞が名詞と動詞からなる単語のみを使用した場合の結果であり，参考として掲載しておく．表から，文末素性はイベント抽出に有効であることが確認でき，正解率で 0.869 を得た．

### 5.2 時間帯連想語を用いたイベントの時間帯判定

#### 5.2.1 1 段階目（時間情報なしフィルタ）の結果

時間情報なしフィルタの学習には，SVM を用い，対象文内の全形態素情報を素性として用いた．また，タグが〈無〉の文を正例として 11,668 文，〈朝〉〜〈夜〉の文を負例として 2,552 文使用して学習した（表 2 参照）．また，ソフトマージンパラメータは，5.1 節と同様に，10 分割交差検定によるパラメータ推定で決定した．

実験結果を**表 5** に示す．表の値は，10 分割交差検定によって得られた結果である．時間情報なしフィルタは，ブログテキストから〈無〉となる文を削除し，残りの〈朝〉〜〈夜〉の生起時間帯を持つイベント文のみを 2 段階目へ通過させる役割を持つ．表 5 から，再現率 0.969 で，時間帯情報のない文を削除できることが確認できる．

**表 5** 時間情報無フィルタの結果
Table 5 Classification results by the time-unknown filter.

| | |
|---|---|
| 正解率 | 0.878 |
| 精度 | 0.838 |
| 再現率 | 0.969 |
| $F$ 値 | 0.899 |

**表 6** 時間帯 4 値分類器の結果
Table 6 Results of time-slot classification.

| | 正解率 | |
|---|---|---|
| | 標準 | 文脈 |
| 明示的時間表現 | 0.109 | |
| ベースライン | 0.406 | |
| ナイーブベイズのみ | 0.567 | 0.464 |
| ナイーブベイズ+ EM | 0.673 | 0.670 |

#### 5.2.2 2 段階目（時間帯 4 値分類器）の結果

ナイーブベイズ分類器と EM アルゴリズムの組合せの学習には，ラベルありデータとして，表 2 の内訳で示した〈朝〉〜〈夜〉の各データを使用した．また，ラベルなしデータには，ラベルありデータとは重複しない未知のブログテキストから得た 64,784 文を使用した．EM アルゴリズムのハイパーパラメータ $\lambda$ と $\beta$ の値は 10 分割交差検定によるパラメータ推定で決定した．

学習に使用した素性には，標準素性セットと，標準素性セットに文脈情報を加えた文脈素性セットの 2 種類がある．標準素性セットは対象文内の単語（名詞，動詞）で構成される．文脈素性セットには，さらに，対象文の前後 1 文内の単語が入る．文脈素性セットによって，3.1.2 項で述べた (2) のような事例に適切に対応できると期待している．

評価の際は，ブログエントリ内の各文を時間情報なしフィルタに通し，フィルタを通過した文に対してのみ，時間帯 4 値分類器で時間帯を判定する．実験結果を**表 6** に示す．精度評価には，10 分割の交差検定によって得られた結果を用いている．また，ベースラインは，時間帯〈朝〉〜〈夜〉の中で最頻の〈夜〉をつねに出力した場合の結果である．

表 6 から，まず，明示的な時間表現を利用した手法の正解率が低いことが分かる．この結果は，冒頭でも述べたように，「ブログテキストでは明示的な時間表現があまり現れない」という事実を反映している．次に，ナイーブベイズ分類器単独での正解率は，標準素性セット，文脈素性セットのどちらについてもベースラインを上回っていることが確認できる．ナイーブベイズ分類器と EM アルゴリズムを組み合わせた場合，さらに正解率が向上し，最大でベースラインから

**表 7** 総合結果
Table 7 Results of total step.

| | |
|---|---|
| 明示的時間表現 | 0.833 |
| ベースライン | 0.821 |
| 提案手法 A（2 段階） | 0.864 |
| 比較手法 B（1 段階） | 0.823 |

0.267 上回った．この結果から，EM アルゴリズムによって，ラベルなしデータからイベントの生起時間帯に関する情報，すなわち，時間帯連想語に関する情報を適切に抽出できていることが分かる．

標準素性セットと文脈素性セットの正解率を比較すると，文脈素性セットでは精度が低下しており，今回の素性設計法では，文脈情報が分類に悪影響を与えていることが分かった☆．

#### 5.2.3 総合結果

5.2.1 項と 5.2.2 項で述べた，時間情報なしフィルタと時間帯 4 値分類の結果を合わせた 2 段階からなる時間帯分類器（手法 A）の結果を，比較手法とともに**表 7** に示す．ベースラインは，〈朝〉〜〈夜〉および〈無〉の中で最頻である〈無〉をつねに出力した場合の結果である．なお，2 段階からなる提案手法 A の正解率は次の式を使用して計算している：

$$\frac{\begin{array}{l}\text{時間情報なしフィルタによって正解できた}\\ \text{〈無〉の文の数 + 時間帯 4 値分類器によって正解できた〈朝〉〜〈夜〉の文の数}\end{array}}{\text{〈無〉および〈朝〉〜〈夜〉の文の数}}.$$

手法 B では，表 2 の内訳で示したデータを利用して学習し，10 分割交差検定によって評価精度を算出した．その他の実験条件は手法 A と同様である．

まず，時間帯 4 値分類の結果（表 6）とは逆に，明示的時間表現を利用した手法がベースラインを上回っていることが分かる．これはベースラインとして採用する時間帯情報が時間帯 4 値分類の場合とは異なること，また，2 段階全体の総合結果を考える際には，〈無〉となる文に明示的な時間表現が存在しない場合は正解として数えられることから，データの〈無〉への偏りが総合結果における正解率の向上に反映された結果であると考えられる．

☆ ここで，対象文の前後の文の単語情報を利用する以外に，前後の文への時間帯判定結果を動的に素性として利用することも考えられる．たとえば，ブログエントリの先頭文から順に時間帯判定を実施した場合，第 2 文目の時間帯判定に第 1 文目に対する時間帯判定結果が利用できる．もちろん前後 1 文を超えた周辺情報を利用してもかまわず，利用する周辺情報の範囲（窓枠の大きさ）は事前に自由に指定できる．このような素性設計法は，系列タグ付け問題で頻繁に見られる．追加実験として，系列タグ付け問題が容易に実現できるソフトウェア YamCha を使用し，窓枠の大きさ，解析順等を変更したいくつかの設定条件のもとで判定実験を試みたが，提案手法 A の正解率を上回る結果を実現するには至っていない．

**表 8** 分割表
Table 8 Confusion matrix.

| | | 提案手法 A の結果 | | | | | |
|---|---|---|---|---|---|---|---|
| | | 朝 | 昼 | 夕 | 夜 | 無 | 計 |
| 正解 | 朝 | 332 | 14 | 1 | 37 | 327 | 711 |
| | 昼 | 30 | 212 | 1 | 44 | 312 | 599 |
| | 夕 | 4 | 5 | 70 | 18 | 110 | 207 |
| | 夜 | 21 | 19 | 4 | 382 | 609 | 1,035 |
| | 無 | 85 | 66 | 13 | 203 | 11,301 | 11,668 |
| | 計 | 472 | 316 | 89 | 684 | 12,659 | 14,220 |

手法 B は，明示的な時間表現を利用した手法よりも正解率が下回っており，4.4 節で指摘した問題点の影響が直接現れている．それに対し，提案手法 A は，明示的な時間表現を利用した手法に比べ正解率で 0.043 上回っており，時間情報無フィルタと時間帯 4 値分類器を合わせた 2 段階からなる提案手法 A の有効性が確認できる．

提案手法 A の出力結果の詳細を見るため，分割表（confusion matrix）を**表 8** に示す．どの時間帯においても〈夜〉あるいは〈無〉に誤分類する傾向が示されており，時間帯ごとの事例の偏りの影響が依然として残っていることが示唆される．また，定性的な誤り分析の結果，3 章で示した例文 (3) のような，生起時間帯の異なる複数のイベントが 1 文で述べられている場合には特に誤り率が高く，このような事例に対しては今後改善の余地が残っている．

#### 5.2.4 取得できた時間帯連想語の例

提案手法によって取得できた時間帯連想語を**表 9** に示す．表 9 は，時間帯ごとに $p(c|w)$ の値が高い上位 20 件を示したものである．「朝」（朝：2 位），「お昼」（昼：1 位）等，上位には明示的時間表現も現れているが，「通勤」（朝：7 位），「花火」（夜：3 位）等，明示的ではないが時間帯が連想できる単語が取得できた．また，20 位以下にも，「寝癖」（朝：1,582 位），「閉館」（夜：503 位）等があった．なお，データ中の異なり単語数はおよそ 22,000 語である．

#### 5.2.5 取得できた時間帯連想語の被覆率

**図 4** はブログテキストにおける，時間帯連想語の数と時間帯連想語を含む文の数の関係を表している．$p(c|w)$ の値の高い上位 $N$ 個の時間帯連想語を選択することを考えて，$N$ を 1 から 100 まで変動させ，各 $N$ に対して，その時間帯連想語集合のいずれかの要素を含む文の数を求めた．また比較のため，明示的時間表現を含む文の数，および，IREX[7] で定められた NE-TIME タグを含む文の数をあわせて示す．NE-TIME

**表 9** 取得した時間帯連想語の例
Table 9 Examples of time-associated words.

| | 朝 | | 昼 | | 夕 | | 夜 | |
|---|---|---|---|---|---|---|---|---|
| 順位 | 連想語 | $p(c\|w)$ | 連想語 | $p(c\|w)$ | 連想語 | $p(c\|w)$ | 連想語 | $P(c\|w)$ |
| 1 | 今朝 | 0.729 | お昼 | 0.728 | 夕方 | 0.750 | 昨夜 | 0.702 |
| 2 | 朝 | 0.673 | 昼過ぎ | 0.674 | 夕日 | 0.557 | 夜 | 0.689 |
| 3 | 朝食 | 0.659 | 午後 | 0.667 | アカデミー | 0.448 | 花火 | 0.688 |
| 4 | 早朝 | 0.656 | 昼間 | 0.655 | 夕暮れ | 0.430 | 夕食 | 0.684 |
| 5 | 午前 | 0.617 | ランチ | 0.653 | ヒルズ | 0.429 | 就寝 | 0.664 |
| 6 | 圧雪 | 0.603 | 昼飯 | 0.636 | 乗り上げる | 0.429 | 晩 | 0.641 |
| 7 | 通勤 | 0.561 | 昼休み | 0.629 | 道案内 | 0.429 | 弓 | 0.634 |
| 8 | 化す | 0.541 | 昼食 | 0.607 | 松ぼっくり | 0.429 | 残業 | 0.606 |
| 9 | パレード | 0.540 | 昼 | 0.567 | 住職 | 0.428 | 忘年会 | 0.603 |
| 10 | 起床 | 0.520 | ちょうちょ | 0.558 | 砂浜 | 0.428 | 夕飯 | 0.574 |
| 11 | 出港 | 0.504 | 中華 | 0.554 | カジ | 0.413 | ビーチ | 0.572 |
| 12 | 寝坊 | 0.504 | 昼前 | 0.541 | 大森 | 0.413 | カクテル | 0.570 |
| 13 | 荷役 | 0.504 | 授乳 | 0.536 | 扇風機 | 0.413 | あっし | 0.562 |
| 14 | 目覚まし | 0.497 | 昼寝 | 0.521 | 羽田 | 0.412 | 知之 | 0.560 |
| 15 | クラ | 0.494 | オムツ | 0.511 | 下見 | 0.402 | 帰宅 | 0.557 |
| 16 | 朝焼け | 0.490 | 日本食 | 0.502 | 雲 | 0.396 | 閉店 | 0.555 |
| 17 | ホイール | 0.479 | 七夕 | 0.502 | 主 | 0.392 | 更かす | 0.551 |
| 18 | 起きる | 0.477 | 湯麺 | 0.502 | すべる | 0.392 | 今夜 | 0.549 |
| 19 | パーマ | 0.474 | 薬局 | 0.477 | 試飲 | 0.391 | 夜中 | 0.534 |
| 20 | 朝刊 | 0.470 | 麺 | 0.476 | 巣 | 0.386 | 毎晩 | 0.521 |

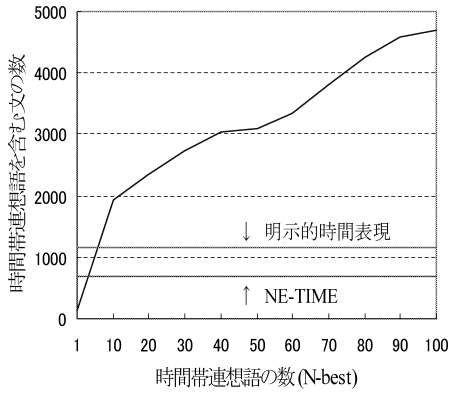


**図 4** 時間帯連想語を含む文の数の変化
Fig. 4 Number of sentences includeing time-associated words.

タグ情報は，CaboCha☆ によって取得した．図 4 から，時間帯連想語を利用することによって，明示的な時間表現を利用するよりもより広範囲なブログテキスト内の文について，イベントの生起時間帯判定が行えることが分かる．

## 6. おわりに

テキスト内に記述されたイベントに対して，実世界においてそのイベントが生起した時間帯を朝，昼，夕，夜の粒度で判定する手法を提案した．イベントの生起時間帯の学習時に時間帯を連想させる語を同時に取得可能な学習アリゴリズムを用いることによって，86.4%の正解率を達成した．今後は，各素性の最適な組合せを検討する予定である．また，判定対象文の周辺情報をより適切に取り込む方法として，系列ラベリング問題でよく利用される Conditional Random Fields[3] を適用することを考えている．

☆ http://chasen.org/~taku/software/cabocha/から入手可能.

## 参 考 文 献

1) Dempster, A.P., Laird, N.M. and Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society Series B*, Vol.39, No.1, pp.1–38 (1977).
2) Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis, *Machine Learning*, Vol.42, No.1-2, pp.177–196 (2001).
3) Lafferty, J., McCallum, A. and Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Proc. International Conference on Machine Learning* (*ICML-2005*), pp.282–289 (2001).
4) Mani, I. and Wilson, G.: Robust Temporal Processing of News, *Proc. 38th Annual Meeting of the Association for Computational Linguistics*, pp.69–76 (2000).
5) Mitchell, T.: *Machine Learning*, McGraw-Hill (1997).
6) Nigam, K., McCallum, A., Thrun, S. and Mitchell, T.: Text classification from labeled

and unlabeled documents using EM, *Machine Learning*, Vol.39, No.2/3, pp.103–134 (2000).
7) Sekine, S. and Isahara, H.: IREX project overview, *Proc. IREX Workshop* (1999).
8) Setzer, A. and Gaizauskas, R.: A Pilot Study on Annotating Temporal Relations in Text, *Proc. ACL-2001 Workshop on Temporal and Spatial Information Processing*, pp.88–95 (2001).
9) Vapnik, V.N.: *The Nature of Statistical Learning Theory*, Springer (1995).
10) 倉島 健，手塚太郎，田中克己：Blog からの街の話題抽出手法の提案，第 16 回データ工学ワークショップ論文集，2C-i10 (2005).
11) 横山憲司，難波英嗣，奥村 学：Support Vector Machine を用いた談話構造解析，情報処理学会自然言語処理研究会（NL-155），pp.193–200 (2003).
12) 土屋誠司，渡部広一，河岡 司：連想メカニズムを用いた時間判断手法の有効性の検証，情報処理学会自然言語処理研究会（NL-168），pp.113–118 (2005).
13) 南野朋之，鈴木泰裕，藤木稔明，奥村 学：blog の自動収集と監視，人工知能学会論文誌，Vol.19, No.6, pp.511–520 (2004).
14) 小倉牧人，田村直良：文間の時間制約モデルと事象の時系列化への応用に関する研究，情報処理学会自然言語処理研究会（NL-140），pp.111–118 (2000).

(平成 19 年 5 月 15 日受付)
(平成 19 年 7 月 3 日採録)

**野呂 太一**

1981 年生．2006 年東京工業大学大学院総合理工学研究科修士課程修了．現在，富士通株式会社に勤務．

**乾 孝司**（正会員）

1976 年生．1999 年九州工業大学情報工学部卒業．2001 年同大学大学院情報工学研究科修士課程修了．2004 年奈良先端科学技術大学院大学情報科学研究科博士課程修了．日本学術振興会特別研究員等を経て，2006 年より東京工業大学統合研究院特任助教．博士（工学）．主に自然言語処理に関する研究に従事．言語処理学会，ACL 各会員．

**高村 大也**（正会員）

1974 年生．1997 年東京大学工学部計数工学科卒業．2000 年同大学大学院工学系研究科計数工学専攻修了（1999 年はオーストリアウィーン工科大学にて研究）．2003 年奈良先端科学技術大学院大学情報科学研究科博士課程修了．博士（工学）．2003 年より東京工業大学精密工学研究所助教．自然言語処理，特に学習理論等の応用に興味を持つ．ACL 会員．

**奥村 学**（正会員）

1962 年生．1989 年東京工業大学大学院情報理工学研究科計算工学専攻博士後期課程修了．1989 年より東京工業大学大学院情報理工学研究科助手．1992 年より 2000 年北陸先端科学技術大学院大学助教授．1997 年より 1998 年トロント大学客員助教授．2000 年より東京工業大学精密工学研究所助教授．自然言語処理，自動テキスト要約，コンピュータによる語学学習支援，テキストデータマイニングに関する研究に従事．工学博士．AAAI, ACL，JSAI，JCSS 各会員．