平成 17 年度知能システム科学専攻修士論文(研究室用)

テキスト中のイベントの生起時間帯判定

野呂　太一

Temporal Processing of Events in Text

Noro　Taichi

提出年月日　平成 18 年 3 月 8 日

主査教員　　奥村　学

審査教員　　新田　克己

審査教員　　橋田　浩一

テキスト中のイベントの生起時間帯判定

野呂太一

Temporal Processing of Events in Text

Taichi Noro

We propose a machine learning-based method for identifying when an event in weblog texts occurs: morning, daytime, evening, and 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 naïve bayes classifiers backed up with the EM algorithm, and support vector machines. Additionally, we utilize the information of "time slots sequence" for improving the performance of classification.

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．そしてそれらの中には，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い．その代表としてニュース記事が挙げられる．

このようなテキスト群を“時間”に注目して整理する研究がある．その目的は，実際のイベントの発生とは異なった順序で記述されるニュース記事などを，実際の発生順に時系列に並べることで事実の理解を深めること，あるいは，ある一連のイベントについて複数回記述されている場合，時間情報をもとに整理することで要約の助けとすることである．

上記のような従来研究の多くはニュース記事を対象としていた．その理由として，ニュース記事が“3 日午後 3 時ごろ”や“今週金曜日”のように，明示的に時間情報を表記していることが多く，研究対象として扱いやすかったことが考えられる．しかし，イベントを記述しているテキストの全てがニュース記事のように明示的に時間情報を記述しているわけではない．例えば Web 日記，ブログなど（以下まとめてブログと呼ぶ）もイベントを記したテキストと言える．

ブログは個人が自由に情報を発信できるものであるため，ニュース記事と比べ，その内容の正確性，重要な情報の密度などが劣ることは確かである．しかし近年，その内容から読み取れる消費者動向，一般消費者による製品の評判情報などが，企業によるマーケティングの対象として調査されることも多く，その重要性を増している．

個人が日常のイベントを記すブログでは，ニュース記事とは性質が異なり，イベントの記述の際に，その生起時間を明示的に示すことは稀である．

しかし，イベントの生起時間が明示されていないにも関わらず，人間がブログを読んだときには，その内容からおおよその生起時間が特定できる場合は少なくない．さらに，ブログの内容からイベントの生起時間の特定が可能になれば，検索における新たな軸としての時間情報の利用や，時間帯ごとの人々の行動統計（例えば“人は朝何を食べているのか”）などを把握することが可能になると考えられる．

そこで本研究は，ブログテキストを対象にし，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を目的とする．もう少し具体的には，イベントを朝，昼，夕，夜の粒度に分類する．

明示的な時間表現がない場合に対応するために，機械学習手法を用いて自動的に時間帯を連想させる表現（以下，時間帯連想語）の情報を取り入れて，イベントの時間帯を特定する．具体的には，ナイーブベイズ分類器を EM アルゴリズムで補強する semi-supervised な手法を SVM と組み合わせ，時間帯情報分類器を作成する手法を提案する．さらに，“朝→昼→夕→夜”という時間の流れの情報を利用し，判定精度を高める試みも行う．

既存研究では，イベント情報の処理単位として，文書レベルから単語レベルまでさまざまなものがあるが，本研究では文を処理単位とし，文毎に時間帯を特定する．ただし，ブログのような日記形式のテキストでも，全ての文がイベントを表した文というわけではない．そして当然のことながら，イベントを表していないその他の文に対して，イベントの生起時間帯判定を行っても意味がない．

これを踏まえ本研究は，

Step1: テキストからイベントについて記述された文を抽出する“イベント文抽出”，

Step2: 抽出された文のイベントの生起時間帯を判定する“イベント文の時間帯判定”，

の２つの課題を逐次的に行うことによって，文内で表現されたイベントの生起時間帯を判定する．

2. 関連研究

Setzer ら[1]や Mani ら[2]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発生の絶対時間の決定を可能にするとともに，時間情報・イベント同士の相対的な順序関係に着目し，イベントの整列を行うことも目指している．小倉ら[3]は，ニュース記事を対象とし，一文章中のイベントの時系列化を目指した．特に，イベント同士の前後関係を求める時間推論に焦点を置き，データを絶対時間を持つものと，相対情報を持つものに分け，イベント群の時系列化を行っている．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[4]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．本研究では，辞書にあたるものを利用せず，人々の行動のデータ（ブログ）から時間情報の取得を目指している．これにより，辞書には記載されないような，日常生活に基づいた時間連想情報を得られると考えている．

また，テキスト中のイベント文を抽出する課題と関連したものでは，倉島ら[5]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をしている．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. コーパス

1 節でも述べたように，本研究ではブログを解析の対象とする．本節では，ブログエントリのテキストから作成するコーパスについて説明する．このコーパスは，4 節で説明する機械学習手法において，訓練・評価データとして使用する．

南野ら[6]が収集したブログデータを，1 文毎に自動的に切り出したものを使用する．ただし，ブログのデータは文末に正確に句点が記述されないことも多いので，正確に文に分割することは難しい．よって，ここでの 1 文とは，句点や HTML タグの情報によって単純にテキストを分割したものであり，不適切に分割されたもの*も含まれている．

3.1. タグ

各文に“event”，“time slot”の 2 種類のタグを付与した．それぞれについて説明する．

3.1.1. event タグ

event タグは文がイベントを表しているか否かを“1”，“0”の 2 値で表したものである．文がイベントを表しているときは 1 を付与し，表していないときは 0 を付与する．イベントを表していない文とは，何らかの説明をしている，主張・感想を述べているものなどである．

event=1 である文の例を例 1 に示す．

例 1

a. 福岡に行くために，羽田に行きま

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。

例 1-b は，文末で“行った”が省略されている場合

* 文分割エラーはおよそ 5%程度である．

である．このように文の一部が省略された場合，用言をひとつのみ補ってイベントであるか否かを判断する．また，例 1-c のように心情が変化したこともイベントとした．例 1-d は，“割れる”というイベントは起きなかったが，“割れなかった”こともイベントであると考え，これもイベント文とする．

続いて event=0 である文の例を例 2 に示す．

例 2

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. 20 名さまにプレゼント！(広告の文句)

今回の対象とするデータは，個人のブログと企業のブログの区別をしていないため，例 2-e のような文も存在する．

ここで，イベント文の判定に迷った事例について紹介する．イベント文の定義は，“イベントを表す文”であるのだが，人によるタグ付けの際，その判断に迷うものが多数存在した．その一部を例 3 に示す．

例 3

a. 今日までだったので、忘れないで、良かった。
b. めずらしく以前の食欲と同じくらい。
c. 端のとこまで走る。

例 3 はどれもイベント文の例である．例 3-a は心情の説明のようにもとれるが，良かったと感じた瞬間のことを書いたもので，イベントである．例 3-b は一見分かりづらいが，“以前と同じくらい食べる”飼い猫の様子を書いたものなので，イベントである．また，例 3-c のように“走った”と表現せずに，“走る”と書く場合も多数見受けられた．このパターンは，正確に過去形で書かず，現在形・進行形のように表現し，イベントの臨場感を高めて表現してある場合である．以上のように，イベント文の判定には，人手による判断の時点でも迷うもの*が多数存在する．これがイベント文抽出の精度に悪影響を及ぼすことが考えられる．

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”，“情報無し（時間帯不明）”の 5 値で表したものであり，event=1 の文にのみ付与される．これらのタグの包含関係を表すものを図 1 に示す．ただしこの図において，面積は実際のタグ数とは無関係である．

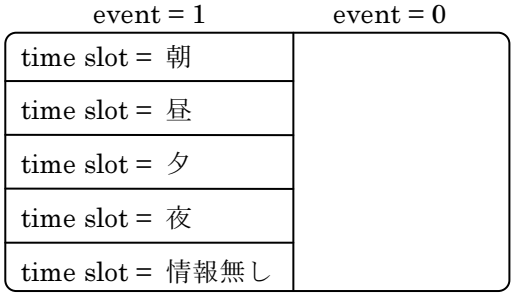

図 1：コーパスにおけるタグの包含関係

各時間帯の目安として設定した定義を以下に示す．

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

上記の時刻は目安であり，ブログの著者が認識している時間帯を重視して判断する．つまり“今朝 3 時ごろ”の場合は，“今朝”という表現から，著者が朝として認識していることが分かるので，夜ではなく朝と判断する．文にそれぞれの値をつける判断は，文内の比較的明示的な表現によって付与可能になるものと，文内には明示的な表現がないが前後の文脈情報を見ることによって付与可能になるものがある．前者の例を例 4 に示す．

例 4

a. 朝から自転車で郵便局へ行く。（朝）
b. 昼は，定食屋で豚丼を食べた。（昼）
c. １６時過ぎには帰路につく。（夕）
d. 鍋を作り，その日の夕食とした。（夜）

後者の例を例 5 に示す．ここで，例 5-文 1,2 は連続してブログに出現したものとする．この場合，例 5-文 1 を朝と判定し，それに続いて出現した例 5-文 2 も話の流れから朝だと判断できる．

例 5

文 1．朝から自転車で郵便局へ行く。（朝）
文 2．郵便局の帰りに某ショップへ。（朝）

例 6 のように，一文で複数のイベントを記述して

* タグ付与は 1 人で行ったため，一致率の調査は不可能である．

いる文も存在する．

例 6

今日は朝学校に行って，昼には弁当を食べ，夕方帰った。(夕)

このような場合は，文の末尾側に記述されているイベントのみに注目して，そのイベントの時間帯で判断することとした．つまり，例 6 では time slot=夕となる．ただし，“朝から晩まで”のように，複数の時間帯にまたがった文の場合には，time slot=情報無しとした．

3.2. コーパス統計

コーパスは人手で作成している．ブログエントリの数は 7,413 であり，そこから分割した文の総数は 70,775 文である．著者数は 267 人である．各タグの内訳を表 1，2 に示す．

表 1：event タグ内訳

event=1	14,220
event=0	56,555
計	70,775

表 2：time slot タグ内訳

time slot=朝	711
time slot=昼	599
time slot=夕	207
time slot=夜	1,035
time slot=情報無し	11,668
計	14,220

表 1 から，event=1 の文と event=0 の文の量は偏っており，event=1 の文の割合が少なく event=0 の文が多いことが分かる．同様に表 2 から，time slot タグについても情報無しの文が，他の文に比べてかなり多いことが分かる．これらの偏りは，後の実験に影響を与えることも考えられる．この偏りへの対処法については，4.2.3 節で詳しく説明する．

4. 提案手法

提案手法では，まずテキストからイベントについて記述された文を抽出する“イベント文抽出”を行い，次に，抽出された文のイベントの生起時間帯を判定する“イベント文の時間帯判定”を行う．本節ではこれらについて順に述べる．

4.1. イベント文抽出

文がイベントを表すか否かを，機械学習を用いて判定する．具体的には前述のコーパスを利用し，event=1 の文を正例クラス，event=0 の文を負例クラスとし，SVM によって分類器を作成する．

分類に利用する素性情報について説明する．文がイベントを表すか否かの判定には，文末に現れるモダリティ情報などが有効であると考えられる．これを次の 2 つの事例を用いて説明する．

例 7

a. 朝，トーストを食べた。(event=1)
b. 朝，トーストを食べる。(event=0)

例 7-a はイベントを表すが，例 7-b は習慣の説明をした文であり，イベントではない．この場合，それぞれの文を構成する単語にはほとんど差異がないが，文末の表現によってイベント文か否かが判定される．このように，文末表現のタイプや，品詞の種類などがイベント文判定には有効だと考えた．

これを踏まえ，単語（名詞，動詞）に加え，表 3 に示すものを素性として使用した．

表 3：イベント文判定に使用した素性

最終文節内の情報
“助詞-格助詞”の種類（9 種類）
名詞・記号のみで構成されているか
動詞の有無
末尾の記号の種類
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）（横山[7]）
文末 n 形態素

最終文節に係る文節内の情報
“助詞-格助詞”の種類（9 種類）
“助詞-係助詞”の種類（9 種類）
末尾が副詞か
末尾が“助詞-連体化”か

文節位置に関係のない情報
挨拶表現が存在するか（6 種類）
“助詞-格助詞”の種類

ここでの文末表現タイプには横山[7]が使用したものを用いた．文末 n 形態素の n は 2~4 を試し，最も良い結果を示したものを使用する．挨拶表現の存在とは，“ありがとうございます”，“お世話になりました”など，主に挨拶で使用される表現が文に含まれているか否かの情報である．品詞体系は ChaSen[8]に従う．文末表現タイプと文末 n 形態素は，情報として重複する部分もあるが，それぞれが有用な情報を持つ．文末表現タイプはあらかじめ用意した 19 種のタイプに文を分類するもので，“過去，現在，断定，推量，理由，要望，叙述，伝聞，状態”など，文の種類を正確に取得することができる．一方，文末 n 形態素は実際のイベント文としての使用例から学習するので，柔軟性のある情報として利用できる．よって，双方を同時に利用するメリットはあると考えた．

4.2. イベント文の時間帯判定（時間帯連想語）

「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，例えば「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は，時間帯のタグが付けられたイベントコーパスを種として，時間帯のタグが付いていない大量のイベントコーパスを併せて利用する，半教師付学習を用いればよい．そこで，教師付き学習手法のナイーブベイズ分類器を Expectation Maximization(以下 EM)アルゴリズム[9]で補強する semi-supervised な方法を適用する．ナイーブベイズ分類器を用いたのは，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[10]によって示されているからである．

4.2.1. ナイーブベイズ分類器による時間帯分類

ここではまずナイーブベイズ分類器（Naïve Bayes classifiers）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，

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

となる．ここで，$P(|x|)$は長さ$|x|$の文が生起する確率であり，$N(w,x)$は文 x 中での素性 w の出現頻度である．文の生起は，全語彙の中から単語を一つ選び出す試行の繰り返しとして，モデル化される．

ナイーブベイズ分類器を文の時間帯分類に適用した場合，各文が事例 x に相当し，カテゴリ c は，朝，昼，夕，夜，情報無しのいずれかの値をとる．使用される素性は，文に出現する単語などである．素性の詳細は 4.2.4 節で述べる．

4.2.2. ナイーブベイズ分類器と EM アルゴリズムの組み合わせ

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：

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

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

以降，モデルのパラメータ群をまとめてθ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：

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

ここで，$P(\theta) \propto \prod_c \left(P(c)^{\alpha-1}\prod_w \left(P(w \mid c)^{\alpha-1}\right)\right)$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：

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

M-ステップ：

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

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

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[11]がある．この変種では，モデルの複雑さを調整することが出来る．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：

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

ここで，β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有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無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$\lambda(0\le\lambda\le1)$を用いて，ラベル無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：

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

ここで，D^lはラベル付きデータ，D^uはラベル無しデータである．この式が示すように，λの値が小さいほどラベル無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．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．

4.2.3. “time slot=情報無し”の文の問題点

ここで，具体的な手法に入る前に，time slotの値に情報無しが付与された文に関する問題点を2点述べる．

１つ目は，時間帯を連想させる表現が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含まれているが，この文にはそれが含まれていない可能性が高い．これにより，情報なしが付与された文では，他の値が付与された文と比べて素性の分布の特徴が著しく異なり，提案手法に悪影響を与えることが予想される．

２つ目は，他の値が付与された文と比べて，量が非常に多いという特徴である．3.2節で示した表2を見ても分かるように，他のものと比べ，10倍以上の量の文が存在している．この差が，生起確率に影響を及ぼすことが予想できる．

以上のように“time slot=情報無し”の文は，分類器の学習において悪影響を及ぼす可能性が高いため，この問題点を考慮した分類器の作成を行う．

4.2.4. 時間帯分類手法

前述した問題点を考慮した手法について説明する．具体的には，2段階で分類器を作成する手法(以下，手法A）を提案する．

１段階目の分類器（以下，時間情報有無分類器）は，time slotの値が情報無しの文と，それ以外の文を分類する．この学習にはSVMを使用する．使用した素性は，対象文内の全形態素である．

そして，時間情報有無分類器によってtime slotの値が時間帯情報有り（朝～夜）だと判定された文を，２段階目の分類器（以下，時間帯４値分類器）で朝，昼，夕，夜に分類する．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．使用した素性は対象文内の単語（名詞，動詞）である．この2段階分類器による，分類の流れを表したものを図2に示す．この図において，楕円は分類器を表し，矢印は分類されるテキストの流れを表す．

また，実験の比較対象として4.2.3節の問題点を考慮しない手法（以下，手法B）を試す．これは，時間情報有無分類器を使用せずに，time slotの値が朝，昼，夕，夜，情報無しの文を分類する５値分類器（以下，時間帯５値分類器）を作成する手法である．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．使用した素性は単語（名詞，動詞）である．

4.3. 時間の流れを考慮したイベント文の時間帯判定

本節では，テキスト中の時間の流れの利用について述べる．ブログのような日記タイプのテキストにおいて，複数のイベントが記されている場合，それ

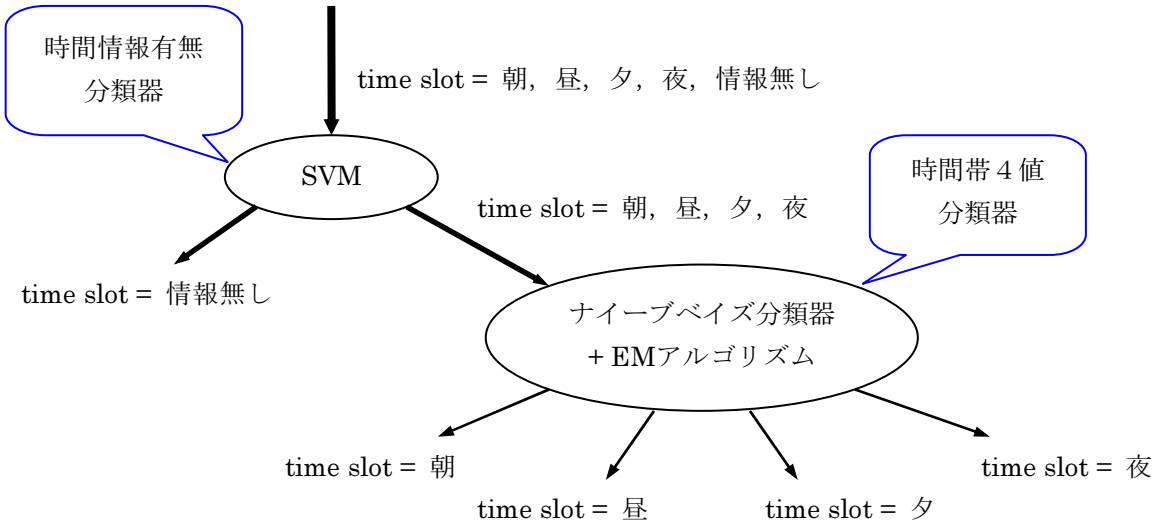

図 2：２段階分類器による分類の流れ

らが発生した順にテキストに出現しやすいのではないかと考える．例えば，一日の出来事をその日の夜にまとめてブログに書く場合，朝のイベントから順に記されやすいのではないかという考えである．つまり，“朝→昼→夕→夜”のような“時間の流れ”がブログテキストには存在するという仮説をもとに，それを時間帯判定に利用する手法の提案である．

4.3.1. 時間の流れの存在検証

まず，実際にブログテキスト中に時間の流れが存在するかについて検証する．まず time slot タグについて，テキスト中での出現順の 2gram, 3gram を取得し，頻度で降順にソートした結果を表 4 に示す．ただし，同じタグが連続して出現した場合はひとつにまとめて計算した．

この結果より，2gram，3gram ともに“夜→夕”，“昼→朝”のような時間が“逆戻り”するパターンは，出現頻度上位には存在しないことが分かった．夜→朝というパターンが上位にきているが，これは時間的に繋がっているので逆戻りではないとした．

次に，time slot タグの出現順の 2gram から遷移確率を求め，状態遷移表を作成した．ただし，連続した同じタグのまとめは行っていない．これを表 5 に示す．

また，この結果をもとに作成した状態遷移図を図 4 に示す．ただし，この図では解析の便宜上，自己ループ及び，“time slot=情報無し”への遷移は省いた．

これら表 5, 図 4 から以下のことが分かる．まず，朝からの遷移確率が一番高いのは昼である．これは，時間の流れの仮説のとおりである．次に遷移が高いのは夜であるが，朝と夜のイベントのみをブログに記す場合のことを考え，同じく仮説のとおりであるといえる．昼からの遷移は夜が一番高い．仮説から考えると，夕への遷移が高くあるべきだが，夕のタ

表 4：time slot タグ出現順 ngram

	2gram		3gram	
順位	種類	割合	種類	割合
1	夜–無	0.216	無–夜–無	0.237
2	無–夜	0.178	無–朝–無	0.146
3	朝–無	0.149	無–昼–無	0.145
4	昼–無	0.130	夜–無–夜	0.097
5	無–朝	0.108	昼–無–昼	0.067
6	無–昼	0.106	朝–無–朝	0.061
7	夕–無	0.039	無–夕–無	0.045
8	無–夕	0.037	朝–無–昼	0.024
9	朝–昼	0.010	朝–無–夜	0.015
10	夜–朝	0.009	夕–無–夕	0.014
11	昼–夜	0.006	昼–無–夜	0.014
12	夕–夜	0.004	朝–昼–無	0.014
13	朝–夜	0.003	夜–朝–無	0.013
14	昼–夕	0.002	夜–無–朝	0.013
15	夜–昼	0.002	夕–無–夜	0.011
16	夜–夕	0.001	無–夜–朝	0.009
17	昼–朝	0.001	昼–夜–無	0.008
18	夕–昼	0.000	昼–無–夕	0.008
19			無–朝–昼	0.007
20			夕–夜–無	0.005

表 5：time slot タグ状態遷移確率表

		遷移先タグ				
		朝	昼	夕	夜	情報無し
遷移元タグ	朝	0.264	0.052	0.004	0.025	0.655
	昼	0.018	0.229	0.011	0.027	0.716
	夕	0.005	0.021	0.298	0.069	0.606
	夜	0.023	0.009	0.005	0.219	0.743
	情報無し	0.033	0.031	0.011	0.055	0.871

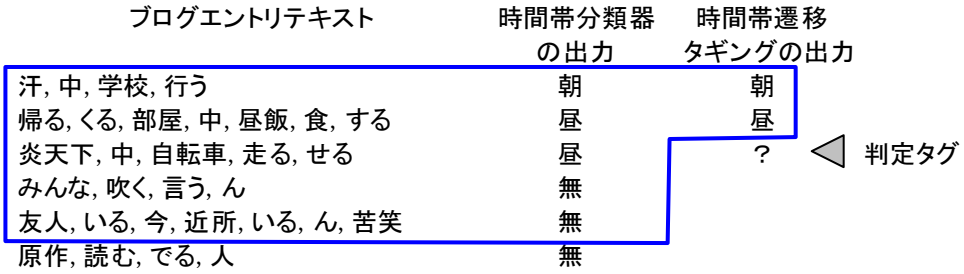

図 3：時間帯遷移タギング使用素性例

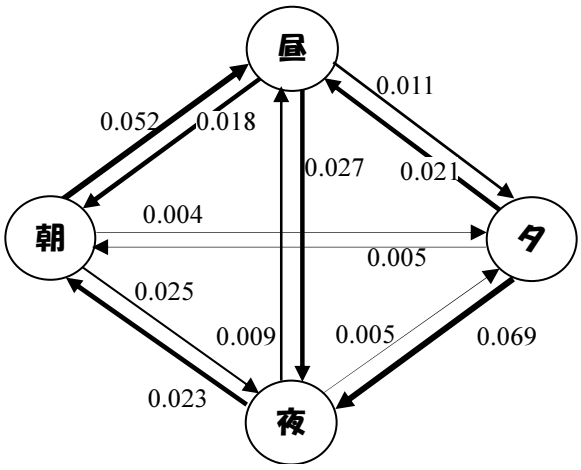

図 4：time slot タグ状態遷移図

グ数自体が少ないことの影響だと思われる．夕からは，夜への遷移確率が朝に比べて約 14 倍，昼に比べて約３倍と著しく高い．同じように，夜からは朝朝への遷移が高い値を示している．以上のように，概ね仮説の通り，時間の流れが存在していると思われる．

4.3.2. 時間の流れの利用手法

ここではブログテキスト中の時間の流れを利用する手法について述べる．前節で述べたように，時間帯タグには，時間の流れに沿った出現傾向があると考えられる．そこで，その前に出現した文群にどの時間帯タグが付与されたかを考慮することで，時間の流れを利用した，文の時間帯判定が可能になると思われる．

具体的には，判定対象文の前後に出現した文の素性を利用し，エントリでの出現順に前から文の time slot タグを判定していく（以下この手法を，時間帯遷移タギングと呼ぶ）．素性には，ブログエントリテキスト，時間帯分類器によるタグなどを使用する．使用素性の例を図 3 に示す．

この図は，あるブログエントリの一部を抜き出したもので，列の左から順に，ブログエントリテキスト，時間帯分類器によるタグ，時間帯遷移タギングによって付与された動的タグを表している．ただし，エントリテキスト中で利用する素性は，単語の名詞・動詞のみなので，図ではそれらを抜き出したものになっている．例えば前後２文の情報をタグの判定に利用する場合，図の枠で囲まれた部分が素性となる．つまり，前後２文の本文中の名詞・動詞，時間帯分類器によるタグ，前２文に時間帯遷移タギングによって付与された時間帯タグである．実験用のツールとしては，SVM の結果に基づいてテキストのチャンキングを行う，汎用テキストチャンカーの YamCha[12]を使用する．

5. 実験と考察

5.1. イベント文抽出

5.1.1. イベント文抽出の実験結果

event=1 のデータを正例クラスとして 14220 文, event=0 のデータを負例クラスとして 56555 文使用して, SVM による分類実験を 10 分割交差検定で行った．なお，SVM の学習には TinySVM[13]を使用した．ソフトマージンパラメータの値は 10 分割交差検定によるパラメータ推定によって決定する．以下，このパラメータ推定手法について詳しく述べる．

分類実験での 10 分割交差検定は，実験データ全てを 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．パラメータ推定での 10 分割交差検定は，この訓練データをさらに 9 対 1 に分割し, それぞれを訓練データと評価データとすることで予測正解率を算出する．そして, その予測正解率が最高となるパラメータ値を, 推定パラメータ値とするものである．なお，ここでは F 値が最高となるようにパラメータ値の推定を行った．

結果を表 6 に示す. 表 6 右列の単語のみとは, 4.1 節で提案した素性（表 3 参照）を使用しない，単語（名詞，動詞）のみの場合の実験結果である．表 3 の素性を使用する場合と比べ，F 値で 0.298 ポイント低い値を示した．これにより，4.1 節で提案した素性は，有効であることが示された．なお，文末形態素数は 2 を使用した. 次節で分類性能の向上を目指す実験を行う．

表 6 ：イベント文分類結果

	提案手法	単語のみ
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

5.1.2. イベント文事例数同数学習

event=1 と event=0 のデータ量には大きな差が存在する（表 1 参照）. これが, 再現率の低下を招き, 分類結果を悪くしている可能性がある．そこで, event=1 と同数の event=0 のデータ（ランダムに選択）を使用して学習した分類器による実験を，10 分割交差検定で行った．その結果を表 7 に示す．なお，事例数未調整との比較のために，評価データは前節と同じものを使用した．

再現率が 0.223 上がり F 値はわずかに 0.012 上がった．しかし，精度が 0.17 ポイント下がった．精度を重視しない場合は，正負例の事例数のバランスを取ることが重要であることがわかった．

表 7：イベント文事例数同数分類結果

正解率	0.825
精度	0.550
再現率	0.802
F 値	0.651

以上の実験より, 最終的に F 値で 0.639 の結果を得た．これでは充分な分類性能とはいえない．しかし，イベント文判定は，人手でのコーパス作成の際にも，判断が困難な（曖昧な）文が多数含まれていたため，分類も困難になったと思われる．

5.2. イベント文の時間帯判定（時間帯連想語）

5.2.1. 時間帯判定結果

まず, 時間帯分類手法 A の結果について説明する. 時間情報有無分類器は time slot の値が情報無しのデータを正例クラスとして 11668 文, 時間帯情報有り（朝～夜）のデータを負例として 2552 文使用して, SVM による分類実験を 10 分割交差検定で行った．結果を表 8 に示す．なお，ソフトマージンパラメータ値は 5.1.1 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．

表 8：時間情報有無分類器結果

正解率	0.878
精度	0.838
再現率	0.969
F 値	0.899

この分類器は，２段階目の分類器へ time slot=情報無しの文を渡さないことが目的であり，“time slot=情報無しフィルター”としての役割を期待するものである． 0.969 という高い再現率は，このフィルターとしての役割を十分に果たすものであると言える．

時間帯４値分類器は時間情報有無分類器により，time slot が時間帯情報有り（朝～夜）だと判断し

たデータを用いてナイーブベイズ分類器＋EM アルゴリズムによる分類実験を 10 分割交差検定で行った．結果を表 9 に示す．

表 9：時間帯 4 値分類器結果

手法	正解率
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
ナイーブベイズ＋EM	0.673

ラベル無しデータには未知のデータ 64784 文を使用した．また，λ，β 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．

表 9 の“明示的時間表現”とは，明示的な時間表現を正規表現*によって抽出する，簡単な時間帯 4 値分類器によるものである．ベースラインは全ての文の time slot が，朝～夜の中で一番数の多い夜だと判断した場合の正解率である．EM アルゴリズムの適用が成功していることが分かり，正解率でベースラインを 27%上回った．

次に，時間帯分類手法 B について説明する．time slot の値が朝，昼，夕，夜，情報無しのデータをそれぞれ，711 文，599 文，207 文，1035 文，11668 文用いて，ナイーブベイズ分類器＋EM アルゴリズムによる分類実験を 10 分割交差検定で行った．結果を表 10 に示す．ラベル無しデータには未知の（タグの付けられてない）データ 64784 文を使用した．λ，β 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．

表 10：時間帯 5 値分類器結果

手法	正解率
ベースライン	0.821
ナイーブベイズのみ	0.823
ナイーブベイズ＋EM	0.786

ベースラインは全ての文の time slot が情報無しと判断した場合の正解率である．EM アルゴリズムによって正解率は低下してしまった．また，ナイーブベイズのみでもベースラインをわずかに上回るのみである．

次に，λ の値を変化させた場合の正解率の変化をグラフで示す．図 5 は時間帯 4 値分類の正解率の変化を，図 6 は時間帯 5 値分類の正解率の変化をそれぞれ表す．なお，どちらも β は 0.04 で実験した．図 5 のグラフより，時間帯 4 値分類ではλの値を上げることで正解率も上昇していることが分かる．一方図 6 のグラフより，時間帯 5 値分類ではλの値を上げると，正解率が減少していることが分かる．

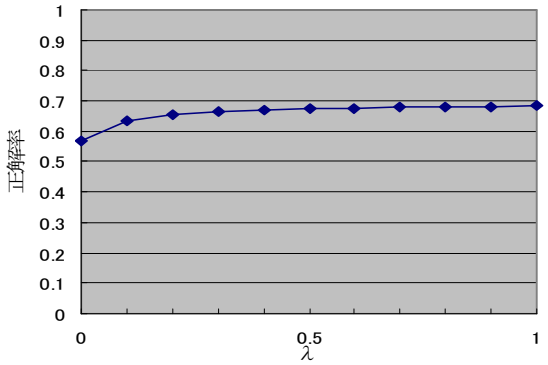

図 5：λ変化による正解率の推移(時間帯 4 値分類)

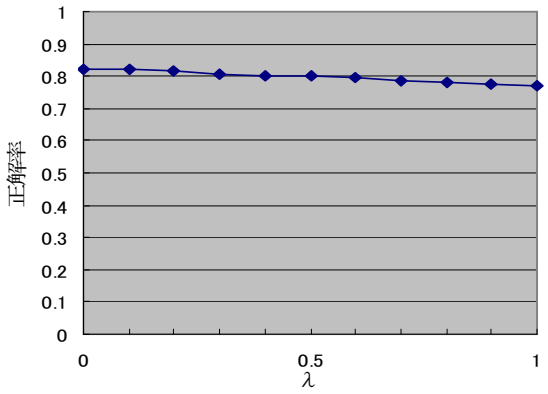

図 6：λ変化による正解率の推移(時間帯 5 値分類)

最後に，時間情報有無分類器の出力を時間帯 4 値で使用した，2 段階の分類器によって得られた最終的な手法 A の分類の正解率を，手法 B と比較して表 11 に示す．

表 11：手法比較

手法	最終正解率
明示的時間表現	0.833
ベースライン	0.821
手法 A	0.864
手法 B	0.823

* 例えば，次の正規表現にマッチしたものを朝に分類する：[(午前)(午前の)(朝)(朝の)(am)(AM)(am の)(AM の)][456789(10)]時, [(04)(05)(06)(07)(08) (09)]時, [(04)(05)(06)(07) (08) (09)]:[0-9]{2,2}, [456789(10)][(am)(AM)].

表 12：時間帯分類器出力の混合表

		時間帯分類器の出力					計
		朝	昼	夕	夜	時間情報無し	
time-slot タグ	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	1035
	情報無し	85	66	13	203	11301	11668
計		472	316	89	684	12659	14220

手法 A の正解率は次の式を使用して計算した．

$$\frac{\begin{pmatrix}\text{時間情報有無分類器によって正解できた}\\ \text{"}timeslot\text{= 情報無し"の文の数}\end{pmatrix}+\begin{pmatrix}\text{時間帯4値分類器によって正解できた}\\ \text{"}timeslot\text{= 朝, 昼, 夕, 夜"の文の数}\end{pmatrix}}{timeslot\text{タグに値が付与されている文の数}} \quad (9)$$

時間帯分類手法 A と B の比較では，手法 A の方が良い結果（正解率で 4.1%上回った）を示した．これらの結果から，4.2.3 節で述べた time slot=情報なしの文の問題点が分類器の学習に悪影響を与えていることが分かり，2 段階に分類器を作成する提案手法が有効であることが示せた．

次に，手法 A の結果による混合表を表 12 に示す．この表より，朝・昼・夕への分類は概ね成功しているものの，夜・情報無しへの分類が若干悪い結果となっていることが分かる．事例のうち 11.7%が，誤って夜・情報無しへと分類されてしまっている．

5.2.2. 時間情報有無分類素性追加実験

本節では，いくつか素性を追加して実験を行う．以下の 2 種類の素性を使用する．

素性セット 1:　全形態素＋形態素の 2gram

素性セット 2:　素性セット 1＋形態素の 3gram

表 13：時間情報有無分類素性追加実験結果

	素性セット 1	素性セット 2
正解率	0.866	0.869
精度	0.678	0.728
再現率	0.487	0.440
F 値	0.558	0.541

結果を表 13 に示す．なお，パラメータ推定は行わず，ソフトマージンパラメータは 0.1 で実験した．2gram，3gram を追加したものは，どちらも表 8 の結果と比較して，値を下げてしまった．

5.2.3. 時間帯４値分類ラベル無しデータの比較

EM アルゴリズムに影響を与える，ラベル無しデータとして最適なデータを調べる比較実験を行った．ラベル無しデータとしては，様々な種類の事例が含まれているもの，ラベル有りデータにできるだけ類似したものの適用が考えられる．

様々な種類の事例が含まれているものとして，タグ付与がされておらず，ブログ中の全種類の文が含まれているデータ（以下未知データ）が考えられる．未知データにはイベント文ではない説明文などが多量に含まれるため，一見ラベル無しデータとしては不適切であるように思われる．しかし，イベント文以外にも時間情報を持つ文，例えば習慣の説明（“私は毎朝，トーストを食べます．”）などが含まれるため，正解率の向上に有効なデータである可能性がある．ラベル有りデータと類似したものは，データの素性の分布が類似しているので，正解率の向上が期待される．これには，時間情報有無分類器によって“time slot=情報無し”以外（朝，昼，夕，夜）だと判断されたデータが適当だと考えられる．そして，以上の 2 つの中間に位置するものとして event=1 であるデータもラベル無しデータとして有効である可能性がある．そこで，以下の 3 種類のデータで実験を行った．

データ 1：未知データ

データ 2：未知データからイベント分類器を使用して抽出した event=1 のデータ

データ 3：データ 2 から時間情報有無分類器によ

って“time slot=情報無し”以外だと判断されたデータ

結果を表 14 に示す．訓練・評価データは 5.3 節の時間帯 4 値分類器で使用したものと同じであり，β の値は 0.01 である．

ここでのデータ 1 ：未知データは，表 9 での実験のラベル無しデータと同じ種類のものであるが，データ 2 に合わせサイズを調整（少なく）してあり，パラメータ推定も行っていないものである．

表 14：ラベル無しデータ比較実験結果

λ	正解率		
	データ 1	データ 2	データ 3
0.1	0.625	0.624	0.604
1.0	0.640	0.634	0.588

データ 2 は λ の値をあげるにつれて正解率は上昇したものの，表 9 の結果と比べて，効果的とは言えない結果となった．データ 3 は，λ の値をあげるにつれて正解率が悪くなってしまった．データ 1 が 3 種類の実験の中で最も良い結果を示した．

データそれぞれに正解率の向上を期待したが，結果的にイベント文とそれ以外の文が混合されたデータが最もふさわしいことが分かった．

5.2.4. 時間帯 4 値分類使用素性比較

本節では，手法 A の時間帯 4 値分類器の学習において，いくつかの素性を追加した実験を行う．追加した素性は，次の 3 種類である．

素性セット 1： 係り受け関係にある名詞-動詞，名詞-形容詞のペア

素性セット 2： 文内での位置情報（最終文節，最終文節にかかる節）

素性セット 3： 前後の文の情報

β=0.01 で実験した結果を表 15 に示す．

表 15：素性比較実験結果

素性タイプ	正解率
素性セット 1 （係り受け）	0.674
素性セット 2 （文内位置）	0.661
素性セット 3 （前後文）	0.660

結果はどれも基本的な素性（名詞，動詞）使用でパラメータ推定を行わない場合の正解率 0.686 を下回ってしまった．

素性セット 1 は，“会社に行く”のような名詞-動詞の組み合わせから，“朝”を連想するような場合を想定したのだが，元の正解率を超えることは出来なかった．素性セット 2 は，例 6 のような場合を考え，最終文節内の情報が分類に有効だと考えたが，同じく元の正解率を下回った．3.1.2 節において説明したように，time slot タグには前後の文脈を見ることによって時間帯の判定が可能になった文が，多数存在する．素性セット 3 は，これを踏まえて追加した素性だったが，良い結果は出なかった．前後の文の情報が有用である可能性は高いが，単純に前後の文を素性に加えるだけでは，その情報をうまく生かすことが出来なかったためだと思われる．

5.2.5. 取得連想語例

提案手法で得られた時間帯連想語の例を表 16 に示す．

これは，素性 w が与えられたときにカテゴリ c のどこに出現しやすいかを表す，$P(c \mid w)$ の値で単語をカテゴリごとに降順に並べたものである．ラベル無しデータのみに出現した時間帯連想語の例としては，“寝癖(朝:1582 位)”，“通学(朝:1989 位)”，“閉館(夜:503 位)”，“酔い潰れる(夜:2852 位)”などがあった．なお，訓練・評価・ラベル無しデータの出現単語総数はおよそ 22600 語であった．

図 7：時間帯連想語を含む文の数の変化

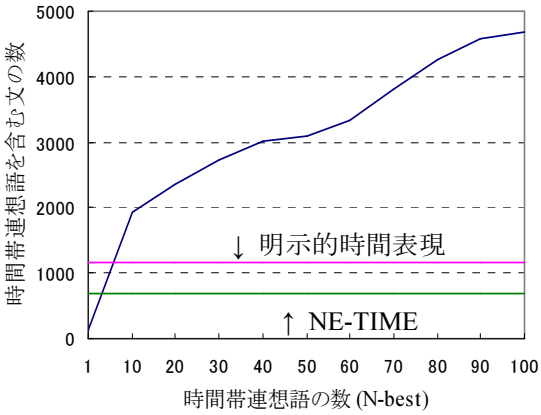

図 7 はブログテキストにおける，時間帯連想語を含む文の数を表したものである．横軸は時間帯連想語の数を表したもので，$P(c \mid w)$ によって降順にソートした N-best を選択したものである．縦軸は N-best のいずれかの時間帯連想語を含む文の数を表したものである．また比較として，明示的時間表

表 16：取得連想語リスト

	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
21	点検	0.467	定食	0.464	集金	0.384	昨晩	0.511
22	朝方	0.467	配る	0.454	帰路	0.382	二次会	0.509
23	集合	0.456	P	0.446	受話器	0.374	クレア	0.507
24	開会	0.454	ユキコ	0.445	ほんま	0.369	散らかす	0.506
25	毎朝	0.450	卒	0.436	姪	0.368	助	0.504
26	おっさん	0.447	パイ	0.433	配線	0.368	弦	0.498
27	朝っぱら	0.446	扇ぐ	0.431	夕焼け	0.368	一口	0.495
28	早起き	0.444	給食	0.431	松浦	0.362	ナポリ	0.494
29	新幹線	0.439	炎天下	0.431	内定	0.362	外食	0.494
30	バス	0.438	天王寺	0.431	途切れる	0.359	詩人	0.493

現を含む文の数，NE-TIME タグ[14]を含む文の数も示した．明示的時間表現は 5.2.1 節で説明したものを使用した．NE-TIME タグ情報は，日本語係り受け解析システム CaboCha[15]によって取得した. この表より, 対象とすることが出来る文の数が, 既存の手法に比べ, 提案手法では大幅に増加していることが分かる.

5.3. イベント文の時間帯判定（時間の流れ）

5.3.1. 時間帯判定実験

この節では，使用する最適な素性の枠（以下，ウインドウ）を求めるための実験を行う．まず，以下の５種類を試す.

素性セット１： 前４文

素性セット２： 前３文＋後ろ１文

素性セット３： 前２文＋後ろ２文

素性セット４： 前１文＋後ろ３文

素性セット５： 後ろ４文

以上の素性で実験した結果を表 16 に示す．YamCha のパラメータは, 全てデフォルトのまま使用した（2 次多項式カーネル使用，ソフトマージンパラメータは 1.0 である）. 以降, 特に明記しない限り, デフォルトのパラメータを使用するものとする．なお, 使用したデータは event=1 であり, 時間帯分類器によって time slot タグが付与されているデータ 14220 文である．なお，表 17 右列の時間帯遷移タギング無し正解率とは, 時間帯遷移タギングによる動的タグ付与情報を使用しない場合の結果である.

また，素性セット 5 に通常の正解率がないのは，後ろの文の情報しか使用しない場合は，もともと動的タグ付与情報を使用していないからである．

表 17：最適ウインドウ実験結果

素性セット	正解率	時間帯遷移タギング無し正解率
1: 前４	0.837	0.855
2: 前３＋後１	0.844	0.862
3: 前２＋後２	0.846	0.864
4: 前１＋後３	0.855	0.863
5: 後４		0.851

動的タグ付与情報を使用する正解率では，前１文＋後ろ３文を使用した素性セット 4 の値が一番高く，前の文の情報を使用する素性セットほど正解率が下がることが分かった．しかしどの素性セットでも，時間帯遷移タギング無し正解率が動的タグ付与情報を使用する正解率を上回り，時間帯遷移タギングの動的タグ付与情報はノイズとなることが分かった．また，一番高い正解率のものは前後２文の情報を利用する素性セット３の結果であった．

次に，前後の情報を利用するものを，ブログエントリテキスト，時間帯分類器によるタグ，それぞれに限定した実験を行う．つまり，前者の場合ならば，エントリテキストについて前後文の情報を利用し，時間帯分類器によるタグは判定対象文の情報のみを利用した実験を行い，後者の場合ならば，時間帯分類器によるタグについて前後文の情報を利用し，エントリテキストは判定対象文の情報のみを利用した実験を行う，ということである．前後文の範囲は，表 17 の実験と同じであり，動的タグ付与情報は使用していない．結果を表 18 に示す．

表 18：前後情報限定実験結果

素性セット	エントリテキスト前後利用正解率	時間帯分類器タグ前後利用正解率
1: 前４	0.862	0.842
2: 前３＋後１	0.861	0.838
3: 前２＋後２	0.862	0.841
4: 前１＋後３	0.861	0.839
5: 後４	0.860	0.833

前後文情報としてエントリテキストのみを利用した結果は，使用素性範囲が変化しても正解率に差がなく，時間帯分類器タグのみの場合は，使用素性の範囲で正解率に差が出た．これにより，素性の範囲選択には，時間帯分類器タグの情報が影響を与えることが分かった．しかし，どちらの正解率も前述の実験より正解率が下がってしまったため，前後の情報を利用する素性を限定する手法は，有効ではないことが分かった．

5.3.2. 後ろ向きタグ付与及びエントリ書き込み時間の利用

前節では，ブログエントリにおいての文の出現順にタグ付与をした．時間の流れの利用において，“夕の次は夜の出現確率が高い”ではなく，“夕の前には夜は出現確率が低い”という利用も当然考えられる．そこで，この節ではタグを後ろから付与する実験を試す．素性は前節と同じ素性セット 1～4 を使用する．

ここで，さらにエントリの書き込み時間の情報も素性として利用することを考える．エントリ中のイベントの生起時間と，エントリの書き込み時間とは基本的には一致しないと思われる．それは，発生したイベントについて，リアルタイムにエントリを記述できる場合は稀だからである．しかしながら，エントリ中の最後のイベントと書き込み時間は，ある程度一致する可能性がある．つまり，朝・昼・夜の複数のイベントをまとめて夜に書く場合，最後の夜のイベントと書き込み時間は近い可能性がある，ということである．

そこで，後ろ向きタグ付与において，書き込み時間の情報が，エントリ最後の文のタグ付与に有用であるかを検証する実験を行う．エントリ最後のタグ付与精度が向上することにより，エントリ全体のタグ付与の正解率が向上することを期待する．結果を表 19 に示す．

書き込み時間を利用したものは，ほぼ全てにおいて，利用しないものより正解率が高いことが分かり，後ろ向きタグ付与において，書き込み時間が有効であることが分かった（ちなみに，予備実験によって，前向きタグ付与に書き込み時間を利用すると，正解率が下がることが分かっている）．しかし，どれも 5.3.1 節の正解率を上回ることは出来なかった．

表 19：後ろ向きタグ付与

素性セット	正解率	書き込み時間利用正解率
1: 前４	0.823	0.820
2: 前３＋後１	0.822	0.852
3: 前２＋後２	0.823	0.856
4: 前１＋後３	0.826	0.859

5.3.3. 正解率上限の検証

5.3.1 節，5.3.2 節の実験では，時間帯分類器の正解率を上回ることは出来なかった．これは，時間帯分類器が，既にある程度高い正解率を出していることが原因の一つだと思われる．ここで，時間帯遷移タギングによるタグ付与で正解している文が，時間帯分類器で正解している文のみだった場合，時間帯遷移タギング使用による正解率の上昇は期待できないことになる．

そこで本節では，時間帯分類器と時間帯遷移タギングを理想的に組み合わせることが出来た場合の，正解率の上限を求める．

具体的には，ある文の時間帯分類器によって付与されたタグと，表 17 の実験で時間帯遷移タギングによって付与されたタグのどちらか一方でも正しいのなら，その文の判定結果を正解として正解率を算出した．結果を表 20 に示す．

表 20：時間帯遷移タギング正解率上限実験結果

素性セット	正解率	時間帯遷移タギング無し正解率
1: 前４	0.895	0.882
2: 前３＋後１	0.893	0.879
3: 前２＋後２	0.892	0.879
4: 前１＋後３	0.893	0.878
5: 後４		0.876

結果はどの素性の場合でも，時間帯分類器による正解率 0.864 を上回った．これは，時間帯遷移タギングでのみ正解できる文が存在することを示している．以上より，時間帯遷移タギングにより正解率が上昇する可能性があることが分かり，理想的に組み合わせることができれば，0.895 まで正解率の向上が見込めることが分かった．また，表 17 では時間帯遷移タギング無しの正解率が高かったが，この結果では逆転している．つまり，時間帯遷移タギングでのみ正解できる文においては，時間帯遷移タギング自身によるタグが有効に働くということであり，時間の流れの情報が有効であるということである．

5.3.4. 時間帯遷移タギングのみ正解可能パターンの解析

確信度を利用した 5.3.4 節の実験では，結局正解率を向上させることは出来なかった．そこで，時間帯遷移タギングのみが正解できる文のパターンを解析することで，タグの適切な選択を可能にすることを目指す．

表 21：時間帯遷移タギングのみ正解 ngram

	2gram		3gram	
順位	種類	割合	種類	割合
1	無-無	0.282	無-無-無	0.236
2	昼-昼	0.249	昼-昼-昼	0.180
3	夜-夜	0.247	夜-夜-夜	0.177
4	朝-朝	0.100	朝-朝-朝	0.071
5	無-夜	0.035	無-夜-夜	0.071
6	夕-夕	0.020	朝-朝-無	0.068
7	朝-無	0.017	朝-昼-昼	0.028
8	夜-無	0.017	無-昼-昼	0.025
9	無-昼	0.012	無-無-夜	0.025
10	無-朝	0.010	無-夕-夕	0.016

表 22：時間帯分類器のみ正解 ngram

	2gram		3gram	
順位	種類	割合	種類	割合
1	無-無	0.523	無-無-無	0.440
2	夜-無	0.074	無-無-夜	0.055
3	朝-無	0.067	朝-無-無	0.050
4	無-夜	0.067	夜-無-無	0.048
5	昼-無	0.060	昼-無-無	0.046
6	夜-夜	0.038	朝-朝-無	0.039
7	無-昼	0.026	夜-夜-夜	0.031
8	無-朝	0.026	昼-昼-無	0.029
9	昼-夜	0.018	夜-夜-無	0.029
10	無-夕	0.018	無-夜-無	0.022

表 21 は時間帯遷移タギングでのみ正解できたパターンで，正解したタグの１つ前と２つ前の統計を取った，2gram と 3gram である．表 22 には比較対照として時間帯分類器のみ正解できたパターンを

示した．なお，同じタグが連続した場合のまとめは行っていない．

表 21 を見ると，同じタグが連続して出現した場合が上位に存在することが分かる．表 22 では，1 位の time slot=情報無しの連続を除くと，そのような傾向は上位には存在しない．これにより時間帯遷移タギングでのみ正解できるパターンは，情報無し以外のタグが連続した場合である可能性がある．そこで，これをもとにして，時間帯遷移タギングによるタグが 2 回連続，及び 3 回連続で同じタグを付与した場合に，時間帯遷移タギングのタグを採用する実験を行う．結果を表 23 に示す．

表 23：時間帯遷移タギング連続同タグ採用実験

手法	正解率
2 回連続	0.844
3 回連続	0.846

残念ながらどちらの手法でも，正解率の向上は見られなかった．今回は，単純な手法によってのみ時間帯遷移タギングの採用を行ったが，結果は不充分であった．この方法によって正解率の向上を目指すには，より詳細なパターンの解析を行い，より高度な時間帯遷移タギングの採用手法を確立することが必要である．

6. おわりに

本研究では，テキストからイベント文を抽出し，そのイベントの生起時間帯を判定することを行った．連想語によって時間帯を判定するという考えを，機械学習の手法によって実現し，正解率で 86.4%という結果を出すことが出来た．また，さらにブログテキスト中の“時間の流れ”を判定に利用する手法を試した．残念ながら結果を向上させることは出来なかったが，時間帯分類器による手法と理想的な組み合わせが可能になれば，最高で 89.5%の正解率を得ることが出来る可能性を示した．

今後は，素性の改善などのほかに，時間の流れを利用して正解率を向上させるために，時間帯分類器と時間帯遷移タギングの理想的な組み合わせ方法を確立することを考えている．また，時間帯遷移タギングの代わりに，データ系列を学習し，品詞タグ付けなどで用いられる Conditional Random Fields[16]を適用することも考えている．

参考文献

[1] Andrea Setzer, Robert Gaizauskas. A Pilot Study on Annotating Temporal Relations in Text. In *Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing*, Toulose, France, July, pp.88–95, 2001.

[2] Inderjeet Mani, George Wilson. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp.69–76, 2000.

[3] 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. 情報処理学会研究報告「自然言語処理」，No.140-16, pp.111–118,2000.

[4] 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. 情報処理学会研究報告，2005-NL-168, pp.113–118, 2005.

[5] 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(DEWS2005)論文集，2C-i10, 2005

[6] 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.

[7] 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.

[8] http://chasen.naist.jp/hiki/ChaSen.

[9] Arthur P. Dempster, Nan M. laird, and Donald B. Rubin. 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.

[10] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, Vol. 39, No.2/3, pp.103–134, 2000.

[11] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learnin*g, 42:177–196, 2001.

[12] http://www.chasen.org/~taku/software/YamCha.

[13] http://www.chasen.org/~taku/software/TinySVM.

[14] Satoshi Sekine, Hitoshi Isahara. IREX project over view. *Proceedings of the IREX Workshop*, 1999.

[15] http://chasen.org/~taku/software/cabocha/

[16] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data, In *Proc. of ICML*, pp.282–289, 2001.