特集号
招待論文

大規模トラフィックを処理する記事推薦サービスへの機械学習の応用事例

大倉 俊平[†1]　村尾 一真[†1]　田頭 幸浩[†1]　小野 真吾[†1]　田島 玲[†1]

[†1] ヤフー（株）

記事推薦システムにおいて，機械学習を用いてユーザと記事をマッチングする手法については，すでにさまざまな研究が成されている．一方で，大規模トラフィックを処理する実サービスにそれらを適用しようとすると，システムの応答速度に関する制約や，季節性によるデータ分布の変化などさまざまな問題が浮上する．また，機械学習の「精度」が利用者の満足に直接結びつくとは限らない．本稿では，筆者らが実際の記事推薦システムを構築する上で，それらの問題にどのように対処したかを事例と実験結果を合わせて紹介する．

1. はじめに

ニュース媒体において，ユーザに世の中のすべてのニュースを提供することは現実的に不可能であるため，何らかの手法によって記事を選択して提供する必要がある．

テレビや新聞などの媒体では，この選択は人手によって行われ，その結果選ばれた記事をすべてのユーザに向けて提供するのが一般的である．

一方で，インターネット上のニュースサイトにおいては，Cookieなどを用いて，ユーザを特定する情報をページ表示前に取得することができ，それによってページの内容を動的に構成することが可能である．そのため，ユーザの行動から好みを推定し，それぞれのユーザにより適切な記事を配信する試みが成されてきた．古くは，人手で作成した簡単なルールに基づいて配信されていたが，現在では記事量の増加やユーザの嗜好の多様化に伴って，機械学習を用いて自動的に記事選択を行うのが主流となっている[1],[2].

我々も，この流れに従っており，サービス初期ではYahoo!ニュース トピックス（以下，トピックス）として，編集者が記事の選定と掲載順序の制御をすべて行っていたが，現在は一部で「あなたへのおすすめ」として機械学習を用いたパーソナライズを行っている．ユーザが提示された記事を実際に読んだか否かというフィードバックを使って，よりそのユーザが読みたい記事を選定できるように最適化を進める．

このようなパーソナライズは，記事のクリック率を上げ，ユーザのサービス利用を促すためには非常に重要な要素となっている．特に実用上では，ユーザ数も候補記事数も多く，かつ情報の鮮度も求められるため，推薦の精度だけでなく，大量のリクエストを処理する前提のモデル構築や，重複情報を含む記事をうまく除去する仕組みも必要とされている．

一方で，筆者らが事前に行ったテストでは，共通の記事を減らし，個人の興味に則した記事の比率を増やすと，クリック率は増加するものの，サービスからの離脱率も促進されるという結果が得られた．ユーザはパーソナライズ部分によって自身の興味のある記事を読みつつも，「トピックス」によって世間の動きを把握するといった使い分けをしているようである．

つまり，全ユーザにとって共通に有益な情報を選定するという枠組みにおいては，編集者の中にはクリック率という指標だけでは計れない，別種の記事選定基準がノウハウとして蓄積されているといえる．たとえば，政治や経済の記事は全般的にクリックされづらい傾向にあるが，「トピックス」では必ず掲載されている．編集者によれば，「トピックス」掲載基準にはユーザの関心の高さに加えて，「公共性」という軸があるという[3]．我々は編集者の知見をうまく取り込むことで，「トピックス」の果たす役割を持つ記事も機械学習によって半自動的に選定する仕組み（以下，オートトピックス）も開発している．

図1はスマートフォン版Yahoo! JAPANトップページの画面イメージである．2016年5月現在，検索ボックス，各サービスへのリンクなどを含むヘッダに続いて，前述した「トピックス」，「オートトピックス」，「あなたへのおすすめ」の3段構成になっている．

本稿では，第2章，第3章で「あなたへのおすすめ」における学習と重複記事排除の事例を紹介し，第4章で

図1　Yahoo! JAPAN トップページの画面イメージ（2016年5月現在）

「オートトピックス」における取り組みを紹介する．最後に，第5章でまとめと今後の展望を述べる．

2. 閲覧履歴に基づくパーソナライズ推薦「あなたへのおすすめ」

本章では，「あなたへのおすすめ」と称する記事のパーソナライズの仕組みについて述べる．前章でも述べたように，ここでは精度の高い推薦であることに加えて，次々と入稿される新着記事を即座に反映しつつも大量のリクエストに応答できる構造になっていることが求められる．

たとえば，自動推薦においてよく知られた手法としてCollaborative filteringやMatrix factorization[4]といった，ユーザ–アイテム行列を用いたIDベースの手法が挙げられる．しかし，Zhongらは，これらの手法は記事推薦には適さないと述べている[2]．なぜなら，即時性が重要なニュース記事は記事の入れ替わりが激しく，IDベースの学習では新着記事に対して迅速な対応ができないからである．つまり，記事の過去の履歴に頼るのではなく，記事の内容をきちんと捉えた上で推薦することが重要である．

すなわち，ニュース記事の推薦は，以下の3つのステップに分解される．

- 記事内容の理解
- ユーザの興味の推定
- それらを考慮した記事引き当て

以下では，最初にこれらの簡単な実装を示した上で，その問題点を挙げ，それら改善した手法との比較実験を紹介する．

2.1　ベースラインモデル

まず，我々が初期に作成した簡潔なシステムを，ベースラインモデルとして説明する．

記事内容の特徴は，その本文中に含まれる形態素の分布として現れると考えられる．そこで，記事を形態素の集合として表現した．ただし，補助的な語は記事内容と直接関係ない場合が多く，ノイズとなることが想定される．そのため品詞解析により，名詞句のみを抜き出して使用することとした．

また，ユーザの興味は，以前閲覧した記事に傾向として現れているものと考えられる．そこで，そのユーザが過去の一定期間に閲覧した記事すべてを1つの記事と見なし，記事の場合と同様に形態素の集合を抽出した．

配信時には，ユーザが持つ形態素の集合と多くの重複語を持つ記事を優先的に表示することにする．ただし，形態素の中には，名詞句に絞ったといえども，その中でも個人の興味を強く表す重要な語もあれば，そうでない語も存在する．たとえば「試合」のような多くのスポーツに共通する語よりも，「野球」のようなジャンルを限定する語はより強い情報となり，特定の選手名や球団名はさらに限定した興味を表す．そこで，配信結果を用いた回帰学習によって，形態素ごとの重みを学習し，その重みに従った形態素の重複度合いに

より，配信優先度を決定する．

このような，重み付き形態素集合による引き当ては従来の検索エンジンと相性が良く，ブーリアン検索を改良したWAND（Weak AND）アルゴリズム[5]などを用いることにより充分高速な引き当てが可能である．

2.2　ベースラインモデルにおける課題点

前節で述べたような，形態素ベースの引き当てでも，各形態素の重みをうまく学習することで，良い推薦は可能である．しかし，以下の2点が課題として挙げられる．

- 単語の表記ゆれ
  似た内容を表す記事であっても，著者や媒体により表現がさまざまであり，略称や言い換えによって，マッチしない場合がある．
- 閲覧頻度や順序の欠損
  ユーザの履歴を集合として扱うため，頻度や順序の情報が欠損する．実際は，数週間前に閲覧した記事より，直前に閲覧した記事の方が，現在の興味に強く相関していると考えられる．

前者を解消するため，記事の分散表現化を導入した．ベースラインモデルで用いた，記事に出現した単語の有無を1/0のベクトルで表現する表現方法（bag-of-words表現）は局所表現と呼ばれるのに対して，分散表現は1単語を複数の次元に分散する連続値として表す表現方法を指す．一般的に局所表現は語彙の数と同じだけの次元を持つ高次元疎ベクトルであるが，分散表現は低次元ですべての次元に数値を持つ密ベクトルとなる．これにより，意味の近さを，表記が一致するか否かの離散値ではなく，連続値として表現することが可能になる．分散表現化について，詳しくは2.3節で述べる．

後者については，閲覧履歴を入力とするRecurrent Neural Network（RNN）を学習させることによって，閲覧順序を学習のフレームワークの中で考慮できるようにした．RNNを用いたモデルについては2.4節で詳しく述べる．

2.3　記事の分散表現の生成

本節では，記事内容を分散表現へ変換する手法について述べる．

既存研究でも文書から分散表現を得るものはいくつかある[6],[7]．しかし，それらは記事の良い特徴量を得ることを目的としており，作成された特徴量を用いて，後続で何らかの分類器等を作成することを前提としている．筆者らのケースでは，大量のユーザからのリクエストに対して，それぞれ数千〜数万の候補記事を評価する必要があるため，応答速度の制約上複雑な分類器を通すことは現実的ではない．実用上は，分散表現に対する簡易な演算（本稿では内積）によって，記事間の類似度が定義できると望ましい．

そこで，筆者らはDenoising Auto Encoder（DAE）[7]をベースに，内積での類似度評価がうまくいくように改良を加えた手法[8]を開発し，使用している．以下で，具体的に変換手法について述べる．

まず，ベースとなる一般的なDAEは以下のように定式化される．

$$
\begin{aligned}
&\tilde{x} \sim q(\tilde{x}|x)\\
&h = f(W\tilde{x} + b)\\
&y = f(W'h + b')\\
&\theta = \underset{W, W', b, b'}{\operatorname{argmin}} \sum_{x} L_R(y, x)
\end{aligned}
$$

W, W', b, b'はパラメータ行列，$f(\cdot)$は活性化関数，$q(\cdot|x)$はxに確率的なノイズを与える分布，$L_R(\cdot,\cdot)$は2つの引数の近さを与える損失関数である．入力xは記事のbag-of-words表現に相当する疎ベクトルを想定している．hが得られる分散表現であり，hから元の入力を復元したyが，ノイズを加える前の入力xにできる限り近づくようにパラメータ群θを学習する．

式からも分かるように，hはxを復元するための情報をできる限り保持することが要請されているが，h自身が成す空間の内積構造は保障されていない．そこで筆者らは，記事に付与されていたカテゴリラベルを元に，内積構造に関する以下のような損失項L_Tを加え，内積が分散表現同士の近さを表すように拡張した．

$$
\begin{aligned}
&\tilde{x}_n \sim q(\tilde{x}_n|x_n)\\
&h_n = f(W\tilde{x}_n + b) - f(b)\\
&y_n = f(W'h_n + b')\\
&\theta = \underset{W, W', b, b'}{\operatorname{argmin}} \sum_{(x_1, x_2, x_3)} \sum_{n=1}^{3} L_R(y_n, x_n) + \alpha L_T(h_1, h_2, h_3)\\
&L_T(h_1, h_2, h_3) = -\log(\sigma(h_1^{\mathrm{T}} h_3 - h_1^{\mathrm{T}} h_2))
\end{aligned}
$$

記事x_1とx_2は同一のカテゴリの記事，記事x_3は異なるカテゴリの記事を用いて学習させる．このとき，h_1とh_2の類似度がh_1とh_3の類似度よりも大きくなることが望まれる．そこで，内積の大小関係が逆になったときには，L_Tが大きな損失を与えるように設計されている．$\sigma(\cdot)$はシグモイド関数である．

この拡張により，記事引き当て時に，コストの低い低次元ベクトルの内積計算のみでも，うまく動作するようになる．

2.4 RNNを利用したユーザモデル

本節では，前節で生成した記事表現の空間に対して，各ユーザの興味・関心を表す表現uを生成する．すなわち，記事xの分散表現をhとするとき，ユーザのxへの興味が強ければ$u^{\mathrm{T}}h$が大きくなるようなベクトルuをユーザごとに生成したい．

そのユーザが過去に閲覧した記事の列を$\{x_k\}_{k=1}^{n}$とし，対応する分散表現を$\{h_k\}_{k=1}^{n}$とする．ユーザの興味は記事を見ることによって移り変わっていくと仮定すると，記事x_kを見た直後の興味状態をu_kと表現すれば，下記のような再帰的な関係があると考えられる．

$$u_k = F(h_k, u_{k-1})$$
$$u = u_n$$

筆者らは，以下に述べるように，この関数FをRNNにより学習させることを試みた．単純なRNNでは，

$$F(h_k, u_{k-1}) = f(W_1 h_k + W_2 u_{k-1} + b)$$

の形で表される．$f(\cdot)$は活性化関数である．図2はこのモデルを表したイメージ図である．

また，関数$F(\cdot,\cdot)$の部分に，LSTM[9]やGRU[10]と呼ばれるユニットを用いることで，より複雑な再帰関係を表現できる．これらは，計算が複雑になる一方で，勾配降下法における勾配爆発・消失問題[11]を緩和する効果があり，学習の収束を促進する．

2.5.1節ではLSTM，2.5.2節ではGRUに全結合層を1層追加したモデルを採用してテストを行っている．活性化関数はいずれもtanhを採用した．

サービスに適用する場合には，記事の表現hとユーザの表現uは事前に計算しておく．リクエストがあった時点で，配信可能なすべての候補記事に対して内積$u^{\mathrm{T}}h$を計算し，その値が高いものから優先的に配信する．低次元の内積は，同一の演算命令を複数のデータに対して並列に実行するSIMD（Single Instruction/Multiple Data）演算などを用いることにより，実際の応答速度の制約下でも計算可能である．

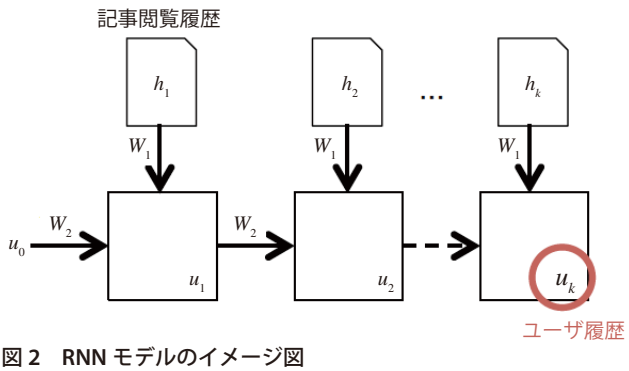

図2 RNNモデルのイメージ図

2.5 実験結果

2.5.1 失敗例

筆者らは，はじめに2015年11～12月の2カ月間の配信ログを用いて，RNNによるユーザモデルの学習を行い，2016年2月に2.1節のベースラインとA/Bテストによる比較を行った．A/Bテストとは，実際のシステム上で，ユーザからのリクエストをランダムにモデルAとモデルBに振り分け，それぞれのクリック数などの指標を比較するテスト手法である．

モデル作成時のシミュレーションでは，AUC（Area Under the ROC Curve）等のランキング指標が8%程度向上することを確認していたが，A/Bテストにおいてクリック数の向上は確認できなかった．

テスト後の分析によると以下のことが分かった．

- 学習期間と同時期の別のユーザにはよく適合するが，時期がずれると精度が大きく下がる．
- RNNモデルは，ユーザの興味・関心よりも，その時期の流行記事を強く学習している．

また，別の実験ではユーザモデルと記事の分散表現の生成をつなげたネットワークにファインチューンを施すパターンも検証したが，上記の問題をさらに加速させる結果となった．これらは，RNNモデルの表現力の高さに対して，学習データの収集期間が短く，季節要因が強く反映されてしまったことが原因の一端であると考えられる．

2.5.2 成功例

前節の失敗を踏まえて，2015年5月～2016年2月の10カ月間のログから生成した長期間の学習データを用意し，2016年5月に再度A/Bテストを実施した．

その結果，2.1節のモデルに比べて，2.4節のモデルは全体で約4%のクリック数向上を確認した．また，ユーザの利用頻度に応じたセグメントごとの分析を行った結果，表1の結果を得た．

これは，閲覧履歴の豊富なヘビーユーザに対しては，形態素ベースのモデルでも，多くの形態素が付与されるため良い推薦が可能であるが，履歴の少ないライト

表1 セグメント別のクリック数増加率

名称	利用頻度	クリック数増加率
ヘビー	週6日以上	+0.9%
ミドル	週2～5日	+11.6%
ライト	週1日以下	+23.9%

層には2.2節で挙げた問題点が顕著であり，それが改善された結果であると考えられる．

3. 重複排除

本章では，機械学習を用いて記事選定を行った場合に発生する記事重複の問題について述べる．

3.1 記事重複の問題

我々のWebサイトでは，多くの媒体から記事の提供を受け配信を行っている．たとえば，ある重要な出来事がニュースになると，それぞれの媒体が個別に記事を作成して，配信記事候補として入稿を行う．すると，ほぼ同様の内容が書かれた記事が複数配信候補に含まれることになる．

これに対して，前章で構築したユーザの興味・関心に基づいたランキングを行った場合，それらの類似記事はお互いに近い評価値・順位になる．

特に携帯端末では，表示できる記事数が限られているため，直接それらを表示すると，似た記事のみで画面が占有されることがある．この現象は，ユーザの効率的な情報収集を妨げ，満足度の低下を引き起こす．そのため，重複記事を適切に省くことが必要となる．

3.2 重複排除手法

重複情報を排除するタイミングとして以下の2つが考えられる．

- 記事入稿時
  すでに候補として保持している記事と，重複した記事が入稿された場合，いずれか一方を残して，他方を候補から除外する．
- 配信時
  ランキング後に，すでに表示済みの自身より上位の記事と比較して，重複した情報である場合に，表示せずにスキップする．

筆者らは，後者の方法を採用した．なぜなら，特にパーソナライズにおいては，前者の手法で記事を選択する場合どちらを残すべきかが明らかではないからである．同じ内容で著者の異なる記事A，Bがあるとする．このとき，あるユーザXはより詳しく書かれたAが良いと言い，他のユーザYはより分かりやすく書かれたBが良いと言う可能性もある．この場合，XにはAを表示し，YにはBを表示するのが望ましいが，記事入稿時の重複排除では，このような対応はできない．

配信時に重複排除を行う場合は，先に表示されるランキング上位の記事から重複判定を行うため，2記事間の類似度が求まれば，適切に閾値を設定することで目的が達成される．

3.3 予備実験

記事間類似度指標として，以下の3つを比較する．

- 記事タイトルの形態素単位でのcosine類似度
- 記事本文の形態素単位でのcosine類似度
- 2.4節の分散表現のcosine類似度

ここでも，分散表現を利用するためには，配信時に評価するため短時間での計算が求められ，2.4節で述べた工夫が重要であることを再度触れておく．

評価に際して，分散表現の学習時に用いたものとは投稿日の異なる記事を用意し，投稿時間のある程度近い記事でランダムにペアを生成した．このうち，明らかに内容が無関係なペアを除外し，評価対象400ペアをサンプリングした．これらに対して，「トピックス」編集者に5段階の類似度ラベルの付与を依頼した．「1」は2つの記事がまったく異なることを表し，「5」は2つの記事がほぼ同一の内容であること表す．

編集者の付与したラベルと，3種類の類似度をプロットした図を図3に示す．記事本文の形態素を用いた場合は，多くの記事に共通で現れる，直接記事の内容

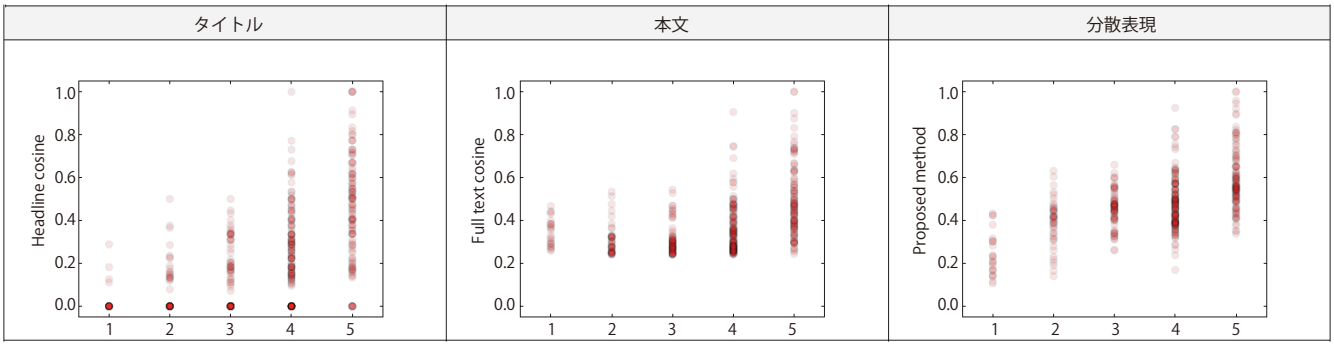

図3　編集者によるラベルと類似度の相関

を表さない語がノイズとなり，うまく類似度が測れていないことが分かる．タイトルを用いた場合は，一見強く相関していそうではあるが，類似度0のところに多くのペアが固まっている．これはタイトルのみだと形態素数が少ないため，似た内容の記事であっても形態素が重複しない場合が多いことを示している．

一方で，分散表現を用いて類似度を測った場合には，すべてのペアに対して連続的に類似度の値を付与できている．また，編集者の評価で比較的類似度が低いとされた1～3においても，cosine類似度がラベルと相関を持って変化していることが見てとれる．

3.4 A/Bテストによる比較

前節の予備実験で，人手のラベルと相関の見られた，タイトルのcosine類似度，分散表現のcosine類似度の2つを実際のシステムに導入し，ユーザの行動について比較実験を行った．結果を表2に示す．

分散表現を用いた重複排除を行った方が，1セッションあたりのクリック数が増加している．これは重複排除がうまくいったことで，より多様な記事が表示されたことによるものと考えられる．また，閾値を下げると，クリック数を維持したまま，スクロール量は減少している．これは，必要な記事を残したまま記事リストを圧縮できていることを表す．

表2 類似度指標によるユーザ行動の変化

類似度	閾値	スクロール量増加率	クリック数増加率
タイトル	0.4	基準値	基準値
分散表現	0.6	+5.25%	+2.32%
分散表現	0.5	+3.31%	+2.69%
分散表現	0.45	+1.61%	+2.99%

4. 編集者の知見に基づいた半自動記事推薦「オートトピックス」

第1章で述べたように，事前のテストの結果，ユーザはパーソナライズされた「あなたへのおすすめ」によって自身の興味のある記事を読みつつも，「トピックス」によって世間の動きを把握していると考えられる．

「トピックス」では，人々に知ってもらうべき政治や経済などの重要なニュースを，1日数千本以上提供される記事の中から適切にピックアップする必要があり，その質を担保するためには，高い技能を持った編集者が不可欠である．一方，高い頻度でユーザがアクセスするスマートフォン用のサービスでは，これまでのPCでのサービスと比較して，より高い更新頻度が求められる．

そこで「オートトピックス」では，一部の機械学習の学習データとして，ベテラン編集者の記事選定結果を用いることで，選定基準の「公共性」にあたる部分を数値化し，それを参考にして経験の浅い編集者でも選定作業が可能となるようにした．加えて，記事の予測閲覧数や，ユーザの訪問間隔などを考慮して，動的に掲載記事を変更することで，訪問頻度の高いユーザに対しても，更新感のある表示を提供している．

4.1 システム概要

図4は「オートトピックス」編成フローのイメージ図である．編成フローは大きく3つのパートからなる．

1つ目のパートは，入稿された記事を自動的に優先順位付けするパートである．ここでは，ベテラン編集者によって選定された「トピックス」を学習することで

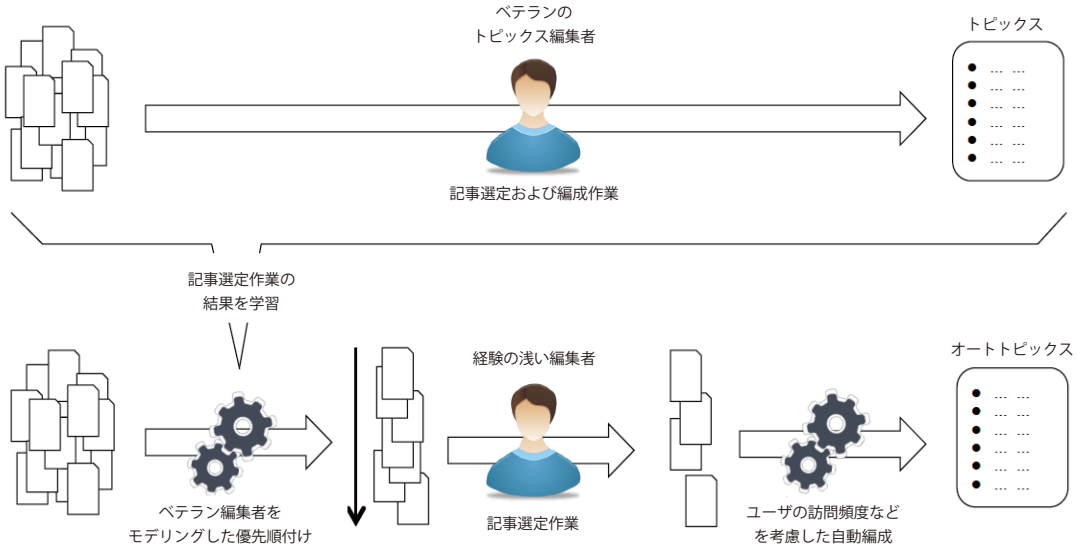

図4 オートトピックス編成作業イメージ

モデル化された「公共性」に関するスコア(次節で詳述する)と，時系列データを用いた各記事の閲覧数の推定による，ユーザ全体の「関心度」を示すスコア[12]によって，記事に優先度を付与する．

2つ目のパートは，編集者によって記事の選定を行うパートである．ただし，前段のシステムによって優先度付けされているため，効率的に確度の高い記事のみに目を通せるのに加えて，比較的経験の浅い編集者であっても，まったく的外れな記事を選んでしまうリスクが軽減される．また，人の目を介することで，機械学習の誤判定で致命的な記事が選ばれてしまうリスクも回避することができる．

最後のパートでは，編集者の選定した記事から自動的に出面の編成を行う．前段の選定作業では多めに記事を選んでおき，ここではユーザの訪問間隔によって記事の出し分けを行っている．頻度の低いユーザには候補の中でより優先度の高い記事を表示し，頻繁に利用するユーザには，少し優先度が低くても，そのユーザが前回目にしていない記事を表示する．これによって，記事の品質と更新感のバランスを担保している．

4.2 「公共性」モデルの検証実験

本節では，最初のパートで用いられる「公共性」を表現するモデルについて筆者らが実施した評価実験を紹介する．現在までのベテラン編集者による記事の選定行動から未来の選定行動を当てる予測問題を，2クラス分類問題と見なして検証を行った．2014年11月～2015年2月の4カ月間で「トピックス」に選定された13,210記事と，同期間に入稿されたが選定されなかった記事をサンプリングした10,000記事を訓練データとし，評価データとして訓練データ収集最終日の翌日の選出記事95記事，非選出記事100記事(サンプリング)を用意した．

AUCによる性能評価の結果を表3に示す．記事タイトルおよび本文のbag-of-words表現(BOW)からの予測のみでも，ある程度高い予測性能が認められたが，記事の入稿媒体や記事のジャンルを加味することでさらに高い予測精度が出ることが分かった．特に，得られた素性の重みを見ると，たとえば記事ジャンルにおいては，社会や政治などのスコアが高くなっており，ベテラン編集者のピックアップ行動における公共性の観点をある程度模倣できているのではないかと考えられる．

なお，実システムにおいては，時事性も重要な要素であるため，公共性モデルはオンライン学習によって常に更新している．今後の課題として，記事の表現として分散表現を導入した場合にこの精度がどのように変化するのか検討していきたい．

表3　使用した素性と選定予測の精度の関係

素性	AUC
BOW	0.871
BOW + 入稿媒体名	0.889 (+2.06%)
BOW + 入稿媒体名 + 記事ジャンル	0.909 (+4.31%)

5. まとめと今後の展望

本稿では，Yahoo! JAPANのスマホ版トップページにおける記事推薦についての，機械学習を用いた取り組みを紹介した．

機械学習においては，クリック率などの数値指標を上げることのみに注力しがちである．一方で，冒頭でも触れたように，クリック率には直接現れないが，ユーザがニュースサイトを利用する目的の1つとなっている記事も存在する．そこには，まだまだ経験を培ってきたニュース編集者の感覚と，機械の間にはギャップがあるといえる．

筆者らは，第3章，第4章の取り組みを始めとして，ニュース編集者の知見と機械学習での最適化をうまく組み合わせることにより，より良いユーザ体験の創出を目指している．

加えて，第2章でも述べたように，特に実サービスでリアルタイムに稼働する場合は，システムの計算リソースや許容時間を十分に考慮して，フレームワークを考えることが重要である．2.5.1節のように，オフラインの評価時には気づかなかった点が，実装し稼働してみると問題になることもあるため，十分に注意しなくてはならない．

また，本稿では詳しく触れなかったが，モデルの更新頻度と精度のトレードオフの問題も重要である．特に多層のニューラルネットワークモデルを学習するためには，大量のデータと長い時間の学習を要する．

一方で，リアルタイムにフィードバックデータを得られるシステムの場合は，直近の傾向を追うことが精度向上によく効く場合が多い．そのため，表現力が高く精度の高い複雑なモデルを採用するよりも，学習が容易な簡単なモデルを用いて頻繁にモデル更新を行うほうが，最終的に結果が良いケースもあり得る．それらのトレードオフを踏まえて，うまく組み合わせて利用することも課題の1つである．

我々は，現在もさまざまなA/Bテストを通して，日々改善を行っている．今後も，ユーザのさらなる満足度向上のため，研究・改善を続けていく予定である．

謝辞　本稿の作成にあたり，日々 Yahoo! JAPANの各サービスを利用し，フィードバックをくださっている，すべてのユーザの皆様に深謝いたします．

参考文献

1) Das, A. S., Datar, M., Garg, A. and Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering, In Proceedings of the 16th International Conference on World Wide Web, WWW '07, pp.271–280 (2007).
2) Zhong, E., Liu, N., Shi, Y. and Rajan, S.: Building Discriminative User Profiles for Large-scale Content Recommendation, In Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 2277–2286 (2015).
3) 毎日新聞経済プレミア：月間閲覧 100 億超 王者ヤフーニュースの圧倒的強さ，http://mainichi.jp/premier/business/articles/20150925/biz/00m/010/004000c（2016 年 5 月 12 日現在）
4) Koren, Y., Bell, R. and Volinsky, C.: Matrix Factorization Techniques for Recommender Systems, Computer, 42(8), pp.30–37 (2009).
5) Broder, A. Z., Carmel, D., Herscovici, M., Soffer, A. and Zien, J.: Efficient Query Evaluation Using a Two-level Retrieval Process, In CIKM '03, Proc. of the Twelfth Intl. Conf. on Information and Knowledge Management, New York, ACM, pp.426–434 (2003).
6) Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, In Proceedings of The 31st International Conference on Machine Learning, ACM, pp.1188-1196 (2014).
7) Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A.: Extracting and Composing Robust Features with Denoising Autoencoders, In Proceedings of the 25th International Conference on Machine Learning, ACM, pp.1096–1103 (2008).
8) Okura, S., Tagami, Y., and Tajima A.: Article De-duplication Using Distributed Representations, In Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16, pp.87-88 (2016).
9) Hochreiter, S. and Schmidhuber, J.: Long Short-term Memory, Neural Computation, 9(8), pp.1735–1780 (1997).
10) Chung, J., Gulcehre, C., Cho, K. H. and Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, In NIPS 2014 Deep Learning and Representation Learning Workshop (2014).
11) Hochreiter, S.: Untersuchungen zu Dynamischen Neuronalen Netzen, Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München (1991).
12) 大倉俊平 , 田頭幸浩 , 小野真吾 , 田島　玲：タイムライン形式での記事配信における課題と記事閲覧数推定による改善，第 8 回データ工学と情報マネジメントに関するフォーラム（DEIM Forum 2016），B7-5 (2016).

大倉 俊平（非会員）sokura@yahoo-corp.jp
2012 年東京大学大学院数理科学研究科修士課程修了．同年ヤフー（株）に入社．広告関連のシステム開発などの業務を経て，2014 年よりニュースレコメンドのロジック開発に従事．2016 年度言語処理学会若手奨励賞受賞．

村尾 一真（非会員）kmurao@yahoo-corp.jp
2012 年東京大学大学院情報理工学系研究科システム情報学専攻修士課程修了．同年ヤフー（株）入社．自然言語処理プロダクトの開発などを経て，2013 年よりスマートフォン版トップページにおける機械学習プロダクトのアルゴリズム開発に従事．

田頭 幸浩（非会員）yutagami@yahoo-corp.jp
2010 年東京工業大学大学院情報理工学研究科計算工学専攻修士課程修了．同年ヤフー（株）入社．ディスプレイ広告のクリック率予測モデルの構築など，機械学習技術の実サービス適用に携わる．現在，京都大学大学院情報学研究科知能情報学専攻博士課程に在籍中．

小野 真吾（非会員）shiono@yahoo-corp.jp
2009 年東京大学大学院情報理工学系研究科数理情報学専攻博士課程修了．同年ヤフー（株）に入社．Web 検索サービスの開発を経て，オンライン広告や E コマースサイトへの機械学習技術の適用に取り組む．博士（情報理工学）．

田島 玲（非会員）atajima@yahoo-corp.jp
1992 年東京大学大学院工学系研究科航空学専攻修士課程修了．日本アイ・ビー・エム（株）東京基礎研究所，A.T. カーニーを経て, 2012 年よりヤフー (株) Yahoo! JAPAN 研究所所長．2000 年に東京大学大学院理学系研究科情報科学専攻博士課程修了．博士（理学）．

採録決定：2016 年 8 月 5 日
編集担当：颯々野学（ヤフー（株））