シソーラスや時制を考慮した電子文書からの知識獲得に関する研究

上嶋, 宏 / UEJIMA, Hiroshi

(発行年 / Year)
2005-03-24
(学位授与年月日 / Date of Granted)
2005-03-24
(学位名 / Degree Name)
修士(工学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

2004年度　修士論文

シソーラスや時制を考慮した電子文書からの知識獲得に関する研究

STUDIES ON KNOWLEDGE DISCOVERY FROM ELECTRIC DOCUMENTS WITH THESAURUS AND TENSE

指導教官　三浦 孝夫 教授

法政大学大学院工学研究科
電気工学専攻修士課程

03R3207　上嶋(うえじま) 宏(ひろし)

Hiroshi UEJIMA

目 次

第1章　序論

1.1　問題の背景

近年，目覚しい情報化の進展により，情報技術を活用する動きが急速に進んでいる．従来の紙媒体やフィルムの電子化は，スペースの有効利用できることや，管理、検索が容易であること，また情報を共有化できる事などから非常に有益である．

例えば，電子政府や電子商取引（e コマース）、電子カルテ、電子図書館など数え上げればきりがない．このように，政治，ビジネス，医療から教育，芸術にいたるまで，幅広い分野において情報技術が活用されている．

また，近年の急速なインターネットの普及や，情報公開の潮流もあり，取得可能なデータ量は爆発的に増加している．さらに，計算機の記憶装置の低価格化や大容量化に伴い，大量のデータがデータベースとして蓄積されるようになった．

現在、世界に存在するデータは毎年約 2 倍のペースで増加している．対して，意味のある情報の量は急速に減少しているという事実が報告されている．これは，情報の量が増え続けているという状況において，求められる意味のある情報を見つけることが次第に難しくなる為である．どの時代においても人間が一生に読める情報量には変化がないため，これは当然である．また，多くの場合において，情報が増える理由は，機械的に文書が作られるからである．

このような状況から，大量のデータから，自明ではない有用なパターンや傾向を抽出する技術が注目されている．これは，**データマイニング (Data Mining)** と呼ばれ，知識獲得 (KDD) のパターン発見の段階と定義される．情報検索では，ユーザは明確な答えを持っているのに対し，データマイニングでは自明ではない新しい知識を発見するという点が，単なる検索とは異なる．

有名な例として，米国ウォルマートの紙オムツとビールの話がある．POS データから，データマイニング技術により,「紙おむつを買う人はビールを買うことが多い」という思いもよらない傾向が発見でき，実際にそれらの棚を近くに配置することで売り上げが上昇したというものである．この傾向は 1 つの知識であり，このような知識を機械的，自動的に見つけようというものがデータマイニングである．上記例は比較的単純であるが，より大量のデータの取り扱いや，複雑なルールの発見は人手では不可能であり，データが電子化される事，すなわち計算機で

扱える事で初めて可能になる．

データマイニングは古くから機械学習 (machine learning) と区別される，双方の目的は知識の獲得で同じであるが，データマイニングは「実用技術」としてのパタン (知識) の発見に相当し，機械学習は，そこで使われている有用な技術，手法として位置づけられる．

実際にデータマイニングが応用されている例として，上記の POS データの分析による市場分析 (バスケット分析) や，顧客関係管理（CRM）と呼ばれる顧客データの分析による事業戦略を図るもの，あるいはレントゲンからの異常発見，ジャンクメールのフィルタリング等がある．

データマイニングの代表的な手法として，相関関係分析 (Association)，時系列分析 (Sequential pattern)，クラスター分析 (Clustering)，クラス分類 (Classification)，などが挙げられる．

現在，企業に蓄積されているデータの約 8〜9 割はテキストデータであると言われている．そこで近年，膨大な文書の中から欲しい情報を探したい，あるいは大量の文書データを分析して内容を瞬時に把握したいというニーズが高まりつつある．そこで，**テキストマイニング**と呼ばれる分野が注目されている．

一般に，データマイニングは，格納するデータが，明示的に属性が与えられたテーブルであり，さらに，値域も制限できる構造データであるのに対して，一方，テキストマイニングは，格納するテキストデータは属性が与えられず値も自由に取り得る非構造データである．テキストマイニングは自然言語処理やデータマイニング技術，機械学習技術など多様な技術を組み合わせた複合技術である．ただし、単にこれらの技術を組み合わせただけでは有効な結果は得られない．近年，多くの研究者が，さまざまなアプローチや要素技術を用いて，テキストマイニングという新しい領域に参入し，研究領域も拡大してきている．

テキストマイニングは，情報検索に加え，コーパスに基づく言語処理 (computational linguistics) と区別しなくてはいけない．テキストマイニングにおいてマイニングという言葉を，新しい情報を発見するという意味ではなく，大きなデータ集合のなかから自動的に傾向やパターンを発見するという意味で使うなら，これはテキストデータを対象にしたテキストデータマイニング，すなわちコーパスに基づく言語処理になってしまう．コーパスに基づく言語処理は，情報検索など文書集合内の処理には貢献しているが，文書集合の傾向をつかむといった文書集合自身を超えた一般的な知識の発見には不十分である．単なるパターンではなく，個別の文書には含まれない新しい情報を発見するものがテキストマイニングと定義される [4].

このような定義から，テキストマイニングの概念に相当する代表的なものとして，文書の特定の内容をあらかじめ定められたカテゴリのいずれかに要約するも

のである文書分類 (text categorization) が挙げられる．この定義では，文書分類自体は新しい情報を発見するものではない．しかし，文書分類を用いて，より一般的な使用目的のためにテキストデータ内の傾向やパターンを発見することは，テキストマイニングの概念に適合する．

例えば，Web ディレクトリの構築においては，収集した大量の Web ページを「ニュース」や,「政治」,「スポーツ」,「計算機」,「中古車情報」などあらかじめ定められたカテゴリへ分類する作業が必須である．従来は人間が過去の経験や知識から内容を読み取り，手作業で分類を行っていた．これは膨大な時間と費用がかかる．テキストマイニング技術により人間の経験や知識等，分類の判断基準をルールとして自動で発見，抽出することができるなら，より大量のデータを素早く安価に分類することが可能である．更に，文書集合が分類されたカテゴリ分布を比較することで，予期せぬ傾向を発見するためにも使われる．また，電子メールの自動フィルタリングや，文献データベースの索引付け等も自動分類の一種とみなせる．

文書集合の傾向を発見する文書分類は，幅広い適用可能性を備え，大規模なデータ処理にも適用できる可能性が高い．そのため，電子化されたコンテンツが増加し続けている現在，非常に必要とされている技術である．

一般に文書分類は，人間に与えられた少数の訓練例 (教師データ) から，分類規則を学習する．これは教師付学習 (supervised learning) と呼ばれる．

データマイニングにおいては，扱うデータが時間軸上のデータである場合が多々ある．POS データからの売り上げの季節による変動や，テキストデータにおいては，オンラインで配信されるニュース記事などはまさに時間に依存している．また掲示板，メールマガジンなどの中には時系列的関連を持つものが少なくない．このようなデータにおいて，新規話題の発生を自動検出したり，興味のある話題の関連記事を自動追跡・収集することができれば，情報過負荷に陥りがちな情報活用において非常に強力な武器になることが期待できる．そこで近年，時制を考慮したデータマイニング手法や，次々と絶え間なく到着するストリームデータからリアルタイムに有益な情報を抽出する研究が盛んに行われている．

代表的なものとして，Topic Detection and Tracking(TDT) プロジェクトがある．これはオンラインニュースやニュース放送といったデータストリームから話題構造を自動的に獲得するための技術確立を目的としたものである．TDT プロジェクトには，時間順の一連のニュースストーリーから，ある話題に関する続報記事の抽出や，リアルタイムに，新しい事件や速報の発生を検出するなど，いくつかのタスクが定義されている．[1] このように Web 情報やニュースにおいては，時制の

[1]TDT プロジェクト (TDT2004) では，(1) 事象発見 (New event detection)，(2) リンク発見 (Story link detection), (3) 話題発見 (Topic detection)，(4) 話題追跡 (Topic Tracking) の 4 つのタスクが設定されている．

利用が知識獲得の性能の向上させるために非常に重要であり，時系列的関連の有効利用が注目されている．

1.2　扱う問題

データマイニングにおいては，より情報の多いデータの収集，加えて，テキストマイニングにおいては，非構造のデータからいかに優れた構造を抽出するか等，事前の KDD プロセスによりマイニング性能に大きな差が出てくる．優れたマイニングを行うためには，機械学習手法や，統計解析手法以外にも，どのようなデータをどのような形で，各マイニング手法に適用するかが性能に大きく影響する．

本稿では，シソーラスと時制の見地から，文書データを数値化する際に単語の意味を考慮しない為に生じる問題と，タイムスタンプが取得できない文書が，時系列データを対象とした手法に適用できない問題を扱う．

1.2.1　単語の意味を考慮しない文書データの数値化

基本的にテキストマイニングで行うことは，文書で示されている内容の統計的な分析である．すなわち，どのような内容が多いか少ないか，増えているか減っているか，あるいはどのような内容とどのような内容が統計的に関連性が高いかといった分析を行うことになる．

ここで，まずテキストデータを計算機で統計的に扱えるようにしなくてはいけない．そこで文書の内容として，どのような語句をどのような単位で抽出し，どのような重みをつけて数値化するかが分析の有効性に大きく影響する．ここで，一般的な文書のクラス分類やクラスタリングでは，“set of words”や “bag of words” と呼ばれる文書を単語の集まりとして考える方法が一般的である [15]．句単位での文書分類は単語単位に比べて良い性能は示さず，文書内での単語の出現順序は分類に重要な意味を持たない [6, 15]．

最小二乗法アルゴリズムや，Support Vector Machine(SVM) のような線形分類規則を用いる場合，文書 x は重み $x_1, ..., x_d$ をもった単語の連続として，ベクトル $\vec{x} = (x_1, ..., x_d)$ と表現される．ここで d は文書集合内で出現した単語の数である．また，各単語の重み（重要度）の指標として，$tf \times idf$ 値 (Term Frequency $\times$ Inverse Document Frequency) が良く用いられる．

一方，確率論による文書分類の場合，ある単語 x_k の文書集合全体での出現確率 $P(x_k)$ や，あるカテゴリ c_i での単語 x_k の出現確率 $P(x_k|c_i)$ など，訓練データから観測できる確率により，分類規則を生成する．また，文書ベクトル $x = (x_1, ..., x_d)$ の各単語の重み x_k の値を，文書内で単語 x_k が現れた時は $1(P(x_k = 1|c_i))$，現

れなかったときは $0(P(x_k = 0|c_i))$ とする二項独立モデル (Binary Independence Model)[7] が一般的である．

このように，通常の文書分類では単語の持つ意味などは考慮せず，単語を単に記号的に扱う．しかし，通常文書内には同じ意味を持つ複数の単語（同義語）や，複数意味を持つ単語（多義語）が存在するため，これは問題である．例えば，“student” と “pupil” はほぼ同じ意味を持っているにもかかわらず，通常の文書分類では，まったく異なったものと扱われてしまう．また，“bank” は「銀行」や「堤防」といった意味を持つが，これらの複数の意味は考慮されない．このように，同義語や多義語を含む文書に対して文書分類を行うと，異なるベクトル表現として処理される結果，分類の一貫性の低下や分類精度の低下が生じる可能性が高い [24]．

1.2.2　時系列データを扱うタスク

一般に, ニュースストリームを含め，オンラインで配信される文書データは, 配信される話題や内容が時刻に対応するものが多い．

TDT 等，時系列データを扱った手法では通常，配信される内容が時刻に対応しているため，各文書はタイムスタンプ（発行時間）あるいは発行順序にしたがって処理される．例えば，ある話題についての一連の流れを観測するタスクや，過去をさかのぼって，ある話題の最初の記事を見つけるタスクなのでは，タイムスタンプや順序が必要不可欠である．また，タイムスタンプを考慮することにより，精度の高いクラスタリングが可能になることが知られている [1]．そのため，これらのタスクでは, 全データの発行時間あるいは発行順序が取得可能である状態を想定しているため, タイムスタンプを持たないデータが存在する場合, そのデータはこれらのタスクに貢献することができないという問題がある．

また, 複数のソースから文書が配信されている場合, ソース間の速報性の差などから，全ソースが必ずしも同じ基準のタイムスタンプを持つとは限らない．内容時間と発行時間が大きく異なる文書は，これらのタスクにおいてノイズになる可能性が高い．そのために，TDT 等の各タスクにおける性能の低下が考えられる．

1.3　提案手法

同義語や，多義語の存在による精度低下を解決するためには，単語の意味を効果的に考慮して，文書を重み付けることが望ましい．

本稿では *WordNet* と呼ばれるシソーラス辞書を用い，同義語，多義語を考慮した文書分類手法を提案する．辞書が持つ単語の意味情報を効果的に用い，単語の意味単位で文書を重みづけることで，同義語だけでなく，多義語を考慮する文書

分類手法を構築する．シソーラス辞書とは単語ではなく，意味で整理した辞書を意味する．

本手法により，意味のあいまいさにより正しく分類できなかった文書や，今まで抽出できなかった構造を抽出することが可能となり，より高度な知識獲得の実現を期待できる．

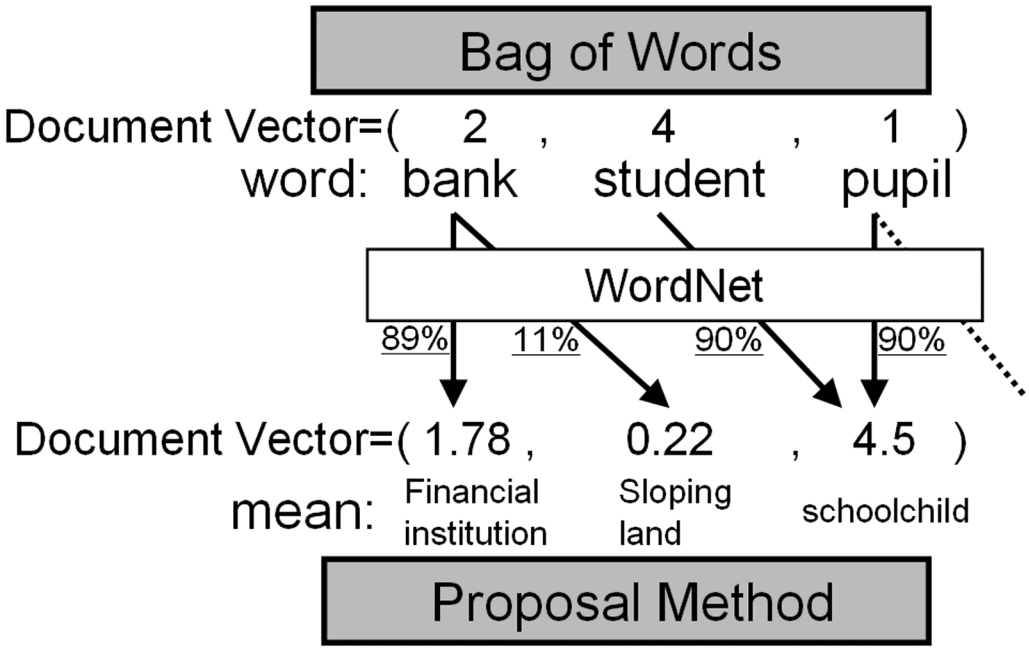


図 1.1: シソーラス辞書を用いた意味単位での重み付け

タイムスタンプを持たない文書を時系列データを対象としたマイニング手法に適用できるようにするために，文書のタイムスタンプ推定手法を提案する．タイムスタンプが既知の文書から事象 (Event) とタイムスタンプを学習し，忘却の概念を用いたタイムスタンプ予想曲線によりタイムスタンプを推定する．

Web やニュース情報ストリームにおいては，新しい話題の発生を抽出することなど，リアルタイム性が必要とされるタスクが少なくない．そこで，逐次処理的にタイムスタンプを推定する手法を提案する．

しかし，実際のニュースソースの状況においては，限られた話題や事象のみが配信される状態はまれであり，話題としての集合をなさない単発的な記事も非常に多数あることが考えられる．そのため，学習できる訓練データが十分でないことがある．このような状態は，逐次的な処理においてた完全に対応することは難しい．TDT タスク等において，逐次的な処理が要求されないタスクに対しては，一括処理 (batch) により，より正確なタイムスタンプの推定手法が適用可能と考えら

れる．そこで本稿では，訓練データが非常に少数である場合や，訓練データ（話題が既知のデータ）内に存在しない話題が出現する場合に対応したタイムスタンプ推定手法を提案する．ここでは，EM アルゴリズムを用いることにより，訓練データが不完全な状況に対応する．

本手法により，タイムスタンプを持たないデータや，ストリーム以外から配信される文書，また過去の事象について述べた文書など，内容時間と発行時間が大きく異なる文書のタイムスタンプを，その内容に基づき推定することが可能となり，今まで時系列上の文書として利用できなかった記事も利用することが可能になる．また，複数のソースを 1 つの時系列へと統一することが可能となる．これにより，よりスムーズに話題構造を把握することができ，TDT タスク等に非常に有用である．

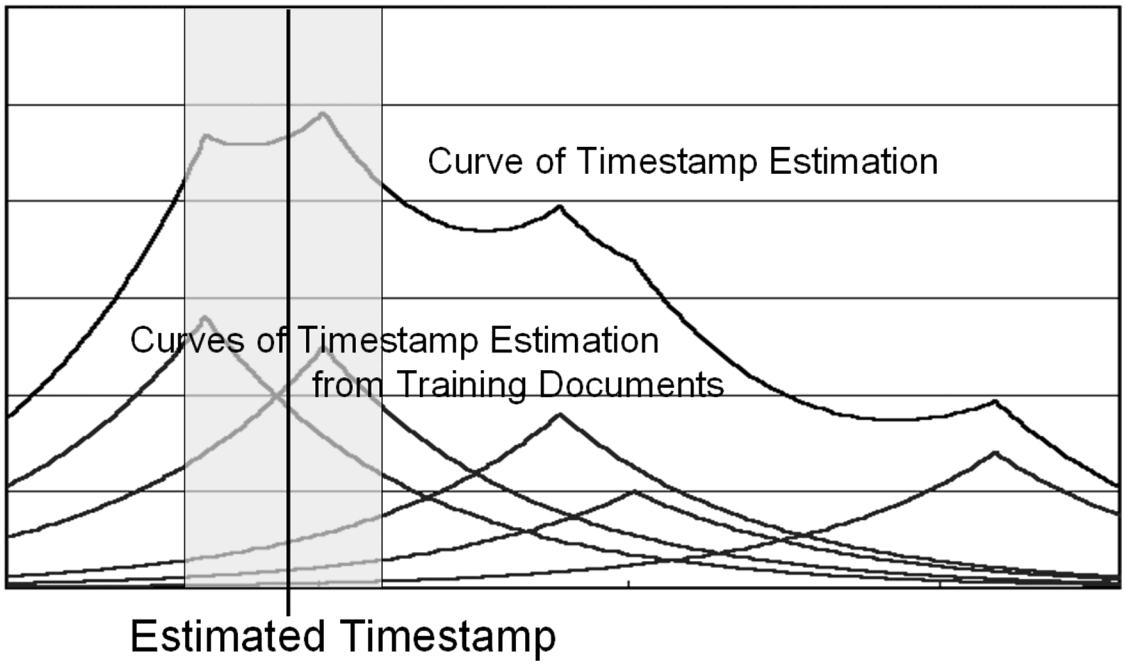


図 1.2: 忘却関数を用いたタイムスタンプの推定

1.4　論文の構成

本論文は，提案手法について以下の構成で論じる．

次章においては，同義語，多義語を効果的に考慮した文書分類手法を提案する．また実験により，多義語を考慮した手法が，単語を記号的に扱う場合，同義語の

みを考慮する場合の双方よりも優れた性能を示すことを証明する．なお，これは電子情報通信学会論文誌 VOL.J87-D1 No.2 などで発表した．

第 3 章では，逐次的に文書のタイムスタンプを推定する手法を提案する．また，本手法の有効性を TDT2 コーパスを用いた実験により証明する．これは筆者が IEEE in International Conf. on Tools with Artificial Intelligence (ICTAI) などにおいて発表した．

第 4 章では，学習できるデータが不完全である場合における，タイムスタンプの推定手法を提案する．TDT2 コーパスを用いた実験により，話題が既知のデータが，非常に少数である場合においても，本手法が有効であることを示す．これは筆者がデータ工学ワークショップ (DEWS) において発表した．

第 5 章では，本論文をまとめ，また本論文で扱えなかった課題について言及する．

1.5　発表論文

1.5.1　論文誌

1. 上嶋宏，三浦孝夫，塩谷勇: 同義語, 多義語の考慮による文書分類の精度向上, 電子情報通信学会論文誌 VOL.J87-D1 No.2 pp.137-144, 2004 年
   単純ベイズ法に基づいて類義性・多義性の双方を意識し、ワードネットを使用した文書分類を行う。

2. Uejima,H.，Miura,T.，Shioya,I.: Improving Text Categorization By Resolving Semantic Ambiguity, Wiley Systems and Computers in Japan, Vol 36,No. 4.
   単純ベイズ法に基づいて類義性・多義性の双方を意識し、ワードネットを使用した文書分類を行う。

1.5.2　研究発表 (査読付き)

1. 上嶋宏，三浦孝夫，塩谷勇: 同義語, 多義語の考慮によるテキストカテゴライゼーションの精度向上, データ工学ワークショップ (DEWS), 電子情報通信学会データ工学研究会，2003,
   単純ベイズ法に基づいて類義性・多義性の双方を意識した文書分類を行う。これまで知られている同種の分類と同等かそれ以上の性能をえることを述べる。

2. Uejima,H.，Miura,T.，Shioya,I.: Improving Text Categorization By Resolving Semantic Ambiguity, IEEE Pacific Rim Conference on Communications, Computers and Signal processing (PACRIM' 03), pp. 796-799, 2003,
   Wordnet を利用した Baysian 文書分類手法の提案。

3. 上嶋宏，三浦孝夫，塩谷勇: 時系列ニュース記事集合に基づくニュース記事の順序付け, データ工学ワークショップ (DEWS), 電子情報通信学会データ工学研究会, 2004,
   TDT2 コーパスを用いたタイムスタンプの推定アルゴリズムの提案。

4. Uejima,H.，Miura,T.，Shioya,I.: Giving Temporal Order to News Corpus, IEEE International Conf. on Tools with Artificial Intelligence (ICTAI), 2004,
   ニューススストリームの逐次クラスタリングから、未知の記事のタイムスタンプを推定する手法の提案と実験による検証を論じている。

5. Uejima,H.，Miura,T.，Shioya,I.: Giving Temporal Order to News Corpus (Extended Abstract), International Symposium on Computational and Information Sciences (CIS04), 2004,
   ニューススストリームの逐次クラスタリングから、未知の記事のタイムスタンプを推定する手法の提案と実験による検証を論じている。

6. 上嶋宏，三浦孝夫，塩谷勇: 不完全なニュース集合からのタイムスタンプ推定, データ工学ワークショップ (DEWS), 電子情報通信学会データ工学研究会, 2005,
   EM アルゴリズムを用いた，不完全なニュース集合からの，効果的にタイムスタンプの推定手法の提案.

第2章　同義語、多義語の考慮による文書分類の精度向上

2.1　前書き

本稿では, ベイズ学習に基づく文書分類法を提案する．ここでは同義語，多義語を利用し，単語の意味を考慮して，分類の一貫性の向上や分類精度の向上を目的としている．

文書分類（テキストカテゴライゼーション）とは，文書をそれが属するカテゴリへ割り当てることをいう．この文書分類では，構造化されていないテキストデータを扱う．構造化されていないデータを扱うには，ある単位での情報の抽出が必要である．文書分類では，“set of words”や “bag of words”と呼ばれる文書を単語の集まりとして考えるのが一般的である [15]．文書 x は重み $x_1, ..., x_d$ をもった単語の連続として，ベクトル $x = (x_1, ..., x_d)$ と表現される．ここで d は文書集合内で出現した単語の数である．句単位での文書分類は単語単位に比べて良い性能は示さない [6, 15]．通常の文書分類では，単語の持つ意味などは考慮せず，単語を単に記号的に扱う．また，文書内での単語の出現順序は分類に重要な意味を持たない．

通常，文書内には同じ意味を持つ複数の単語（同義語）や，複数意味を持つ単語（多義語）が存在する．同義語や多義語を含む文書に対して文書分類を行うと，異なるベクトル表現として処理される結果，分類の一貫性の低下や分類精度の低下が生じる可能性が高い [24].

本稿では，ベイズ学習を用いた文書分類法を提案する．一般に, ベイズ手法に基づく場合，高次元の入力により過学習が起こりやすいとされているが，本稿で用いる次元縮小法はこの問題を顕在化させない．本稿の目的は同義語，多義語を考慮する文書分類法を提案することにある．ワードネットを用いた実験を行い，その有効性を示す.

ワードネットを用いた関連研究としては福本ら [25] が代表的である ．ここでは，ワードネットにより，同義語クラスの上位関係を利用する手法を提案している．ワードネット内で，名詞は “entity”や “location”などの 25 種類の意味クラスに分類され，意味クラスごとに階層構造を形成している．各ノードは synset である．

この手法では，意味クラスごとに文書中の単語が unique beginner とよばれる root ノードからどの深さまでの synset を用いて表現するかのパラメータを設定し，各カテゴリの分類規則ごとにそれぞれの最適なパラメータを求め分類を行っている．しかし，単語が多義の場合は，各意味を示す階層構造のうち，unique beginner から単語までの深さが最大である階層構造を使用している．このため，多義語による精度低下の問題を解決していない．

また，Rodriguez らは，文書分類を Rocchio 法（関連フィードバック）や Widrow-Hoff アルゴリズムによる機械学習法で行うときに，ワードネットを利用する方法を提案している [14]．しかし、本稿の提案と異なり，“カテゴリを構成する単語” に対して同義語を展開し，テストに使う Reuters 記事そのものは同義語展開していない．

2 章ではベイズ学習と文書分類について述べる．3 章ではワードネットについて同義語，多義語に重点を置いて述べる．4 章で同義語，多義語を考慮した文書分類について述べ，5 章では実験結果を示し，6 章では結論を示す．

2.2　ベイズ学習による文書分類

2.2.1　文書分類

文書分類とは，文書をその内容に応じて，あらかじめ与えられたいくつかのカテゴリに自動的に分類することである．この技術は，文書検索，分類，ルーティング，クラスタリング，Web ページのディレクトリ構造の作成，電子メールなどのフィルタリング等に有用で，これらの作業を人手により行う場合に比べ，はるかに時間やコストを削減できる．

文書分類には，本来自然言語で書かれていることによるあいまいさ，高次元データ故の粗データ表現と計算量，個々の文書固有の表現方法の差異など多くの問題がある．特に，単語が同義語，多義語のようなあいまい性を持っており，実際に単語がどの意味として働いているかを理解することは困難で，分類精度を下げている要因と考えられる．本稿では教師付きベイズ学習を用いる．ベイズ学習は分類手段として広く使用されている手法である．教師付き学習とは，人手によりカテゴリを割り当てられたカテゴリが既知のデータ（訓練集合）から訓練集合の中に潜むパターンを学習し，そのパターンを用いて分類規則を作成し，その分類規則により未知のデータが属するカテゴリを予測するものである．

2.2.2　ベイズ学習

ベイズ学習は文書分類に有用な手段の一つである．ベイズ学習は「興味の対象となっている物理量は確率分布によって支配されており，最適な意志決定は，訓練データの確率を推論することで達成される」という考え方に基づいており，特徴は「最終的な仮定成立の確率計算に事前知識を使用する」という点にある．事前知識とは訓練データにおいての，仮説が成り立つかどうかの事前確率，訓練データにおいて，ある仮説が成り立つとした場合の，他の仮説が成り立つとする条件付確率の 2 つからなる．

ベイズ学習の利点は，仮説が成立する確率を明示的に扱い，確率的な仮説の表現を許すことにある．すなわち出力結果が真か偽ではない点が挙げられる．他方，欠点としては，事前確率分布のために多くの初期知識を必要とし，計算コストが大きいことが挙げられる．しかし，文書分類を行うとき，あらかじめ分類するためのデータは揃っており，量も十分にある．また，計算機の性能も計算量に対し十分であることが多い．

教師付きベイズ学習による文書分類の場合，訓練データから以下の 4 つを求める [6].

1:$P(c_k)$　文書がカテゴリ $c_k \in C$ に属する事前確率

2:$P(x)$　何の事前知識をもたないときに，訓練例において文書ベクトル x を観測する確率

3:$P(x|c_k)$　カテゴリ c_k に属するときに，文書ベクトル x を観測する条件付確率 (尤度)

4:$P(c_k|x)$　文章ベクトル x が観測されたという条件下でカテゴリ c_k に属するという事後確率

尤度 ($P(x|c_k)$) と事前確率 ($P(c_k)$) から結合確率 $P(c_k)P(x|c_k)$ を得る．結合確率を正規化し，以下のようなベイズルールを得る．

$$P(c_k|x) = P(c_k) \times \frac{P(x|c_k)}{P(x)} \tag{2.1}$$

カテゴリ集合 C の要素 c_k で，$P(c_k|x)$ の値が最も大きいもの（最大事後確率を与えるカテゴリ）は，以下のように示される．

$$\begin{aligned} c &= MaxArg_{c_k} P(c_k|x) \\ &= MaxArg_{c_k} P(c_k) \times \frac{P(x|c_k)}{P(x)} \end{aligned} \tag{2.2}$$

ベイズルールでは，最大事後確率を取るカテゴリ c_k を文書 x が属するカテゴリとすることで予想される分類エラーの数が最少になると考える．従って，ベイズ

ルールでの分類規則の作成は，訓練データから $P(x|c_k)$,$P(c_k)$,$P(x)$ の値を求めることである．

2.2.3　単純ベイズ分類

ベイズルールでは $P(c_k), P(x), P(x|c_k)$ の確率を組み合わせ $P(c_k|x)$ を求める．$P(x), P(x|c_k)$ で出現する文書ベクトル $x = (x_1, ..., x_d)$ は，ほぼすべての文書で異なり，極めて多数になることを想定する必要がある．このため単純 (naive) ベイズ分類では，ベクトル x を，すべての c_k に対して各要素 x_j を（同時的でなく）独立事象とみなすことで，計算量の削減を行う [7].

$$P(x|c_k) = \prod_{j=1}^{d} P(x_j|c_k) \tag{2.3}$$

この仮定により $P(x_j|c_k)$ は比較的少ないパラメタで構成され，$P(c_k|x)$ は以下のように表すことができる．

$$P(c_k|x) = P(c_k) \times \frac{\prod_{j'=1}^{d} P(x_j|c_k)}{P(x)} \tag{2.4}$$

また，本稿では 2 項独立モデル（BIM）を用いる．2 項独立モデルとは，文書ベクトル $x = (x_1, ..., x_d)$ のすべての単語の重み x_d の値を文書内で単語 x_d が現れた時は 1，現れなかったときは 0 とするものである [7]．2 項独立モデルを使うことにより，$P(x_j|c_k)$ は以下のように表すことができる．

$$P(x_j|c_k) = p_{jk}^{x_j}(1 - p_{jk})^{1-x_j} = (\frac{p_{jk}}{1 - p_{jk}})^{x_j}(1 - p_{jk}) \tag{2.5}$$

ここで $p_{jk} = P(x_j = 1|c_k)$ である．式（1）と式（5）より対数をとることにより，以下の式を得る．

$$\begin{aligned} \log P(c_k|x) = \log P(c_k) + \sum_{j=1}^{d} x_j \log \frac{p_{jk}}{1 - p_{jk}} \\ + \sum_{j=1}^{d} \log(1 - p_{jk}) - \log P(x) \end{aligned} \tag{2.6}$$

この 2 項独立モデルを利用することで，単純化されたベイズルールにより文書分類を行うことができる．2 項独立モデルでは，文書内での単語の出現回数や文書を占める割合を考える必要はなく，単語の出現の有無のみを調べればよい．

2.2.4　分類規則の次元縮小

文書分類において，高い次元 d を持つベクトル x は，粗な表現を生むことが多く，また多量の計算量を発生させるため問題である．通常，文書分類では，この問題解決のために x の次元 $|d|$ のサイズを縮小する試みを行う [15]．すなわち次元縮小とは，分類規則の単語を減らし，分類に使用する単語を限定したベクトルを生成することである．次元縮小は計算量の減少や，訓練データの過学習を避けるのに有用である．しかし次元縮小は用語を削除するので，潜在的に有用な情報が削除されるため，精度を低下させる可能性がある．次元縮小には，局所的次元縮小と大域的次元縮小の 2 つの方法が提案されている [15]．局所的次元縮小とは，それぞれのカテゴリ c 毎に異なった用語セット $\hat{d}_c$ を使用し，分類を行う．大域的次元縮小とは，すべてのカテゴリ上で同一の用語セット $\hat{d}$ を使用する．

本稿では “DIA asociation factor”[2, 15] と呼ばれる用語選択による局所的次元縮小を用いる．この方法は $P(c_k|x_j)$ の値の大きな x_j だけを分類規則に用いる．文書分類に使用する用語数としては局所的次元縮小の場合，$\hat{d}_c$ のサイズは 10 ないし 50 程度がよいとされている [15]．Reuter コーパス (新聞記事）における “gas” など 3 つのカテゴリ毎の $P(c_k|x_j)$ の大きな単語とその値を表 2.1 に示す．

カテゴリ	“gas”		“gold”		“fuel”	
1	0.825	gasoline	1.0	gold	1.0	fuel
2	0.6	oil	0.5	ounces	0.6923	pct
3	0.6	mln	0.4681	mine	0.6923	oil
4	0.5	crude	0.4468	pct	0.6154	dlrs
5	0.475	pct	0.4042	ton	0.5385	petroleum
6	0.425	year	0.4043	ounce	0.5385	corp
7	0.425	petroleum	0.3936	year	0.5385	barrel
8	0.35	fuel	0.3936	company	0.4615	prices
9	0.325	distillate	0.3723	mln	0.4615	crude
10	0.3	energy	0.3194	ore	0.4615	barrels
11	0.3	barrel	0.3085	production	0.3846	mln
12	0.275	stok	0.3085	mining	0.3846	heavy
13	0.275	product	0.3085	dlrs	0.3846	energy
14	0.25	refinery	0.2872	short	0.3846	effective

表 2.1: $P(c_k|x_j)$ の上位の単語とその値

2.3　ワードネットによる同義語，多義語の利用

本稿では語彙参照のためワードネットを用いる [10]．ワードネットはオンライン形式で語彙を参照することが可能であり「英語用語彙データベース (lexical database for the English language)」とも呼ばれる．

ワードネットは特定の分野の知識を持たず，一般的な知識を持つ．本稿では，特定のカテゴリに特化して優れた性能を示すのではなく，すべてのトピックに関して均等な性能を示す文書分類を対象としており，ワードネットを使用することは妥当である．

ワードネットはシソーラス辞書とは異なり，辞書を構成する基本単位として *synset*(synonym set) を用いる．synset とは同義語の集合であり，これより単語の意味や概念の記述および分類を行う．また synset には反義語，上位語，下位語，部分語，全体語などが定義されている．

本稿では，複数の単語が同じ概念を定義しているとき，これを同義語という．例えば “lofty” は “high” の同義語である．多義語は複数の意味を持つ単語をいう．例えば，“picture” は “movie”，“figure”，“photo”，“illustration” を意味する．当該語がどの意味を表現するかを考慮しないならば，明らかに同義語や多義語は検索や分類性能を低下させる．

図 2.3 に “human” をワードネットで検索した結果を示す．これより頻度や意味，同義語の単語の集合などを得る．またワードネットは品詞毎に結果を出力する．例えば，“human” は名詞と形容詞の両方に使われ，名詞としては “person” と “human” の 2 つの意味を持つ．図 2.2 では “person” が “human”，“body”，“grammatical category” の意味として使われる事を示す．通し番号の後ろの () 内の数字は，ワードネットの作成に使われた文書集合で使用された意味の数を表し，意味の使用頻度として扱うことができる．例えば，図 2.3 より “human” は文書集合内で $7 + 5 + ... = 94$ 回出現しており，そのうち 7 回は “person” という意味で使用されていたことを示す．また，使用頻度の低い意味は頻度がつけられていない．

これらの例が示すように，synset は複数の単語を含む場合があり，それらの単語は同義語である．また，複数の意味を持つ単語は，図 1 の単語 “human” のように複数の synset を持つ．また synset は固有の識別番号を持つ．

“human”，“person” はそれぞれ，synset 番号 5303 を持つ．この synset 番号は図 2.3，図 2.2 での第 1 番目の意味に対応している．即ち，“human” と “person” は 5303 という同じ意味を持つ．これらの単語が持つ識別番号を表 2.2 に示す．

The noun human has 2 senses (first 2 from tagged texts)

1. (7) person, individual, someone, somebody, mortal, human, soul -- (a human being; "there was too much for one person to do")
2. (5) homo, man, human being, human -- (any living or extinct member of the family Hominidae)

The adj human has 3 senses (first 3 from tagged texts)

1. (47) human -- (characteristic of humanity; "human nature")
2. (20) human -- (relating to a person; "the experiment was conducted on 6 monkeys and 2 human subjects")
3. (15) human -- (having human form or attributes as opposed to those of animals or divine beings; "human beings"; "the human body"; "human kindness"; "human frailty")

図 2.1: ワードネットによる検索結果 (human)

The noun person has 3 senses (first 2 from tagged texts)

1. (7229) person, individual, someone, somebody, mortal, human, soul -- (a human being; "there was too much for one person to do")
2. (11) person -- (a person's body (usually including their clothing); "a weapon was hidden on his person")
3. person -- (a grammatical category of pronouns and verb forms; "stop talking about yourself in the third person")

図 2.2: ワードネットによる検索結果 (person)

2.4　単語の意味を考慮した文書分類

2.4.1　同義語を考慮した文書分類

通常の文書分類では，単語の意味を考慮せず，単に記号的に解釈する．実験ごとに，訓練データ，テストデータの選択方法や件数，ストップワード（停止語) の選択等が多少異なるので，分類精度は一概には比較できないが，同義語，多義語を考慮しないベイズルールでの Reuter コーパスの分類精度は約 74 %〜79.5 %であることが知られている [15]．実際，筆者らが行った実験と同じ条件でも 79.26 %を

	“human”	“person”
名詞	1:5303	1:5303
	2:2130996	2:4465544
形容詞	1:324678454	
	2:2634237	
	3:2634331	

表 2.2: 単語が持つ意味番号

カテゴリ	gas			trade		
1	0.825　gasoline			0.930667　trade		
	意味番号	出現数	頻度	意味番号	出現数	頻度
	12402140	1	1	831317	394	0.502
				6962272	124	0.158
				844555	107	0.136
				454930	94	0.12
2	0.6　oil			0.586667　year		
	12653557	11	0.647	12867473	832	0.962
	3347615	5	0.294		・	
		・			・	
		・			・	
3	0.6　mln			0.477333　billion		
	mln	0	1	11604853	1	1

表 2.3: 分類規則にワードネットを適用

得た.

同義語を考慮した文書分類の研究も行われている [24]．例えば,「生徒」という意味の単語 “student” と “pupil” の両方が “school” というカテゴリの文書集合で出現していた場合，この 2 つの単語を同じと見なすことで，これらの単語の “school” カテゴリでの出現確率を増やすことができ，重要性を増すことができる．このように，意味を考慮して重みを与えることで，カテゴリに関係が深い単語の重要度が増加する．また「生徒」を “pupil” と表現する文書が少なく，単語 “student” のみが “school” カテゴリで重要性を持っている場合も,「生徒」を pupil と表現している文書に対してより多くの重要性を付与することができる．

しかし，同義語だけを考慮して文書分類を行った場合，使用頻度が低い同義語を扱うときに分類精度が低下することがある．例えば，文書内で “man” という多義の単語が出現したとき，この “man” は主に “人間（human)” という意味で使われているが，“チェスの駒” という意味も持つ．このため “piece” という単語を同義語として扱い，この単語に “man” と同様の重要性を与えてしまう．通常これらの単語はまったく違う意味で使われているので，分類精度の低下が生じる．

2.4.2　同義語，多義語と頻度を考慮した文書分類

本研究では単語の同義性だけでなく，単語が持つ多義性と，その複数の意味の使用頻度を利用した文書分類を提案する．ここで使用頻度とは，ワードネットにより得ることのできる意味の出現回数の総和に対する比率である（図 2.3，図 2.2）．2 項独立モデルでは，特徴ベクトル $x = (x_1, ..., x_d)$ の値を，すべて 0 か 1 で表現する．しかし，本研究では前述の多義語の使用頻度の低い意味による問題を解決するために，単語の意味の使用頻度を利用して文書ベクトル $x' = (x'_1..., x'_d)$ の重み x'_j を決定する $(0.0 \leq x'_j \leq 1.0)$．これにより (5）式を以下のように変更する．

$$P(x_j|c_k) = p_{jk}^{x'_j}(1 - p_{jk})^{1-x'_j} \tag{2.7}$$

$$= (\frac{p_{jk}}{1 - p_{jk}})^{x'_j}(1 - p_{jk}) \tag{2.8}$$

このため本稿での単語の重みは，もはやバイナリ値ではないが，文書内での単語の出現回数に基づくものでもない．重み付けの具体的な方法を以下に示す．

本稿では分類規則にワードネットを適用し，意味の出現頻度により文書ベクトルの重みを与える．しかし，1 つの単語は複数の意味を持つので，ワードネットにより多義語をその意味に分解することで，文書ベクトルの次元の増加が考えられる．このため本研究では，“DIA asociation factor”による局所的次元縮小を行った分類規則（表 2.1）にワードネットを適用する（表 2.3）．これにより単に多義語を意味に分解して分類規則を作る場合に比べ，次元の増加を抑制することができる．また，通常の文書分類と同じ分類規則を利用する．更に本稿では，テストデータにもワードネットを適用する．テストデータ内の単語を最大出現頻度を持つ synset へ置き換え，それぞれの単語の主要な意味のみを扱う（表 2.4）．特に，意味の頻度が分散している単語の分類への重要度は低いと見なし，最大頻度が低い単語の重要性は減少させる．

例えば，“oil is gasoline”という文を考える．“gas”カテゴリについては，表 2.3 より “oil”(最大頻度 id 12653557) が 0.647 回，“gasoline”(最大頻度 ID 12402140) が 1 回出現すると見なす．他に分類規則内の synset id の出現がなかった場合，文書ベクトル (oil,gasoline,...) は (0.647,1,...) となる．表 2.3 の “mln”のようにワードネットに生じていない単語が分類規則にある場合，頻度を 1 とし，通常の文書分類と同様，出現するか否かの 0 か 1 により重み付ける．

最大頻度の意味が同じ単語，例えば “student” と “pupil” の最大頻度の意味番号は両方とも 8734996 で同じである．テストデータは表 2.4 のように最大頻度の synset に置き換えるので，テストデータでこの 2 つの単語が出現した場合は全く同じものと見なす．

oil	12653557
gasoline	12402140
trade	831317
craft	454930
student	8734996
pupil	8734996

表 2.4: 最大の出現頻度を持つ synset

類似の意味を持つ単語がテストデータ内で出現するとき，例えば文書内に “trade” と “craft” が出現した場合，それぞれの最大頻度の意味番号は trade:831317，craft:454930 である．本稿では表 2.3 の “trade” カテゴリに関して，“trade” の 831317 により “trade” が 0.5019 回出現し，また “craft” の 454930 により同じく “trade” が 0.1197 回出現すると見なし，“trade” の出現回数 x'_{trade} は $x'_{trade} = 0.5019 + 0.1197 = 0.6216$ 回とする．本稿ではテストデータの単語が持つ最大頻度の意味と分類規則の単語が持つ意味を比較し，同じ意味を持つ場合，分類規則での意味の頻度を単語の類似度とする．この類似度により単語の出現回数を重み付け，これにより前述の意味のあいまい性を解決し，より精度の高い分類を得ることができる．

分類規則にワードネットを適用するとき，その意味番号と使用頻度を使用する．しかし，そのすべての意味を使用するのではなく，表 2.5 のように，頻度の高い上位 S 個の意味を使用する．これにより，上位頻度の意味の重みが増加し，意味の頻度が分散している単語にも比較的重要性を持たせることができ，前述の多義語の同義語の問題をより的確に扱うことができる．しかし，使用する意味を限定することは，使用頻度が S 番目以下の意味を無視することであるため，本実験では S=1，2，3，5 とすべての意味を使用する（S=ALL）計 5 パターンを比較する．以下でこれを S 値と呼ぶ．

2.5　実験と評価

2.5.1　実験に使用するコーパス

本実験には Reuter21578 を用い，ApteMod に従い訓練文書集合とテスト文書集合を作成する．本実験での Reuter の設定を以下に示す．この設定も以下の様に標準的手法に従う [6].

・Reuter の “TOPICS” を分類するカテゴリ，すなわち分類の答えとして扱う．

human		
S=ALL		
意味番号	出現数	頻度
2634237	47	0.5
2634331	20	0.2128
1220852	15	0.1596
5303	7	0.0745
	・	
	・	

(a)S=ALL の場合の頻度

human		
S=2		
意味番号	出現数	頻度
2634237	47	0.7015
2634331	20	0.2985

(b)S=2 の場合の頻度

表 2.5: S=ALL と S=2 の場合の頻度の変化

・“TOPICS”を持たない記事は使用しない.
・訓練集合で出現回数が 5 回以下の “TOPICS”にしか属さない記事は削除する
その結果，本実験で使用する訓練文書集合は 7907 件，テスト文書集合は 3081 件となり，総カテゴリ数は 73 となる．またストップワードについてはあらかじめ削除しておく．本稿では，通常のベイズ学習による文書分類（以下，通常の手法）と，提案手法での文書分類による文書毎の重み付けの変化や精度の変化を比較する．このため，しきい値の設定の必要がない文書主導の単一ラベル文書分類を行う．分類精度は正解率により評価する．

$$\text{正解率} = \frac{\text{正解した文書数}}{\text{全文書数}} = \frac{\text{正解した文書数}}{3081}$$

文書が複数のカテゴリ（答え）を持つ場合は，そのうちのどれか 1 つに当てはまれば正解とする．文書分類では精度と再現率による評価が一般的であるが，本実験では文書を必ず 1 つのカテゴリに割り当てているため，精度と再現率，F 値[1]は，正解率と同じになる．

2.5.2　実験手順

2-1 で述べたように，教師付き学習での文書分類の手順は一般に（1）訓練データ，テストデータの作成,（2）訓練データから分類規則の作成,（3）分類規則をテストデータに適用であり，最後に,（4）その結果から性能評価を行う.（1）,（2）は通常の手法，提案手法のどちらの場合も同じである．提案手法では通常の手法に

[1]文書分類の性能評価によく使われる値．精度（r）と再現率（p）から求める $F = \frac{2rp}{r+p}$

より作成された分類規則にワードネットを適用する．また 3 で使用するテストデータにもワードネットを利用し最大頻度の意味のみを考慮する．本実験では縮小した次元（語）数は 10，20，30，40，50 の 5 種類，また S 値は 1，2，3，5 と ALL の 5 種類の分析を行う．

2.5.3　実験結果

通常の手法による実験結果を図 2.3，表 2.6 に示す．結果は 4-1 で述べたように最高で 79.26 %の正解率を示した，このときの分類規則の次元数は 40 である．

分類規則の次元 (語)	10	20	30	40	50
正解率 (%)	76.43	78.94	78.77	79.26	78.77

表 2.6: 通常の手法による分類結果

次に提案手法による分類結果を図 2.3，表 2.7 に示す．この結果では S=5，分類規則の次元数が 50 のときに 82.31 %の正解率を示した．

通常の手法による分類で最高の正解率を示した分類規則は次元数 40 であった．これをこのままを使用した場合の提案手法での結果は，S=3 のときに 81.89 %の正解率を示した．

分類規則の次元 (語)	10	20	30	40	50
ワードネット (S=1)	76.40	78.77	79.42	80.59	80.69
ワードネット (S=2)	77.47	80.98	81.08	81.82	81.99
ワードネット (S=3)	77.83	81.17	81.08	81.89	82.02
ワードネット (S=5)	77.99	81.17	81.14	81.76	82.31
ワードネット (S=ALL)	78.03	81.78	81.21	81.82	82.18

表 2.7: 提案手法による分類結果

2.5.4　考察

図 2.3 及び表 2.7 からわかるように，提案手法では正解文書数が 2442 件から 2536 件と増加し，正解率が約 3.04 %向上する．誤分類された文書数は 639 件から 545 件と約 15 %減少している．この結果より，同義語，多義語とその頻度の考慮により，より正確に文書分類を行うことができる．

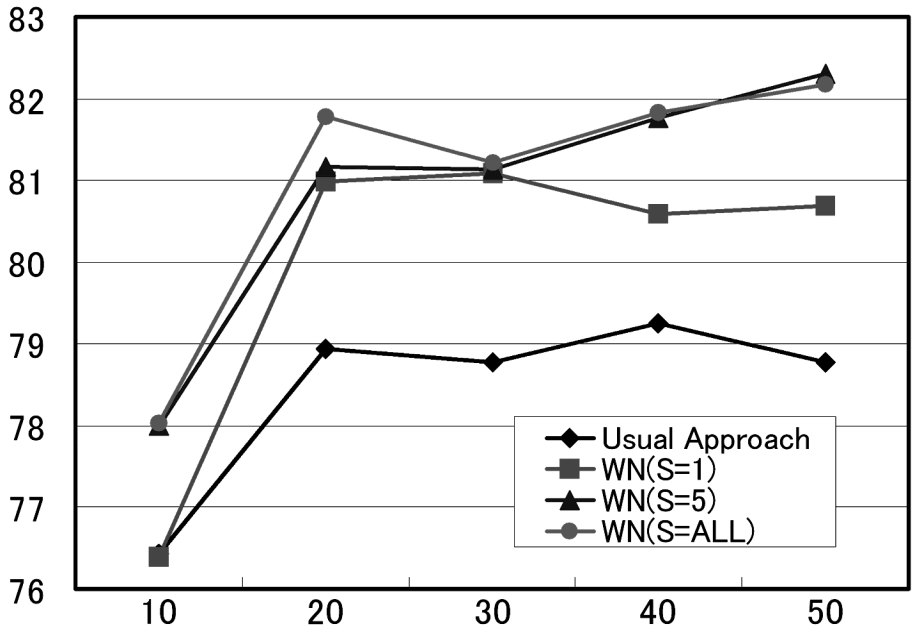


図 2.3: 分類結果

提案手法では，すべての S 値において，通常の手法の性能を上回った．また，$S \geq 2$ の場合は，ほぼ同じ値を示した．S=1 の場合，多義性を扱っておらず，最大頻度の synset しか考慮しないことに注意したい．この結果，単語の同義性と多義性を同時に考慮することで，より正確な分類が行えるようになる．本実験ではすべての意味を考慮した場合（S=ALL) の場合でも，大きな精度の低下は得られなかった．しかし，S=5 の場合に最大の精度を示したため，使用する意味を限定することは有効である．有効な結果を導くために，意味の使用頻度に対するしきい値として S 値を設定するなど，詳密な限定方法が必要である．

通常の手法と提案手法の正解文書の変化を表 2.8 に示す．

		通常の手法	
		正解	間違い
提案手法	正解	2339	197
	間違い	103	472

表 2.8: 通常の手法と提案手法の正解数，間違い数

3081 の文書中 2339 の文書に関しては，双方の手法のどちらでも正確に分類された．また，通常の手法では正確に分類され，提案手法により間違って分類された文書が 103 件あった．この内容を分析すると，提案手法により間違って割り当てら

れたカテゴリは，正解カテゴリに類似したものが多数あることがわかった（例えば正解のカテゴリが “trade” の文書を “yen” カテゴリに誤分類）．また正解カテゴリも必ず上位にランクされていた．この変化の原因としては，上記分析より（1）ワードネット作成に使われたコーパスと Reuter との意味頻度の違いなどコーパス間の相性によるもの（2）分類精度は手作業により割り当てられたカテゴリに依存する（3）手作業によるカテゴリ分類の基準のあいまいさ等の原因が考えられる．

双方の手法のどちらでも正しく分類できない 472 件のデータに関しては（1）他の文章と内容がかけ離れている（2）人手によるトピック割り当ての問題（3）ベイズルールによる分類の限界等の原因が考えられる．

2.6　結び

本研究では単語を単に記号的に認識する通常の文書分類とは異なり，同義語，多義語と，その意味の使用頻度を考慮する文書分類を行った．Reuter コーパスを使用した実験により評価した結果，従来の文書分類よりも有用性を示した．実際に，間違って分類される文書数が 15 %減少し，正解率も 82.3 %となることから，本手法の有用性を確認した．

今後はベイズ学習以外の学習アルゴリズムや，教師なしのクラスタリングへの本手法の適用を行う予定である．また，単語をワードネットで調べることは非常に大きな計算時間を要するので，ワードネットの高速化や，技術文書などの，より専門的な文書を対象にした性能評価などが今後課題となる．

第3章　時間によるニュース記事の順序付け

3.1　前書き

インターネットの普及により, ニュースなどの報道記事が複数のソースを通じて幅広く利用可能になっている. そこで大量の日々報道されるニュースの動向を容易に素早く把握するための研究が, 近年行われている. その代表的なものとして Topic Detection and Tracking (TDT) プロジェクトがある [1]. TDT はオンラインニュースなどの文書データストリームから話題構造を自動で抽出する技術の確立を目指すプロジェクトで, “話題分割”, “話題追跡”, “話題検出”, “事象検出”, “リンク検出”の 5 つのタスク設定している [11]. これらのタスクにおいては, 記事が時間順に並んでいること, あるいはタイムスタンプを持っていることが非常に重要である. 例えば, 話題追跡において, ある話題に関する続報記事を抽出するためには, 話題が時間とともに変化していく過程を的確に捉えなくてはならない. このように, ニュースや話題の動向を把握するには記事の時間順序を取得できることが必要である. そのため, 一般にこれらのタスクでは, あらかじめ時間順に並んだデータ, あるいはタイムスタンプを持っているデータを想定している. 逆に言うとタイムスタンプや時間順序を持たないデータは, このようなタスクに貢献できないことを意味する.

タイムスタンプを持たないデータのタイムスタンプを推定することができるなら, 今まで, 時系列上の文書として利用できなかった記事も利用することが可能になる為, データ空間が密になり, ニュースの話題や動向の変化をよりスムーズに把握することができ, 話題追跡や話題検出に非常に有用である.

通常, 文書は, その内容に関する時間（有効時間）に従って理解されるが、必ずしも文書の内容時間が文書の作成された時間（トランザクション時間）と一致するものではない. 文書の内容と生成時間に大きな差のある文書 (過去の事件を振り返った記事や, 月刊誌等) などの, タイムスタンプを予測することは, 生成時間ではなく, 記事の内容時間によりタイムスタンプを与えることを意味し, 複数のニュースソースを一つの時系列において順序付けることができる. この点におい

ても，タイムスタンプを推定することは非常に有用である．

そこで，本稿ではタイムスタンプを持たない文書 (ニュース記事)n にタイムスタンプを割り当てる手法を提案する．本稿ではタイムスタンプを持つ時系列文書集合 M を学習することにより，文書 n のタイムスタンプを推定する．基本的な考えとしては，文書集合 M 内で，n に最も近い文書 m_1 のタイムスタンプを n に割り当てる．これにより, タイムスタンプを持たない文書にタイムスタンプを割り当てる．しかし，n に一番近い記事が，常に同じ内容 (事象) を述べているとは限らず, このままでは，正しく内容を考慮してタイムスタンプを与えたことにはならない. そのため，本稿では，文書集合 M を事象により逐次クラスタリングし (事象検出), n を生成されたクラスタ C に割り当てる．これにより，n の示す事象 (属するクラスタ) を決定し，クラスタ C を用いて，n にタイムスタンプを割り当てる．

TDT での話題発見や追跡タスクにおいて，時間を考慮したクラスタリングは非常に高い性能を示す [18]．すなわち，ニュースの話題や事象の出現は時間に大きく依存することを意味する．さらに，ここで M はテキストストリームであり，時間とともに記事数が増加する．ストリームデータでは，常に最適な結果を取得可能であることが重要であるため，逐次的な手法により処理を行うことが必要である. そこで，本稿では，時間を考慮した逐次的な手法により文書集合 M のクラスタリングを行う．

本稿での一連の処理は，教師付き学習と考えることができる．M を逐次クラスタリングする事により生成されたクラスタは訓練データに対応し，n へのタイムスタンプの割り当ては、分類に相当する．ここで，n へのクラスタ (事象) 割り当ての精度やタイムスタンプ割り当ての精度は，M のクラスタリングの精度に大きく依存するために，この 2 つの割り当て精度と同様，クラスタリング精度についても評価を行う．

2 章で，文書ストリームデータ (ニュース記事) の逐次クラスタリングについて示し，3 章で，タイムスタンプを持たない文書のクラスタ (事象) への割り当て方法について示す．4 章で，タイムスタンプを持たない文書に対するタイムスタンプ割り当て手法を提案する，5 章で TDT2 コーパスを用いた提案手法の実験と結果を示し, 6 章において関連研究について述べたあと，7 章で結びとする．

3.2　逐次クラスタリング

最初に，本稿では逐次的な手法により，タイムスタンプを持った時系列文書集合 M を事象によりクラスタリングする．ここでのクラスタはタイムスタンプを考慮して作成され, 各クラスタはそれぞれ事象に対応する．プロセスは逐次的なので, アルゴリズムは現在のクラスタを維持し、追加されたデータだけを差分的に考慮

する．これにより，その都度，最適な結果を得ることが可能であり，また，タイムスタンプを持たない文書が来るたびに, タイムスタンプを推定することができ，話題追跡等へ利用することが可能になる.

3.2.1　文書表現

文書とクラスタの表現は，ベクトル空間モデルを利用する. 文書を単語の集合と考え，文書とクラスタは各単語を属性としたベクトルで表現する．本稿では単語をステミングし，BrillTagger により, 名詞と固有名詞のみを抽出して利用する [3].

文書ベクトル $\vec{X}$ の各属性の値は記事内での単語の重みで, 文書 $\vec{X}$ での j 番目の単語 t_j の値は, 単語 t_j の文書 X における出現頻度 $TF(j)$（term frequency;TF) で表す. 文書ベクトル $\vec{X}$ は以下のように示される

$$\vec{X} = \frac{(TF(1), ..., TF(n))}{\sqrt{TF(1)^2 + ... + TF(n)^2}}$$

ここでベクトルは $\sum_{j=1}^{n} TF(j) = 1$ のように正規化される.

クラスタはクラスタ内の文書の重心で表現する. ストリームデータのように逐次データが増える場合, TF・IDF 値による文書表現を利用するためには，新しい文書が追加されるたびに IDF 値の再計算が必要である．Yang ら [18] や，石川ら [5] は，話題検出タスク内での逐次クラスタリングで逐次 IDF を更新する方法を提案しているが, IDF を更新することは, 過去のクラスタリング基準が変る可能性があり, クラスタの割り当てやタイムスタンプ割り当ての結果が時間により変わる可能性がある．このため本稿では，IDF 値を利用しない逐次クラスタリングを行う.

3.2.2　単一パスクラスタリング

本稿では, タイムスタンプを持つ時系列文書集合を逐次クラスタリングするために，単一パスクラスタリング法を用いる．[13, 18]. 単一パスクラスタリング法は逐次クラスタリングに適しており，また，非常に単純な手法である.

この単一パスクラスタリングの実行手順を以下に簡単に示す.

1. しきい値 h を設定する
2. 最初は空の集合 S から始め,1 つ目のデータ X_1 だけからなるクラスタ C_1 を構成する
3. 次の文書 $X_i (i > 1)$ が追加されると，文書 X_i と既存の全クラスタ C との類似度 $sim(\vec{X_i}, \vec{C})$ を求める.

4. ここで、最も近いクラスタをC_jとし，その類似度をscとする．($sc = \mathtt{MAX}\{sim(\vec{X_i}, \vec{C})\}$)
   もし$sc > h$なら，X_iをクラスタC_{j}のメンバとし, クラスタC_jの重心を更新する．もし$sc < h$なら,X_iだけからなるクラスタC_{x_i}を新たに生成する．

5. 手順 (3)，(4) を繰り返す

文書XとクラスタCの類似度は余弦尺度と呼ばれる基準で求め，以下のように示される．クラスタCの重心をV_Cとあらわす．

$$sim(\vec{X}, \vec{C}) = \frac{\vec{X} \cdot \vec{V_C}}{|\vec{X}||\vec{V_C}|} \tag{3.1}$$

単一パスクラスタリングは，データが追加されるたびに，逐次クラスタリングを行い，その差分だけを計算する．すなわち, 一度決定されたクラスタ結果は永久に変更されることはなく，再クラスタリングも行わない．

3.2.3　忘却関数

本稿では, 文書とクラスタ間の時間距離を考慮した類似度計算を行うために，忘却関数$w_\lambda(t)$を用いる [5, 19]．ここでtは時間距離を示し，その単位は日数である．λは忘却速度を表し，$0 < \lambda < 1.0$で$w_\lambda(t) = \lambda^t$とする．

ある期間に集中して発生するニュース記事では, 2 つ文書の類似度が高くても, タイムスタンプが離れていると，同じ事象について述べている可能性は低くなる．逆にタイムスタンプの非常に近い 2 つの文書がある場合、文書の類似度は多少低くても同じ事象について述べている可能性は高い．

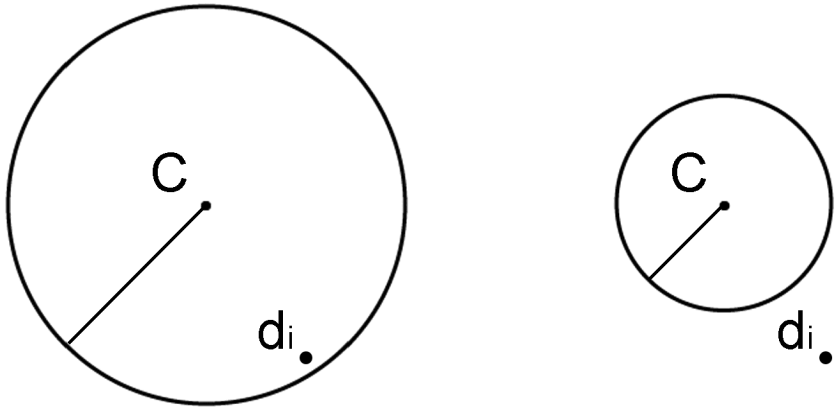
忘却関数を用いることにより，これらの問題を考慮する．文書とクラスタのタイムスタンプが離れているほど忘却関数の値は小さくなり，類似度を小さくする．図 3.1 は，$\lambda = 0.97$の時の，現在 ($w_\lambda(0)$) と 30 日後 ($w_\lambda(30)$) のクラスタの状態を示している．忘却関数により類似度が小さくなることはクラスタが小さくなることに対応する．

また, 本稿ではクラスタCのタイムスタンプ$time_C$はクラスタ内の一番新しい文書のタイムスタンプとする．“現在”を$time_{now}$とすると, 現在到着した文書XとクラスタCの忘却関数を考慮した類似度sim'は，以下のように与えられる．

$$sim'(\vec{X}, \vec{C}) = w_\lambda(|time_{now} - time_C|) \times sim(\vec{X}, \vec{C})$$

この類似度sim'を用いて, 上記の単一パスクラスタリングを行う．文書XとクラスタCの類似度sc'は以下のように求める．

$$sc' = \mathtt{MAX}\{sim'(\vec{X}, \vec{C})\} \tag{3.2}$$



(a)$\omega_\lambda(0) = 0$　　(b)$\omega_\lambda(30) = 0.5454$

図 3.1: 忘却関数によるクラスタと文書関係の変化

3.3　クラスタ割り当て

上記の，タイムスタンプを持つ文書集合 M の逐次的なクラスタリング結果に基づき，タイムスタンプを持たない文書 n が示す事象 (クラスタ) を決定する．

本稿では，文書 n が属するクラスタを，文書集合 M 内において，n との類似度の高い上位 10 個の文書による投票で決定する．つまり, この 10 個の文書内で一番多くを占めているクラスタを n の属するクラスタと決定する. 図 4.2 の場合、文書 n が属するクラスタはクラスタ C_1 となる．この方法を**上位 *10* 点法**と呼ぶ．また，上位 10 点法の有効性を評価するために，**最近点法**と比較する．最近点法とは，n に最も類似した文書 m が属するクラスタを n が属するクラスタとする，非常に単純な方法である．図 4.2 の場合，最近点法による n の属するクラスタは C_3 クラスタとなる．

本稿では，逐次手法によるクラスタリング結果を評価するために，一般的なバッチ手法によるクラスタリングとの比較を行う．ここで，バッチ手法として k-means 法を用い，逐次手法による単一パスクラスタリングの結果と比較する．ここで使用する k-means 法では，単一パス法と同様，余弦尺度と忘却関数を用い文書間距離を計算する．また，忘却関数で使用するパラメタ λ は単一パス法と等しい値とする．k-means 法の結果に基づいた文書 n のクラスタ割り当ても逐次手法と同様，最近点法と上位 10 点法により行う. (2 つのクラスタリング手法を各 2 つの割り当て方法による計 4 パターンを比較する．)

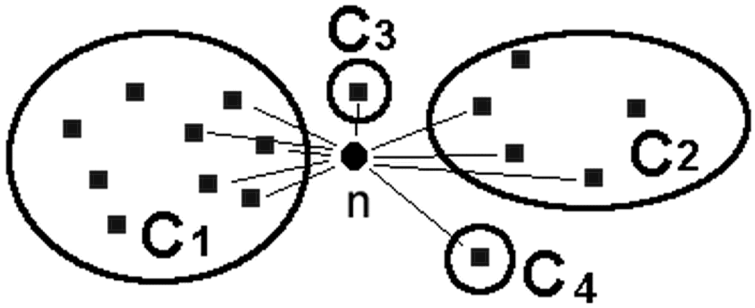


図 3.2: 文書ベクトル $\vec{n}$ の投票によるクラスタ割り当て

3.4　タイムスタンプ割り当て

本稿では，上記のクラスタリング結果と，文書 n のクラスタ割り当ての結果に基づいた，文書 n の示す事象を考慮した，逐次タイムスタンプを割り当てる手法を提案する．また，本稿ではニュース記事の内容時間と生成時間は一致すると考える．上位 10 点法と最近点法のそれぞれの基づき，タイムスタンプを割り当てる．本稿では，上位 10 点法と最近点法の割り当て方法はそれぞれ異なり，2 種類の割り当て結果を比較する．

最近点法によるタイムスタンプの割り当ては，前章と同じく，n に最も近い文書を n のタイムスタンプとする非常に単純な手法である．図 4.2 の場合、最も近い，クラスタ C_3 内の文書のタイムスタンプを n のタイムスタンプとする．この場合のタイムスタンプ割り当ては，文書集合 M のクラスリング結果には依存しない．

次に上位 10 点法によるタイムスタンプ割り当て方法について述べる．上位 10 点法では，文書 n に, 最も近い 10 個のデータのうち, データ n が属するとしたクラスタ C に属するデータ集合 T_C を用い，タイムスタンプを割り当てる. 図 4.2 の場合，文書 n に近い上位 10 個の内，クラスタ C_1 に属する 5 つの文書から n のタイムスタンプを予想する．

データ集合 T_C 内の 1 つのデータ t_{C1} による n のタイムスタンプの予測 $TS_{n,t_{C1},\lambda}$ を以下の式により与える．

$$\begin{aligned} & TS_{n,t_{C1},\lambda}(date) \\ & \quad = sim \quad (t_{C1}, n) \times distr(C, time_{t_{C1}}) \\ & \quad \times w_\lambda \quad (|time_{t_{C1}} - date|) \end{aligned}$$

ここで $distr_C(time_{t_{C1}})$ は，クラスタ C に属する文書集合のタイムスタンプ分布の時間 $time_{t_{C1}}$ における密度を示す．この式は，文書 n と t_{Ci} との類似度と，クラ

スタ内のタイムスタンプ分布密度の積を頂点とした時間距離により減少するタイムスタンプ予想曲線を与える．

ニュース記事において, 大抵のニュースはある期間に集中して起こる特性があり, タイムスタンプ分布を考慮することは重要である [1]. クラスタ内のタイムスタンプの分布は n が持つタイムスタンプの発生確率と考えることができる．そのために，本手法では，タイムスタンプ分布を考慮したタイムスタンプ予想曲線を T_C 内の各データ t_{Ci} 毎に求め総和を取る。すなわち, 以下の式になる．

$$TS_{n,T_C,\lambda}(date) = \sum_{t_{ci} \in T_C} TS_{n,t_{ci},\lambda}(date) \tag{3.3}$$

図 4.2 の場合，T_{C_1} 内の 5 つの文書から，図 3.3 のようなタイムスタンプ予想曲線を得る．

ここで，タイムスタンプ割り当ての精度を評価するために，タイムスタンプの**誤差許容範囲**を設定する．これは，実際のタイムスタンプ（正解）と，予測したタイムスタンプの差の許容範囲であり，この許容範囲に従って予想曲線からタイムスタンプを求める．すなわち，予想曲線の誤差許容範囲内の総和が最大になるタイムスタンプを文書 n に割り当てる．n のタイムスタンプ $date_n$ は以下の式で表すことができる．

$$date_n = \mathtt{MaxArg}_{date} \int_{date-m}^{date+m} TS_{n,T_C,\lambda}(date) \tag{3.4}$$

例えば許容誤差範囲が m 日の場合，前後 m 日間のタイムスタンプ予想曲線の総和が最大になる日付を文書 n のタイムスタンプとする [図 3.3].

本稿では，逐次手法，バッチ手法の双方に基づいた上位 10 点法によるタイムスタンプの割り当てと，最近点法に基づいたタイムスタンプ割り当ての，計 3 つのタイムスタンプの割り当て方法を比較する．

3.5　実験

3.5.1　TDT2 コーパス

本稿では, 実験に TDT2 コーパスを用いる [17]. TDT2 コーパスは，放送されたニュース (音声) を書き写したものと，ニュース通信の 2 種類からなり，それぞれ 1998 年 1 月から 6 月までの 6ヶ月間分のデータからなる．また，TDT コーパスには英語と中国語のデータが含まれるが, 今回は英語の記事のみを利用する．放送データとして ABC，CNN，VOA，PRI，ニュース通信として APW，NYT の計 6 つの

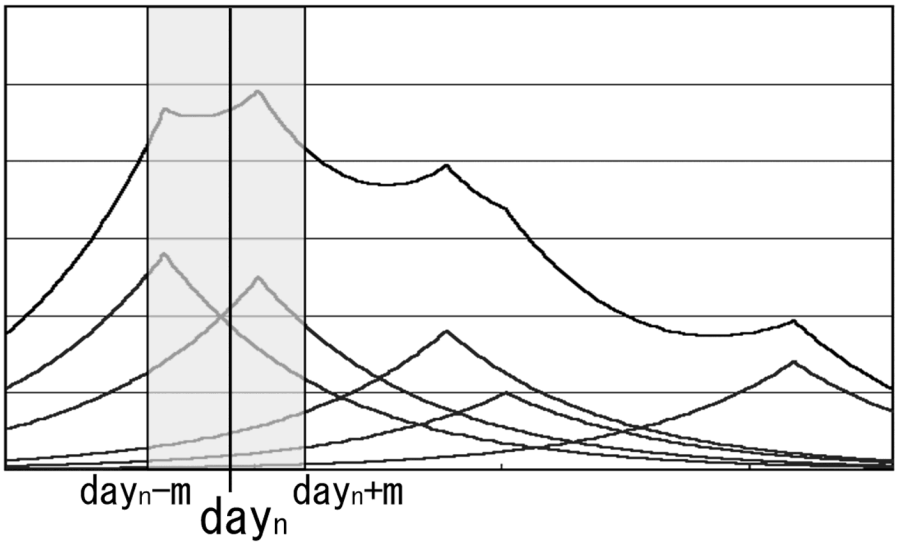


図 3.3: 5 つの文書によるタイムスタンプ予想曲線

ニュースソースを利用する．本稿では，放送データは一般的に速報性が高く，内容時間と発行時間は一致すると考えるため，放送データをタイムスタンプを持ったデータ（訓練データ）とする．そして，放送データに比べ比較的速報性が低いと思われるニュース通信のデータを, タイムスタンプを予測するデータ (テストデータ) とする．そして, 実際のタイムスタンプと予測結果を比較することにより, 提案する手法を評価する.

また, 本稿では, 放送記事, ニュース通信記事の双方に 200 件以上の YES タグが付与された記事を持つ 4 つの話題（$T_1 - T_4$) を利用して実験を行った。実験に使用した話題とそのデータ数を以下に示す.

	話題	放送	ニュース通信
T_1:	アジアの経済危機	476	657
T_2:	モニカルインスキ事件	747	222
T_3:	1998 年冬季オリンピック	222	318
T_4:	イラクとの対立	1022	464
	合計	2467	1661

3.5.2　実験手順

本稿では TDT2 コーパスの発行時間によるタイムスタンプ割り当てを行う．逐次方式，バッチ方式の 2 つの方法による上位 10 点法と，最近点法の合計 4 つの方

法をに基づいたクラスタ割り当てと，2 つの上位 10 点法と最近点法の計 3 つのタイムスタンプ割り当てを行い，比較する．

時間 tm までの放送データをクラスタリングするごとに，tm 以前のすべてのニュース通信データのタイムスタンプ割り当てを試みた．また，評価は 1 月単位で行う．すなわち，最初の 1 月分の全放送データを「単一パス法により発行時間順に」あるいは「k-means により一括で」クラスタリングした時点で，1 月のニュース通信記事のタイムスタンプ割り当てを行う．次に，2 月の終わりまでの全放送データをクラスタリングした時点で，1 月，2 月両方のニュース通信記事のタイムスタンプ割り当てを行う．これを 6 月分まで繰り返す．k-means 法による結果のほうが，全体を考慮したクラスタリング結果を得ると考えられる．

3.5.3　評価方法

本稿では 3 つの評価を行う．まず最初に，M の “クラスタリング”結果について評価を行う.「逐次法」によるクラスタリング結果を,「バッチ法」による結果,「TDT2 コーパスにあらかじめ与えられている話題」のそれぞれと，どれほと一致しているを評価する．2 番目に，“事象割り当て”について評価を行う.「逐次法」による文書 n のクラスタ (事象) 割り当て結果が,「バッチ法」,「TDT2 に与えられている話題」のそれぞれと，どれほど一致するかを評価する．最後に “タイムスタンプ割りあて”について，予測したタイムスタンプが許容誤差範囲にどれほど収まっているかの評価を行う．これらの 3 つの精度の評価を “クラスタリング精度”，“割り当て精度”，“タイムスタンプ精度”により行う．

最初に，クラスタリング精度による逐次クラスタリングの評価方法について述べる．

ここで，単一パス法によるクラスタを $C_1 - C_x$ とおき，k-means 法によるクラスタを $K_1 - K_y$ とする．まず，クラスタ K_i が $C_1 - C_x$ のどのクラスタに対応するかを決定する．ここでは，K_i 内の文書集合の属する割合がもっとも多いクラスタ C_j を K_i が対応するクラスタとする．図 3.4 の場合，K_1, K_2 は両方とも C_1 に対応する．逐次，バッチのそれぞれのクラスタリング結果に基づき，文書 n が属するクラスタ C_j と K_i を比較して，K_i の対応するクラスタが C_j ならば正解とする．

[1]α を全文書数とし，β を正解文書数とすると，バッチ方式との比較による**クラスタリング精度**は β/α と定義される．図 3.4 の K_2 において，クラスタリング精度は 14/20 となる．

また，TDT2 のあらかじめ与えられている話題と比較することで，各クラスタが，話題を細分化した事象にどれほど正しく対応しているかを評価する．クラスタ

[1]本実験では k-means 法の k の値は単一パス法により生成されたクラスタ数の 2 倍とした．

C の対応する話題を，クラスタ内の文書が示す話題のうち，最も多い話題を，そのクラスタが対応する話題とした．このために β をクラスタに属する記事の内，記事が示す話題と，クラスタの対応する話題が同じである記事数として，話題との比較による**クラスタリング精度**は β/α と定義される．

クラスタリング評価は一ヶ月毎にその時点でのクラスタリング結果を評価する.[表 3.1]

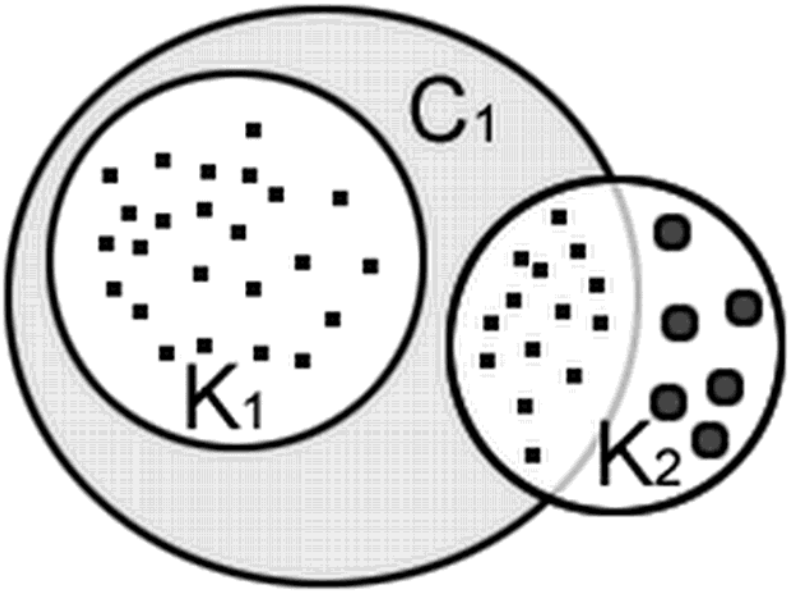


図 3.4: クラスタリング精度の例

次に，テストデータへの事象（クラスタ）割り当ての評価を行う．
「逐次方式」を「バッチ方式」,「TDT2 の話題」の双方においてそれぞれ上位 10 点法と，最近点法の計 4 つの比較に基づいた，クラスタ割り当て結果を評価する．

逐次法とバッチ法を比較する場合の比較方法は，文書 n が k-means 法に基づき割り当てられたクラスタ K_i と，逐次法に基づき割り当てられたクラスタ C_j のそれぞれが，同じ話題に対応していれば $(K_i = C_j)$ 正解とする．この割り当ては**割り当て精度**により評価し，α を文書数，γ を正しく割り当てられた $(K_i = C_j)$ 文書数とすると，バッチ法との比較による**割り当て精度**は γ/α と定義される．

話題との比較する場合，文書 n に割り当てた事象 (クラスタ）[図 4.2] の対応する話題が，文書 n にあらかじめ与えれている話題（答え）と一致するかどうかを評価する（話題追跡）．[表 3.2] ここで全文書数を α とし，γ を正しい話題が割り当てられた記事数とすると，話題との比較による**割り当て精度**は γ/α と定義される．

最後に，**タイムスタンプ精度** によりタイムスタンプ割り当ての評価を行う．逐次手法とバッチ手法それぞれのクラスタリング結果に基づいた上位 10 点法による

結果と，最近点法による結果の計 3 つを比較する．[2]α を全記事数とし，δ を正しくタイムスタンプが与えられた記事の数とすると，タイムスタンプ精度 はδ/α と定義される．[表 3.3,3.4]

ここで，3 章で示した許容誤差範囲を指定する．許容誤差範囲は 1 週間（7 日）と 1ヶ月 (30 日）とする．つまり，予想されたタイムスタンプと実際のタイムスタンプの差が 1 週間以内なら正解 [表 3.3 図 4.2], 1ヶ月以内なら正解 [表 3.4 図 4.1] とする計 2 パターンの許容誤差範囲を比較する．

1 月毎にタイムスタンプ割り当ての性能を評価するが，日数が増えるごとに，タイムスタンプを割り当てる範囲が広くなるので，タイムスタンプ精度は低下すると考えられる．

3.5.4　実験結果

最初に，クラスタリング精度を表 3.1 に示す。

	Jan.	Feb.	Mar.	Apr.	May	June
k-means(%)	86.16	79.81	77.72	81.98	79.26	88.68
topic(%)	96.89	96.11	96.38	96.55	96.7	97.04

表 3.1: クラスタリング精度

これより，非常によい精度を得ていることがわかる．ここでは，単一パスクラスタリングにより生成されたデータ集合のうち，データを 5 つ以上含む集合をクラスタとみなした．また、6 か月分クラスタリングした後の最終的なクラスタ数は 21 クラスタであった．すなわち K-means 法では最終的には $k = 42$ としてクラスタリングを行った．

次にクラスタ割り当て精度の結果を示す．[表 3.2，図 3.5]．ここにおいても、高い精度を得た．

続いて，これらの結果に基づいたタイムスタンプ精度を，誤差を 1 週間とした場合 [表 3.3, 図 4.2] と 1ヶ月とした場合 [表 3.4, 図 4.1] のタイムスタンプ精度をそれぞれ以下に示す．

[2]最近点法は逐次，バッチにかかわらず同じ結果しか生成しないためここでは手法差を評価しない。

		Jan.	Feb.	Mar.	Apr.	May	June
k-means	TOP10	83.72	85.78	75.95	81.29	81.93	86.03
k-means	NN	80.23	83.74	71.57	75.83	74.62	74.59
topic	TOP10	93.80	96.62	96.53	96.67	97.07	96.81
topic	NN	95.16	96.31	95.90	96.40	96.25	96.69

表 3.2: 割り当て精度

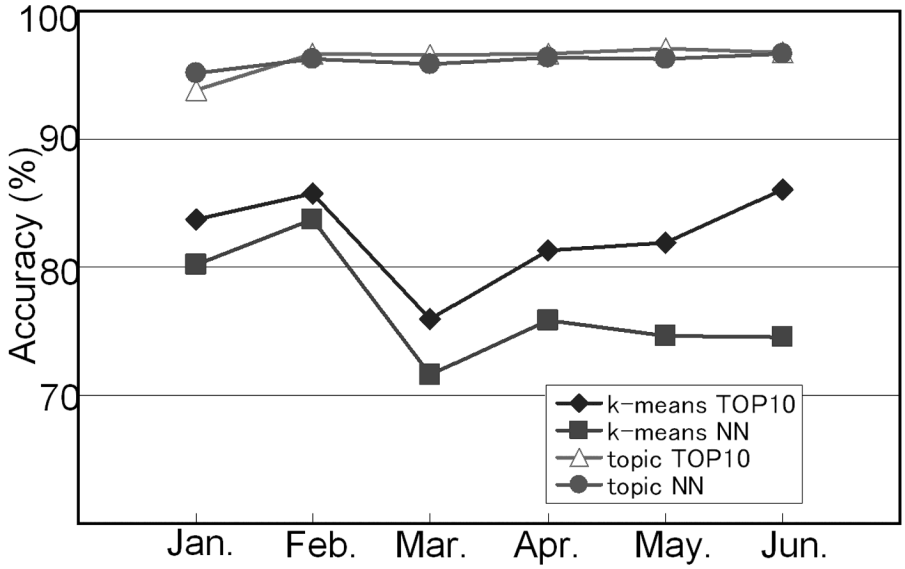


図 3.5: 割り当て精度

3.5.5　考察

表 3.1 のクラスタリング精度より，バッチ手法による結果と比較しても，80%以上一致し，非常に高い精度でのクラスタリングが可能なことがわかる．また，話題と比較した場合も，常に 96%以上の正解率を得ていることから，話題を細分化した，事象によりクラスタリングができていることがわかる．

表 3.2 より，逐次手法により，テストデータ n に割り当てた事象の正解率も，バッチ手法の場合と 80%程度一致しており，高い精度で，事象の割り当てが可能であることがわかる．表 3.1 により，事象（クラスタ）が正しく話題を細分化しているため，話題割り当ての正解率も，95%以上を示している．

これらより, 逐次, バッチどちらの場合でもほぼ同性能のタイムスタンプ割り当て結果を得た.[図 4.2, 図 4.1] タイムスタンプ精度は誤差範囲が 7 日で, 6 か月分のデータにタイムスタンプを割り当てた場合, 精度が約 50%と高い性能で，タイムス

		Jan.	Feb.	Mar.	Apr.	May	June
	NN	70.93	57.11	51.91	48.27	44.91	43.59
k-means	TOP10	69.19	59.15	53.96	50.93	49.55	46.6
incremental	TOP10	73.06	60.88	55.30	53.00	49.81	47.38

表 3.3: タイムスタンプ精度（誤差許容範囲 7 日）

		Jan.	Feb.	Mar.	Apr.	May	June
	NN	100	93.72	87.27	80.83	75.13	72.55
k-means	TOP10	100	96.78	90.95	83.69	81.36	77.24
incremental	TOP10	100	98.59	91.09	85.62	82.12	78.33

表 3.4: タイムスタンプ精度（誤差許容範囲 30 日）

タンプを割り当てることが可能となった. また，誤差範囲が 1ヶ月の場合，最低の場合でも 70%以上の精度でタイムスタンプを割り当てることができた.

上位 10 点法と最近点法によるタイムスタンプ割り当てを比較した結果, 上位 10 点法による結果の方が優れた性能をもち, この手法を取ることは妥当であるといえる. TDT コーパスにおいて，事象は時間に深く依存している．上位 10 点法は，事象に基づいてタイムスタンプを割り当てているために，図 4.2，図 4.1 より，最近点法に比べ，上位 10 点法は割り当て期間の増加による影響が少なく，期間の増大に対応できることがわかる.

ここで表 3.5 に，6ヶ月分のデータを逐次クラスタリングした後の，クラスタ割り当て精度とタイムスタンプ割り当て精度を話題ごとに示す.

これより，タイムスタンプ割り当て精度が，クラスタ割り当て精度に依存していることがわかる．タイムスタンプ割り当て精度が低い T_1 は，文書発生頻度の時間的ピークがあまりない話題であった．時間を考慮し，クラスタリングしているために，ピークを持たない話題は，事象により細分化することが困難であると考えられる．しかしながら，ニュース記事の特性として，短期間に集中して現れるという特性があり，T_1 のような，ピークの分散している話題は少ない [18].

次に，話題の割り当てとタイムスタンプ割り当ての関係について考察する．逐次的な上位 10 点法により全文書データのタイムスタンプ割り当て処理を行ったとき，誤差範囲 7 日の場合において，正しく話題が割り当てられた 1608 件中，誤ったタイムスタンプが割り当てられた文書は，約 48%の 778 件であるのに対し，間違った話題が割り当てられた 53 件の文書中，約 83%を占める 44 件の文書が間違ったタイムスタンプを割り当てられた．誤差許容範囲 30 日の場合でも，同じような

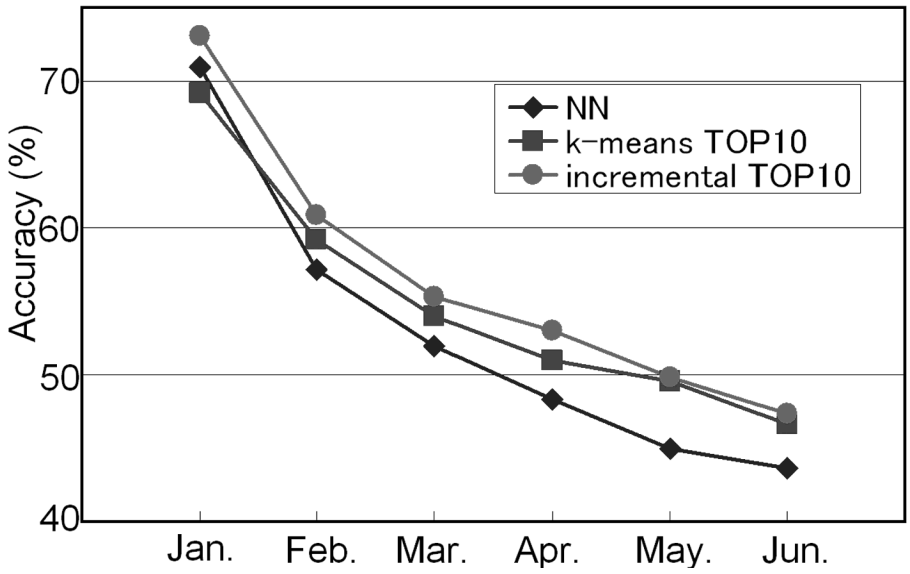


図 3.6: タイムスタンプ精度（誤差許容範囲 7 日）

Topic	event accuracy	Error 7 days	Error 30 days
T_1	94.67	24.66	63.77
T_2	97.3	48.65	74.32
T_3	98.74	72.96	96.23
T_4	98.28	61.42	88.58

表 3.5: 割り当て精度とタイムスタンプ精度

傾向を見ることができ，誤ったタイムスタンプを割り当てられた文書には例外的な文書の存在も考えられる．

本実験では，単一パス法において，しきい値 $th = 0.24$, 忘却関数 $\lambda = 0.97$ の時にもっとも良い結果を示した．しかし今回の実験では, クラスタリングに関して，忘却関数は結果にはそれほど大きく影響しなかった．これは，TDT2 コーパスは 6ヶ月間という期間のデータで，話題の時間的大きさに対して，データの期間が短いために，あまりうまく働かなかったようである．今後より長い期間のデータによる評価も必要と考えられる．

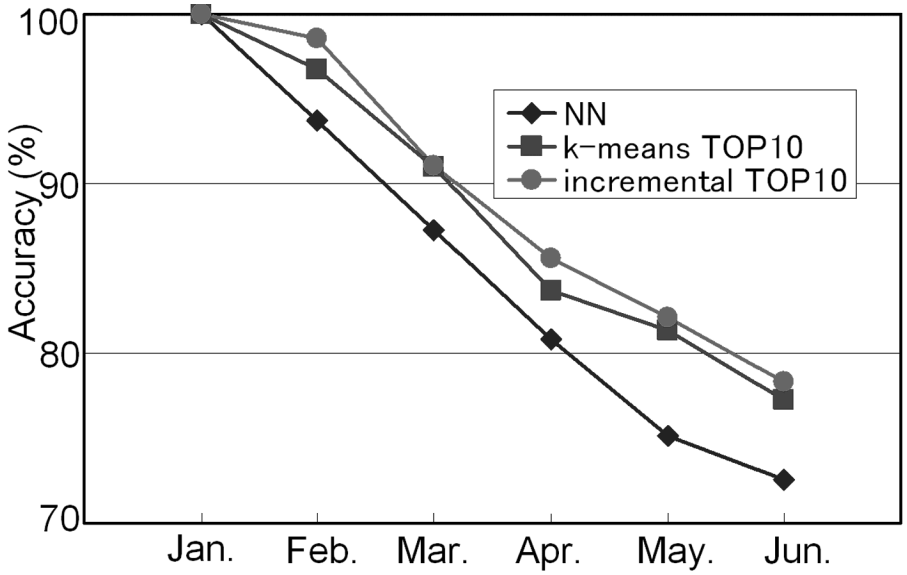


図 3.7: タイムスタンプ精度（誤差許容範囲 30 日）

3.6　関連研究

文書に時間を割り当てる研究として，Mani らの研究がある [9]．Mani らは, 文書内の時制表現を抽出する手法を提案している．ここでは文書の発行時間から，文書内の時制表現が示す相対的な時間を抽出し, 文書内の文の時間を求めるが，他の文書との関係や, 文書自体の発行時間は考えない.

また，Papka らは Inquery による単一パス法に基づいた話題検出法を提案している [13]．しきい値を時間距離により設定し逐次クラスタリングする点は，本稿で提案するクラスタリング手法と類似しているが，文書と質問の類似度の求め方やしきい値の設定方法は異なる．本稿では IDF 値は使用しないが，Papka らは，補助コーパスから求めた IDF 値を使用している．

石川らは，忘却の概念を利用した，逐次的な文書クラスタリング手法を提案している [5]．C^2ICM という文書クラスタリングアルゴリズムによる逐次クラスタリングを行っている．ここでは，少ない計算量での差分によるクラスタの更新を行っている．

これらの手法のほとんどは，すべての文書がタイムスタンプを持っていることを想定しており，時系列文書とタイムスタンプを持たない文書を同時に扱う手法はあまり提案されていない.

3.7　結び

本稿では, ニュースストリームを実時間でクラスタリングしつつ、タイムスタンプを持たないデータにタイムスタンプを割り当てる手法を提案した．TDT2 コーパスによる実験の結果，1 週間という狭い誤差許容範囲にもかかわらず, 50%程度の精度でタイムスタンプを割り当てることが可能であった。

今後は，忘却関数でのパラメタ λ や，しきい値 th などのパラメタ設定手法の提案が必要であると考えられる．また，記事の内容時間よる並べ変えの処理を行うことで，複数のソースを一つの時系列へと統一することにより，今までに抽出できなかった情報を抽出することを考えている．

第4章　不完全なニュース集合からのタイムスタンプ推定

4.1　前書き

近年，インターネット等を通じて利用可能な電子文書やメールマガジン，オンラインニュースが増加している．これまで，次々と配信される大量の文書データから，有用な情報を抽出する為のクラスタリングや要約手法が提案されている．その中でも，ニュースストリームから話題構造を自動的に獲得し，動向を容易に把握することを目的とした話題抽出と追跡（Topic Detection and Tracking, TDT) プロジェクトがある [11]．TDT2004 では，(1) 事象発見 (New event detection)，(2) リンク発見 (Story link detection), (3) 話題発見 (Topic detection)，(4) 話題追跡 (Topic Tracking) の 4 つのタスクが設定されている．

一般に, ニュースストリームを含め, オンラインで配信される文書データは, 配信される話題が時刻に対応するものが多い．そのため, これらは時系列文書と呼ぶことができる．配信される話題が時刻に対応しているため，時系列文書のクラスタリングや, TDT の各タスクにおいて, 各文書はタイムスタンプ（発行時間）あるいは発行順序にしたがって処理される．また，タイムスタンプを考慮することにより，精度の高いクラスタリングが可能になることが知られている [1].

しかし，これらのタスクでは, データの発行時間あるいは発行順序が取得可能である状態を想定しているため, タイムスタンプを持たない場合, そのデータはこれらのタスクに貢献することができない．また, 複数のソースから時系列文書が配信されている場合, 各ソースにより速報性に差がある場合があり, 必ずしも同じタイムスタンプを持つとは限らない.

本稿では，ストリーム以外から配信されている文書や，内容時間と発行時間が大きく異なる，過去の事象について述べた記事など，タイムスタンプを持たない文書のタイムスタンプを推定する手法を提案する．このことにより，今までに貢献できなかったデータが使用可能になることや，複数のソースを一つの時系列に統一することが可能となり，よりスムーズに話題の発見や追跡が可能になる．

筆者らは，事象に基づいた教師学習データからのタイムスタンプ推定が有効に

働くことを示した [16]．しかし，ここでは記事に含まれている話題情報がすべて学習可能で，かつ限定されていることを想定している．また，タイムスタンプの推定に，ある一定量の，話題とタイムスタンプの両方が取得可能である完全なデータを必要とした．

しかし，実際のニュースソースの状況においては，限られた話題や事象のみが配信される状態はまれであり，話題としての集合をなさない単発的な記事も非常に多数あることが考えられる．

本稿では，話題とタイムスタンプの両方が取得可能であるデータ集合 ($D_{topic,time}$) に加え，話題のみが既知のデータ集合 (D_{topic}) や，発行時間のみが取得可能なデータ集合 (D_{time})，またそれらの両方が不明であるデータ集合 (D_u) などを総合的に利用して，タイムスタンプが不明であるデータ集合 (D_{topic}，D_u) のタイムスタンプを推定する．

本稿で提案するタイムスタンプ推定のアルゴリズムの概略は以下のとおりである．

(1) 話題が既知のデータ集合 ($D_{topic,time}$，D_{topic}) から，EM アルゴリズムを用いたベイズ規則により，各データ集合 (D_{time}，D_u) を話題へと分類する．

(2) 1) の結果に基づき，発行時間が取得可能なデータ集合 ($D_{topic,time}$，D_{time}) を用いて，単一パス法により，各話題を時間に依存した事象へとクラスタリングする．

(3) 1),2) の結果に基づき，k-NN によりタイムスタンプが未知のデータ集合 (D_{topic}，D_u) を 1 の話題情報を用いて 2 で生成された各事象へと割り当てる．

(4) 3) で割り当てられた事象からデータ D_{topic}，D_u のタイムスタンプを推定する．

EM アルゴリズムとは Expectation(期待値)，Maximization（最大化) を意味し，観測したデータからは直接観測できない確率変数があるような，直接，最尤推定法を適用できない場合に有効な推定方法である [21]．文書分類において，EM アルゴリズムは，訓練データが少数であるときに効果的に働くことが知られている．

本稿で用いる手法は，EM アルゴリズムを用いたベイズ規則による分類に基づく．これにより，訓練データが非常に少数である場合や，訓練データ（話題が既知のデータ）内に存在しない話題が出現する場合においても，対応することができ，その分類結果に基づいた，時間が既知のデータを用いたクラスタリングによ

り，文書集合を時間依存の事象に分割することができる．さらに，この結果に基づきタイムスタンプを推定することで，不完全なニュース集合からでも，効果的にタイムスタンプの推定を行うことができる．

次章では関連研究について述べ，3 章では EM アルゴリズムを用いたベイズ分類について，4 章で単一パス法による逐次的なクラスタリングについて述べる．5 章でタイムスタンプの推定手法を提案し，6 章において TDT2 コーパスを用いた提案手法の実験と結果，考察について示す，最後に 7 章で結びとする．

4.2　関連研究

文書に時間を割り当てる方法としては，Mani らの研究がある [9]. Mani らは，文書内の時制表現を抽出する手法を提案しており，例えば，文章内の “yesterday” などの索引的な表現からその文書の発行時間が 4 月 20 日であった場合，その時制表現は 4 月 19 日について述べていると推定する．このように，文書から文章内の時制表現を抽出する．しかし，これらの研究は文書のタイムスタンプや，他の文章や文書との関係から時制表現が示す時間を推定するもので，根本的な文書のタイムスタンプを推定する本手法とは異なる．

Papka らは Inquery による単一パス法に基づいた話題検出法を提案している [13]. また，Yang らも単一パス法を用い，逐次，バッチ双方における話題検出を行っている [18]．逐次的な処理において，Papka らは，文書ベクトル表現に，補助コーパスから求めた IDF 値を使用しており，Yang らは補助コーパスによる IDF 値に加え，逐次 IDF 値を更新する手法を提案している．しかし，本稿では，すでにタイムスタンプを推定するデータはすべて揃っていることを想定しているため，従来の IDF 値を用いる．

石川らは，忘却の概念を利用した文書クラスタリング手法を利用している [5]. ここでは，C^2ICM を用いることにより文書データを逐次クラスタリングしており，少ない計算量での差分によるクラスタ更新を行っている．ここでは，統計的な確率に基づき文書間の類似度を求めている．また，Papka らや，Yang らも類似の忘却の概念を利用している．

これらのように，時間を考慮した時系列文書のクラスタリング手法は多く提案されている，しかし，これらの手法のほとんどは，文書のタイムスタンプが利用可能であること，あるいはリアルタイムで処理することを想定している．そのため，タイムスタンプ自体を予測する手法は，筆者らの知る限り提案されていない．

4.3　EM アルゴリズムを用いた分類

本稿では，話題が取得可能であるデータを教師データとして，話題が未知であるデータを各話題へと分類する．ここで，EM アルゴリズムを用いたベイズ規則に基づき分類を行う．

EM アルゴリズムでは，混合正規分布を仮定し，不完全な観測データ $x_1, ..., x_N$ が，モデル $P_\theta(x)$ から得られたとき，この不完全な観測データ $x_1, ..., x_N$ から，未知のパラメータ θ の値を推定する．現時点での θ を使用して，条件付確率モデルから，完全データにおいてのサンプル数，期待値を求める（E ステップ），続いて，ここで求めたサンプルから期待値を最大化するパラメタ $\hat{\theta}$ を求める（M ステップ）．この E ステップと M ステップを繰り返すことにより，モデルの対数尤度を最大化するパラメータを求める．

一般的に，文書分類においては，分類規則の訓練に，少数のラベル付の訓練データだけでなく，話題情報を持たない，大量の不完全なデータを使用し，分類規則の精度を高める手法として使用される [12]．そのため，一般に EM アルゴリズムにより精度を高める場合は，大量のラベルなしデータが述べている話題のすべてが，ラベルを持つデータに含まれていることを想定している．しかし，ラベルを持たないデータは，ラベルを持つ訓練データからは学習できない話題について述べている可能性が高い．本稿では通常の精度を高める目的ではなく，EM アルゴリズムにより，訓練データには含まれない話題について述べているデータに対しても，既存の話題のうち近いものに割り当て，EM アルゴリズムを適用することにより，既存の話題を汎化させ，より意味的に大きな話題を生成することを目的としている．

4.3.1　単純ベイズ法にる文書分類

今，文書 d をカテゴリ集合 $C = \{c_1, ..., c_n\}$ に分類することを考える [22]．ベイズ規則による分類は，文書 d がカテゴリ c に属する確率 $P(c|d)$ の確率分布を求めることである．また，排他的な分類の場合，最大事後確率をとるカテゴリ $c_k(c_k = argmax_{c \in C} P(c|d))$ へ文書 d 分類することで誤分類を抑えると考える．

ここでベイズ規則は以下のように与えられる．

$$P(c_k|d) = P(c_k) \times \frac{P(d|c_k)}{P(d)}$$

すなわち，ベイズ規則での分類規則生成（訓練）は訓練データ集合から，確率分布 $P(c_k)$，$P(d)$，$P(d|c_k)$ を推定することである．

しかし，文書ベクトル $d = (w_1, ..., w_m)$ はほぼすべての文書で異なり，$P(d|c_k)$ や $P(d)$ の推定が問題であるため，一般に，文書内での単語 w_j の出現は，統計的に他の単語の出現とは独立であるという仮定をおき，各文書を単語の集合と考る，単純ベイズが使われる．単純ベイズでは，$P(d|c_k)$ を以下の形式に分解して考える [7].

$$P(d|c_k) = \prod_{i=1}^{|d|} P(w_j|c_k)$$

これにより，文書主導の排他的分類の場合，ベイズ規則は以下のように書くことができる．

$$P(c_k|d) = P(c_k) \times \prod_{i=1}^{|d|} P(w_j|c_k) \tag{4.1}$$

また，ここでは文書内での単語の出現回数は考慮せず，単語が出現したか否かのみを考えるバイナリ独立モデルを用いる．

4.3.2　EM アルゴリズム

EM アルゴリズムを文書分類適用する場合，アルゴリズムは以下のように定義される．

(1)　入力：ラベル付文書，ラベルなし文書

(2)　ラベル付文書のみから単純ベイズ分類規則 $\hat{\theta}$ を生成

(3)　以下のステップを分類規則のパラメタが収束するまで繰り返す
　・(E-step) 現在の分類規則 $\hat{\theta}$ を使用してラベルなし文書を各カテゴリ分類する ($P(c_j|d_i;\hat{\theta})$)
　・(M-step) 推定された事後確率（分類結果）を利用して，分類規則 $\hat{\theta} = P(D|\theta)P(\theta)$ を再度生成する．

(4)　出力：分類規則 $\hat{\theta}$

具体的に，本稿では以下の式を用いる．

$$P(w_i|c_k) = \frac{1 + \sum_{j=1}^{|D|} N(w_i, d_j)P(c_k|d_j)}{|V| + \sum_{i=1}^{|V|}\sum_{j=1}^{|D|} N(w_i, d_j)P(c_k|d_j)} \tag{4.2}$$

ここで D は文書データ全体を表し，w_i はデータ内の各単語を表す．また $N(w_i, d_j)$ は文章 d_j における単語 w_i の発生回数であるが，前述のとおり，本稿では出現の有無により 0 か 1 の値をとる．さらに，$P(c_k|d_j)$ は前述の文書 d_j がカテゴリ c_k に属する確率であり，ラベル付けされたデータに関しては，そのラベル付けられたカテゴリ c_m においては，$P(c_m|d_j) = 1$ であり，それ以外のカテゴリに対しては 0 をとる．対して，ラベルなしデータに関しては，最初は全カテゴリに対して 0 であるが，最初は通常のベイズ分類により，その後は EM アルゴリズムにより，徐々に適切な値へと更新される．式 4.1 と式 4.2 により EM アルゴリズム内で分類規則を生成する．同様に $P(c_j)$ は以下のように与えられる．

$$P(c_j) = \frac{1 + \sum_{j=1}^{|D|} P(c_k|d_j)}{|C| + |D|} \tag{4.3}$$

式 (4.2),(4.3) は，それぞれ $P(w_i|c_k), P(c_j)$ のスムージングを行っている．

4.4　時間距離を考慮したクラスタリング

本章では，EM アルゴリズムによる話題への分類結果に基づき，各話題を事象へと分割するステップを論じる．

TDT や時系列文書クラスタリングにおいて，時間を考慮したクラスタリングは良い性能を示すことが知られている [1, 18]．これはニュースの話題や事象の出現は時間に大きく依存することを意味する．そのため，文書がどのような事象について述べているかを求めることで，より正確なタイムスタンプの推定が行えると考えられる．そこで本稿では，文書とクラスタ間の時間距離を考慮してクラスタリングを行う．これは各クラスタが事象を示し，ニュース的意味を持つことを示す．事象は，各話題に基づいたクラスタリング結果から生成されるので，時間を考慮した話題の部分集合である．

4.4.1　文書表現

文書とクラスタの表現は，“Bag of words” と呼ばれる，文書を単語の集合考える従来のベクトルスペースモデルを利用する．また，ベクトルを構成する単語として本稿では不要語 (stop-word) を削除した後，BrillTagger[3] により，名詞と固有名詞のみを抽出して使用する．さらにステミングを行うことにより語幹を取り除く．ここで文書 d_i における単語 t_j の重み $w(t_j, d_i)$ は，従来の tf*idf 法を拡張した ltc で表す．これは SMART システムにより提供される重みで，TDT タスクにおいてよい性能を示すことが知られている [18]．

$$w(t_j, d_i) = (1 + log_2 TF_{t_j,d_i}) \times IDF_{(t_j)} / \|\vec{d_i}\|$$

ここでTF_{ij}は文書d_iにおける単語t_jの出現頻度(Term Frequency)を表し，$IDF_{(t_j)}$は文書集合での単語t_jを含む文書の割合の逆数である．これにより，文書ベクトル$\vec{d_i}$は以下のように表すことができる．

$$\vec{d_i} = (w(t_1, d_i), ..., w(t_n, d_i))$$

クラスタはクラスタに含まれるベクトルの重心で表現する.

4.4.2　単一パスクラスタリング

本稿では, タイムスタンプを持つデータを単一パス法によりクラスタリングする [18]. 単一パス法は非常に単純な方法で、文書とクラスタの類似度がしきい値以上ならばクラスタに追加し, 超えない場合は文書を新しいクラスタとする方法である. 通常は，リアルタイムの学習に使われる手法であるが，本稿ではタイムスタンプを持つデータの発行順序を考慮したクラスタリングを行うために，この手法を用いる．

この単一パスクラスタリングは, 以下の手順で実行される.

(1) しきい値thを設定

(2) 最初は空の集合Cから始め，1 つ目の文書d_1自身をクラスタの重心とする

(3) 次の文書d_iを読み込み，既存の全クラスタCのそれぞれとの類似度$sim(\vec{d_i}, \vec{C})$を計算する

(4) 最も類似したクラスタc_{d_i}との類似度sim_{max}をが，しきい値より大きい$(sim_{max} > th)$なら，d_iをクラスタC_{d_i}に追加し，クラスタC_{d_i}の重心を再計算する．もし$sim_{max} < th$なら，d_iを新しいクラスタの重心とする

(5) 3-4 をデータがなくなるまで繰り返す

ここで，sim_{max}は以下のように定義される．

$$sim_{max} = MAX(sim(\vec{d_i}, \vec{C}))$$

文書とクラスタ間の類似度はコサイン尺度と呼ばれる方法を用い，以下の式で与えられる．ここで V_C はクラスタ C の重心を表す．

$$sim(\vec{d}, \vec{C}) = \frac{\vec{d} \cdot \vec{V_C}}{|\vec{d}||\vec{V_C}|}$$

4.4.3　忘却関数とタイムウィンドウ

本稿では，文書とクラスタ間の時間距離を考慮した類似度計算を行うために，忘却関数を適用する [5, 18]．

ニュースの話題は，ある一定の期間に集中して発生することが多く，話題の出現が記事のタイムスタンプに大きく依存する．[18]．そのため，例えばある 2 つの文書の文書ベクトル間の類似度が高くても，発行時間に大きな差がある場合，その 2 つの文書は同じ話題について述べている可能性は低くなる．逆に，発行時間が非常に近い場合，この 2 つの文書が同じ話題について述べている可能性は非常に高くなる

本稿では，忘却関数により文書が古くなればなるほど分類への重要度を減少 (忘却) させる．すなわち，時間的に近いものほど重要と考えクラスタリングを行う．クラスタと文書間の類似度に忘却関数を適用するということは，クラスタが時間により小さくなることに対応する [図 4.1]．また，タイムウィンドウ [18] を導入し，ウィンドウ内にあるクラスタのみと比較する．類似度比較を行う期間を限定することで，完全に忘却する期間を決定する．これによりクラスタリングや話題発見を行う期間の長期化に対応でき，ノイズや計算量の減少が考えられる．また，本稿においてはタイムウィンドウのサイズを 90 日とした．

本稿では以下のように忘却関数を定義する．

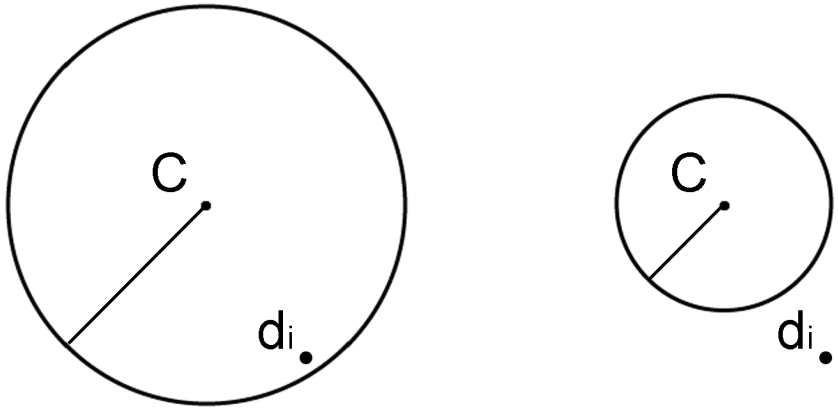
$$\omega_\lambda(t) = \lambda^t (0 \leq \lambda \leq 1.0)$$

ここで t は時間距離を示し，その単位は日数である．本稿では，この忘却関数と，コサイン尺度から以下のように時間距離を考慮した新しい距離基準 sim' を定義する．

$$sim'(\vec{d_i}, \vec{C}) = \omega_\lambda(|time_{d_i} - timeC) \times sim'(\vec{d_i}, \vec{C})$$

ここで $time_{d_i}$ と $time_C$ はそれぞれ文書 d_i のタイムスタンプ，クラスタのタイムスタンプを示す．クラスタのタイムスタンプは，クラスタに含まれる文書集合中最新の文書のタイムスタンプとする．

図 4.1 は，$\lambda = 0.97$ の時の，時間距離が 0 日 ($\omega_\lambda(0)$) の場合と，30 日 ($\omega_\lambda(30)$) の場合のクラスタの状態を示している．文書ベクトル d_i とクラスタ C 間のベクトルの類似度が同じでも，時間距離が離れている場合，クラスタには追加されない．



(a)$\omega_\lambda(0) = 0$　　(b)$\omega_\lambda(30) = 0.5454$

図 4.1: 忘却関数によるクラスタと文書関係の変化

この類似度を用いて, 上記の単一パスクラスタリングを行う.

4.5　タイムスタンプ推定

本章では，タイムスタンプをもたない文書 n のタイムスタンプを推定するために，n の事象を決定するステップについて論じる．タイムスタンプを持つ文書のクラスタリング結果と文書 n を比較することで，n が述べている事象を推定し，この事象に基づいて文書 n のタイムスタンプ推定を行う.

4.5.1　k 近傍法によるクラスタの決定

n のクラスタ (事象) の決定は，k 近傍法による投票で決定する．ここで，全文書集合 D は，前述のベイズ規則に基づき，いずれかの話題に割り当てられている．各事象は，時間を考慮したクラスタリングによる，話題の部分集合なので，各事象は話題に属すると考えることができる．そのため，文書 d が述べる話題の事象だけと比較，投票することにより決定する.

例えば図 4.2 において，k=10 の場合，文書 n がベイズ規則により話題 $topic_1$ に割り当てられた時，タイムスタンプを持つ文書中，話題 $topic_1$ について述べている文書集合 D_{topic_1} だけから投票を行い，n に近い 10 個の文章が属するクラスタ

のうち，もっとも多いものに属するとする．この場合，n が属するクラスタは C_1 となる．

また，最近傍文書のみを使う場合 $(k = 1)$ の場合は，事象へのクラスタリング結果は反映されず，ベイズ規則による話題の分類結果だけが反映される．

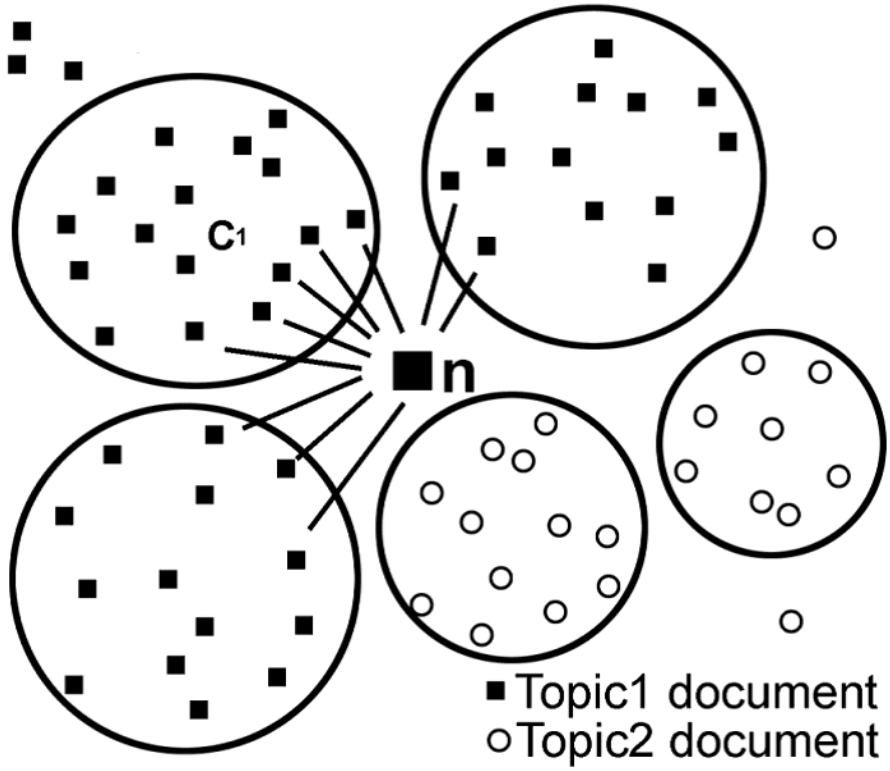


図 4.2: 文書ベクトル $\vec{n}$ の所属クラスタ決定

4.5.2　タイムスタンプ推定

クラスタ (事象) 割り当て結果に基づき，文書 n の示す事象を考慮したタイムスタンプ推定を行う．文書 n のタイムスタンプの割り当てには, もっとも近い k 個のデータのうち, 文書 n が属すると判断されたクラスタ (事象) に属する文書のみを使用する．図 4.2 の場合，n に近い上位 10 個のうち，n が属するクラスタ C_1 に属する 5 つの文書を使用して n のタイムスタンプを推定する．

タイムスタンプ予想曲線

ここで，本稿では，タイムスタンプ予想曲線を用いることで文書 n のタイムスタンプを推定する．図 4.2 における 5 つの文書のうちの 1 つの文書 d_{C_11} から，文書 n のタイムスタンプ予想曲線を以下の式により与える．

$$TS_{n,d_{C_11},\lambda}(day) = sim(d_{C_11}, n) \times dist \quad r_{C1}(time_{d_{C_11}}) \times \omega_\lambda(|time_{d_{C_11}} - day|)$$

$TS_{n,d_{C_11},\lambda}(day)$ は，文書 n のタイムスタンプが時間 day である推定の度合いであり，文書 d_{C_11} と n，それらが属するクラスタ C_1 から与えられる．

クラスタリング同様，忘却関数を使用し，教師文書 d_{C_11} の発行時間から離れるほど，文書 d_{C_11} の影響力は小さくなる．

ここで，$distr_{C1}(time_{d_{C_11}})$ は，クラスタ C_1 に属する文書集合における，時間 $time_{d_{C_11}}$ においてのタイムスタンプの分布（例：図 4.4) である．ニュース記事において，大抵の話題や事象は，ある期間に集中して起こる特性があるため，事象のタイムスタンプ分布を考慮することは重要であり，クラスタ内のタイムスタンプ分布は，そのクラスタが示す事象について述べている文書のタイムスタンプの発生確率と考えることができる．

これにより，文書 d_{C_11} が文書 n に与えるタイムスタンプ予想曲線は，文書 d_{C_11} と文書 n の類似度と，クラスタ C_1 の時間 $time_{d_{C_11}}$ におけるタイムスタンプ分布の積を頂点とし，時間距離により減衰する図 4.3 のような曲線をあたえる．

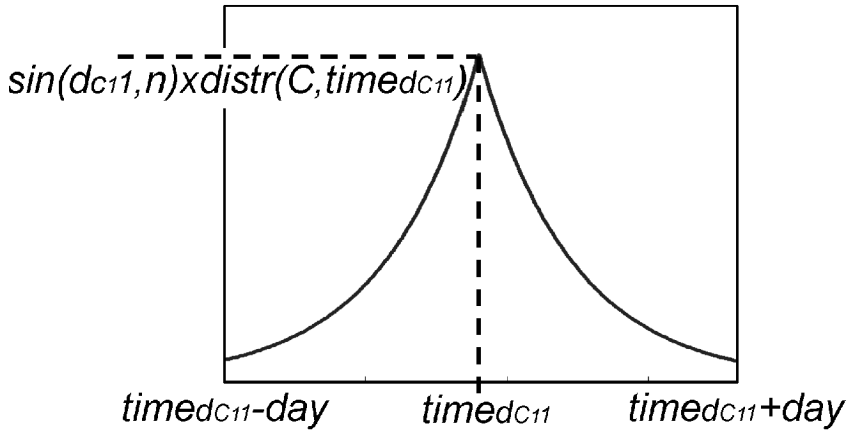


図 4.3: タイムスタンプ予想曲線の例 ($\lambda = 0.97$)

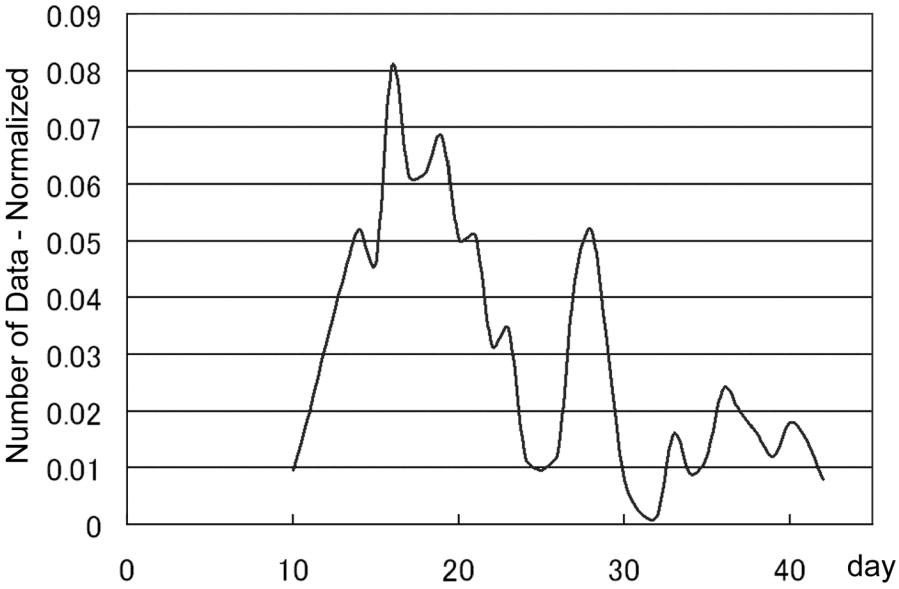


図 4.4: あるクラスタ内のタイムスタンプの分布例

このタイムスタンプ予想曲線を，文書 n のタイムスタンプ推定に使用する教師文書のそれぞれに対し求め，各曲線の総和をとる．すなわち，以下の式になる．

$$TS_{n,T_C,\lambda}(date) = \sum_{t_{ci} \in T_C} TS_{n,t_{ci},\lambda}(date)$$

例として図 4.2 の場合，文書 n に近い 10 個の文書のうち，n と同じクラスタ C_1 に属する 5 つの文書から求まる値の合計を取り，図 4.5 のようなタイムスタンプ予想曲線を得る．

誤差許容範囲

本節では，タイムスタンプ割り当ての精度を評価する為の誤差許容範囲を設定する．これは，実際のタイムスタンプと予測したタイムスタンプの差の許容範囲であり，この許容範囲に基づき，予想曲線からタイムスタンプを求める．すなわち，これは求めるタイムスタンプの細かさの基準を表す．

本節では，予想曲線の誤差許容範囲内の総和が最大になる日付を文書 n のタイムスタンプと推定する．例えば，図 4.5 において，誤差許容範囲が m 日の場合，前後 m の期間のタイムスタンプ予想曲線の総和が最大になる日付を文書 n のタイムスタンプとする．

形式的には，文書 n のタイムスタンプ day_n は以下の式で推定する．

$$day_n = MaxArg_{day} \int_{day_m}^{day+m} TS_{n,T_C,\lambda}(day)$$

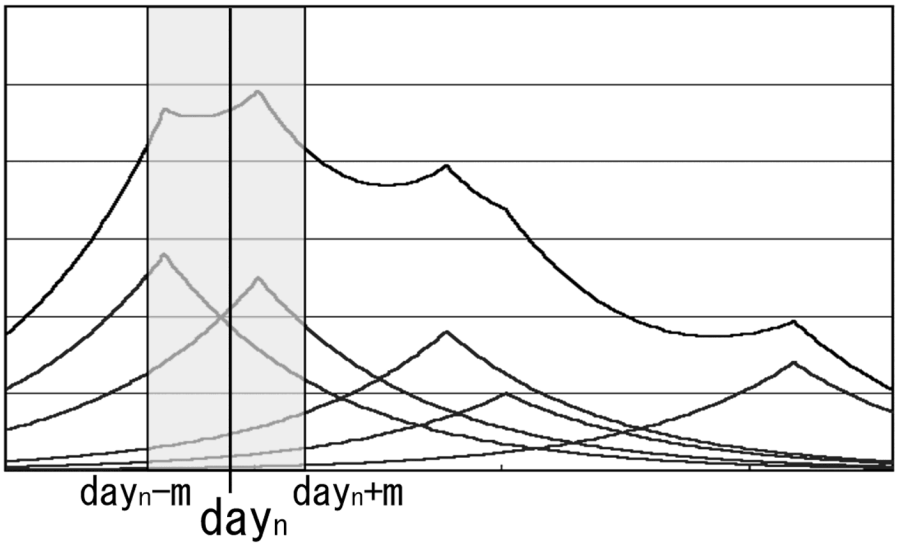


図 4.5: タイムスタンプ予想曲線

4.6　実験

4.6.1　TDT2 コーパス

本稿では, 実験に TDT2 コーパスを用いる [11].

TDT2 コーパスは，放送されたニュースをテキストへ書き写したものと，ニュース通信の 2 種類のニュースソースからなり，1998 年 1 月から 6 月までの 6ヶ月間分のデータを含む．また,TDT コーパスには英語と中国語のデータが含まれるが, 今回は英語のソースである 4 つの放送データ (ABC,CNN,VOA,PRI) と 2 つのニュース通信 (APW,NYT) の記事の計 6 つのソースを使用する．

ここで，本稿では上記ニュースソースはいずれも速報性が高く，それぞれのソースにおいて，事象や話題の発生や変化に差はないものと想定している．

また TDT2 コーパスには，100 個の話題が定義されており，代表的なものとして,「冬季長野オリンピック」や「モニカルインスキ事件」等がある．この各話題について述べている記事には，話題との適合の度合いから “YES”（完全に適合している），“BRIEF”（一部で関連している）の 2 種類のタグが人手によって付与

されている．本稿では，“BRIEF”タグは無視し，“YES”タグを持つ，あるどれかの話題についてのラベルを持つ 8040 件の記事と TDT2 コーパスで定義されている話題について述べていない 45580 件の記事の計 53620 件の記事を利用する．

また，話題の大きさは，10 件に満たない記事しかないものから，1000 件以上の記事が含まれるものまで様々である．

図 4.6 に “ Yes”タグが付与された記事の時間毎の分布と，それ以外の記事の分布を示す．記事の分布は全体としては一定であるが，話題を持つ記事の分布には偏りがあることがわかる．

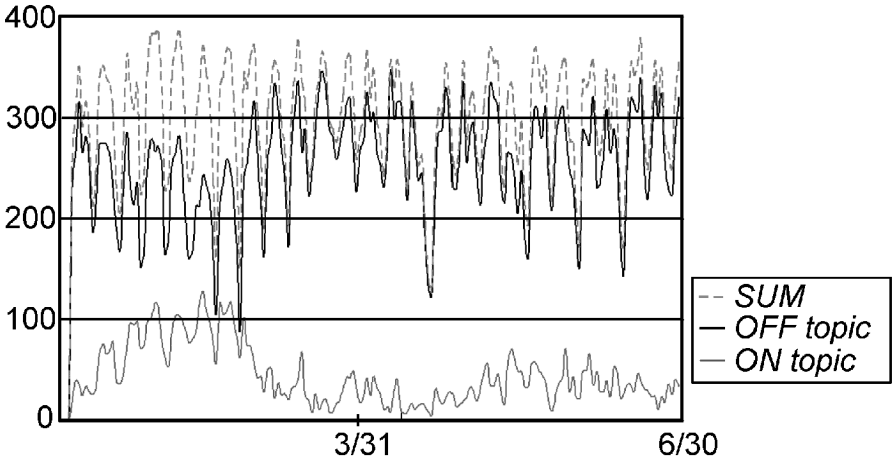


図 4.6: 記事の時間ごとの分布

4.6.2　実験手順

．本提案手法の評価を，以下 3 点において行う.,

- 不完全な状態での有効性
- EM アルゴリズムの精度への影響
- 話題が限られている状態での有効性

まず最初に，本手法の，不完全な状態においての有効性を評価する．不完全な状態，すなわち TDT2 コーパスの約 8 割を占める，どの話題にも属さない，話題情報を持たない状態のデータを使用して実験を行う．これは，教師データにより，学習

できない話題が多数ある状態である．実験は，TDT2 コーパス全データ（約 53000 件) を使用して行い，訓練データ，テストデータの分割は，以下のように行う．

- $D_{topic,time}$:話題、タイムスタンプの双方が取得可能なデータ: 話題が取得可能な記事集合全体の 1%(約 80 件)
- D_{topic}:話題のみ取得可能なデータ: 話題が取得可能な記事集合全体の 1%(約 80 件)
- D_{time}:タイムスタンプのみ取得可能なデータ:全体の 18%(約 10000 件)
- D_u:すべてが未知のデータ:全体の 80%

上記データを話題が割り振られているデータ集合からランダムに抽出し，D_{topic},D_u のタイムスタンプ推定を行い，精度を評価する．

ここでは，5 回の抽出作業を行い，それぞれの実験の精度の平均値を本手法の有効性とする．また，誤差許容範囲については，1 週間 (7 日)，2 週間 (14 日)，1ヶ月（30 日）の 3 パターンを評価する．例えば，誤差許容範囲が 1 週間の場合，予測したタイムスタンプと実際のタイムスタンプの差が 1 週間以内なら正解とする．また，k 近傍法の k の値は，1,2,5,10,30 の計 5 パターンと，最近点法について比較を行う．ここで，1-NN はベイズ規則による話題の分類結果に基づいて推定するのに対し，最近点法は，分類，クラスタリングの結果に関係なく，最も近いものを割り当てる為，NN と 1-NN を区別する．

続いて，EM アルゴリズムがどれほどタイムスタンプ推定に影響しているかを評価する．ここで，EM アルゴリズムによる収束回数は 20 回を最大とし，前述の実験と同じ条件において，EM アルゴリズムを適用しない，通常のベイズに基づいたタイムスタンプ推定 (収束回数 0 回) と，EM アルゴリズムによる収束回数 5 回，10 回，20 回の計 4 パターンを評価する．

最後に，全データの話題が限定されている状態での本手法の有効性について評価を行う．これは [16] において行われたように，タイムスタンプを推定する記事の話題が学習可能である状態についての本手法の評価を行う．そのため，実験には話題が割り当てられている約 8000 件の記事のみを使用し，以下のような状態でデータを用いる．

- $D_{topic,time}$:話題、タイムスタンプの双方が取得可能なデータ:全体の 1%
- D_{topic}:話題のみ取得可能なデータ:全体の 1%
- D_{time}:タイムスタンプのみ取得可能なデータ:全体の 18%
- D_u:すべてが未知のデータ:全体の 80%

ここで，EM アルゴリズムを使用した場合と使用しなかった場合について比較を行う．

4.6.3　実験結果

表 4.1, 図 4.7 に不完全な状態においての有効性を示す．

(EM=10)

(%)	1week	2weeks	1month
NN	31.80	40.64	54.55
k=1	36.83	45.40	58.46
k=3	36.98	46.05	60.39
k=5	35.17	44.57	59.50
k=10	32.33	42.33	58.31
k=20	14.85	30.37	57.88

表 4.1: 不完全な状態においてのタイムスタンプ推定精度

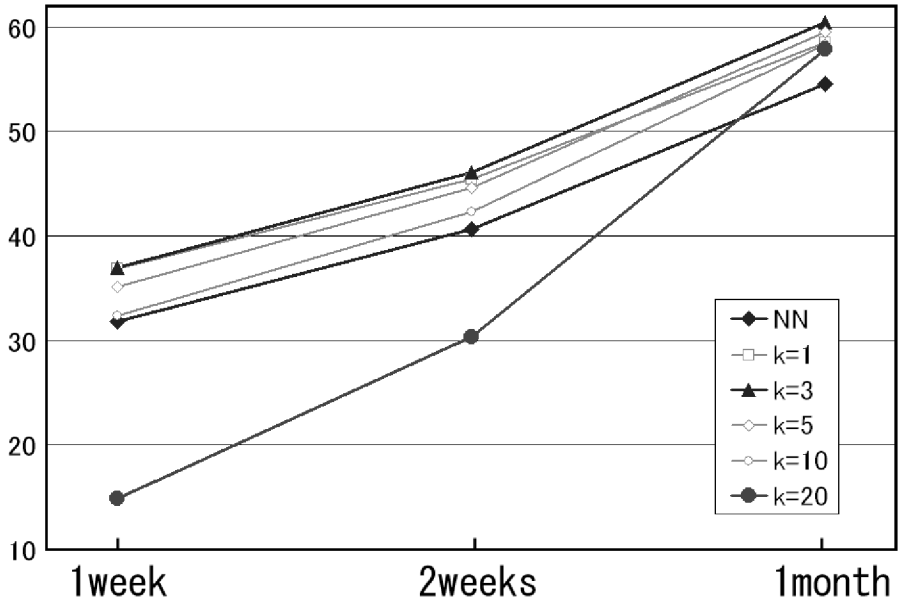


図 4.7: 不完全な状態のいてのタイムスタンプ推定精度 (グラフ)

続いて，表 4.2, 図 4.8 に EM アルゴリズムの精度への影響を評価する．
最後に，表 4.3, 図 4.9 に話題が限られている状態での精度を評価する．

(K=3)

(%)	1week	2weeks	1month
Non-EM	35.33	43.97	57.90
EM5	36.07	44.90	58.92
EM10	36.98	46.05	60.39
EM20	36.07	44.84	58.74

表 4.2: EM の収束回数に対する精度の変化

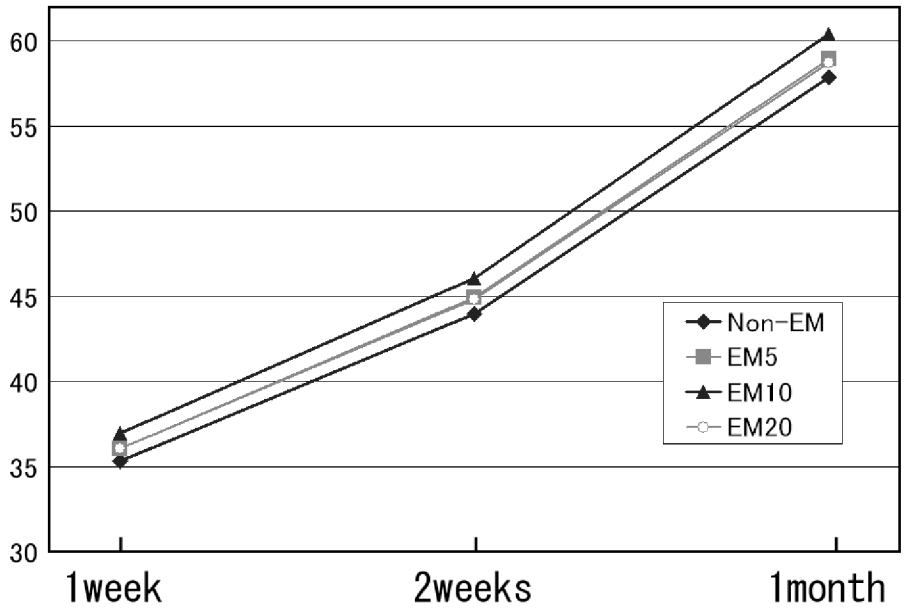


図 4.8: EM の収束回数に対する精度の変化

4.7　考察

表 4.1, 図 4.7 より，教師データが非常に少数で，かつ教師データから学習できない話題が多数存在する状態においても，誤差許容範囲 7 日で，約 37%，誤差許容範囲 30 日では約 60%と，高い精度でタイムスタンプを割り当てることができた．

ここで，事象を割り当てた場合，すなわち NN 以外の場合は，一般的に NN よりも優れており，話題，事象に基づきタイムスタンプを割り当てることで，高い精度でのタイムスタンプ推定が可能であることがわかる．

EM アルゴリズムの収束回数と精度の関係を，表 4.2, 図 4.8 に示す．本実験においては，収束回数が 10 回の時において，最も良い成績を得た，逆に 20 回になると精度が悪化した．

(K=5,EM=10)

	1week	2weeks	1month
OnTopic+EM	56.75	69.17	81.61
OnTopic(NonEM)	50.05	60.87	73.55

表 4.3: 話題が限られている状態での精度

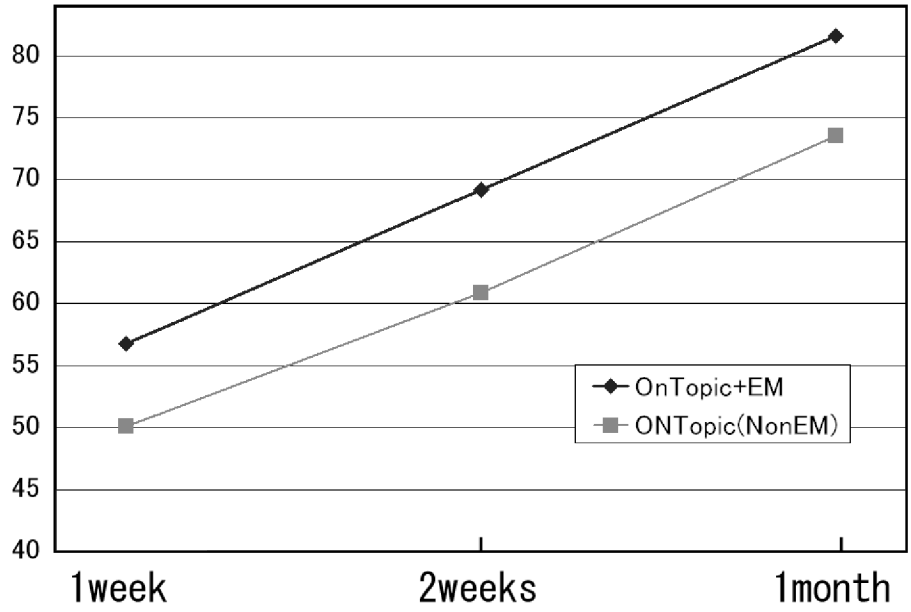


図 4.9: 話題が限られている状態での精度

文書集合においては，話題はその内容から人手により割り当てられているため，単語の出現の確率分布と話題の分類は必ずしも一致しないことが原因と考えられる．文書分類においては，EM アルゴリズムの収束回数に伴い，精度が上がるとは一概には言えず，EM アルゴリズムの最適な収束回数を推定する手法も提案されている [23]．今後はこれらの手法により，ある一定の回数で EM アルゴリズムを終了する基準が必要である．

表 4.3, 図 4.9 より，与えられている文書の話題が限定されている場合も，EM アルゴリズムが有効に働いていることがわかる．また，EM アルゴリズムを用いた場合，誤差許容範囲 7 日において，約 60%，誤差許容範囲 30 日においては，約 83%と非常に高い精度にてタイムスタンプの推定が可能となっている．

いずれの場合においても，EM アルゴリズムを用いた場合のほうが優れた精度を得ており，話題への分類において EM アルゴリズムを用いることは有効であると考えられる．また，最初の実験においては，TDT で定義されている話題に対し

て，EM アルゴリズムの繰り返しにより，対応できたと考えられる．

本実験においては，単一パス法におけるしきい値 $th = 0.1$, 忘却関数 $\lambda = 0.97$ の時に最も良い結果を示した．

次に，表 4.4 に，各期間毎のタイムスタンプ推定の精度を示す．全体的に EM アルゴリズムの使用により各期間で精度は向上しているが，最初の 1 月の期間において，EM アルゴリズムの使用により精度が落ちていることがわかる．これはそれ以前の期間の文書が存在しないことなどから，EM アルゴリズムにより，各話題や事象の重みがデータの多いほうへと収束していったと考えられる．また，話題を持つデータの量が非常に少ない期間においても，他の期間と同程度のタイムスタンプ推定精度を得ていることがわかる．これにより，EM アルゴリズムの有効性と，本手法が，不完全な状態へも対応可能であることがわかる．

(K=5,EM=10,Arrowable 2weeks)

(%)	Jan	feb	Mar	Apr	May	Jun
EM	34.40	50.53	43.63	40.94	43.11	48.42
NonEM	39.10	48.92	41.44	39.37	40.90	44.21

表 4.4: 期間毎のタイムスタンプ推定精度

4.8　結び

本稿では, 不十分で不完全な訓練データを用いた，タイムスタンプ推定手法を提案した．TDT2 コーパスを用いた実験により，本手法の有効性を証明した．今後，この手法を応用することにより，今まで抽出できなかった情報の抽出．例えば，ソース毎に，発行時間に大きな差があるために，違う話題と判断されたものの抽出や，そのような情報の話題追跡への応用，また，より時間情報の欠落や，誤差の大きい Web ページ等についての本手法の応用を考えている．

第5章　結論

本稿では，シソーラスや時制を考慮することにより，テキストデータから優れた構造を抽出する手法，今までタスクに貢献できなかったデータをタスクへと適用可能にする手法を構築することで，テキストマイニングにおいて，より高度な知識獲得を実現した．

まず，同義語，多義語を考慮した文書分類手法を提案した．シソーラス辞書から取得できる意味情報を用い，文書を重み付けることにより，多義性を考慮した文書分類が可能となった．実験により単語を単に記号的に扱う場合に比べ，正解率で約3%向上，語分類率では約15%の減少を得た．また，同義語だけを考慮する場合に比べ，単語の多義性を考慮することにより，より正確な分類が行えることを証明した．これにより，同義語や多義語によるマイニング精度低下の問題を解決し，単語の同義性，多義性の問題を効果的に扱うことが可能となった．

次に，ニュースソースにおいてタイムスタンプを逐次的に推定する手法を提案した．実験により，誤差許容範囲が7日で，推定期間が6ヶ月の場合において，推定精度が約50%と高い性能で，タイムスタンプを推定することが可能となった．これにより，タイムスタンプを持たないためにタスクに貢献できなかった文書，あるいは文書の内容時間と発行時間に大きな差があり，ノイズとなっていた文書等が，TDTタスクを初めとする文書のタイムスタンプが取得可能であることを前提としているタスクに貢献できるようになった．従来のリアルタイムな話題検出や追跡処理にスムーズに適用することができ，より精度の高い話題検出や，スムーズな話題追跡が可能になる事が期待できる．

また本稿では，より実際のニュースソースの状況を考え，学習できる訓練データが十分でない状況に対応したタイムスタンプ推定手法を提案した．訓練データが不完全である状態での話題情報の学習は，EMアルゴリズムを用いることにより対応した．実験により，話題が既知のデータが全データの約1%と非常に少ない場合においても，誤差許容範囲7日で約40%の推定精度を得た．

これにより，話題が既知である情報が非常に少数である場合においてもタイムスタンプ推定が可能となった．また，1つのニュースソースを学習し，そのニュースソースが形成する時系列や話題に，他の多数のニュースソースを統一することが可能になる．

今後の課題としては，まず同義語，多義語を考慮したタイムスタンプ推定手法の構築が挙げられる．本論文においては，すべてニュースソースを対象にしている．ニュースソースは，出現する単語は比較的一般的であり，時間による単語の持つ意味も変化は少なく，シソーラス辞書の更新の必要はないと考えることができ，この両手法はスムーズに統合することが可能と考えられる．

そのため，もう一つの課題として，ニュースソース以外への対応の問題がある．Web ページや掲示板，メールマガジン，あるいは医学文献データベース等の技術文書が考えられる．これらのソースは，ニュースのように，きれいな時系列を形成しておらず，また，出現する単語も一般的ではない為，本稿での提案手法をスムーズに適用できない可能性がある．Web 上の情報においてはノイズへの対処が必要で，学習に使用するデータの取捨選択が問題となる．また，技術文書等などでは，従来のシソーラス辞書では対応できない可能性が高い．しかし，実際に実用することを考えると，これらの問題への対処が重要となる．

謝辞

本研究を遂行し，まとめるにあたり，多くの方にお世話になりました．この場を借りて，感謝の意を述べさせていただきたいと思います．

指導教官である，法政大学工学部情報電気電子工学科　三浦孝夫教授には，日頃から数々のご指導，ご指示を頂きました．心からお礼申し上げます．

また，産能大学経営情報学部　塩谷勇教授には，本研究を進めるにあたり，格別の配慮を賜りました．心から感謝申し上げます．

データ工学研究室の先輩，同級生，後輩には，研究活動，学生生活の両方にわたり大変お世話になりました．

最後になりましたが，このような形で私の研究をまとめることができたのも，多くの皆様方のご支援ご協力の賜物であります．両親を始め，学生生活の中でお世話になったすべての方へ，この場をお借りしまして厚く御礼申し上げます．

関連図書

[1] Allan, J., Carbonell,J., Doddington, G., Yamron,J. and Yang,Y.: Topic Detection and Tracking Pilot Study: Final Report, proc. DARPA Broadcast News Transcription and Understanding Workshop (1998)

[2] Fuhr,N. and Buckley, C.: A Probabilistic Learning Approach for Document Indexing, ACM TOIS 9-3, pp.223-248 ,1991.

[3] Grossman,D. and Frieder,O.: Information Retrieval – Algorithms and Heuristics, Kluwer Academic Press, 1998

[4] Hearst,M.A.: Untangling Text Data Mining, ACM'99,pp.3-10,1999

[5] Ishikawa,Y. ,Chen,Y. and Kitagawa, H.: An On-line Document Clustering Method Based on Forgetting Factors, in proc. 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'01),2001

[6] Lewis,D.D.: Representation and Learning in Information Retrieval, Ph. D. thesis, Department of Computer Science, University of Massachusetts,1992.

[7] Lewis,D.D.: Naive (Bayes) at forty: The independence assumption in information retrieval, proc. ECML-98(10th European Conference on Machine Learning),1998.

[8] Li, Y.H. and Jain, K.: Classification of Text Documents, *The Computer Journal* 41-8, 1998.244, 1990 (revised 1993, Princeton University).

[9] Mani,I. and Wilson,G.: Robust temporal processing of news, proc.of Annual Meeting of the Association for Computational Linguistics (ACL 2000), New Brunswick, New Jersey, 2000, Association for Computational Linguistics.

[10] Miller, G.A., Beckwith, R. et al.: Introduction to WordNet – An On-Line Lexical Database, *Journal of Lexicography* 3(4), pp.235-244, 1990 (revised 1993, Princeton University)

[11] National Institute of Standards and Technology (NIST): http://www.nist.gov/speech/tests/tdt/

[12] Nigam,K. Mccallum,A. Thrun,I. Mitchell,T.: Text Classification from Labeled and Unlabeled Documents using EM, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

[13] Papka, R. and Allan, J.: On-line new event detection using single-pass clustering, Technical Report UMASS Computer Science Technical Report 98 - 21, Department of Computer Science, University of Massachusetts, 1998

[14] Rodriguezd,M.B.,Gomez-Hidalgo,J.M. and Diaz-Agudo,B.: Using WordNet to Complement Training Information in Text Categorization, proc.Recent Advances in Natural Language Processing,pp.150-157,1997

[15] Sebastiani,F.: Machine Learning in Automated Text Categorization, proc.ACM Computing Surveys,Vol.34,No.1,2002 pp.1-47

[16] Uejima,H., Miura,T. SHioya,I.: Giving Temporal Order to News Corpus, The 16th IEEE International Conference on Tools with Artificial Intelligence,2004

[17] Wayne,C., Doddington,G. et al.: TDT2 Multilanguage Text Version 4.0 LDC2001T57, Philadelphia: Linguistic Data Consortium (LDC), 2001

[18] Yang, Y., Pierce, T. and Carbonell,J.: A Study on Retrospective and On-Line Event Detection, proc. SIGIR-98, ACM Intn'l Conf. on Research and Development in Information Retrieval, 1998

[19] 石川 佳治, 北川 博之: 忘却の概念に基づくクラスタリングの改良手法, 日本データベース学会 Letters Vol.2, No.3, 2003

[20] 市村 由美，長谷川 隆明，渡辺 勇，佐藤 光弘: テキストマイニング-事例紹介，人工知能学会誌，16 巻，2 号，2001

[21] 岩崎 学.: 不完全データの統計処理, エコノミスト社,2002

[22] 上嶋 宏，三浦 孝夫，塩谷 勇: 義語，多義語の考慮による文書分類の精度向上, 電子情報通信学会論文誌 Vol.J87-D-I No.2, 2004

[23] 新納 浩幸, 佐々木 捻.: EM アルゴリズムの最適ループ回数の予測を用いた, 語義判別規則の教師なし学習，情報処理学会論文誌 Vol.44,No.12,2003

[24] 那須川 哲哉，河野 浩之，有村 博紀: テキストマイニング基盤技術，人工知能学会誌，16 巻，2 号，2001

[25] 福本 文代，鈴木 良弥：WordNet の同義語クラスとその上位関係を利用した文書の自動分類，情報処理学会論文誌，Vol.43 No.6 2002

[26] 福本 文代, 鈴木 良弥, 山田 寛康: 話題の推移に基づく続報記事の自動抽出, 情報処理学会論文誌 Vol.44 No.07,2003