分散表現を用いたニュース記事の重複排除

大倉 俊平　田頭 幸浩　田島 玲
ヤフー株式会社
{sokura, yutagami, atajima}@yahoo-corp.jp

1　導入

ウェブ上のニュース配信システムでは，ユーザーの興味に合致した情報を限られた表示面積と閲覧時間でより多く提供するために，冗長な記事を配信リストから排除することもまた重要である．例えば，ニュースの提供元が複数ある場合，同じ出来事について記述された複数の記事が同時に配信候補になることがある．この時，単純にユーザーの興味の度合いにしたがって記事をランキングすると，これらの同じ出来事について記述された記事が，配信リストの近い位置に表示されることになるが，ユーザーは似た記事を連続して目にすることになるため，満足度は低下することが予想される．そのため，重複する記事の中から一つを選択し，似た記事を排除するというアプローチが有効だと考えられる．

非常に似た記事を排除する場合には，それらに含まれる単語の共起度合いによって検出することができる [1]．しかしながら，言い換えや異なる文体を用いて書かれた記事間では，正しく類似度を測ることは難しい．一方で，記事に付与されたカテゴリやタグなどの情報は，単語レベルの情報よりも頑健かつ安定しており，類似度を判定するには有用ではあるが，重複排除という目的においては粒度が粗い．

本稿では，上記の問題に対して，記事の bag-of-words 表現から生成された分散表現を用いて記事の重複排除を行うアプローチを提案する．まず，記事間の類似度を，対応する二つのベクトルの内積の値で表現できるように，記事のカテゴリを弱い教師シグナルとして用いた学習を行うことで，低次元の分散表現を生成する．そして，生成された分散表現の内積の値で記事間の類似度を判定することで，配信リスト内から重複排除を行う．

実際のニュース配信システムは，入力となるサイト上でのユーザーの直近の行動や，ニュース記事が随時入稿されることによる候補集合の変化にしたがい，ユーザーがシステムにリクエストを送るたびに処理を行い，その時点で最適な記事の配信リストを返却することが期待される．この要求を満たすため，提案手法は，一つの配信リストに対して数ミリ秒程度で処理が完了するように設計されている．本稿では，Yahoo! JAPAN の実システムに提案手法を適用し，オンライン環境で行った実験結果についても紹介する．

2　提案手法

この章では提案手法の詳細について述べる．提案手法は二段階の手続きから構成される．一段階目では，分散表現間の内積の値が対応する記事間の類似度と連動するように，カテゴリ情報を弱い教師シグナルとして用いて，分散表現を生成するモデルを学習する．二段階目では，一段階目で生成した分散表現を用いて，実際にユーザーに提示する記事リストを作成する．

2.1　分散表現の生成

この節では，記事の分散表現の生成について述べる．記事の分散表現の生成には，denoising auto-encoder（dAE）[5] に弱い教師シグナルを付与したモデルを用いる．

通常の dAE は隠れ層が一層のニューラルネットであり，以下のように定式化される．

$$
\begin{aligned}
\tilde{\boldsymbol{x}} &\sim C(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \\
\boldsymbol{h} &= f(\boldsymbol{W}\tilde{\boldsymbol{x}} + \boldsymbol{b}) \\
\boldsymbol{y} &= f(\boldsymbol{W}'\boldsymbol{h} + \boldsymbol{b}') \\
\boldsymbol{\theta} &= \underset{\boldsymbol{W},\boldsymbol{W}',\boldsymbol{b},\boldsymbol{b}'}{\arg\min} \sum_{\boldsymbol{x}\in X} L(\boldsymbol{y},\boldsymbol{x})
\end{aligned}
$$

ここで，$\boldsymbol{x} \in X$ はオリジナルの入力ベクトル，$C(\cdot|\cdot)$ はノイズ分布である．$\tilde{\boldsymbol{x}}$ は入力ベクトル $\boldsymbol{x}$ に分布 $C(\cdot|\cdot)$ を用いてノイズを加えたベクトルであり，これをパラメータ行列 $\boldsymbol{W}, \boldsymbol{W}'$ とパラメータベクトル $\boldsymbol{b}, \boldsymbol{b}'$，ベクトルに対する活性化関数 $f(\cdot)$ で構成されるニューラル

ネットの入力とする．そして，ニューラルネットの出力として得られたベクトル $\boldsymbol{y}$ が，損失関数 $L(\cdot,\cdot)$ のもとでオリジナルの入力 $\boldsymbol{x}$ に近くなるようにパラメータの学習を行う．

通常，$\boldsymbol{x}$ に対応する分散表現として，隠れ層のベクトル $\boldsymbol{h}$ が用いられる．しかしながら，この分散表現 $\boldsymbol{h}$ は，$\boldsymbol{x}$ を復元するための情報を単純に保持しているだけである．一方で，我々が扱う重複排除の問題設定においては，二つの記事 $\boldsymbol{x}_1$ と $\boldsymbol{x}_2$ の類似度が高くになるにつれ，それに対応する分散表現 $\boldsymbol{h}_1$ と $\boldsymbol{h}_2$ から計算されるある値が連動して大きくなることが望まれる．そこで，記事にあらかじめ付与されているカテゴリ情報を用い，ある記事 $\boldsymbol{x}_1$ と，同一もしくは似たカテゴリに含まれる記事 $\boldsymbol{x}_2$，異なるカテゴリに含まれる記事 $\boldsymbol{x}_3$ で構成される三つ組 $(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3) \in X^3$ を用い，目的関数を以下のように拡張する．

$$\begin{aligned} \tilde{\boldsymbol{x}}_n &\sim C(\tilde{\boldsymbol{x}}_n|\boldsymbol{x}_n) \\ \boldsymbol{h}_n &= f(\boldsymbol{W}\tilde{\boldsymbol{x}}_n + \boldsymbol{b}) - f(\boldsymbol{b}) \\ \boldsymbol{y}_n &= f(\boldsymbol{W}'\boldsymbol{h}_n + \boldsymbol{b}') \\ \phi(\boldsymbol{h}_1,\boldsymbol{h}_2,\boldsymbol{h}_3) &= \log\left(1+\exp(\boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{h}_3 - \boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{h}_2)\right) \end{aligned} \quad (1)$$

$$\boldsymbol{\theta} = \underset{\boldsymbol{W},\boldsymbol{W}',\boldsymbol{b},\boldsymbol{b}'}{\arg\min} \sum_{(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)\in T} \sum_{n=1}^{3} L(\boldsymbol{y}_n,\boldsymbol{x}_n) + \alpha\phi(\boldsymbol{h}_1,\boldsymbol{h}_2,\boldsymbol{h}_3)$$

上記のように目的関数を拡張することにより，分散表現 $\boldsymbol{h}_n$ がそれぞれの入力ベクトルを復元するだけではなく，その内積値がカテゴリ間の近さを表現することが期待される．式 1 から，入力 $\boldsymbol{x}_n$ がゼロベクトル $\boldsymbol{0}$ であれば，それに対応する分散表現 $\boldsymbol{h}_n$ も同様にゼロベクトルになる．つまり，いかなる情報も含まない記事は，その他の記事とは全く似ていないということを表している．関数 $\phi(\cdot,\cdot,\cdot)$ はカテゴリの類似度に基づいた記事の類似度に関する損失関数，α は二つの項のバランスをとるためのハイパーパラメータである．

具体的には，活性化関数 $f(\cdot)$ は要素ごとのシグモイド関数 $\sigma(x)_i = 1/\left(1+\exp(-x_i)\right)$ を用い，損失関数 $L(\cdot,\cdot)$ として要素ごとのクロスエントロピーを，ノイズ関数 $C(\cdot|\cdot)$ としてマスキングノイズを用いる．また，ミニバッチの SGD（Stochastic Gradient Descent）で上記のモデルを学習する．

2.2 記事の重複排除

この節では，前節で記述した分散表現を用いて，ユーザーに提示する記事の配信リストから重複する記事を排除する手続きについて述べる．

Algorithm 1 重複排除の手続き

Input: 候補となる記事の順序リスト: l_c, 閾値: s
Output: 重複排除後の記事の順序リスト: l_d

```
l_d ← 空リスト
for i in l_c do
    if max_{j∈l_d} cos(h_i, h_j) < s then
        l_d に i を追加
    end if
end for
return l_d
```

重複排除の手続きの疑似コードを Algorithm 1 に示す．まず，候補となるベースの記事の順序リスト l_c は他のランキングシステムによって与えられているものとし，そのリストの先頭から一つ一つの記事に対して，ユーザーに表示するかを貪欲的な手続きにより決定する．ある記事を表示するかを決めるために，対象の記事と，既に表示することが決定された全ての記事との類似度の計算を行う．もし，類似度の最大値が定められた閾値 s を超えていれば，その記事をスキップして表示しないこととする．この手続きを繰り返し，ユーザーに提示する記事の順序リスト l_d を構成する．

3 実験

この章では，提案手法の評価実験について述べる．まず，分散表現の学習方法について述べ，その後，複数の編集者が評価したデータによるオフライン評価と，実システムに提案手法を適用したオンライン評価について報告する．

3.1 学習

分散表現の学習には，Yahoo! JAPAN のトップページに 2015 年の 3 月に掲載されたニュース記事を用いた．語彙集合として，ストップワードを除き，頻度上位 1 万語の名詞を採用した．入力ベクトル $\boldsymbol{x} \in X$ は，それぞれの次元が上記の語彙集合内の各単語に対応する，1 万次元の二値ベクトルとして表現した．分散表現 $\boldsymbol{h}$ は 500 次元のベクトルとし，ノイズ分布 $C(\cdot|\cdot)$ のマスキング割合は 0.3 を用いた．

3.2 オフライン評価

提案手法で生成された分散表現の定量評価のために，2015 年 9 月に入稿されたニュース記事の中から，同

図 1: 編集者による評点と各手法による類似度．それぞれ，記事タイトルの単語ベクトルのコサイン類似度（左），記事全体の単語ベクトルのコサイン類似度（中央），提案手法により生成された分散表現のコサイン類似度（右）に対応する．

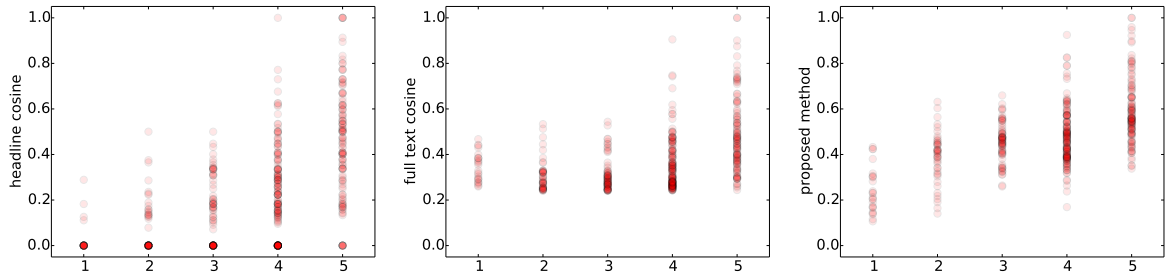

表 1: 編集者による評点とその評価基準

評点	評価基準
5	同じ話題でほぼ同じ内容
4	同じ話題だが書き方やタイミングが違う
3	同じではないが関連する話題
2	同じジャンルの話題
1	関係のない話題

じ日に入稿された記事のペアを作成した．これは，異なる日に入稿された記事は同時に表示されることはないためである．しかし，このようにして作成されたペアの大部分には関連性がないため，以下の評価では，記事全体の単語ベクトルに対するコサイン類似度の上位 0.2%のペアを用いることにした．

これらの記事のペアに対して，複数の編集者に，1 から 5 までの評価付けを依頼した．評点の基準を表 1 に示す．この評価の結果，ラベル付けされた 400 のペアを得た．

図 1 は，編集者の評点と各手法による類似度をプロットしたものである．比較手法として，記事タイトルの単語ベクトル間のコサイン類似度と，記事全体の単語ベクトル間のコサイン類似度を用いた．

単語ベースのコサイン類似度を用いた場合は，評点 4 や 5 に対して高い類似度を示す傾向が見られた一方で，1 から 3 の範囲では類似度との相関があまり見られない結果となった．特に，タイトルに含まれる単語のみを用いた場合には，多くのペアの類似度が 0 となった．一方，提案手法で生成されたベクトルのコサイン類似度は，編集者の評点と連動して大きくなる傾向が見られ，特に 1 から 3 の範囲では顕著であった．

表 2 は，編集者の評点を閾値として二値分類問題を作成し，AUC（Area Under ROC Curve）で評価した結果である．上で見られた傾向と同じく，提案手法は [1 vs. 2-5] や [1-2 vs. 3-5] で特に良い結果を示した．

3.3 オンライン評価

提案手法の有効性を検証するため，Yahoo! JAPAN のスマートフォン用トップページ上でオンライン評価を実施した．各ユーザーは，過去のサイト上での行動に基づき，関連した記事の候補リストが作成される．対象のスマートフォン用トップページはタイムライン形式となっており，ユーザーはページをスクロールすることで最大 200 記事を閲覧することができる．しかしながら，大部分のページ訪問では，ユーザーは上位の 10 から 20 記事のみを目にすることがほとんどであるため，リクエストに応じて閲覧する記事数は異なる．評価指標としては，CTR（= クリックされた記事数 / 表示された記事数），depth（= 表示された記事数 / セッション数），セッション CTR（= クリック数 / セッション数）を用い，記事をスキップする条件を変更してこれらの評価指標を比較した．

実験結果を表 3 に示す．条件 1 と 3 はほぼ同数のスキップ量となり，それよりも条件 4 は多く，条件 2 は少なくなった．条件 2 から 4 は条件 1 と比較してセッション CTR が 2 から 3 パーセント高くなった．この結果から，提案手法によりユーザーは自身の興味にあった記事を見つけることができるようになったといえる．

条件 2 から 4 を見ると，閾値 s の値を小さくするにつれて，depth は減少する一方で，セッション CTR は増加した．これは，重複排除の強さを閾値によって効果的に調整できていること，また，強く重複排除を行うことによって効率的に情報を提供できることを示唆している．

上記の実験結果を受け，スマートフォン用の Yahoo! JAPAN のトップページでは，これまでの単語の共起

表 2: 編集者による評点から作成した，二値分類問題に対する AUC（Area Under ROC Curve）.

類似度	[1 vs. 2-5]	[1-2 vs. 3-5]	[1-3 vs. 4-5]	[1-4 vs. 5]
記事タイトルの単語ベクトル	0.832	0.806	**0.778**	0.811
記事全体の単語ベクトル	0.521	0.646	0.719	**0.813**
提案手法	**0.940**	**0.829**	0.749	0.803

表 3: オンライン環境での実験結果．s は重複排除時の閾値.

ID	スキップ条件	CTR[%]	depth[%]	セッション CTR[%]
1	記事タイトルの単語ベクトル ($s = 0.40$)	+0.00	+0.00	+0.00
2	提案手法 ($s = 0.60$)	−2.78	**+5.25**	+2.32
3	提案手法 ($s = 0.50$)	−0.60	+3.31	+2.69
4	提案手法 ($s = 0.45$)	**+1.36**	+1.61	**+2.99**

ベースの手法の代わりに，提案手法を採用することとなった.

4 関連研究

この章では提案手法と関連する研究について述べる.

2.2 節で述べた重複排除の手続きにおいて，Paragraph Vector [2] などの教師なし学習の手法で生成した記事の分散表現を用いることも考えられる．しかし，2.1 節で述べたのと同様に，そのように生成された記事の分散表現が，重複排除という目的に沿った類似度を表現するかどうかは，不明瞭である．一方で，提案手法はカテゴリ情報を弱い教師シグナルとして用いることで，生成された分散表現で記事間の類似度を明示的に表現することが可能である．もちろん，記事のペアに対して人手で類似度の評価付けを行ったデータを大量に用意できれば，完全な教師あり学習が可能であるが，それはデータ作成のコストが高い．そのため，提案手法では，教師なし学習のモデルである dAE の単語の自己復元の項に，カテゴリの類似度に関する損失項を加えるというアプローチを採用することで，データ作成のコストを下げている．また，このように二つの項からなる目的関数を最適化するように生成された分散表現は，単語の共起関係よりは粗く，カテゴリよりは細かい粒度の類似度を表現することが期待される.

1 章で述べたように，ウェブ上のニュース配信システムは短時間でユーザーに記事リストを提示することが期待されており，数ミリ秒程度で一つの配信リストに対する重複排除の手続きが完了することが求められる．単語レベルで似た記事の判定を行う場合には，b-Bit Minwise Hashing [3] やそれを改良した手法 [4] を用いた前処理を行うことで，類似度を素早く計算することが可能である．一方，提案手法では，低次元ベクトル $\boldsymbol{h}$ 間の単純な内積計算によって，類似度を見積もることができる.

本稿で扱ったニュース記事の重複排除に似た問題として，ウェブ上の UGC（User Generated Content）におけるテキストの使い回しの検出が挙げられる [6, 7].

5 まとめ

本稿では，ウェブ上のニュース配信システムにおいて，冗長な記事を配信リストから省くために，記事の分散表現を用いた重複排除のアプローチを提案した．Yahoo! JAPAN の記事をもとに編集者に評価を依頼し，そのデータを用いて提案手法の有効性を示した．加えて，実システム上でオンライン評価を行い，その結果を紹介した.

参考文献

[1] O. Alonso, D. Fetterly, and M. Manasse. Duplicate news story detection revisited. In *Information Retrieval Technology*. 2013.

[2] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[3] P. Li and C. König. b-bit minwise hashing. In *WWW*, 2010.

[4] M. Mitzenmacher, R. Pagh, and N. Pham. Efficient estimation for high similarities using odd sketches. In *WWW*, 2014.

[5] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[6] Q. Zhang, J. Kang, J. Qian, and X. Huang. Continuous word embeddings for detecting local text reuses at the semantic level. In *SIGIR*, 2014.

[7] Q. Zhang, Y. Wu, Z. Ding, and X. Huang. Learning hash codes for efficient content reuse detection. In *SIGIR*, 2012.