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[†††]
[†]東京工業大学大学院 総合理工学研究科 [††]日本学術振興会 特別研究員 [†††]東京工業大学 精密工学研究所
代表連絡先：norot@lr.pi.titech.ac.jp

1. はじめに

近年，増加を続ける電子テキストの中に，ニュース記事やブログなどのように，ある出来事，イベントについて記述されたものが多く存在する．このようなテキストを対象に，“時間”について整理し，事実理解や，テキスト要約などに役立てる研究がある[2][3].

これらの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，イベントの生起時間について明示的時間情報が表記されていることが多く，解析対象として扱いやすいことが挙げられる．

一方，企業のマーケティング対象として，近年その重要性を高めているブログでは，イベントの生起時間を明示的に示すことは稀であり，従来の手法では十分な時間情報を取得することができない．しかし，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も多い．ブログ中のイベント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などを把握することが可能になると考えられる．

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．我々は野呂ら[1]において，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，文単位にイベントの時間帯を特定する手法を提案した．本稿では，この手法を改善する．さらに“朝→昼→夕→夜”という時間の流れの情報を取り入れた新しい手法を提案する．

2. 関連研究

Setzerら[2]やManiら[3]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注釈付けを研究目的とした．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．このような先行研究は，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．

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．

3. コーパス

本節では，解析の対象とする，ブログエントリのテキストからな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．

データには南野ら[5]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．次節で，各文に付与したtime slotタグについて説明する．

3.1. time slot タグ

本節では各文に付与した“time slot”タグについて説明する．これはイベントが生起した時間帯を“朝，昼，夕，夜，情報無し（不明）”の 5 値で表したものである．時間帯の目安として設定した定義を次に示す．

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できる場合と，前後の文脈情報を見ることによって付与できる場合がある．前者の例を例 1 に示す．

例 1　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. 鍋を作り，その日の夕食とした。（夜）

後者の例を例 2 に示す．ここで，例 2-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 2-文 1 を朝と判定し，それに続いて出現した例 2-文 2 も話の流れから朝だと判断できる．

例 2　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）

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，分割した文の総数は 70,775 文である．このうち，本研究の対象となるイベント文*は 14220 文存在した．タグの内訳を表 1 に示す．表 1 から，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，この問題を考慮した手法を提案する．

表 1：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

4. 提案手法

まず，4.1 節で野呂ら[1]の手法を簡単に説明する．その後，4.2 節で時間の流れの情報を考慮した新しい手法を説明する．

4.1. イベント文の時間帯判定（時間帯連想語）

ここでは，野呂ら[1]の手法について簡単に説明する．「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で，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[6]で補強するsemi-supervisedな方法[7]を適用する．

ただし，3.2 節で述べたように，time slot=情報無しの文の問題があるため，時間帯を２段階に分けて判定する．

１段階目では，time slot の値が情報無しの文と，それ以外の文を分類する分類器（以下，時間情報有無分類器）を作成する．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．この手法は，基本的に野呂ら[1]と同じだが，使用素性が異なる．

２段階目では，時間情報有無分類器によって time slot が時間帯情報有り（朝～夜）だと判定された文を，朝～夜に分類する分類器（以下，時間帯４値分類器）を作成する．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．

4.2. イベント文の時間帯判定（時間の流れ）

本節では，時間の流れの情報を利用した手法について述べる（前節の手法では時間の流れが考慮されない）．ブログのような日記タイプのテキストでは，複数のイベントは，実際の発生順に出現しやすいと仮定できる．例えば，一日の出来事をその日の夜にまとめて書く場合，朝の出来事から順に記されやすいだろう．“朝→昼→夜”のような“時間の流れ”がブログには存在するという仮定をもとに，それを時間帯判定に利用する手法を提案する．

4.2.1. 仮定の検証

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．time slot タグについて，テキスト中での出現順の 2gram を取得し，それをもとに遷移確率を求め，状態遷移表を作成したものを表 2 に示す．列ラベルが遷移元を示し，行ラベルが遷移先を示す．

表 2 をもとに，自己ループと情報無しへの遷移を除いて，遷移先の遷移確率の大きさを比較する．朝からの遷移確率が一番高いのは昼である．これは，時間の流れの仮定の通りである．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，同じく仮定のとおりであるといえる．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，夜からは朝への遷移が高い値を示しているが，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．以上より，概ね仮定の通り時間の流れが存在していると思われ，“時間的に次の時間帯への遷移確率が高い”，“時間的に逆行する遷移は起きにくい”ということが分かった．

表 2：time slot タグ状態遷移確率表

	朝	昼	夕	夜	情報無
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	0.033	0.031	0.011	0.055	0.871

4.2.2. 時間の流れの利用手法

ここでは時間の流れを利用する具体的手法について述べる．前節で述べたように，time slot 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，ブログ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

*本研究の目的は，イベントの時間帯判定である．よって，イベントを表していない文に時間帯判定を行っても意味が無い．

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 1 に示す．

ブログエントリテキスト (名詞，動詞のみ抜粋)	時間帯分類器の出力	時間帯遷移タギングの出力
汗，中，学校，行う	朝	朝
部屋，昼飯，食，する	昼	昼
炎天下，自転車，走る	昼	？
みんな，吹く，言う，ん	無	▲
友人，いる，今，近所	無	判定
原作，読む，でる，人	無	タグ

図 1：時間帯遷移タギング使用素性例

この図は，あるブログエントリの一部を抜き出したものある．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，図ではそれらに置換している．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出力，前 2 文に動的に付与された time slot タグである．

また，同様に，エントリの末尾から逆順に判定していく，後ろ向きタギングも行う．なお，後ろ向きタギングにはエントリの書き込み時間の情報も素性として利用した．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を即座に記述できる場合は稀だからである．しかし，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．そこで，後ろ向きタギングでは，書き込み時間情報は有用であると考えた．

5. 実験と考察

5.1. イベント文の時間帯判定（時間帯連想語）の結果

時間情報有無分類器はtime slotの値が時間帯情報有り（朝～夜）のデータを正例（2552 文），情報無しのデータを負例（11668 文）として，SVMによる分類実験を 10 分割交差検定で行った．SVMの学習にはTinySVM[8]を使用した．ソフトマージンパラメータの値は，訓練データを使用した 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．結果を表 3*に示す．

表 3：時間情報有無分類器結果

正解率	0.878
精度	0.750
再現率	0.493
F 値	0.588

*コーパスサイズ，使用素性，パラメータが異なるため，野呂ら[1]とは結果が異なる．

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．ラベル無しデータにはタグ付与がされていないデータ（以下未知データ）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．結果を表 4 に示す．

表 4：時間帯 4 値分類器結果

手法	正解率
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

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正解率を表 5 に示す．表 5 の 5 値分類手法とは，時間情報有無分類器を使用せずに，ナイーブベイズ分類器+EM アルゴリズムによって，朝～夜，情報無しの 5 つのクラスを一度に分類する 5 値分類器を作成する手法である．なお，最終正解率は次式で算出した．

$$\frac{\begin{pmatrix}\text{時間情報有無分類器によって正解できた}\\ \text{"}timeslot = \text{情報無し"の文の数}\end{pmatrix} + \begin{pmatrix}\text{時間帯4値分類器によって正解できた}\\ \text{"}timeslot = \text{朝,昼,夕,夜"の文の数}\end{pmatrix}}{timeslot\text{タグに値が付与されている文の数}} \quad (1)$$

表 5：手法比較

手法	最終正解率
提案手法	0.864
5 値分類手法	0.823

5 値分類手法との比較では，提案手法の方が良い結果（正解率で 4.1%上回った）を示した．これらの結果から，time slot=情報無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手法が有効であることが示せた．

5.2. イベント文の時間帯判定（時間の流れ）の結果

本節では時間の流れ情報を利用する手法の実験について述べる．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[9]を使用する．5 種類の窓枠を用いて実験した前向きタギングの結果を表 6 に示す．表中の素性セットのラベルは，“前x”が前x文を表し，“後y”は後ろy文を表す．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，ソフトマージンパラメータ= 1.0）．以降，このデフォルトのパラメー

タを使用するものとする．使用したデータは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．表 6 の動的タグ無し正解率とは，時間帯遷移タギングの情報を使用しない場合の結果である．

表 6：前向きタギング実験結果

素性セット	正解率	動的タグ無し正解率
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

前向きタギングでは，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情報を使用する正解率を上回り，動的タグ情報はノイズとなった．一番高い正解率は前後２文の情報を利用する素性セット３の結果であった．

続いて後ろ向きタギングの結果を表 7 に示す．

表 7：後ろ向きタギング実験結果

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

後ろ向きタギングでは，書き込み時間を利用したものは，ほぼ全てで，利用しないものより正解率が高いことが分かり，後ろ向きタギングにおいて，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，前向きタギング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 5.1.1 節の正解率を上回ることは出来なかった．

5.3. 組み合わせ手法

最後に，これまで述べた２つの手法を組み合わせる手法について述べる．まず，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た．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 6 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 8 に示す．

表 8：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	動的タグ無し正解率
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．表 6 では動的タグ情報無しの正解率が高かったが，表 8 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，動的タグ情報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．

ここで，時間帯分類器と時間帯遷移タギングの結果のうち，コーパス中での出現頻度の高いタグを選択する，簡単な組み合わせ手法を試す．これは，単純にコーパス中での出現頻度の高いほうのタグを選択すれば，正解する可能性が高いという考えがもととなる．しかし，この手法による正解率は 0.863 となり，残念ながら正解率の向上は見られなかった．

6. おわりに

本研究では，テキスト中のイベント文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，ブログ中の“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．

参考文献

[1] 野呂太一，乾孝司，高村大也，奥村学．イベントの生起時間帯判定. 情報処理学会研究報告, 2005-NL-170, pp.7–14, 2005.

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

[3] Inderjeet Mani, George Wilson. Robust Temporal Processing of News. In *Proc. of the 38th ACL*, pp.69–76, 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] 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.

[6] 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.

[7] 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.

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

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