一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NLC2020-1(2020-09)

ランキング学習を用いた関連記事候補の抽出

塩田　　宰† 　嶋田　和孝† 　野上　真司†† 　福山　修平††

† 九州工業大学大学院 情報工学府　〒820–8502 福岡県飯塚市川津 680–4
†† 株式会社西日本新聞社 経営企画局 新メディア戦略室　〒810–8721 福岡県福岡市中央区天神 1–4–1
E-mail: †{t_shiota,shimada}@pluto.ai.kyutech.ac.jp, ††{shinji.nogami,shuhei.fukuyama}@nishinippon-np.jp

あらまし　オンラインで発行されるニュース記事に付与されている関連記事は，読者が有益な情報にアクセスすることを容易にする役割があり，読者の情報収集の効率化に貢献している．そのため，発行記事に対して適切な関連記事を付与することは新聞社にとって重要な課題の 1 つである．しかしながら，関連記事候補の中から適切な関連記事を人手で選択することは多大な労力を要する．そこで本研究は関連記事選択のコスト削減・自動化に向け，発行記事に対して関連記事候補をランキング形式で提示する手法について提案する．複数のランキング評価指標を用いた各モデルの精度比較を行い，最も精度の高かったモデルによる実際の入出力例について報告する．
キーワード　関連記事抽出，ランキング学習，情報検索，業務支援

Related Article Extraction Using Learning to Rank

Tsukasa SHIOTA†, Kazutaka SHIMADA†, Shinji NOGAMI††, and Shuhei FUKUYAMA††

† Kyushu Institute of Technology　680-4 Kawazu, Iizuka-shi, Fukuoka, 820–8502 Japan
†† The Nishinippon Shimbun Co. Ltd.　1–4–1 Tenjin, Fukuoka-shi Chuo-ku, Fukuoka, 810–8721 Japan
E-mail: †{t_shiota,shimada}@pluto.ai.kyutech.ac.jp, ††{shinji.nogami,shuhei.fukuyama}@nishinippon-np.jp

Abstract　Related articles of a news article help readers to access other beneficial information efficiently, so it is an important task for newspaper companies to link appropriate related articles to a news article on news websites. However, it costs to select related articles from the vast news articles that have been published in the past. To solve the issue, we introduce learning to rank method to the related article extraction task to support business operation. We propose some ranking methods and report the accuracy of each model with three evaluation metrics. We also show some input-output examples by the system that scores the highest accuracy in our experiments.
Key words　related article extraction, learning to rank, information retrieval, business support

1. はじめに

タブレットやスマートフォンに代表される電子デバイス，および Web 技術の発展とともに，多くの新聞社が自社サイトやオンラインポータルで Web 版のニュース記事を発行することが一般的になってきている．新聞社は新たな記事を Web 版として発行する際に，過去に発行した類似トピックの記事や，発行記事の背景や前提知識を補完する先行記事を関連記事として付与する．関連記事は，読者が現在読んでいる記事に関連する有益な情報へのアクセスを効率化し，読者の素早い情報収集に貢献している．また新聞社としても，クリックスルー率（CTR）の高くなる記事を関連記事として付与することは，自社の記事をより多くの読者に届ける有効な手段の一つである．そのため，発行する記事に対して読者の満足する適切な関連記事を過去の記事から選出し，発行記事に付与することは，読者および新聞社に関係する重要な課題の 1 つである．関連記事の選択業務は新聞社の記者の知識・経験に基づき人手で行われている．しかしながら，新聞社の保有する記事は膨大で，かつ新聞社は 1 日に大量の記事を発行するため，1 つ 1 つの記事に対して適切な関連記事を選出する作業は多大な労力を要する．

そこで本研究は，記者の関連記事選択のコスト削減，更には関連記事選択の自動化に向け，関連記事候補となる過去の記事を入力記事に応じたランキング形式で出力する機械学習モデルの構築を目指す．本研究では，Yahoo!ニュース Insights から得られる「発行記事」「発行記事に付与されている関連記事」「関連記事のクリックスルー率（CTR）」の三つの情報を組とする発行記事・関連記事のペアデータセットを構築する．そして，情報検索の分野で用いられるランキング学習 [1] に倣った，複数のランキング生成モデルの構築について説明する．その後，構築したモデルのランキング精度を複数の評価指標で比較した結

果を報告する．加えて，高い精度となったモデルの出力した関連記事ランキングについても確認する．

2. 定 式 化

本研究では新しく発行する記事へ付与する関連記事の選択を，ランキング学習 [1] によって関連記事候補群をランキングする問題として定式化する．ランキング学習とは，検索システムなどに用いられる手法で，ある入力クエリに対して検索結果の並び順を学習させる手法である．入力クエリと検索結果のペアは以下のように定式化される．

ある入力クエリ q とそれに対して M_q 件の検索結果 $r \in R_q$ という一対多のデータが N 件存在するとする．更に，各検索結果に対して任意の尺度でユーザの応答に関するスコア $y_{q,r}$ が付与されているとする．このとき，入力クエリ q と検索結果 r から得られる特徴量 $\mathbf{X}_{q,r}$ とユーザの応答に関するスコア $y_{q,r}$ のペアを $\sum_N M_q$ 件作成することができる．

$$Dataset = \{(\mathbf{X}_{q,r}, y_{q,r}) : q = 1, ..., N, r = 1, ..., M_q\} \quad (1)$$

特長量 $\mathbf{X}_{q,r}$ は入力クエリおよび検索結果それぞれから独立に得られる情報や，入力クエリと検索結果の組み合わせによって得られる情報から算出される．学習時は（1）の対をモデルが学習し，推論時はモデルが（1）の $\mathbf{X}_{q,r}$ から $y_{q,r}$ を推測し，その結果に基づいて検索結果を並び替えることでランキングを獲得できる．

ランキング学習の手法は損失関数の定義によって pointwise，pairwise，listwise の 3 つに分類される．pointwise な手法は，モデルの出力値 $f(\mathbf{X}_{q,r})$ が各検索結果に対するスコア $y_{q,r}$ を正しく推定できれば，出力結果をソートすることで正しいランキングが得られるという仮定を基にした手法である．損失は式（2）のように予測値と正解値の誤差を表現する形で定義される．

$$L_{pointwise} = \sum_{q,r} L(f(\mathbf{X}_{q,r}), y_{q,r}) \quad (2)$$

pairwise な手法は，任意の 2 ペアの順序関係を正しく推定できれば，それらの順序関係が正しいランキングになるという仮定を基にした手法である．損失は式（3）のように，2 ペアの順序関係が尤もらしいかを確率値などで表現する形で定義される．

$$L_{pairwise} = \sum_{q} \sum_{y_{q,i} > y_{q,j} : i,j \le M_q} L(f(\mathbf{X}_{q,i}), f(\mathbf{X}_{q,j})) \quad (3)$$

listwise な手法は，出力ランキング自体を MAP や NDCG などのランキング評価指標で直接最適化する手法である．損失は式（4）のように，クエリに対する全ての予測値と正解値の誤差を表現する形で定義される．

$$L_{listwise} = \sum_{q} L(f(\mathbf{X}_{q,1}), y_{q,1}, ..., f(\mathbf{X}_{q,M_q}), y_{q,M_q}) \quad (4)$$

本研究では，「入力クエリ」を「発行予定の記事」，「検索結果」を「関連記事候補」，「ユーザの応答」を「関連記事のクリックスルー率（CTR）」と対応させることで，発行記事に対する関連記事の抽出を目指す．

表 1　発行記事・関連記事ペアデータセット統計値

	発行記事数	記事ペア数	利用する発行記事数	利用する記事ペア数
訓練データ	2,800	16,704	2,800	16,704
検証データ	330	1,650	330	1,650
テストデータ	330	1,650	299	1,495

3. データセットの構築

本節では，ランキング学習を用いたランキング推定モデルの構築に必要な「発行記事」「関連記事」「関連記事のクリックスルー率（CTR）」の三つ組ペアである，発行記事・関連記事ペアデータセットの構築について述べる．加えて，ランキング推定モデルを用いた関連記事抽出実験に必要な関連記事候補データセットの構築について述べる．

3.1 発行記事・関連記事ペアデータセット

発行記事および関連記事は「ヘッドライン」「本文」「配信日時」「カテゴリ」の 4 つの属性から構成されている．データの獲得には Yahoo!ニュース Insights を用いて，2019 年 1 月 1 日から 2019 年 12 月 31 日までに西日本新聞社が掲載したものの中で利用可能な記事計 3,460 件，20,004 ペア分のデータを獲得した．Yahoo!ニュースに掲載する記事に付与する関連記事は一般には 5 件であるが，関連記事を任意のタイミングで差し替えることが可能であるため，1 つの発行記事に対して 5 件以上の関連記事が付与されている場合がある．精度検証・分析のために，それらのデータは全て訓練データに割り振り，残りのデータを訓練データ，検証データ，テストデータの 3 つに 8:1:1 の割合となるようランダムに分割した．表 1 に獲得した発行記事数，および発行記事・関連記事のペア数の統計値を示す．なお，テストデータのうち全ての関連記事の正解スコア（CTR）が 0 のデータは一部の評価指標の値を不当に低下させるノイズとなってしまう．そこで本研究では，分割後のテストデータの中で上記の条件に該当するデータが計 31 件観測されたため，それらを取り除いた 299 件で精度評価を実施する．

3.2 関連記事候補データセット

記者が関連記事を選択する際に利用している西日本新聞社の保有する記事リストが存在する．そのリストに登録されている記事，および発行記事・関連記事ペアデータセット中に存在する全ての記事を統合し，関連記事候補データセットとした．関連記事候補データセットに収録されている合計記事数は 53,023 件である．

4. 抽 出 手 法

本研究では，教師無しのランキング手法をベースラインとし，pointwise な教師あり手法 2 つ，pairwise な教師あり手法 1 つの出力結果を実験によって比較する．それら 4 つの手法について説明する．

（1） cos 類似度によるソート（CosSim）

発行記事のヘッドラインと関連記事候補のヘッドラインから名詞，動詞，形容詞，副詞を抽出し，それらの分散表現の相加平

均をそれぞれのヘッドラインの分散表現とし，cos 類似度を算出する．その後，類似度が降順になるよう関連記事候補を並び替えることでランキングを作成する．これは「内容の似ている記事は関連記事になりやすい」という人の直感を基にした手法である．形態素解析器は MeCab[1]を利用する．分散表現の獲得には，発行記事・関連記事ペアデータセットおよび関連記事候補データセットに含まれるヘッドラインと本文から skip-gram を用いて事前学習した Word2Vec [2]（300 次元）を利用する．

（2） Linear Regression（LR）

pointwise で線形な手法として Linear Regression（LR）を利用する．LR は特長量ベクトルから CTR を直接推定するように学習を行う．実装には scikit-learn [3] の LinearRegression を利用する．モデルの予測した CTR を降順にソートすることで関連記事のランキングを作成する．

（3） Support Vector Regression（SVR）

pointwise で非線形な手法として SVM [4] を利用する．SVM も同様に特長量ベクトルから CTR を直接推定するように学習を行う．実装には scikit-learn [3] の SupportVectorRegression を利用し，カーネルは rbf，C および γ はそれぞれ 1，0.01 とする．LR と同様に，モデルの予測した CTR を降順にソートすることで関連記事のランキングを作成する．

（4） LambdaMART

pairwise な手法として LambdaMART [5] を利用する．LambdaMART は 2010 Yahoo! Learning to Rank Challenge [6] で最高精度を獲得したランキング学習のモデルで，LambdaRank [7] と MART [8] を組み合わせた手法である．LambdaMART では正解ラベルに直接 CTR を利用できないため，学習の際には CTR を四捨五入した値（0 から 20：20% を超えるものについては全て 20 で統一）に変換したものを CTR ラベルとして学習を行う．実装には LightGBM [9] を利用し，評価の metric は NDCG@k（k=1，2，3，4，5），learning_rate は 0.05，bagging_fraction および feature_fraction は共に 0.9 に設定する．

LR，SVR，および LambdaMART に利用する特長量を発行記事，関連記事，そして発行記事と関連記事の組み合わせから 40 次元算出する．以下に特長量のリストを示す．なお，形態素解析器および分散表現獲得のための事前学習モデルは CosSim と同様のものを利用する．

- ヘッドライン・本文の文字列長（4 次元）
- ヘッドライン・本文の内容語数（4 次元）
- 本文の文章数（2 次元）
- 文字列長・内容語数・文書数の差（5 次元）
- ヘッドライン・本文の分散表現間の cos 類似度（4 次元）
- ヘッドライン・本文の内容語間のシンプソン係数（4 次元）
- ヘッドライン・本文の形態素 uni-gram，bi-gram，tri-gram の重複率（12 次元）
- 配信日時の日付差，時間差，日時差（3 次元）
- カテゴリの重複フラグ，重複率（2 次元）

（注1）：https://taku910.github.io/mecab/

5. 実　　験

3. 節で構築したデータセット，および 4. 節で述べた手法を用いて 2 種類の実験を実施した．1 つ目の実験では，発行記事と正解関連記事 5 件を与え，モデルが関連記事を CTR の高い順に並べ替える精度を確認した（正解関連記事ランキング実験）．つまり，正解関連記事ランキング実験を通して，人が選んだ記事の CTR の順序関係を各提案手法が適切に識別できるかを確認する．2 つ目の実験では，発行記事と正解関連記事を含む関連記事候補データセットを与え，モデルが CTR の高い順に並べ，かつ正解関連記事を上位に持ってくることができるかを確認した（関連記事抽出実験）．つまり，関連記事抽出実験を通して，運用環境を想定し，各提案手法が膨大な記事の中から関連記事を抽出する検索エンジンとしての性能を確認する．はじめに，本実験の結果を評価するために利用した評価指標について述べる．その後，正解関連記事ランキング実験，および関連記事抽出実験における各モデルの精度をまとめる．最後に，最も精度の高いモデルに関して，検証データを用いた出力結果についても報告する．

5.1 評 価 指 標

本研究では定量的な評価指標として，NDCG@k，MRR，Recall@k を用いる．それぞれの評価指標の定義を以下に示す．

（1） NDCG@k

NDCG@k（Normalized Discounted Cumulative Gain）はモデルの予測結果上位 k 件から得られる DCG（Discounted Cumlative Gain）を，正解の上位 k 件から得られる理想的な DCG で割って正規化した値である．NDCG@k は 0 から 1 の間の値を取り，1 に近いほどモデルが正解に近いランキングを出力できていると解釈できる評価指標である．式 5 に NDCG@k の定義を示す．ここで，N はテストデータ数，k は上位何件で評価値を計算するかを示すインデックス，$dcg@k$ はモデルの出力結果による DCG 値，$ideal\_dcg@k$ は正解による理想的な DCG 値を表す．

$$NDCG@k = \frac{1}{N}\sum_{i=1}^{N}\frac{dcg_i@k}{ideal\_dcg_i@k} \tag{5}$$

DCG の定義はいくつかあり，本研究では Järvelin ら [10] によって定義されたものを DCG1，Burges ら [11] らによって定義されたものを DCG2 とし，NDCG1 および NDCG2 を算出する．DCG1 および DCG2 の定義をそれぞれ式 6 および式 7 に示す．ここで rel_i はランキング中の i 番目の記事の正解スコア（$= y_{(q,r),i}$）を表す．

$$DCG1@k = rel_1 + \sum_{i=2}^{k}\frac{rel_i}{log_2 i} \tag{6}$$

$$DCG2@k = \sum_{i=1}^{k}\frac{2^{rel_i}-1}{log_2(i+1)} \tag{7}$$

なお，本研究において NDCG 算出に利用する正解スコアは LambdaMART に向けて作成した CTR を四捨五入した整数値としている．

表 2　各手法による正解関連記事ランキングの精度

	NDCG1@k				
	k = 1	*k = 2*	*k = 3*	*k = 4*	*k = 5*
CosSim (baseline)	0.602	0.724	0.782	0.827	0.881
LR	0.484	0.598	0.706	0.765	0.832
SVR	0.638	0.749	0.792	0.843	0.891
LambdaMART	**0.758**	**0.811**	**0.838**	**0.875**	**0.915**
	NDCG2@k				
	k = 1	*k = 2*	*k = 3*	*k = 4*	*k = 5*
CosSim (baseline)	0.525	0.626	0.696	0.747	0.801
LR	0.389	0.479	0.594	0.660	0.733
SVR	0.557	0.653	0.703	0.762	0.815
LambdaMART	**0.707**	**0.761**	**0.794**	**0.836**	**0.872**

（2） MRR

MRR（Mean Reciprocal Rank）とは，モデルの出力したランキングを上位から順番に辿って最初に正解記事が出現した順位の逆数を全データ分足し合わせ，データ数で割った平均値のことである．MRR は 0 から 1 の間の値を取り，1 に近いほどモデルが正解記事を上位に持ってくることができていると解釈できる評価指標である．式 8 に MRR の定義を示す．ここで N はテストデータ数，$rank_i$ は初めて正解記事が出現した順位を表す．

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{8}$$

（3） Recall@k

Recall@k は発行記事に対する正解関連記事が，モデルによって出力されたランキング上位 k 件中に何割含まれているか示す値である．Recall は 0 から 1 の間の値を取り，1 に近いほどモデルが人の選ぶ記事を上位 k 件に選出することができていると解釈できる評価指標である．式 9 に Recall@k の定義を示す．ここで，N は評価データ数，$item_{gs}$ は正解記事の集合，$result_{pred}$ はモデルの出力した記事の集合を表す．

$$Recall@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|item_{gs,i} \cap result_{pred,i}|}{|item_{gs,i}|} \tag{9}$$

5.2 実験結果

発行記事と正解関連記事 5 件を与える正解関連記事ランキング実験，および発行記事と正解関連記事を含む関連記事候補データセットを与える関連記事抽出実験を各モデルで実施した．それぞれの結果を 5.2.1 節および 5.2.2 節にまとめる．

5.2.1 正解関連記事ランキング実験

表 2 に NDCG1@k および NDCG2@k による各モデルの正解関連記事ランキングの精度を示す．結果より，ヘッドラインの意味的類似性のみを考慮して単純に並び替えるよりも，SVM や LambdaMART のような教師あり学習による CTR 推定モデルを利用した手法の方が適切に関連記事をランキングできていることがわかる．続いて，ベースラインを上回った SVR と LambdaMART の NDCG を比較すると，全体を通して LambdaMART の NDCG が高かった．つまり，正解ラベルの単純なランキングを評価する NDCG1 と，正解ラベルの重みを考慮してランキングを評価する NDCG2 の両方で上回っている点を踏まえ，LambdaMART の方が SVR よりも優先度の高い関連記事を適切に並べる能力が高いと考えられる．以上の結果から，提案手法のモデルは CTR の順序関係を適切に学習できていると結論づけられる．

表 3　各手法による関連記事抽出の精度（1,000 件）

		NDCG1@k				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)		0.163	0.180	0.204	0.217	0.234
LR		0.003	0.011	0.029	0.033	0.040
SVR		0.126	0.181	0.250	0.251	0.251
LambdaMART		**0.218**	**0.236**	**0.257**	**0.269**	**0.281**
		NDCG2@k				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)		0.168	0.185	0.213	0.226	0.245
LR		0.003	0.011	0.029	0.034	0.041
SVR		0.128	0.184	0.254	0.255	0.255
LambdaMART		**0.249**	**0.270**	**0.293**	**0.306**	**0.318**
	MRR	*Recall@k*				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)	0.331	**0.123**	0.157	0.229	0.288	0.363
LR	0.038	0.006	0.025	0.079	0.099	0.139
SVR	0.241	0.110	**0.222**	**0.441**	**0.443**	**0.445**
LambdaMART	**0.395**	0.118	0.152	0.211	0.260	0.321

表 4　各手法による関連記事抽出の精度（10,000 件）

		NDCG1@k				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)		0.072	0.081	0.094	0.102	0.110
LR		0.000	0.000	0.000	0.001	0.004
SVR		0.028	0.034	0.047	0.063	0.094
LambdaMART		**0.125**	**0.136**	**0.146**	**0.155**	**0.163**
		NDCG2@k				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)		0.081	0.090	0.103	0.112	0.122
LR		0.000	0.000	0.000	0.001	0.004
SVR		0.036	0.041	0.057	0.074	0.105
LambdaMART		**0.143**	**0.156**	**0.167**	**0.177**	**0.187**
	MRR	*Recall@k*				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)	0.167	0.052	0.071	**0.107**	**0.131**	0.168
LR	0.004	0.000	0.000	0.001	0.005	0.022
SVR	0.069	0.017	0.027	0.061	0.125	**0.270**
LambdaMART	**0.226**	**0.056**	**0.077**	0.096	0.123	0.156

5.2.2 関連記事抽出実験

関連記事候補データセットのデータ数を 1000 件，10000 件，53023 件（全件）としたときの精度を表 3 から 5 に示す．なお，ここでの評価値は「1 つの記事に対して過去の記者が選出した関連記事 5 件が（たとえ，結果的に CTR が低くても）最も最適であり，それ以外の記事を付与した場合は全て CTR が 0% であった」という制約を基にして算出したものである．NDCG@k および MRR の 2 つの評価指標において LambdaMART が最も高

表 5　各手法による関連記事抽出の精度（53,023 件）

		NDCG1@k				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)		0.028	0.032	0.040	0.044	0.051
LR		0.000	0.000	0.000	0.000	0.000
SVR		0.016	0.017	0.021	0.023	0.029
LambdaMART		**0.081**	**0.086**	**0.093**	**0.098**	**0.104**
		NDCG2@k				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)		0.028	0.033	0.042	0.047	0.055
LR		0.000	0.000	0.000	0.000	0.000
SVR		0.020	0.022	0.026	0.029	0.035
LambdaMART		**0.092**	**0.099**	**0.108**	**0.115**	**0.120**
	MRR	*Recall@k*				
		k = 5	*k = 10*	*k = 25*	*k = 50*	*k = 100*
CosSim (baseline)	0.051	0.025	0.034	0.052	0.071	**0.099**
LR	0.001	0.000	0.000	0.000	0.000	0.001
SVR	0.034	0.007	0.009	0.017	0.027	0.050
LambdaMART	**0.140**	**0.035**	**0.043**	**0.057**	**0.072**	0.092

い精度を示しており，LambdaMART が他の手法に比べ人が選んだ記事を上位に抜き出し，関連記事の並びの適切性が高いことがわかった．また，NDCG1 と NDCG2 のスコアを比較すると，NDCG2 の方が全体的に高い値となっていることを確認した．一方で，Recall@k については SVR やベースラインの CosSim の方が評価値が高い場合が多いという結果になった．以上のことから，SVR や CosSim のように発行記事と関連記事のペアのスコアを独立で算出しているモデルが人の選ぶ記事を網羅的に上位に選ぶ性質がある．その一方，LambdaMART は高い CTR を見込めないと判断した記事については下位に配置しながらランキング全体を最適化する傾向にあると推測できる．本実験では運用環境を想定した膨大な候補の中から関連記事の抽出を行い，制約付きでの精度評価を行った．したがって今後は，記者とシステムそれぞれが選んだ記事を基に運用した結果を比較分析などを実施し，各提案手法の更なる精度検証を行っていく必要がある．

5.3　検証データによる出力例

本節では，検証データを入力として受け取り，LambdaMART が実際に出力した結果について正例（表 6）および負例（表 7）を示す．表 6 の入力記事は，福岡知事選で自民県議が自民党推薦の立候補者ではなく，現職の知事を推薦するために会派を退会したことを報じた記事である．これに対して，LambdaMART のモデルは本文の欠損があるにも関わらず，記者の選んだ CTR の高い記事を同様に上位に選出しており，他の正解関連記事についても 100 位以内に選出している．また，記者に選ばれていない記事で LambdaMART のモデルが抽出した記事の中に党内での派閥争いなどを扱った記事が上位にランキングされていることが確認できた．表 7 の入力記事は有明海花火フェスタが終了することを報じた記事である．花火に関連する記事が候補中に存在するにも関わらず，福岡県や佐賀県の火災を報じる記事を選出しており，ヘッドラインに含まれる「福岡県」「柳川市」といった表層情報や，「花火」と「火災」が火に関する情報ということを認識し，誤った予測を行ったと推測できる．本研究で用いたデータセットの多くは本文データを自動獲得できずに欠損している割合がとても高い．そのため，抽出手法で利用しているモデルが本文の情報よりもヘッドラインの情報に重みを置いてランキングを生成している可能性があると考えられる．従って今後は，欠損している本文データの追加などデータセットの品質向上を行った上で再実験することで，これらの事象が解決されるか確認をすることも重要な課題の 1 つである．

6.　おわりに

本研究では，発行予定の記事に付与する関連記事を選択する問題をランキング学習として定式化し，ベースラインを含む 4 つの手法を提案した．5 件の正解関連記事を CTR が高い順に並べ替えを行う正解関連記事ランキング実験，および関連記事候補から適切な関連記事候補を上位に選出する関連記事抽出実験の 2 つを実施し，各モデルの精度を確認した．また，最も精度の高い LambdaMART を用いたランキング生成モデルによる入出力の事例についても報告した．今後は，データセットの品質向上およびモデルの再学習・再実験を実施することで，現状のモデルとしての課題をまとめていきたい．また，本研究で利用した手法を検索エンジンとし，記者が関連記事を選択する業務支援システムの開発，それを用いた運用を行いながら出力の評価・分析に取り組みたい．

文　献

[1] T.-Y. Liu, Learning to Rank for Information Retrieval, Springer, 2011.

[2] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS2013), pp.3111–3119, 2013.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research (JMLR), vol.12, pp.2825–2830, 2011.

[4] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 2013.

[5] C.J. Burges, “From Ranknet to Lambdarank to Lambdamart: An Overview,” 2010.

[6] O. Chapelle and Y. Chang, “Yahoo! Learning to Rank Challenge Overview,” Proceedings of the Learning to Rank Challenge, pp.1–24, 2011.

[7] C.J. Burges, R. Ragno, and Q.V. Le, “Learning to Rank with Nonsmooth Cost Functions,” Proceedings of the 19th International Conferences on Neural Information Processing Systems (NeurIPS2006), pp.193–200, 2006.

[8] J.H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” Annals of Statistics, vol.29, no.5, pp.1189–1232, 2001.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A Highly Efficient Gradient Boosting Decision Tree,” Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS2017), pp.3146–3154, 2017.

[10] K. Järvelin and J. Kekäläinen, “Cumulated Gain-based Evaluation of IR Techniques,” ACM Transactions on Information Systems (TOIS), vol.20, no.4, pp.422–446, 2002.

[11] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to Rank Using Gradient Descent,” Proceedings of the 22nd International Conference on Machine Learning (ICML2005), pp.89–96, 2005.

表 6　検証データにおけるランキング例（正例）

発行記事		
ヘッドライン	福岡知事選で現職支持、自民県議が退会届　「一身上の都合」	
本文	（欠損）	
配信日時	2019 年 2 月 14 日 (木)　9:30	
カテゴリ	社会	
LambdaMART が選出した関連記事上位 10 件		
予測順位	ヘッドライン	
1	現職支援を二階氏容認？　福岡知事選、自民議員対応巡り「発言に注文付ける必要はない」	
2	"面従腹背"予告も　自民分裂の福岡知事選　議員の「結束」早くも不協和音	
3	私怨拭えず看板倒れ　麻生氏「心からおわび」　福岡知事選、自民推薦候補落選	
4	「反麻生」うねり一気　権力争い拒否感あらわ　福岡知事選	
5	自民「ねじれ」組織困惑　知事選、現職路線に揺らぎ　「麻生氏の私怨」反発も	
6	「推薦だめなら副総理やめる」首相に迫った麻生氏の"執念"　自民分裂の福岡知事選	
7	福岡知事選は自民分裂　党本部、異例の新人推薦を決定　現職の要請退ける	
8	「トトロ」に見える？　高さ 120 メートル、岩山が話題に　長崎	
9	「麻生色」払拭したい新人、現職は自民全面の「県民党」　福岡知事選、対照的な戦略　【福岡コンフィデンシャル】	
10	【混戦 参院選福岡】（上）しこり　自民、挙党態勢に不安	
記者の選んだ関連記事		
正解順位	予測順位	ヘッドライン
1	2	"面従腹背"予告も　自民分裂の福岡知事選議員の「結束」早くも不協和音
2	6	「推薦だめなら副総理やめる」首相に迫った麻生氏の"執念"
3	28	「勝負あり」「意味ない」　保守分裂の福岡知事選、一気に拡散した"秘密"の調査
4	94	麻生氏と「大御所たち」の覇権争い　福岡知事選、狭まる"包囲網"
5	17	「与野党対決よくない」現職知事が異例の推薦辞退　福岡知事選、立憲民主が取り下げへ

表 7　検証データにおけるランキング例（負例）

発行記事		
ヘッドライン	夏の風物詩「有明海花火」が終了　観覧者 10 万人超、渋滞が深刻化　福岡県柳川市	
本文	柳川市橋本町の干拓地で８月に開かれる「有明海花火フェスタ」が今年を最後に終了することが分かった。市観光課によると近年、観覧者数が１０万人を超え、...（中略）... 今夏の第２０回大会では、世界最長クラスとなる全長２キロの打ち上げ方式の仕掛け花火「スカイナイアガラ」が人気を呼んだ。	
配信日時	2018 年 12 月 12 日 (水)　9:22	
カテゴリ	社会	
LambdaMART が選出した関連記事上位 10 件		
予測順位	ヘッドライン	
1	福岡県飯塚市で建物火災か	
2	福岡県みやま市の火災は鎮火	
3	佐賀県鹿島市で林野火災か	
4	福岡県久留米市で建物火災か	
5	マイクロバス追突、13 人けが	
6	福岡県の大曲副知事が再任へ	
7	吉本興業が福岡に新劇場	
8	福岡県大川市で建物火災発生か	
9	トヨタ 18 万台リコール	
10	福岡ボート・オールスタークオカを 10 人に	
記者の選んだ関連記事		
正解順位	予測順位	ヘッドライン
1	1350	「1 杯 1 万円」福岡の喫茶店に特別なコーヒー
1	7899	「今なら絶対に付けられない名前」　親しまれて 35 年、福岡市の焼き鳥店が閉店
1	8905	「注文は必ずカツ丼」佐賀県唐津市に若い女性が押し寄せるワケ
1	13075	壁一面に「うまい棒」　月額 540 円でカレー食べ放題の店
1	43735	84 年の歴史、時が止まったビル　シャッター開けてみたら