DEIM Forum 2019 G2-2

文書のトピック検出と局面追跡

井上　眞乙† 　三浦　孝夫††

† 法政大学理工学研究科システム理工学専攻　〒 184–8584 東京都小金井市梶野町 3–7–2
†† 法政大学理工学研究科システム理工学専攻　〒 184–8584 東京都小金井市梶野町 3–7–2
E-mail: †maoto.inoue.7h@stu.hosei.ac.jp, ††miurat@hosei.ac.jp

あらまし　本稿では文書のトピック検出と局面追跡を論じる. 大規模な量のデータの中には断片的な情報が多く存在しているが, これらから人手でトピック (話題) の抽出, 内容の変遷を把握する事は困難である. 本研究では, 検出では確率的 TF-IDF によるクラスタリング, 追跡では隠れマルコフモデル (HMM) を用いる, 特定のトピックに対する文書を抽出・追跡し, 全容を正確に取り出す事を目的とする. また, パラメータやデータ数によるモデルの良さやクラスタ内外の纏まりの良さに基づいて打ち切る閾値を設け, 各推定の終了を判断する.

キーワード　隠れマルコフモデル,HMM, トピック検出, トピック追跡

1 前　書　き

現在, インターネットの普及により膨大な量の情報が日々増加しているが, 中には断片的な情報も多く存在し, 時間の経過と共にその内容が変化していく. 情報の大部分は文書であり, どの語がどの文書に含まれているかを調べる必要がある. トピック追跡は既知のトピックに適した文書の変遷を (時系列で) 捉える事を目的としている. 一般的に文書分類や k-NN のようなクラスタリングを用いた教師付き学習である. このため自動的にモデル化する手法が提案されてきたが, 重い計算を避ける事は困難であり, 効率的かつ正確で有用な技術が必要である.

トピック検出は, 同様のトピックを議論する文書のクラスタを (時系列で) 検出する事である. これは教師無し学習であり, 遡及的検出 (蓄積された文書集合内で新規のイベントを検出) およびオンライン検出 (次々と流れてくる文書から新規のイベントを検出) の 2 つの方法がある [5].

トピックに関する代表的なアプローチの 1 つとして,Topic Detection and Tracking(TDT) がある [5].TDT はある時点で発生する固有の事象を検出し, 状況の発展を追跡する事を可能にする. 一般的な情報源はラジオニュースや新聞記事である.TDT における New Event Detection は, 新規又は以前に未確認の事象に関連する文書を特定することであり, 主な技術は教師無クラスタリングである.Event Tracking は, 特定のトピックに対するストーリーを追跡する.Allan らは適合性フィードバックなど 2 つのアプローチを適用し, 情報フィルタリングに類似した追跡手法を提案している [5].Yang らは決定木 (dtree) と k 近傍法 (k-NN) を用いた手法を提案している [9]. しかしこれらは事前にある程度の学習データを必要としている.

本研究では時系列データを扱う. 単純な構造のデータとは異なり, 時間の流れによって事象が変化するデータ, すなわち新聞記事や音声データである. これに対し隠れマルコフモデル (HMM), 最大エントロピー (MEMM), 条件付確率場 (CRF) などの確率論に基づく高度な手法が存在する.HMM は少量の学習データがあればより上手く機能する助けとなる.MEMM や CRF は系列データに対し強力なモデルであるが事前に学習データが必要である. 学習データの取得は多くの時間とコストがかかるため容易ではない事は明らかであり, 大量の学習データを得る事はほぼ出来ない.

本研究では HMM を用いた尤度によるトピックの局面追跡を行う手法と, 語の増分を考慮した確率的 TF-IDF を用いたクラスタリングによるトピック検出を行う手法を提案する. これにより正確なオンラインでの検出/追跡のモデル化を実現する. 本研究の貢献は以下の 3 つである.

(1)　文書トピックの局面を追跡し, 内容の変遷を正確に捉える.

(2)　正確なオンライン検出のモデル化を可能にする確率的 TF-IDF を用いたトピック検出について議論する.

(3)　各推定に打ち切り閾値を設け終了の判断をし, 計算量の削減を図る.

本論文は次のように構成されている.2 章ではトピックの局面追跡とトピック検出について,3 章では HMM について述べる.4 章では HMM によるトピックの局面追跡と確率的 TF-IDF によるトピック検出のアプローチを提案し,5 章では実験結果を示し,6 章では結論を述べる.

2 トピック検出と局面追跡

TDT においてイベントは特定の時間と場所で発生する事象として定義され, トピックはある一連の事件を表す特定の話題のことである. 例えば,「千葉・市川女性刺殺事件」と「山口・周南放殺人事件」では「何らかの事件が発生した」「犯罪者が逮捕された」などのニュース記事を含むトピックだが, 本質的な違いは「致命傷となった凶器」「犯罪場所」である. それぞれの「殺人事件」は異なるストーリーを含んでいるため, 別々の事件を表す特定の話題として捉える方が自然である.

トピック検出は同様のトピックについて議論するイベントを, 話題の変化を検出する技術である. 主に 2 つのアプローチで行

う. 全ての記事が用意された上で行う遡及的検出およびストリーム内の記事を順に処理するオンライン検出である.

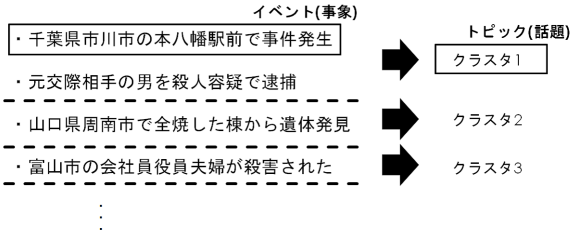

図 1　トピック検出の例

上記の図 1 には 3 トピックの記事が時系列に並んでおり, オンライン検出を想定している. これらの事件が発生した固有の場所, 時間, 事象の違いにより前後の記事の話題の変化を捉え, 各記事が述べているトピック毎に分ける.

ストーリーライン（またはストーリー）は, 一貫した意味のある事柄で記述される幾つかの適切な構造に言及する 1 つまたは複数のトピックに関連するイベントの系列である. 言い換えれば, 関連するイベントはストーリーの局面を構成する.

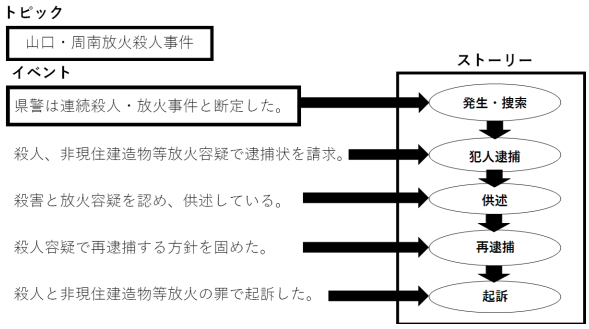

図 2　山口・周南放火殺人事件のストーリー

例えば,「山口・周南放火殺人事件」トピックは「発生・捜索」,「犯人逮捕」,「供述」,「再逮捕」,「起訴」の 5 つの局面で構成されている. それらを追跡することによりトピックのストーリーを把握し, 発生から起訴までの全体を分析することが可能である.

局面追跡と呼ばれるストーリー抽出 (ニュース記事から) は, 特定のトピックの下でイベントを抽出し, これらのイベントが時間の経過と共にどのように発展するかを明らかにすることを目的としている [10]. しかしここでのトピック追跡は, 単語の特徴などの共通項を基に議論しているのであり, 本研究が目指す局面追跡の” 前述・後述の記事の整合性に重きを置いたトピックの抽出および知識構造の発見” とは目的が異なる. たとえ時系列的に記事が与えられたとしても関連するイベントを正確に抽出し, それらを首尾一貫したストーリーに繋げるための技術が必要である.

局面追跡には主に 2 つの問題がある. 第 1 に, ストーリーの決まった構造がないため学習データを事前に用意する方法が無く, 同ジャンルのトピックであればストーリーの変遷が似てしまう. そのため, 特定のトピックに対するストーリーの捕捉が難しい. 第 2 に, トピック内であるイベントがバースト状態 (急激に記事数が増加する) が生じる事がある. 急激な変化に対する対応が難しく, 同じトピックでありながら「異なるトピックの記事である」と間違った追跡を行う場合がある.

また, トピック検出にも語分布の変化への対応が困難な問題がある. ニュース記事は常に増加し, それに伴い語数も語頻度も増加する. 従来の TFIDF や遡及的検出での処理は難しく, 語の増分を考慮したオンラインでの検出方法が必要となる.

本研究は追跡問題に対し,HMM による確率論的アプローチをとる. 状態遷移をストーリーとして考え, マルコフモデルを構築し, トピックの構造の可視化を行う. さらに, 尤度計算でトピックの局面追跡を行う. 尤度計算では Viterbi アルゴリズムを拡張した K-Viterbi アルゴリズム, 後ろ向き K-Viterbi アルゴリズムを使用し, 尤度上位 K 個の経路を得る. また, 検出においては増分を考慮した確率的 TFIDF および遡及性を考慮したクラスタリングによるオンラインでの検出方法を提案する.

例としてトピック「山口・周南放火殺人事件」を行う. 図 3 に記事の一部を示す.1 行 1 記事に対応しており, 時系列順に並んでいる. 語は名詞を抽出している.

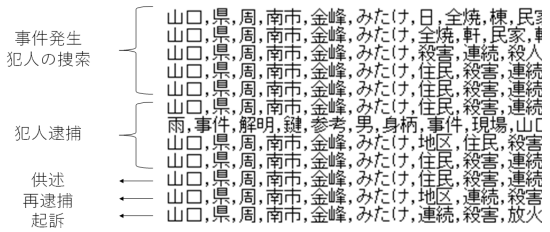

図 3　山口・周南放火殺人事件の記事

3　隠れマルコフモデル

3.1　マルコフモデル

確率論において確率過程は, 確率変数が一組の状態に関連付けられ, 時間の経過と共に変化する数学的な手法を意味し, 確率的有限オートマトン (観測出力値付き) として考えられる. このアプローチは, 様相が無作為に変化する現象およびシステムの数学的モデルに広く使用されており, 一般的な例として「音声認識」が挙げられる.

マルコフモデル (MM) は, 未来の状態が現在の状態にのみ依存し, それ以前に発生した事象・状態には依存しないという仮定を有する確率的枠組みの 1 つであり, マルコフ性と呼ばれる.MM は (N, M, A, B, Π) で定義され, ここで $N = \{1, 2, .., n\}$ は状態数,$M = \{1, 2, .., m\}$ は出力記号数であり,$A = \{a_{ij}, i, j \in N\}$, $0 \leqq a_{ij} \leqq 1$, $\Sigma_j a_{ij} = 1$,$B = \{b_{ij}(o), i, j \in N, o \in M$, $\Sigma_o b_{ij}(o) = 1$,$\Pi = \{\pi_i, i \in N\}$, $\Sigma_i \pi_i = 1$ である.a_{ij} は状態 i から状態 j への遷移確率を意味し,$b_{ij}(o)$ は状態 i から状態 j への遷移での出力記号 o の確率, また π_i は開始状態 s_i の初期確率を意味する. 一般的にマルコフの仮定は $P(s_{t+1}|s_1..s_t) = P(s_{t+1}|s_t) s_1, .., s_{t+1} \in N$ であるので, 扱いに

くいシステムへの推論と計算を可能にし, 単純で効率的な方法で予測モデリングおよび確率的予測を容易にする [4].

3.2 隠れマルコフモデル

マルコフモデルには主に 2 つの問題がある. 学習データの取得, および学習データに適したマルコフモデル (A, B および Π) の推定方法である. 学習データを調べる事によりモデルの構築が可能となるが, 真の問題は多くの学習データを正しく効率的に取得する方法である.

隠れマルコフモデル (HMM) は, 状態が部分的にしか観測できないマルコフモデルの一種である [3]. 観測値の出力はシステムの状態に関連しているが, 状態を正確に判断するには不十分である. ここでいう「隠れ」とは, 最適な解釈のための観測に関連付けた状態の事を示す.

HMM は 3 つの問題に対する解決策を提供する. 観測系列 $O = o_1...o_T, o_j \in M$ と仮定する. 第 1 に, 観測系列 O の確率を得ることが出来る. 実際, マルコフモデルを仮定すると,O を生成する全ての経路を探索し, その確率を合計する. 第 2 に, 最も可能性の高い経路 $S = s_1...s_T, s_j \in N$ を取得することが可能である.Viterbi アルゴリズムは動的計画法を用いて効率的に最尤の経路を推定する.

HMM の真の貢献は, マルコフモデルにおける状態遷移確率 A, 出力記号確率 B および 初期状態確率 Π を推定する事が可能である. マルコフモデルを推定するために EM アルゴリズムを適用する. ゼロから EM アルゴリズムを介して調整過程を繰り返す事で, データの尤度を最大化していく.Baum Welch(BW) アルゴリズムを使用すると, 一連の (未学習の) データから HMM を効率的に推定できる. 十分なサイズの学習データがあれば, 素早い収束と局所的最適解の回避に役立つ. 学習データや適切な初期化が無ければ, これらの問題を避ける事は難しい.

代表的な応用事例に,Viterbi アルゴリズムを利用した形態素解析を行う方法がある. 品詞に基づくマルコフモデルを構築し, 各経路の尤度を Viterbi アルゴリズムにより得て最大尤度となる経路を決定する. 例えば, 品詞のマルコフモデルと”Time flies like an arrow” が与えられたとする. この英文の翻訳が単語”’flies” と”like” の品詞が「名詞または動詞」,「動詞または前置詞」であるかの違いにより,「時蝿は矢を好む」または「光陰矢の如し」というように大きく差が出来てしまう. まず, 出力観測値として考慮したこの英文を単語ごとに区切り, 名詞, 動詞, 冠詞, 形容詞, 前置詞の状態遷移を推定する. 次に, 最大尤度である状態の経路を決定する. この場合, 最尤経路は名詞, 動詞, 前置詞, 冠詞, 名詞であるため最適な翻訳は「光陰矢の如し」であると推測できる [11].

本研究では,HMM を用いてモデルを構築し Viterbi アルゴリズムを用いて尤度計算を行い, トピックの局面追跡を行う. 例として図 3 の記事を用いる. 尤度の計算の前にトピック検出を行い, 得られた記事によりモデルを構築する. 手法については 4 章で述べる. 得られた記事の一部とその記事により構築したモデルを図 4 に示す. 検出で得られた記事 13 件から Baum Welch アルゴリズムにより, モデルを推定する. ここで状態 0 から始まる初期状態確率 0.9969..., および状態間の各遷移確率を取得する (実験では状態名は各状態の出力記号に基づいて決定する).

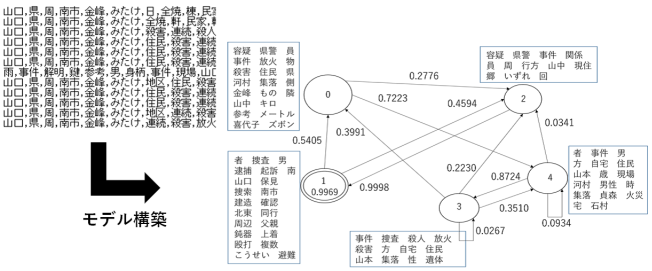

図 4 記事の一部と構築されたマルコフモデル

4 トピック検出と局面追跡に対する提案手法

トピック検出および追跡において,Yang 氏らは余弦類似度に基づく CMU アプローチを提案している [5][9]. 各文書をストーリーとみなし, トピック検出では余弦類似度に基づいてクラスタリングを行い、「新規のイベント」または「既存のトピックのイベント」であるかを閾値に基づいて判断する. 後にオンラインでの語の増分を考慮した改良版を提案している. トピック追跡では k 最近傍法 (kNN) を採用している. 学習データとテストデータ間の類似度を比較し, 最も近い k 個の学習データのトピックを選択する. 共に提案されている決定木の手法よりもやや優れており, 修正版では誤検出率も減少出来ると述べている. しかし, 全体的な精度としてはあまり良好でなく, 追跡に関しては具体的な結果を述べていない.

本研究では文書とする新聞記事に対し,HMM によるトピックの局面追跡および確率的 TFIDF によるオンラインでのトピック検出の新規のアプローチを提案し, それぞれの手法を補完するように組み込んで行う. 局面追跡は, ある記事が特定のトピックにおける後続の記事となるかを 1 つのモデルで判断するため, クラスごとにモデルを構築して最尤原理 (MLP) により分類する分類問題とは異なる点に注意されたい.

以下で述べる各手法での閾値は幾つかの予備実験を通して決定する. 本研究で設定された閾値については次の章で述べる.

4.1 トピック検出

本研究では従来のクラスタリングを拡張し, 以下の 2 つのアイデアに基づいてトピック検出を行う.

(1) 各記事の TF と各クラスタにおける確率 IDF を計算.
(2) 遡及的に各記事間の類似度を計算.

Term Frequency Inverse Document Frequency(IDF) とは, 文書中の単語の重要性を推定するための方法である.TF は文書内における単語 w の出現頻度を意味する.IDF は単語 w を含む文書数の全文書数に対する比の逆対数である. 相沢氏らは, エントロピーの概念を用いる TFIDF のモデル化を議論している [1].

しかし, 新しい文書が継続的に入る事で文書数が変化し続ける状況を考慮すると, 毎回 TFIDF を取得する必要がある. 全て

の文書を処理するには時間がかかるため, ここでは確率的逆文書頻度 (IDF) を適用する. 単語 w の出現確率 $P(w)$ を用いて容易に確率的 IDF を求める事が可能となる. 以下に示す.

$$\begin{aligned} TFIDF &= TF \times IDF \\ TF &= \frac{f_{w_i}+1}{F+N} \\ IDF &= \log \frac{1}{P(w_i)} \\ P(w_i) &= \frac{f'_{w_i}+1}{F'+1} \end{aligned}$$

ここで,f_{w_i} は i 番目の単語 w の頻度.F は全単語の頻度,N は文書内の全単語数である.$P(w_i)$ は, 各クラスタ内の総単語数 F' における単語 w_i の出現確率を意味する.TFIDF の枠組みが動的環境下でも維持することが可能であることに注意されたい. 以下の手順に従い, トピック検出を行う.

(Step A) 各記事に対する TF とクラスタにおける確率 IDF を計算する.

(Step B) 各クラスタ内における記事とテストデータ間の類似度を計算. このとき, クラスタ内の記事の全ての組み合わせ (既存の記事が所属するにあたり, 過去に類似度を求めた際の記事の組み合わせ : 遡及的) で類似度を計算し, 最終的に最小類似度を決定する.

(Step C) 最小類似度が閾値以上の場合は同クラスタの記事とする. 閾値以上となるクラスタが複数ある場合は最も最小類似度が大きいクラスタに所属する事となる. また, 全てのクラスタで閾値以下であった場合は新規のクラスタを作成し, 所属させる.

(Step D)Step A〜Step C を繰り返す.

4.2 局面追跡

以下の 2 つの仮定を提示し, 尤度によるトピックの局面追跡を行う.

(1) トピックには統一した構造がある.

(2) 各トピック内で時系列順に並べられた文書には区別可能な一定の尤度を保持する.

トピックの構造を確率過程の状態遷移とみなし, モデルの構築および尤度計算により類似する記事を選定することで構造抽出が可能と考える. さらに,Viterbi アルゴリズムを拡張した 2 つの方法,K-Viterbi アルゴリズム, 後ろ向き K-Viterbi アルゴリズムを提案し, 尤度計算を行う.

4.2.1 Viterbi アルゴリズムの拡張

一般的に, マルコフモデル (N, M, A, B, Π) と共に出力系列 O=$o_1...o_T$ $(T > 0)$ が与えられると系列 O に対する最大尤度の経路 $s_1, ..., s_T$ を Viterbi アルゴリズムにより推定する. このアルゴリズムに対し, 時刻 t における状態 i の最大尤度を保持する 2 つの関数 $\alpha_t(i)$,$\beta_t(i)$ および $\alpha_t(i)$ の時刻 t における状態 i の状態連鎖をそれぞれ定義し, 以下のような動作を行う.

$$\begin{aligned} \alpha_0(i) &= \pi_i \\ \alpha_{t+1}(i) &= Max_j\{\alpha_t(j) \times a_{ji} \times b_{ji}(o_{t+1})\} \\ \beta_0(i) &= \phi \\ \beta_{t+1}(i) &= ArgMax_{j=1,..,n}\{\alpha_t(j) \times a_{ji} \times b_{ji}(o_{t+1})\} \end{aligned}$$

π_i は状態 i が初期状態となる確率である. 帰納法により, 時刻 t での状態 j における最大尤度 $\alpha_t(j)$ を仮定する. この時, 時刻 $t+1$ で状態 i に到達するために,$\alpha_t(j) \times a_{ji} \times b_{ji}(o_{t+1})$ を最大化する全ての状態 $j \in N$ を調べる. 同様に, 帰納法により時刻 t の状態 j における最大尤度 $\beta_t(j)$ を仮定する. この時, 時刻 $t+1$ で状態 i に到達するために,$\alpha_t(j) \times a_{ji} \times b_{ji}(o_{t+1})$ を最大化する全ての状態 $j \in N$ を調べる. $\beta_{t+1}(i) = j$ は, 尤度を最大化する状態 j から時刻 $t+1$ で状態 i への到達を意味している. 例として,$\beta_2(s_2) = s_1, \beta_3(s_3) = s_2, ...$ と示す. 最終的な結果は後方連鎖 $\beta_T(s_T)$ で保持する.

本研究では Viterbi アルゴリズムを拡張し, 新たに K-Viterbi, 後ろ向き K-Viterbi アルゴリズムの 2 つのアルゴリズムを提案する. これらは尤度の大きい順に K 個の経路を推定するアルゴリズムである. 複数の経路を同時に推定することでモデルの多重化が可能となる.K-Viterbi アルゴリズムの手順を以下に示す.

$$\begin{aligned} \alpha_0^{(k)}(i) &= \pi \ \ (k=1), 1 \ \ (k>1) \\ \alpha_{t+1}^{(k)}(i) &= Max_{h=1,..,K,j=1,..,n}^{(k)}\{\alpha_t^{(h)}(j) \times a_{ji} \times b_{ji}(o_{t+1})\} \\ \beta_0^{(k)}(i) &= \phi \\ \beta_{t+1}^{(k)}(i) &= ArgMax_j^{(k)}\{\alpha_t^{(h)}(j) \times a_{ji} \times b_{ji}(o_{t+1})\} \\ & \quad j = 1, .., n, h = 1, .., K \end{aligned}$$

$Max^{(k)}$ は,k 個の最大値の演算子であり,$K = 1$ ならば $Max = Max^{(1)}$ となる. また,$ArgMax^{(k)}$ は k 個の `argmax` 演算子であり,$K = 1$ ならば $ArgMax = ArgMax^{(1)}$ となる. 前述のように, 帰納法により時刻 t での状態 j における $h(h = 1, ..., K)$ 個の最大尤度 $\alpha_t^{(h)}(j)$ を仮定する. このとき, 時刻 t+1 で状態 i に到達するために,K 個の最大 $\alpha_t^{(h)}(j) \times a_{ji} \times b_{ji}(o_{t+1})$,$h = 1, .., K$ を含む全ての状態 $j \in N$ を調べる. 結果は $\alpha_T^{(k)}(s_T)$ で保持する.

k 個の最大経路に関して, 時刻 t における状態 j での最大尤度 $\beta_t^{(h)}(j)$,$h = 1, .., K$ を仮定する. この時,k 個の最大尤度を保持する時刻 $t+1$ での状態 i に到達するために,k 個の最大 $\alpha_t^{(h)}(j) \times a_{ji} \times b_{ji}(o_{t+1})$, $h = 1, .., K$ を含む全ての状態 $j \in N$ を調べる. ここで $\beta_{t+1}^{(k)}(i) = j$ は状態 j から時刻 $t+1$ の状態 i への到達を意味しているが, 状態 j は常に k 個であるわけではなく,$\beta_2(s_2)^{(1)} = s_1, \beta_3^{(2)}(s_3) = s_2$ という事も考えられる. 後方連鎖 $\beta_T^{(k)}(s_T)$ を取得し, 各ステップで常に k 個の選択 $O(KnT)$ を取得する.

$\alpha_T^{(k)}(s_T)$ および $\beta_T^{(k)}(s_T)$ は全て同時に取得することに留意されたい. この拡張した計算方法は, 効率的かつ同時に上位 K 個の経路を推定するのに役立つ. 例として図 4 のマルコフモデルと記事を用いて上位 2 個の経路を推定する. 記事には" 富山, 市, 会社, 役員,..." という単語が含まれ, 最初の単語" 富山" の出力確率は初期状態 0 で 0.01 である. この時のコストは初期状態確率 0.9933 を掛けた $\alpha_1^{(}1)(s_1) = 0.009933 = 0.9933 \times 0.01$ である. さらに" 山口" から次の単語" 市" へは, 状態 0 から状態 2 および状態 0 から状態 4 の 2 経路の遷移が存在する. 先程求めたコストと状態遷移確率および出力記号確率を掛けると, 新しいコストはそれぞれ 0.000223 と 0.000047 となる. この過程を最後の語まで繰り返していく.

後ろ向き K-Viterbi アルゴリズムは,K-Viterbi アルゴリズムを後ろ向きに行う計算方法である. 系列の全ての情報を後方で

取得する. ここで, 時刻 t で状態 i から始まり経路 s_T へ到達する最大尤度を保持する 2 つの関数 $\gamma_t(i)$,$\delta_t(i)$ および $\gamma_t(i)$ の時刻 t での状態 i から経路 s_T への状態連鎖をそれぞれ定義する.

$$
\begin{aligned}
\gamma_T(i) &= 1\ (i = s_T), 0\ (otherwise) \\
\gamma_t(i) &= Max_{j=1,..,n}\{a_{ij} \times \gamma_{t+1}(j) \times b_{ij}(o_t)\} \\
\delta_T(i) &= i \\
\delta_t(i) &= ArgMax_j\{a_{ij} \times \gamma_{t+1}(j) \times b_{ij}(o_t)\}
\end{aligned}
$$

帰納法により, 時刻 $t+1$ で状態 j から始まり経路 s_T へ到達する最大尤度 $\gamma_{t+1}(j)$ を保持する. 次に, 時刻 t で状態 i に後方に取得するために,$a_{ij} \times \gamma_{t+1}(j) \times b_{ij}(o_t)$ を最大化する全ての状態 $j \in N$ を調べる. 同様に, 時刻 $t+1$ での状態 j において最大尤度 $\delta_{t+1}(j)$ の経路を仮定する. この時, 時刻 t で状態 i を後方に取得するために,$a_{ji} \times \gamma_{t+1}(j) \times b_{ji}(o_{t+1})$ を最大化する全ての状態 $j \in N$ を調べる.$\delta_t(i) = j$ は状態 j から時刻 t で状態 i への到達を意味している. 従って $\delta_2(s_2) = s_1, \delta_3(s_3) = s_2, ...$ となる.Viterbi アルゴリズムと比較して複雑さが増すことは無い. 後ろ向き K-Viterbi アルゴリズムの手順を以下に示す.

$$
\begin{aligned}
\gamma_T^{(k)}(i) &= i\ \ (i = s_T), 0\ \ (otherwise) \\
\gamma_t^{(k)}(i) &= Max_{h=1,..,K,j=1,..,n}^{(k)}\{a_{ji} \times \gamma_{t+1}^{(h)}(j) \times b_{ji}(o_{t+1})\} \\
\delta_T^{(k)}(i) &= 1\ \ (i = s_T), 0\ \ (otherwise) \\
\delta_t^{(k)}(i) &= ArgMax_j^k\{a_{ji} \times \gamma_{t+1}^{(h)}(j) \times b_{ji}(o_{t+1})\} \\
& \quad j = 1, .., n, h = 1, .., K
\end{aligned}
$$

例として,K-Viterbi アルゴリズムと同様に $K = 2$ で図 4 を用いる. 記事には” 富山, 市, 会社, 役員,...” という単語が含まれ,173 番目の最後の単語” 放火” は状態 4 のみから出力記号確率 0.05121... で出力されている. 上位 2 個のコスト $C(s_1 73)$ は 0.05121.. となる.172 番目の単語” 灯油” は後方の遷移により状態 4 から状態 3, 状態 4 から状態 4 への遷移が可能であり, この時の上位 2 個のコスト $C(s_1 72)$ は 0.000502..., および 0.0000094... である. 常に K 個のコストを保持しながらこの過程を繰り返す.

4.2.2 局面追跡の手順

以下の手順に従い,K-Viterbi, 後ろ向き K-Viterbi アルゴリズムを尤度計算に用いたトピックの局面追跡を行う.

(Step A') トピック検出で得られた記事を学習データとする. 記事の時系列は保持する.

(Step B') テストデータ 1 件と共にマルコフモデルを構築する. 構築したマルコフモデルを用いて, 尤度計算により各記事の上位 K 個の尤度を推定する.

(Step C') テストデータのパープレキシティ(PP) 値と学習データの平均 PP 値との差が閾値以下であれば, 追跡対象のトピックに類似する記事とみなし選定. 学習データに追加する. 閾値以上であれば現在のテストデータを捨てる.

(Step D') 次のテストデータを推定する.

(Step E')Step B'〜Step E' を繰り返す.

4.3 トピック検出と局面追跡

本項では提案手法全体の流れについて述べる. 前述で述べたトピック検出および局面追跡を組み込んで行い, 双方の手法の結果を補完し合う. さらに, 各手法の推定に終了の判断である閾値を設定する. これらにより, 検出または追跡漏れを防ぎ, 推定量の削減を図る.

時系列に並べた記事系列を一定の区間で区切り, 以下の手順に従い区画ごとに検出と追跡を行う. なお, 第 1 区画には追跡したいトピックの記事が数記事含まれているものとする.

(Step A") トピック検出を行う. 現区画の最後の記事まで終えた後, 得られた各クラスタ内外の纏まりの良さをそれぞれ求め, 閾値 α と比較する. クラスタ i の纏まりの良さ C_{CD_i} は中心度 C_{C_i} および密度 C_{D_i} に基づいて以下で求める.

$$
\begin{aligned}
C_{D_i} &= \frac{W_d}{W_{C_i}} \\
C_{C_i} &= \frac{1}{C_{N-1}} \sum \frac{(W_{C_{ij}})^2}{W_{C_i} \times W_{C_j}} \\
C_{CD_i} &= \frac{1 - C_{D_i} + C_{C_i}}{2}
\end{aligned}
$$

(Step B") 閾値 α 以上であれば検出を完全終了し, 以下であれば検出は次の区画でも続行する. どちらの場合においても, 得られた学習データのクラスタを用いて, 現区画内のその他の記事 (テストデータ) の局面追跡を行う.

(Step C") 追跡出来た記事がある場合, その時点でモデルの性能 M を以下の式で求め, 閾値 β と比較する. 閾値以上であれば追跡を完全終了し, 以下であれば次の区画でも続行する.

$$
M = D + P + \sum_D PP
$$

(Step D") 現区画内の追跡が終わり次第, 次の区画と合体し Step A"〜Step D" を繰り返す. 検出または追跡が終了している場合はその推定は行わない.

クラスタ内外の纏まりの良さは, 中心度および密度の概念を用いて評価する [7].W_d はクラスタ C_i 内で 2 件以上の記事 d に出現する語数 W,W_{C_i} はクラスタ C_i 内の全語数である.$W_{C_{ij}}$ はクラスタ間で共通する語数であり,C_{N-1} は比較したクラスタ数である. モデルの性能において,D はデータ数,P は推定されたパラメタ数であり,$\sum_D PP$ はパープレキシティ値の合計である.

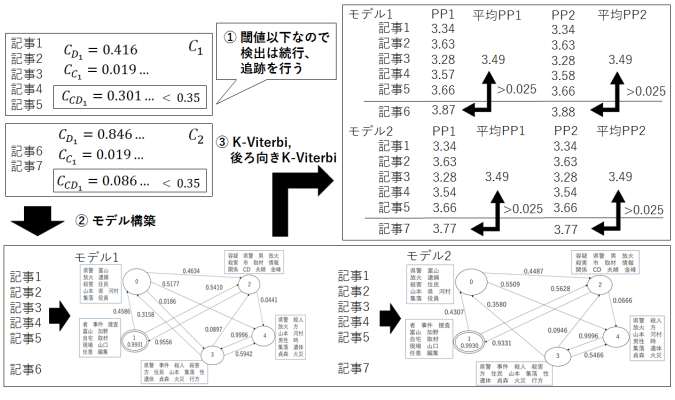

図 5 記事 7 件を用いた手法の流れ

例題として, 記事集合 7 件を用いて行う. 尤度計算の際は $K = 2$ と設定する. 図 5 の記事 1〜5(山口トピック) は追跡したいトピックの記事, 記事 6,7 はテストデータである. まず, トピック検出によりこれらの記事のクラスタリングを行う. その

後, 作成された各クラスタ C_1,C_2 でクラスタ i の纏まりの良さ $C_{CD_1} = 0.301$,$C_{CD_2} = 0.086$, を求める. このとき各値は閾値 0.35 以下であるので, 検出は次の区画でも行う事が決定する. 次に記事 1〜5 と記事 6 および記事 7 によるマルコフモデルを順に構築する. 図中のモデル 1 は記事 6, モデル 2 は記事 7 を含めて構築したものである. さらに K-Viterbi, 後ろ向き K-Viterbi アルゴリズムを用いて尤度計算を行う. 各モデルで求めた平均 PP 値とテストデータの PP 値との差は閾値 0.025 以上であるため, 記事 6 および記事 7 は追跡したいトピックの記事でないとし, 捨てる. このとき, 追跡出来た記事は無いのでモデルの性能 M は計算しない. 第 1 区画内の記事の追跡が終了次第, 次の区画と合体し同様に行う.

5 実　　験

5.1 実験準備

関連研究である Yang 氏らの kNN による手法を追跡のベースライン, 余弦類似度による手法 (以下,CMU) と増分を考慮した IDF を用いた手法 (以下,CMU 最適化) を検出のベースラインとし, 提案手法を評価する.3 つのトピックで検出と追跡をそれぞれ行. ベースラインは複数のトピックを想定しており, 記事は時系列順になっている. 本研究も同様に行う.

全記事にトピックのラベルが含まれているがストーリーの情報は含まれていない. 実験では「CD-毎日新聞 2013 データ集」の 1 月 1 日の先頭から, ストーリーの流れが完結している「千葉・市川女性刺殺事件」(以下, 千葉) の記事全 11 件と,「山口・周南放火殺人事件」(以下, 山口) の記事全 12 件,「富山・放火殺人事件」(以下, 富山) の記事全 17 件を手で抽出する. 事前に形態素解析を行った後, 名詞のみを抽出する. また, 尤度の上位 K 個の経路を常に候補として保有し, それぞれでパープレキシティ (PP) 値を求める. 予備実験を通して幾つかの閾値を取るが, ここでは K-Viterbi・後ろ向き K-Viterbi アルゴリズムでは K=2 を採用する. 各手法の類似度に対する閾値を以下の表 1 に示す.

表 1　各手法における閾値

	提案手法			CMU	
	山口	千葉	富山	CMU	CMU 最適化
検出	0.85	0.9	0.85	0.997	0.95
追跡	0.025	0.05	0.5		
モデル性能	180	200	180		
クラスタ内外の纏まり	0.35	0.34	0.34		

また,

検出の評価には未検出率 (検出に失敗した記事の割合), 誤検出率 (誤って検出した記事の割合), それらに基づく正規化した Detection Cost Function(DCF)[1]の 3 つの基準を用いる [6]. 追跡の評価基準は検出と同様の計算で求めるが, 区別するためにここでは未追跡率, 誤追跡率,Tracking Cost Function(TCF) と定義する.

5.2 実験結果

3 つのトピックを追跡対象としてそれぞれ行う. 各手法におけるトピック検出および局面追跡の結果を表 2, 表 3 にそれぞれ示す.

表 2　各手法における性能評価値 (トピック検出)

富山		ベースライン (CMU 最適化)	ベースライン (CMU)	提案手法
	未検出率 (%)	0.7059 (12/17)	0.2941 (5/17)	0.2353 (4/17)
	誤検出率 (%)	0 (0/23)	0	0
	Cost	0.3	0.125	0.1
山口		ベースライン (CMU 最適化)	ベースライン (CMU)	提案手法
	未検出率 (%)	0.75 (9/12)	0.5833 (7/12)	0.0833 (1/12)
	誤検出率 (%)	0	0	0.0357 (1/28)
	Cost	0.225	0.175	0.1083
千葉		ベースライン (CMU 最適化)	ベースライン (CMU)	提案手法
	未検出率 (%)	0.4545 (5/11)	0.4545 (5/11)	0.6364 (7/11)
	誤検出率 (%)	0	0	0
	Cost	0.125	0.125	0.175

表 3　各手法における性能評価値 (局面追跡)

富山		ベースライン	提案手法
	未追跡率 (%)	0 (0/15)	0.75 (3/4)
	誤追跡率 (%)	0.4211 (8/19)	0 (0/23)
	Cost	0.5333	0.1111
山口		ベースライン	提案手法
	未追跡率 (%)	0.1 (1/10)	0 (0/1)
	誤追跡率 (%)	0 (0/24)	0 (0/28)
	Cost	0.0294	0
千葉		ベースライン	提案手法
	未追跡率 (%)	0.6667 (6/9)	0.8571 (6/7)
	誤追跡率 (%)	0 (0/25)	0 (0/29)
	Cost	0.1765	0.1667

トピック検出の DCF において, 提案手法では富山トピックで最大 0.2, 山口トピックで最大 0.12 改善している. 誤検出している記事はあるものの, 全体的にベースラインよりも正しい記事を多く得られている. さらに, 富山トピックでは 2 区画目でクラスタ内外の纏まり閾値を超えたため, 後の他トピックの記事 28 件を処理することなく終了している.

局面追跡の TCF において, 提案手法では富山トピックで 0.42, 山口トピックで 0.029, 千葉トピックで 0.01 改善している. 富山トピックでは 3 区画目でモデル性能の閾値を超えたため, 他のトピックの記事 25 件を処理することなく終了している. また, 千葉トピックでは検出で得られなかった記事 (番号 5) が追跡で得られており, 検出の補完が行えている.

5.3 考　　察

5.3.1 トピック検出における DCF

提案手法では富山トピックで DCF が 0.1(未検出率 23.5%, 誤検出率 0.0%), 山口トピックで DCF が 0.1083(未検出率 8.3%,

1：未検出率と誤検出率に基づいてシステムの性能を評価する関数. 式は $C_{Det} = C_{miss} \times P_{miss} \times P_{target} + C_{false} \times P_{false} \times (1 - P_{target})$ で与えられる. 正規化の式は,$(C_{Det})_{Norm} = C_{Det}/Min(C_{miss} \times P_{target}, C_{false} \times (1 - P_{target}))$ である.C_{miss},C_{false} はそれぞれ 1 で与えている.P_{miss} は未検出率,P_{false} は誤検出率,P_{target} はコーパス中に含まれる正解トピックの記事集合の割合である.

誤検出率 3.6%) であり, ベースラインより改善している. この原因として, 各記事に適用した確率的 TF-IDF にあると考えられる. ベースラインおよび提案手法での各記事の類似度の比較を, 以下の表 4 に示す.

表 4　各手法における性能評価値 (局面追跡)

提案手法 (閾値:0.85)			ベースライン (CMU)(閾値:0.997)			
cluster1	cluster2		cluster1	cluster2	cluster3	cluster4
		富山 1				
0.993079549		富山 2	0.999676661			
0.9400628		富山 3	0.996670738			
0.94137681		富山 4	0.996750794	0.999308266		
0.934366291		富山 5	0.996082754	0.997054324		
0.83527228		富山 6	0.994927881	0.994148886		
0.933241667	0.830811173	富山 7	0.995707958	0.997407739	0.992469825	
0.913772632	0.81478893	富山 8	0.994909694	0.99760953	0.991617724	
0.881335434	0.79353116	富山 9	0.995370743	0.997213689	0.99212898	
0.857023514	0.771419636	富山 10	0.994045133	0.996231752	0.990655636	
0.858370607	0.783899849	富山 11	0.994945013	0.997571107	0.991700799	0.995510695
0.859017126	0.812699878	富山 12	0.99577382	0.997784505	0.992464027	0.995885737
0.858888619	0.811943235	富山 13	0.995206237	0.997684486	0.991754478	0.994406151
0.85982732	0.783772708	富山 14	0.994925127	0.997045583	0.991458067	0.994084378

ベースラインでは, 各クラスタにおける類似度に大きな差がみられない. また, 提案手法と同様の閾値または下げてしまうと誤検出率が高くなり, 現閾値でも微妙な差で別クラスタに振り分けられてしまう問題が出ている.

しかし提案手法では, 遡及性クラスタリングおよび確率的 TF-IDF に基づく手法により類似度に大きな差が出ている. このため各記事がどのクラスタに所属するかが明確になり, クラスタの不要な増加も防いだと考えられる.

5. 3. 2　局面追跡における検出の補完

提案手法による追跡で, 検出で選びきれなかった千葉トピックの記事 (番号 5) が追跡可能となっている. この原因として, 構築されたマルコフモデルにあると考える. 学習データのみで構築したマルコフモデルおよびテストデータ (記事 5) を含んだ記事集合で構築したマルコフモデルを以下の図 6, 図 7 にそれぞれ示す. なお, ここでの状態名は各状態に付随する出力語を基に記述している.

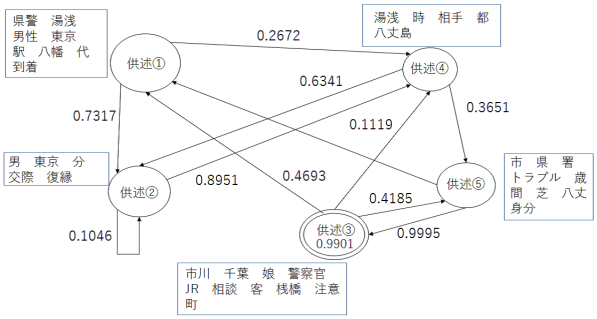

図 6　学習データのみで構築したマルコフモデル

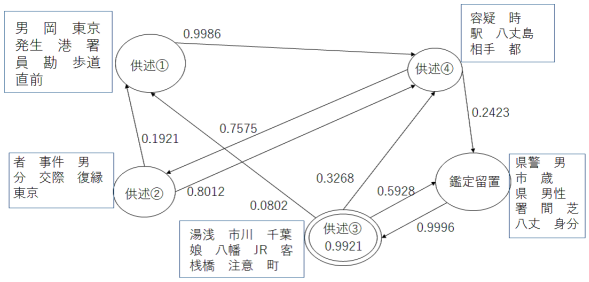

図 7　テストデータを含めて構築したマルコフモデル

各状態に付随する出力語は出力記号確率が大きい語を採用し, 記述している. これら 2 つのマルコフモデルの出力記号をみるとほぼ違いが見られず, 状態遷移にも大きな変化は見られない. このため, 図 7 のモデルでテストデータの PP 値を推定すると, 語の並びや出力語が似ているため学習データの平均 PP 値との差が小さくなり, 追跡可能となったと考えられる. 以下にテストデータと平均 PP 値との差を表 5 に示す.

表 5　各手法における性能評価値 (局面追跡)

	学習データの平均 PP 値	記事 5	差
1-PP	3.626728166	3.636609558	0.009881391
2-PP	3.627053815	3.636609558	0.009555742

6　結　　論

本研究では, 尤度計算による局面追跡と, 確率的 TF-IDF による遡及性を考慮したトピック検出のオンラインでの手法を提案した. 富山, 山口トピックの検出と追跡, 千葉トピックの検出においてベースラインよりも良い性能を示した. 結果の一部として, 富山トピックの DCF では最大で 0.2,TCF では 0.42 以上も改善された. また, 千葉トピックでは, 検出で得られなかった記事が追跡で得られ, 検出と追跡のそれぞれの結果でお互いを補完する事により検出漏れを防ぐことが可能となった.

富山トピックでは検出が 2 区画目, 追跡が 3 区画目で終了した. 各手法で打ち切り閾値を設けた事により後の他トピックの記事を処理することなく終了し, 計算量の削減が可能となった.

文　　献

[1] Aizawa, A., “An Information-Theoretic Perspective of Tf-idf Measures”, Information Processing and Management, Vol.39, No.1, pp.45-65, 2003

[2] Andrew D. Brown, Geoffrey E. Hinton, “Products of Hidden Markov Models”, Gatsby Computational Neuroscience Unit, pp. 1–8, 2001

[3] Bilmes, J.A., “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gausian Mixture and Hidden Markov Models”, ICSI, pp.1–15, 1998

[4] Daniel Jurafsky, James H. Martin, “Hidden Markov Model”, Speech and Language Processing, 2016

[5] James Allan, Jaime Carbonell, Geroge Doddington, Jonathan Yamron, and Yiming Yang, “Topic Detection and Tracking Pilot Study Final Report”, In Proceedings of the DARPA Broadcast News Transcription and Understanding

Workshop, pp. 194–218, 1998
[6] Jonathan G. Fiscus, George R. Doddington, “Topic Detection and Tracking Evaluation Overview”, Topic detection and tracking, pp. 17–31, 2002
[7] M. Callon, J.P. Courtial, F. Laville, “Co-word analysis as a tool for describing the network of interactions between basic and technological research: The case of polymer chemsitry”, Scienmetrics Vol.22, pp. 155–205, 1991
[8] Maoto Inoue, Takao Miura, “Tracking Aspects in News Documents”, IEEE International Conference on Data Mining Workshops (ICDMW) IWSC, pp. 1–8, 2018
[9] Yiming Yang, Jaime G. Carbonell, John Lafferty, Ralf D. Brown, Tom Pierce, Xin Liu, “CMU Report on TDT-2: Segmentation, Detection and Tracking”, In: DARPA Broadcast News Transcription and Understanding Workshop(1999), pp. 3–6, 1999
[10] Zhou D., Xu H., “Unspervised Storyline Extraction from News Articles”, IJCAI 2016, pp. 3014–3020
[11] 井庭 崇, “探索的モデリング”, pp. 5–11, 2004