ニュース制作に役立つtweetの自動抽出手法

宮﨑 太郎[1)]　鳥海 心[2)]　武井 友香[1)]　山田 一郎[1)]　後藤 淳[1)]
1) NHK 放送技術研究所　　2) 東京都市大学
1) {miyazaki.t-jw, takei.y-ek, yamada.i-hy, goto.j-fw}@nhk.or.jp
2) g1683108@tcu.ac.jp

1　はじめに

近年，ソーシャルメディア上には数多くの情報が投稿されている．スマートフォンなどから手軽に投稿できることから，事件や事故などの現場に偶然居合わせた人からの目撃情報が写真付きで投稿されることも多い．これらを活用することで素早い情報収集が可能となることから，現在では重要な取材源のひとつとなっている．NHK でも，ソーシャルメディアからニュース取材に役立つ情報を迅速に抽出するためのプロジェクトを立ち上げており，すでに実際に報道番組などでその成果が利用されている [1]．ここでは，タイムライン上の情報を，複数の検索クエリでフィルタリングした上で，人手で情報を選別しているが，この作業には大きな労力が必要である．

また，抽出したい情報が多岐にわたるため，フィルタリングに用いる検索クエリで全てを網羅することが難しい．例えば電車の事故に関する投稿でも，「○○線が事故で遅れてる」のように路線名が書かれる場合もあれば「○○駅での事故で電車が遅れてる」のように駅名が書かれる場合もある．これらの情報を抽出するには全ての路線名や駅名を網羅したクエリを作成する必要があるが，それは現実的ではない．

我々は，この労力を軽減し，また，キーワード検索だけでは抽出できない情報も取り出すための手法の研究に取り組んでいる．本稿では，tweet がニュース制作に役立つ情報かどうかを自動で判定する手法について述べる．ソーシャルメディアでは，気軽に投稿ができるという性質から，口語体で書かれることが多く，略語やスラングが頻出するなど，従来の自然言語処理で用いられてきた形態素解析などの手法がそのまま適用できない場合も多い．そこで，提案手法では単語分割せずに，文字ごとに RNN (Recurrent Neural Network) に入力し，ニュース制作に役立つか判定する．さらに，アテンションメカニズムとマルチタスク学習を導入することで判定性能が向上したので報告する．

2　ニュース制作に役立つtweetの自動抽出手法

2.1　RNN を用いた tweet の判定

はじめに，tweet を RNN に入力し，tweet 全体の意味を表すベクトル表現を得る．一般に，ソーシャルメディアは気軽に投稿されることから，口語調でかつ，略語やスラング，絵文字などが多く出現する．そのため，一般の形態素解析器では，うまく単語単位に分割できない場合も多い．また，投稿の内容が多岐にわたるため，出現する語彙数が膨大なものになる．そのため，従来，単語単位でテキストデータを扱う手法が一般的であったが，ソーシャルメディアを対象とする場合には文字単位で扱うことで良い性能を得られるという報告がされている [2, 3]．このため，本稿でも tweet を文字ごとに RNN に入力することとする．

図 1 に，RNN を用いた tweet の判定手法の概要を示す．ベクトル表現を得るための RNN は順方向と逆方向の 2 つを用い，該当の tweet をそれぞれの RNN に順方向，逆方向に入力する．全ての文字の入力後，終端記号を入力した時点でのそれぞれの RNN の内部状態を結合したものを，以下では tweet の意味を表すベクトル表現として用いる．得られたベクトル表現は，2 層の Feed-Forward Neural Network によりニュース取材に役立つかどうか判定する．

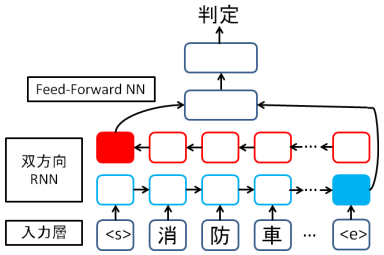
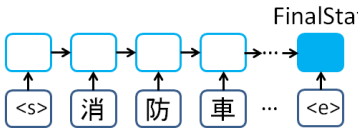
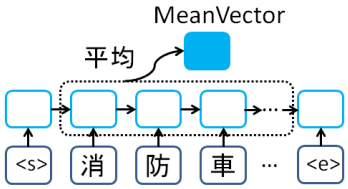


図 1: RNN による tweet の判定



(a) FinalState アテンション



(b) MeanVector アテンション

図 2: アテンション計算に用いるベクトル

2.2 アテンションメカニズムの導入

アテンションメカニズムは，近年，機械翻訳やキャプション生成にも多く用いられる手法で，入力データのうちの重要な部分への重み付けに用いることができる [4, 5]．本稿では，このアテンションメカニズムを利用し，入力 tweet 中の重要部分に重み付けすることで，判定性能の向上を目指す．アテンションの計算方法に，一般に用いられるもの（FinalState アテンション）と，今回のタスクに合わせて改良した提案手法 (MeanVector アテンション）の 2 つを用意し，性能を比較する．

2.2.1 FinalState アテンション

FinalState アテンションは，tweet を文字ごとに全て RNN に入力し終わった時点での RNN の状態を用いてアテンションを計算する (図 2-(a))．以下では順方向 RNN の場合で説明するが，実際には順方向，逆方向それぞれに同様にしてアテンションを求める．

Tweet を文字ごとに RNN に入力した際の，t 番目の文字まで入力した RNN の状態を $\bar{h}_t$，終端記号を含めた全ての文字を入力した後の RNN の状態を h_f とすると，t 番目の文字に与えるスコア $score_t$ は以下のように求められる．

$$score_t = h_f^T \bar{h}_t \tag{1}$$

このスコアを用い，各文字の重み W_t を求める．

$$W_t = \frac{\exp(score_t)}{\sum_{t'} \exp(score_{t'})} \tag{2}$$

なお，t' は対象の tweet に出現するすべての文字の集合を表す．この重みと，その文字に対応する RNN の状態を足し合わせることで，FinalState アテンションのベクトル a_f が得られる．

$$a_f = \sum_{t'} W_t h_t \tag{3}$$

こうして得られた FinalState アテンションのベクトル a_f と，RNN を用いた tweet のベクトル表現 h_f との和を特徴量として用い，tweet がニュース取材に役立つかどうかを判定する．判定には，図 1 に示した RNN を用いた tweet の判定手法と同様，2 層の Feed-Forward Neural Network を用いる．

FinalState アテンションは他の多くのアテンションを用いた手法と同様の考え方に基づいたものである．tweet 全体の文意を表す h_f と，各入力時点での類似度を用いることで，tweet 全体の文意に影響を及ぼしている文字に高い重みが与えられる．

2.2.2 MeanVector アテンション

MeanVector アテンションは，tweet を文字ごとに入力した際に得られる，各文字ごとの RNN の状態の平均を用いてアテンションを計算する (図 2-(b))．以下では，2.2.1 節同様，順方向 RNN の場合の例で説明する．

t 番目の文字に与えるスコア $score_t$ は以下のように求められる．

$$score_t = h_m^T \bar{h}_t \tag{4}$$

$$h_m = \frac{\sum_{t'} \bar{h}_t}{|t'|} \tag{5}$$

以下，FinalState アテンションの計算手順と同様に，MeanVector アテンションのベクトル a_m を求める．

$$W_t = \frac{\exp(score_t)}{\sum_{t'} \exp(score_{t'})} \tag{6}$$

$$a_m = \sum_{t'} W_t h_t \tag{7}$$

得られた MeanVector アテンションのベクトル a_m と，RNN を用いた tweet のベクトル表現 h_f との和を特徴量として用い，2 層の Feed-Forward Neural Network を用いて tweet がニュース取材に役立つかどうかを判定する．

MeanVector アテンションも，tweet 全体の文意に影響を及ぼす文字に高い重みを与えることが目的であるが，FinalState アテンションと比較して文字の出現位置による影響を受けづらい．

表 1: 実験に用いたデータセット

データ種別		分量
学習データ	正例	19,962
	負例	1,524,155
評価データ	正例	426
	負例	1,574

2.3 マルチタスク学習の導入

ニューラルネットワークを用いた手法では，マルチタスク学習と呼ばれる，一つのモデルを複数のタスクで学習することでより汎用的なモデルを作成する手法により性能が向上することが報告されている [6, 7]．今回の提案手法でも，マルチタスク学習を導入することで，性能の向上を試みた．

本稿では，本来のタスクである tweet がニュース取材に役立つかどうかの判定に加えて，入力文字列の次の文字を予測するタスク，すなわち文字単位のニューラル言語モデルを学習することでマルチタスク学習した．これにより，新たなデータを準備すうることなくマルチタスク学習ができる．

入力層と双方向 RNN については２つのタスクで共有し，出力層をタスクごとに使い分けることとした．モデルの学習時には，まず別のタスクを用いて学習する．この結果を初期モデルとして用い，本来のタスクで学習する．

3 評価実験

3.1 実験条件

学習データとして，正例には報道現場で実際に番組制作に使用した tweet を，負例にはランダム抽出した tweet を用いた．評価データとしては，報道現場での使用を想定し，実際に報道現場で用いている検索クエリによりフィルタリングした tweet からランダムに抽出し，1 名の評価者によりそれぞれの tweet について番組制作に役立つまたは役立たないのラベルを付与したものを用いた．データ量を表 1 に示す．

手法の実装には Chainer[8] を用いた．RNN の実装は LSTM(Long Short-Term Memory) を利用し，活性化関数には ELU (Exponential Linear Units) を，学習に用いる誤差計算には softmax cross entropy 法を用いた．中間層のノード数は双方向 RNN が 200，Feed-Forward NN の 2 層は入力層に近い方から順に 200, 100 とした．学習の epoch 数は 10 とし，マルチタスク学習の epoch 数は 3 とした．

表 2: 評価実験結果

手法	Recall	Precision	F 値
RNN	0.615	0.567	0.590
＋ MTL	0.552	0.622	0.585
＋ Final	0.580	0.572	0.576
＋ Final ＋ MTL	0.674	0.573	0.619
＋ Mean	0.650	0.535	0.587
＋ Mean ＋ MTL (提案手法)	**0.650**	**0.606**	**0.627**

3.2 実験結果

表 2 に，評価実験の結果を示す．表中の RNN は 2.1 節で述べた RNN を用いた tweet 判定手法を，Final と Mean はそれぞれ 2.2.1 節で述べた FinalState アテンションと 2.2.2 節で述べた MeanVector アテンションを，MTL は 2.3 節で述べたマルチタスク学習を表す．

RNN を用いた tweet 判定手法に，MeanVector アテンションとマルチタスク学習を組み合わせた提案手法が全ての中で最良の結果となり，RNN を用いた tweet 判定手法を単独で用いた場合と比較して F 値が 3.7 ポイント向上した．また，MeanVector アテンションは，FinalState アテンションと比較して性能が向上した．２種類のアテンションメカニズムは，マルチタスク学習を用いない場合に，RNN を単独で用いる場合と比較して性能の向上は見られなかったが，マルチタスク学習を用いることで性能が向上した．

3.3 考察

2 種類のアテンションの計算方法では，MeanVector を用いた手法のほうが性能が上回った．FinalState アテンションは後方で入力された文字の影響を強く受けやすい性質がある．図 3 は，実際に計算したアテンションの大きさ W_t を文字ごとに表したものである．FinalState アテンションでは，本文最後の「笑」に大きな値が与えられているが，これは本文の内容とは関係の薄い文字である．それに対し，MeanVector アテンションでは，「警察」や「集」などに高い値がつけられている．MeanVector アテンションではこのように文字列中の位置による影響を軽減したアテンションが与えられるため，今回のタスクにおいては効果が得られたものと考えられる．

RNN を単独で用いた場合と比較し，マルチタスク学習をせずにアテンションメカニズムのみを導入しただけでは性能が向上しなかった．アテンションの計算

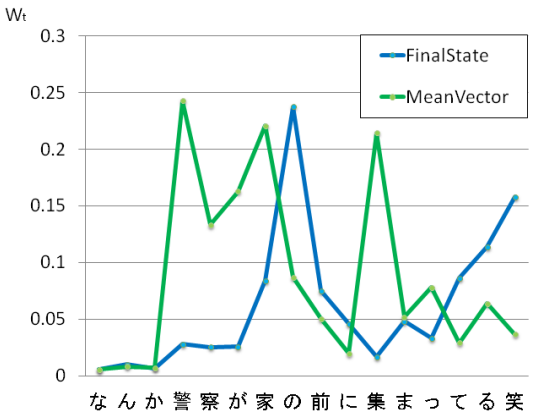


図 3: アテンション計算手法の比較

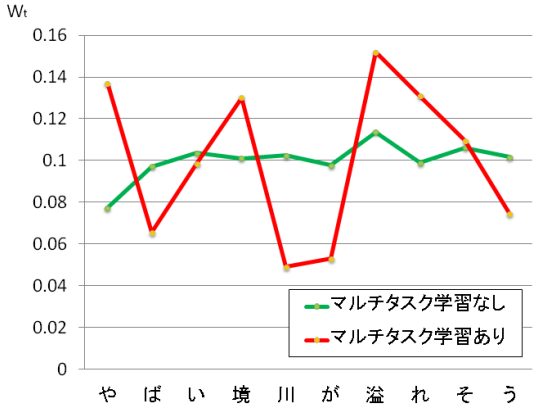


図 4: マルチタスク学習の効果

には，各文字入力ごとの RNN の状態を用いる．それに対し，今回用いた RNN を用いた判定手法では，全文字を入力したあとの最終的な状態のみを用いる手法であるため，学習時に文字ごとの RNN の状態がうまく学習できなかったものと考えられる．マルチタスク学習により，文字入力ごとの RNN の状態を詳細に学習することが可能となり，その結果アテンションメカニズムが機能したものと考えられる．図 4 に，マルチタスク学習の有無によるアテンションの大きさ W_t の比較を示す．マルチタスク学習を行わない場合には値に差が現れなかったが，マルチタスク学習を行ったことで，「溢れ」などの重要な部分に高い値を付与することができた．これにより，ニュースに役立つ tweet かどうかの判定の性能が向上したものと考えられる．

4 おわりに

本稿ではニュース制作に役立つ tweet の自動抽出手法について述べた．tweet を文字ごとに RNN に入力し，その内容をベクトル化してその内容がニュースに役立つかどうかを判定した．さらに，アテンションメカニズム，マルチタスク学習を導入することで，それらを用いない場合と比較して判定性能が F 値で 3.7 ポイント向上した．今回のタスクにおいては，アテンションメカニズムはマルチタスク学習と組み合わせて利用した場合にのみ効果が現れた．

今後，さらなる精度向上を目指して，タスクに特化した特徴量の導入や，その tweet が火事や事故などのどのような種類のニュースになるのかを自動で識別する手法の導入を検討している．

参考文献

[1] 足立 義則, “震災ビッグデータからソーシャルリスニングへ,” 放送メディア研究 No.11, pp. 290–293 (2014).

[2] 萩行 正嗣, “選択式天気情報を用いたソーシャルメディアからの有用投稿抽出，” NLP2016, pp. 397–400 (2016).

[3] Bhuwan Dhingra et al., “Tweet2Vec: Character-Based Distributed Representations for Social Media.” in Proceedings of ACL2016, pp. 269–274 (2016).

[4] Dzmitry Bahdanau et al., “Neural Machine Translation by Jointly Learning to Align and Translaate,” ArXiv: 1409.0473 (2014).

[5] Kelvin Xu et al., “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” ArXiv: 1502.03044 (2015).

[6] Munh-Thang Luong et al., “Multi-task Sequence to Sequence Learning,” ArXiv: 1511.06114, (2015).

[7] Anders Søgaard and Yoav Goldberg, “Deep multi-task learning with low level tasks supervised at lower layers,”” in Proceedings of ACL2016, pp. 231–235 (2016).

[8] Seiya Tokui et al., “Chainer: a Next-Generation Open Source Framework for Deep Learning,” in Proceedings of NIPS 2015 workshop (2015)