オンラインニュースとツイートのリアルタイムマッチング手法

大西　誠† 山口　祐人†† 北川　博之††,†††

† 筑波大学システム情報工学研究科　〒 305–8573 茨城県つくば市天王台 1-1-1
†† 筑波大学計算科学研究センター　〒 305–8573 茨城県つくば市天王台 1-1-1
††† 筑波大学システム情報系情報工学域　〒 305–8573 茨城県つくば市天王台 1-1-1
E-mail: †sei0024@kde.cs.tsukuba.ac.jp, ††yuto@kde.cs.tsukuba.ac.jp, †††kitagawa@cs.tsukuba.ac.jp

あらまし Twitter で投稿されるツイートには，テキスト情報やユーザ情報が含まれている．そのため，ニュースと関連のあるツイートを集めることができれば，ニュースの評判分析やニュースに興味を持つユーザの分析に利用することができる．そこで本研究では，ニュースとそのニュースに関連するツイートをリアルタイムにマッチングする手法を提案する．ニュース毎に内容類似度の閾値を定め，閾値以上の内容類似度を持つツイートを結びつける．先行研究では，ツイートが与えられた時に，そのツイートを関連するニュースに高速に結びつける手法を提案した．具体的には，ツイートに含まれている単語から内容類似度の上限値を計算し，その上限値以上の閾値をもつニュースとの内容類似度の計算を省略した．本稿では，ニュースに含まれている単語を考慮して内容類似度の上限値を計算することで，より多くのニュースとの内容類似度の計算を省略する．

キーワード Twitter，マイクロブログ，オンラインニュース

1. 序　　論

多種多様な情報をリアルタイムに取得できる情報源として，Twitter が注目を集めている．Twitter では，様々な内容のメッセージが投稿されており，それらのメッセージはリアルタイム性の高い情報であることが知られている．Twitter で投稿されるメッセージは「ツイート」と呼ばれ，テキスト情報の他にもユーザ情報や位置情報などを含んでいる．そのため，ツイートをリアルタイムに取得することで，テキスト情報・ユーザ情報・位置情報など，複数の情報をリアルタイムに得ることができる．

Twitter は，ニュースメディアと密接な関係があることが報告されている．例えば，Kwak ら [1] は，Twitter における流行のトピックの分析を行い，85%以上のトピックがニュースのトピックであると報告している．また，Petrovic ら [2] は，ニュースメディアから提供されたニュースのほとんどが，Twitter 上に存在していると報告している．そのため，Twitter 上にはニュースと関連があるツイートが多く存在していると考えられる．

ニュースと関連があるツイートは，ニュースに対する様々な分析に利用することができる．例えば，ニュースと関連があるツイートのテキスト情報は，ニュースの評判分析に利用することができる．ニュースの評判分析を行うことにより，ニュースの分類やニュースのランク付けを行うことができる．また，ニュースと関連があるツイートのユーザ情報は，ニュースに興味をもっているユーザの分析に利用することができる．ニュースに興味をもっているユーザの分析をすることで，ユーザに対してニュースの推薦を行うことができる．さらに，ニュースと関連があるツイートの位置情報は，ニュースに反応を示している地域の分析に利用することができる．ニュースに反応を示している地域の分析することにより，ニュースに地理的な特徴を付与することができる（地域的なニュース，世界的なニュースなど）．

上記のようなニュースの分析は，ニュースが発生してから早期に行う必要がある．例えば，地震・台風・スポーツ実況などのニュースは，リアルタイム性が重要であり，評判の分析や地域の分析を早く行うことが重要である．ニュースを早期に分析するためには，ニュースと関連があるツイートをリアルタイムに結びつけることが重要である．

本研究では，ニュースとツイートをリアルタイムにマッチングする手法を提案する．連続的に投稿されるニュースとツイートを入力として受け取り，ニュースとそのニュースに関連するツイートを出力する．この時のニュースとツイート間の関連性は，内容類似度を用いて計算を行う．

ニュースとツイートのリアルタイムなマッチングを行うには，以下の 3 つの課題に取り組む必要がある：

（1） **人気度の違い：**ニュースと関連のあるツイートの数は，ニュース毎に異なる．一般的に，人気なニュースに関連するツイートは多く，不人気なニュースに関連するツイートの数は少ない．そのため，ニュース毎に適切な数のツイートとマッチングを行う必要がある．

（2） **大量のツイート：**膨大な量のツイートがリアルタイムに投稿されている．Twitter では，1 日に 1 億件以上のツイートが投稿されている [3]．そのため，ニュースとツイートのマッチングにおいて，効率的にツイートを処理する必要がある．

（3） **大量のニュース：**複数のオンラインニュースメディアによって，多くのニュースがリアルタイムに投稿されている．そのため，ニュースとツイートのマッチングにおいて，効率的にニュースを処理する必要がある．

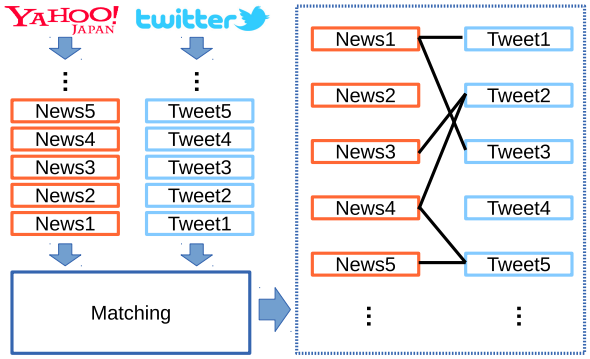

図 1: **入力と出力：**連続的に投稿されるニュースとツイートを入力として受け取り，ニュースとそのニュースに関連するツイートを出力する．

我々は先行研究 [4] において，上記の課題を解決するために以下の 3 つのアプローチを試みた：

（1） **閾値の推定：**ニュース毎に適切な数のツイートとマッチングを行うために，ニュース毎に内容類似度の閾値を定めた．閾値よりも大きな内容類似度をもつツイートとのみ結びつけを行うことで，ニュースと関連のある可能性が高いツイートとのみマッチングを行った．

（2） **効率的なマッチング：**ツイートに対して関連のあるニュースを効率的に結びつけるためのアルゴリズムを提案した．具体的には，ニュースの転置インデックスを利用することにより，マッチングの対象となるニュースの数を削減した．これにより，ニュースとツイートのマッチングにおいて，大量のツイートに対応できるようにした．

（3） **効率的な更新処理：**更新コストの小さな転置インデックスを利用することによって，ニュースの更新にかかる時間を削減した．これにより，ニュースとツイートのマッチングにおいて，大量のニュースに対応できるようにした．

本研究では，先行研究における (2) のアプローチを発展させ，より大量のツイートに対応できるマッチングアルゴリズムを提案する．具体的には，より厳密な条件を用いることで，より多くのニュースをマッチングの対象から取り除く．

本研究の貢献は以下のようになる．

- **効率的なマッチングアルゴリズム：**ニュースとツイートのリアルタイムなマッチングにおいて，大量のツイートに対応できるマッチングアルゴリズムを提案する．

本研究では，提案手法の有効性を示すために実データを用いて評価実験を行う．先行研究でのマッチングアルゴリズムをベースラインとし，マッチングの処理時間の比較を行う．実験により，提案手法はベースラインよりも処理時間を最大で 54.8%削減できることを示す．

2. 関 連 研 究

本節では，本研究と関連のある研究について議論する．

2.1 Twitter を用いた分析

Twitter に含まれる情報を利用して様々な分析を行う研究が多く存在する．Twitter は感情分析や意見抽出など，様々な種類の分析に利用されている．[5] [6] [7] [8]．また，ツイートを用いて特定のトピックの要約を行う研究も数多く行われている．[9] [10] [11] [12]．そのため，ニュースと関連のあるツイートを集めることができれば，感情分析や意見抽出，ニュースの要約などに利用することができると考えられる．

2.2 ニュースと Twitter の関係

Twitter とニュースは密接な関係があることが報告されている．Kwak ら [1] は，Twitter の性質を調べるための分析の 1 つとして，Twitter 上における流行のトピックについて分析を行った．その結果，トピックの 85%以上がヘッドラインニュースや継続的なニュースのトピックであることがわかった．また，Petrovic ら [2] は，Twitter 上のニュースはニュースメディアから提供されるニュースをどの程度カバーしているのかを調査した．その結果，Twitter 上のニュースはニュースメディアから提供されたニュースのほとんどをカバーしていることがわかった．また，Osborne と Dredze [13] は，Facebook，Google Plus，Twitter において，ニュースがどのように報告・伝搬されているのかを分析した．その結果，Twitter は Facebook，Google Plus よりもニュースの報告が早いことがわかった．

2.3 ニュースとツイートの結びつけ

ニュースとツイートのマッチングを行う研究が多く存在する．Shi ら [14] は，ハッシュタグを用いてニュースとツイートを結びつける仕組みを提案した．ニュースに対して機械学習を用いて適切なハッシュタグを見つける．しかし，Shi らの手法はハッシュタグを利用した結びつけであり，内容類似度によって結びつけを行う本研究とはアプローチが異なる．Guo ら [15] は，ニュースとツイートに含まれている単語からグラフを作成し，行列分解を用いてニュースとツイートを結びつけた．グラフを作成する際に，ハッシュタグ・固有名詞・時間情報を考慮したグラフを作成することで高精度を実現した．しかし，Guo らの手法は静的なデータを対象としており，リアルタイムな処理を考慮していない．Stajner ら [16] は，ニュースと関連のあるツイートの中から，「面白い」ツイートを探しだすための方法を提案した．ツイートに含まれている特徴（フォロワー数やリツイート数など）を用いて目的関数を定義し，目的関数を最大にするようなツイートの集合を探した．しかし，Stajner らの手法はニュースと関連のあるツイートの中から「面白い」ツイートを探すための方法であり，ニュースと関連のあるツイートを探す本研究とは大きく異なる．Phuvipadawat ら [17] は，Twitter 上のニュースを収集・グルーピング・ランク付けする手法を提案した．ツイート同士の内容類似度に基づきグルーピングを行い，その後グループ間のランク付けを行う．しかし，Phuvipadawat らの手法はツイートをグルーピングすることによりニュースを発見する手法であり，本研究とは異なる．

3. 問 題 定 義

本節では，用語の説明とニュースとツイートのマッチング問

題の定式化を行う．本研究ではツイートを m，ニュースを s とし，単語を w とする．ツイート m とニュース s は単語の集合として定義する．また，ツイートとニュースの集合をそれぞれ M と S で表す．ツイート m 内における単語 w の出現回数を $n(m,w)$ とする．同様に，ニュース s 内における単語 w の出現回数を $n(s,w)$ とする．これらの記号の定義を以下の表 1 に示す．

表 1: 記号と定義

記号	定義
w	単語
m	ツイート内の単語集合
M	ツイートの集合
$n(m,w)$	ツイート m における単語 w の数
s	ニュース内の単語集合
S	ニュースの集合
$n(s,w)$	ニュース s における単語 w の数

上記の用語を用いてニュースとツイートのマッチング問題を以下のように定義する：

問題定義．（ニュースとツイートのマッチング問題）

入力: 連続的に投稿されるニュース s とツイート m

出力: ツイート m が入力として与えられた時，それまでに入力されたニュース集合 S の中から，ツイート m と関連があるニュース $S' \subset S$

上記の「関連がある」とは，ツイートがニュースの話題に対して言及をしていることを指している．

4. 先 行 研 究

本節では，我々の先行研究 [4] について説明を行う．先行研究では，ニュース毎に適切な数のツイートとマッチングを行うために，内容類似度の閾値を用いたマッチング手法を提案した．具体的には，各ニュースに内容類似度の閾値を定め，その閾値よりも大きな内容類似度をとるツイートとのみ結びつけを行う．

先行研究では，以下のような内容類似度を用いている．

$$sim(s,m) = \sum_{w \in s \cap m} idf^2(w) \cdot \sqrt{\frac{n(s,w)}{|s|}} \qquad (1)$$

ただし，idf はニュースにおける逆文書頻度を表す（詳細は先行研究 [4] を参照）．この内容類似度は，実験によって妥当性が示された [4]．

図 2 に全体の処理の流れを示す．先行研究では，ニュースの転置インデックスを用いることで，ニュースとツイートの効率的なマッチングを行った．先行研究における処理は以下の 3 つに分かれる： (a) **閾値の推定：**ニュースが与えられたときに，ニュースに対して内容類似度の閾値を定める．(b) **マッチング処理：**ツイートが与えられた時に，転置インデックスを用いて，そのツイートを関連するニュースに結びつける．(c) **更新処理：**新しいニュースが到着したときに，転置インデックスの更新を行う．

以降では，まず，4.1 節で閾値の推定方法について説明し，4.2 節でマッチング処理について説明する．その後，4.3 節で更新処理について説明する．

4.1 閾値の推定

ニュースに関連するツイートとのみ結びつけを行うために，ニュースに対して内容類似度の適切な閾値を定める．適切な閾値とは，ニュースに関連するツイートとニュースに関連しないツイートを正しく分類できるような内容類似度の値である．具体的には，ニュースより過去に投稿されたツイート（以降，事前ツイートと呼ぶ）を用いてニュースに閾値を定める．事前ツイートの多くは，ニュースと関連のないツイートであると考えられる．なぜなら，ニュースはリアルタイム性が高い情報であり，ニュースに関するツイートをニュースよりも早く投稿することは難しいためである．そのため，ニュースと事前ツイートの内容類似度を上回るような値を閾値とすることで，ニュースに対して適切な閾値を定める．

4.2 マッチング処理

ツイートが与えられた時に，そのツイートと関連のあるニュースを結びつける．具体的には，ニュースとツイート間の内容類似度を計算し，ニュースの閾値を上回っているかの判定を行う．ニュースとツイート間の内容類似度が，ニュースの閾値を上回っていた場合，そのニュースとツイートを結びつける．

しかし，全てのニュースに対して上記のマッチング処理を行うのはコストが大きい．そのため，内容類似度の計算を行うニュースの数を削減することで処理の効率化を行う．具体的には，ツイートに含まれている全ての単語を利用して内容類似度の上限値を計算し，その上限値よりも大きな閾値をもつニュースとの内容類似度の計算を省略する．

内容類似度の上限値は，転置インデックスを用いることで計算することができる．転置インデックスの各要素に内容類似度の一部（ニュースの TF・IDF 値など）を保持させておくことで，各単語の上限値を計算することができる．ツイートに含まれている全単語の上限値を合計することで，そのツイートがとり得る内容類似度の上限値を計算することができる．この上限値よりも大きな閾値を持つニュースは，ツイートとの内容類似度が閾値を超えることがないため，実際の内容類似度の計算を省略することができる．

4.3 更 新 処 理

ニュースが与えられたときに，転置インデックスの更新を行う．更新処理は，以下の 2 つの処理に分かれる： (a) **追加処理：**ニュースを転置インデックスに追加する．(b) **削除処理：**ニュースを転置インデックスから削除する．削除処理は，古いニュースをマッチングの対象から除外するために行う．先行研究では，効率的なマッチングのためにニュースの閾値によってソートされた転置インデックスを利用している．そのため，追加処理はソートされたリストに要素を追加する処理であり，削除処理はソートされたリストから要素を削除する処理になる．

4.4 先行研究の問題点

先行研究では，マッチング処理において，ニュースに含まれ

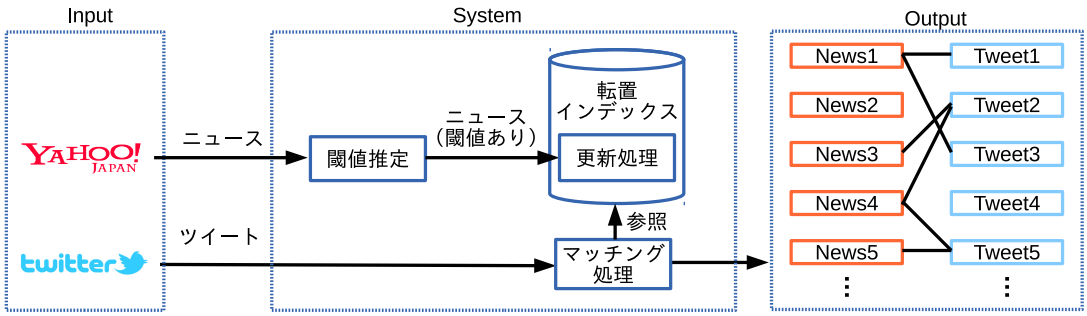

図 2: **既存手法の全体図：**先行研究における全体の処理の流れを示す．ニュースが与えられたときに，ニュースに対して内容類似度の適切な閾値を定め，転置インデックスを更新する．また，ツイートが与えられた時に，そのツイートを関連するニュースに結びつける．このマッチング処理は，転置インデックスを参照することで効率的に行う．

ている単語を考慮せず，ツイートに含まれる全単語を用いて内容類似度の上限値を計算していた．そのため，ツイートの単語数が増えると，内容類似度の計算を省略できるニュースの数が少なくなってしまうという問題がある．次節では，この問題を解決する手法を提案する．

5. 提 案 手 法

本節では，先行研究におけるマッチング処理の高速化手法について説明を行う．マッチング処理は，ツイートが与えられた時に，そのツイートを関連するニュースに結びつける処理である．具体的には，ニュースとツイート間の内容類似度の計算を行い，ニュースの閾値を上回っていた場合，そのニュースとツイートを結びつける．

先行研究では，ニュースに含まれている単語を考慮せず，ツイートに含まれている単語のみを用いて上限値の計算を行った．そのため，ツイートに含まれる単語数が多くなると，省略できるニュースの数が少なくなってしまうという問題がある．そこで提案手法では，ニュースに含まれている単語も考慮して上限値の計算を行うことで，より多くのニュースを計算の対象から除外できるようにする．

以降では，まず，転置インデックスを用いた内容類似度の計算方法について 5.1 節で説明し，その後，5.2 節で効率的なマッチング手法について説明を行う．

5.1 内容類似度の計算方法

ツイートが与えられた時に，転置インデックスを用いて各ニュースとの内容類似度の計算を行う．本研究では，ニュース ID と部分スコアを要素として持つ転置インデックスを利用する [3]．部分スコアとは，ニュースの情報のみであらかじめ計算可能な内容類似度の値である．例えば，本研究では先行研究の内容類似度である式 1 を利用するため，部分スコアは以下のようになる．

$$ps(s,w) = idf^2(w) \cdot \sqrt{\frac{n(s,w)}{|s|}}$$

内容類似度の計算には，従来の情報検索のアプローチである Document at a time (DAAT) [18] を用いる．DAAT は，転置インデックスを用いて文書（ニュース）毎に評価（内容類似度の計算）を行う手法である．転置インデックスの各リストの先頭から後方に向かってポインタ（以降，カレントポジションと呼ぶ）を動かしていき，カレントポジション上のニュースの内容類似度の計算を行う．

DAAT では以下の処理を全ての要素に対して行うことで，ニュースとツイート間の内容類似度を計算することができる．

（1） カレントポジション内で最小のニュース ID を選択．

（2） 選択したニュース ID の内容類似度を計算し（部分スコアの合計），カレントポジションを次の要素に進める．

図 3 に DAAT の処理の例を示す．図 3 は，ニュース $s_1, \ldots, s_5$ と，単語 $w_1, \ldots, w_5$ を含むツイート m を，DAAT を用いてマッチングしている様子を表している．各要素における括弧内の値はニュースの閾値を表しており，右側の値は部分スコアを表している．また，白い矢印は，カレントポジションを表しており，赤色の要素は，カレントポジション内で最小のニュース ID を表している．まず，Step1 では，カレントポジション内で最小のニュース ID が s_1 である．そのため，s_1 との内容類似度の計算を行い（$sim(s_1, m) = 2$），ニュースの閾値と比較を行う（閾値 $= 5$）．ここでは，$sim(s_1, m)$ がニュースの閾値を超えていないため，ニュース s_1 とツイート m の結びつけは行わない．その後，計算を行ったカレントポジションを次の要素に進める．Step2 において，カレントポジション内で最小のニュース ID は s_2 である．先ほどと同様に，s_2 の内容類似度の計算を行い（$sim(s_2, m) = 4$），ニュースの閾値と比較を行う（閾値 $= 12$）．ここでも，$sim(s_2, m)$ がニュースの閾値を超えていないため，ニュース s_2 とツイート m の結びつけは行わない．その後，計算を行ったカレントポジションを次の要素に進める．Step3 において，カレントポジション内で最小のニュース ID は s_3 である．s_3 の内容類似度は 10 であり，ニュースの閾値 8 を上回るため，ニュース s_3 とツイート m の結びつけを行う．その後，計算を行ったカレントポジションを次の要素に進める．ニュース s_4 とニュース s_5 に対しても同様の処理を行うが，ニュース s_4 と s_5 の内容類似度はそれぞれの閾値を超えないため，最終結果としてニュース s_3 のみが出力される．

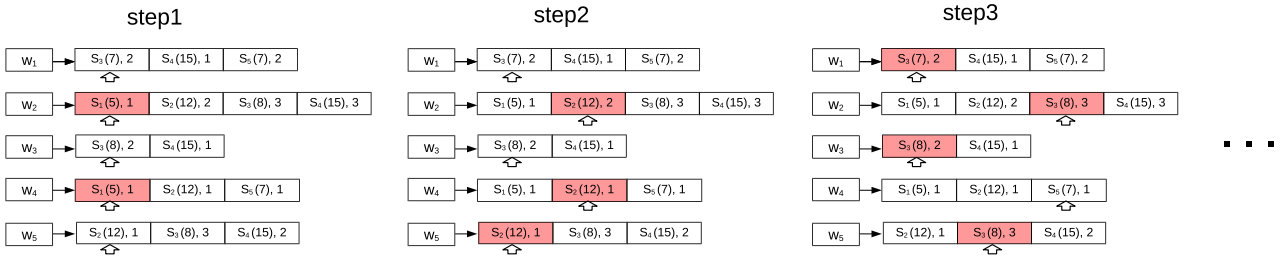

図 3: **DAAT を用いたマッチング処理：**この図はニュース $s_1, \ldots, s_5$ と単語 $w_1, \ldots, w_5$ を含むツイート m を，DAAT を用いてマッチングしている様子を表している．白い矢印は，カレントポジションを表しており，赤色の要素は，カレントポジション内で最小のニュース ID を表している．ニュース s_1 から s_5 の各ニュースとツイート m の内容類似度を計算し，各ニュースの閾値と比較する．最終結果として，ニュース s_3 が出力される．

5.2 効率的なマッチング手法

DAAT では，転置インデックス内の全てのニュースと内容類似度の計算を行う．提案手法では，内容類似度の上限値を用いて，内容類似度の計算が必要のないニュースをスキップしながら，カレントポジションを進めていくことを考える．

内容類似度の上限値は，部分スコアを利用することで計算することができる [19]．まず，部分スコアによって各単語の上限値を計算することができる．単語の上限値とは，単語のリストの中で最大の部分スコアである．例えば，単語 w における上限値 $UB(w)$ は以下のようになる．

$$UB(w) = \max_{s \in L_w} ps(s, w)$$

L_w は単語 w のリストである．内容類似度の上限値は，ニュースとツイートの両方に含まれている単語を用いて以下のように求めることができる．

$$UB(s, m) = \sum_{w \in s \cap m} UB(w)$$

ニュースとツイート間の内容類似度は，UB よりも小さくなるため，UB より大きな閾値をもつニュースとの内容類似度の計算は省略することができる．

提案手法では，転置インデックスをニュースの閾値でソートし，ID 順ではなく，ニュースの閾値の昇順に処理する．こうすることで，内容類似度の上限値が閾値を上回るニュースを高速に検索できるようにする．処理の流れは以下のようになる．

（1） カレントポジションのニュースの閾値で各リストを昇順にソートする．

（2） ソートした順に単語の上限値を加算していき，各単語のリストのカレントポジションにおいて，上限値が閾値を上回るニュースを探す．

（3） (3) のニュースが，先頭のリストのカレントポジションのニュースと同じなら内容類似度の計算を行う．異なる場合，(3) のニュースの閾値以上の閾値をもつニュースまでカレントポジションを進める．

この処理により，内容類似度の上限値がニュースの閾値を上回る可能性があるニュースの内容類似度を計算し，それ以外のニュースとの内容類似度の計算を省略できる．

提案手法のアルゴリズムを Algorithm1 に示す．Algorithm1 では，ツイート m とニュースの閾値の集合 $\Theta = [\theta_{s_1}, \theta_{s_2}, ..., \theta_{s_n}]$ が入力として与えられ，ツイートに関連するニュースの集合 S' を出力する．$L_w.cur$ は，リスト L_w のカレントポジションを表しており，$L_w.curPS$ は，リスト L_w のカレントポジションの部分スコアを表している．まず，ツイート m に含まれる単語のリスト $L_{w_1}, L_{w_2}, ..., L_{w_n}$ を選択し，各リストの先頭をカレントポジションとする（1–3 行目）．次に，上限値を計算していき，上限値が閾値を上回るニュースをリストの中から探す．カレントポジションのニュースの閾値でリストをソートし（5 行目），単語の上限値を加算しながら（9 行目），カレントポジションのニュースの閾値と比較を行う（10 行目）．上限値が閾値を超えるようなニュースがあった場合，そのニュースがあったリストの番号を保持し，for 文から抜ける．上限値が閾値を超えるようなニュースがない場合，処理を終了する（13–14 行目）．その後，そのニュースの内容類似度を計算が必要かどうかを判定する．先頭のリストのカレントポジションにおけるニュースと，保持したリストにおけるカレントポジションのニュースが異なる場合，保持したリストにおけるカレントポジションのニュースの閾値以上のニュースまでカレントポジションを進める（15–17 行目）．同じ場合は，内容類似度を計算し，ニュースの閾値と比較する（19–24 行目）．内容類似度がニュースの閾値を上回っていた場合，そのニュースを結果に加える（25–26 行目）．

図 4 は，提案手法を用いたマッチング処理の例を表している．この図は，ニュース $s_1, \ldots, s_5$ と単語 $w_1, \ldots, w_5$ を含むツイート m を，提案手法を用いてマッチングしている様子を表している．白い矢印は，カレントポジションを表しており，赤色の要素は，内容類似度の上限値が閾値を上回るニュースを表している．各リストをカレントポジションのニュースの閾値でソートし，各単語の上限値を加算しながら，上限値が閾値を上回るニュースを探していく．Step1 では，ニュース s_3 において上限値 UB がニュース s_3 の閾値を上回る．そのため，先頭のリストのカレントポジションのニュース (s_1) と同じか調べる．ニュース ID が異なるため，カレントポジションをニュース s_3 の閾値（閾値 $= 8$）以上の閾値をもつニュースまで進める．Step2 では，ニュース s_3 において上限値 UB がニュース s_3 の閾値を上

回る．先頭のリストのカレントポジションのニュース (s_3) と同じであるため，内容類似度の計算を行う．ニュース s_3 の内容類似度は 10 であり，閾値 8 を超えるため結果として保持する．Step3 では，上限値が閾値を上回るニュースがないため，処理を終了する．最終結果として，ニュース s_3 が出力される．

Algorithm 1: 提案手法

入力: tweet m, $\Theta = [\theta_{s_1}, \theta_{s_2}, ..., \theta_{s_n}]$
出力: S'

$L_{w_1}, L_{w_2}, ..., L_{w_n}$ はツイート m に含まれる単語のリスト
for $w_i \in m$ **do**
    L_{w_i} の先頭をカレントポジションとする
while *true* **do**
    カレントポジションのニュースの閾値でリストを昇順にソート
    $p \leftarrow \perp$
    $UB \leftarrow 0$
    for $i \in [1, 2, ..., |m|]$ **do**
        $UB \leftarrow UB + \max_{s \in L_{w_i}} ps(s, w_i)$
        if $\theta_{L_{w_i}.cur} \leqq UB$ **then**
            $p = i$
            Break
    if $p = \perp$ **then**
        return
    if $L_{w_0}.cur \neq L_{w_p}.cur$ **then**
        for $w_i \in [w_1, w_2, ..., w_{p-1}]$ **do**
            L_{w_i} のカレントポジションを $L_{w_p}.cur$ のニュースの閾値以上のニュースまで進める
    else
        $sim \leftarrow 0$
        $i \leftarrow 0$
        while $L_{w_i}.cur = L_p.cur$ **do**
            $sim \leftarrow sim + L_{w_i}.curPS$
            L_{w_i} のカレントポジションを 1 つ進める
            $w_i \leftarrow w_{i+1}$
        if $sim \geqq \theta_s$ **then**
            s を S' に加える

6. 評価実験

本節では，5. 節で提案したマッチング処理に要する時間を評価した．具体的には，ツイート 1 件当たりのマッチング処理に要する時間の測定を行い，提案手法と先行研究の手法との比較を行った．実験には，Intel®Core™i7-2600 CPU @ 3.40GHz と 32GB のメモリーを使用した．

実験では，以下の 2 つのデータセットを用いた：

- `NEWS`：2014/6/9 に投稿されたニュース 1037 件．ニュースは Yahoo!Japan ニュースの速報ページ (注1) から閲覧可能なニュースを収集した．
- `TWEET`：2014/6/8 から 2014/6/11 に投稿されたツイート 2,575,198 件．ツイートは Streaming API の Sample とよばれるエンドポイントから言語設定を日本語にしているユーザのツイートを収集した．

実験では，以下の 2 つの手法の比較を行った：

- *DAAT/VAT*：先行研究 [4] で用いられている手法．ツイートに含まれている全単語を用いて内容類似度の上限値を計算し，上限値より大きな閾値をもつニュースとの内容類似度の計算を省く．
- *Proposed*：5. 節で説明した提案手法．

実験の方法は以下のようになる：

- `NEWS` 内のニュースを時系列順に転置インデックスへ加える．転置インデックスに加えるニュースの数は，$100, 200, \ldots, 1000$ の 10 通りを試した．
- 転置インデックスに加えたニュースと `TWEET` 内の 100,000 件のツイートのマッチングを行う．ツイートは，一番最後に転置インデックスに加えたニュースの時間を基準とし，その時間以降に投稿されたツイート 100,000 件を利用した．
- マッチング処理にかかった時間を測定し，ツイート 1 件当たりのマッチングにかかった平均時間を比較する．

実験において，ニュースの閾値を定めるためのパラメータは $p = 0.004$，$h = 10$，$\delta = 0.1$ とし，各ニュースに対して 1 時間分の事前ツイートを用いた（閾値の定め方については先行研究 [4] を参照）．

実験の結果を図 5 に示す．図 5 は，各手法を用いた際の 1 ツイート当たりのマッチングにかかった平均時間を表している．x 軸は転置インデックスに加えたニュースの数であり，y 軸は 1 ツイート当たりのマッチングにかかった平均時間を表している．*Proposed*，*DAAT/VAT* の結果はそれぞれ丸線，三角線で表している．*Proposed* の処理時間は，*DAAT/VAT* の処理時間と比べて，最大で 54.8% 削減されている．

DAAT/VAT では，ツイートに含まれている全ての単語を利用して内容類似度の上限値を計算しているため，単語数の多いツイートの場合，内容類似度の上限値が大きくなってしまう．内容類似度の上限値が大きくなると，内容類似度の計算を省略できるニュースの数が少なくなってしまうため，マッチング処理に時間を多くの要してしまう．提案手法では，ニュースとツイートの両方に含まれている単語を用いて内容類似度の上限値を計算しているため，より多くのニュースとの内容類似度の計算を省略できる．そのため，マッチング処理に要する時間が少なくなっている．

7. 結　論

本稿では，ニュースとツイートをリアルタイムにマッチングするための手法を提案した．ツイートに対して，関連のあるニュースを効率的に結びつけることによって，大量のニュース

(注1)：http://news.yahoo.co.jp/flash

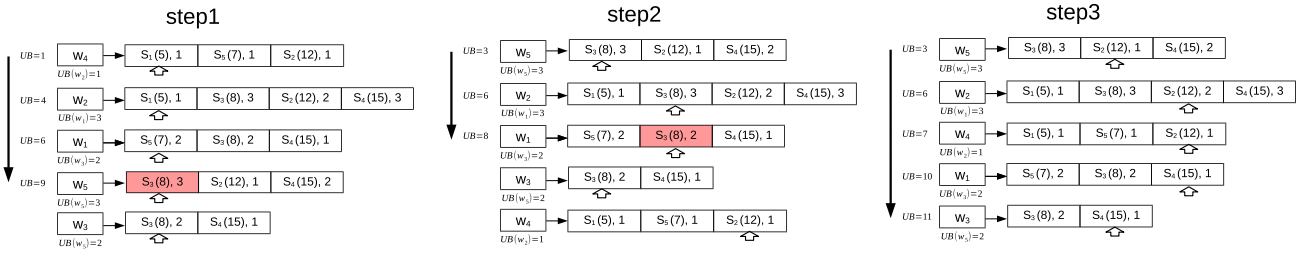

図 4: **提案手法を用いたマッチング処理：**この図はニュース $s_1, \ldots, s_5$ と単語 $w_1, \ldots, w_5$ を含むツイート m を，提案手法を用いてマッチングしている様子を表している．白い矢印は，カレントポジションを表しており，赤色の要素は，内容類似度の上限値がニュースの閾値を上回ったニュースを表している．ニュース s_3 とツイート m の内容類似度のみ計算し，ニュース s_3 の閾値と比較する．最終結果として，ニュース s_3 が出力される．

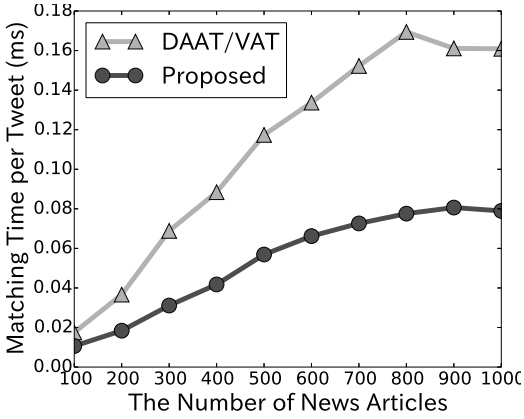

図 5: **マッチング時間：**各手法を用いた際の 1 ツイート当たりのマッチングにかかった平均時間を表している．x 軸は転置インデックスに加えたニュースの数であり，y 軸は 1 ツイート当たりのマッチングにかかった平均時間を表している．*Proposed*，*DAAT/VAT* の結果はそれぞれ丸線，三角線で表している．*Proposed* の処理時間は，*DAAT/VAT* の処理時間と比べて，最大で 54.8% 削減されている．

に対応できるようにした．具体的には，ニュースとツイートの両方に含まれている単語を用いて内容類似度の上限値を計算し，上限値以上の閾値をもつニュースとの内容類似度の計算を省略した．

本稿での貢献は以下の通りである：

- **効率的なマッチングアルゴリズム：**ニュースとツイートのリアルタイムなマッチングにおいて，大量のツイートに対応できるマッチングアルゴリズムを提案した（図 5）．

また，実データを用いた評価実験により提案手法の有効性を示した．ニュースとツイートのマッチングにおいて，ツイート 1 件あたりに処理時間を測定し，提案手法が既存手法よりも処理時間を最大で 54.8% 削減できることを示した．

今後の研究では，マッチング処理のさらなる高速化として，転置インデックス内のリストを分割するアプローチが考えられる．提案手法では，単語のリスト内に部分スコアが大きなニュースがあった場合，単語の上限値が大きくなってしまい，内容類似度の計算を省略できるニュースの数が減少してしまうという問題がある．そのため，転置インデックス内のリストを分割し，単語の上限値が大きくなることを防ぐことで，より多くのニュースを省略できると考えられる．

謝辞 本研究の一部は，JSPS 科研費・基盤研究（B）（26280037）および文部科学省「実社会ビックデータ利活用のためのデータ統合・解析技術の研究開発」による．

文　献

[1] H. Kwak, C. Lee, H. Park, and S. B. Moon, “What is twitter, a social network or a news media?,” in *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp.591–600, Raleigh, North Carolina, USA, April 26-30, 2010.*

[2] S. Petrovic, M. Osborne, R. McCreadie, C. Macdonald, I. Ounis, and L. Shrimpton, “Can twitter replace newswire for breaking news?,” in *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013.*

[3] A. Shraer, M. Gurevich, M. Fontoura, and V. Josifovski, “Top-k publish-subscribe for social annotation of news,” *PVLDB*, vol. 6, no. 6, pp. 385–396, 2013.

[4] S. Onishi, Y. Yamaguchi, and H. Kitagawa, “Real-time relevance matching of news and tweets,” in *On the Move to Meaningful Internet Systems: OTM 2015 Conferences - Confederated International Conferences: CoopIS, ODBASE, and C&TC2015, pp.109–126, Rhodes, Greece, October 26-30, 2015.*

[5] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining,” in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, May 17-23, 2010.*

[6] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, “Sentiment analysis of twitter data,” in *Proceedings of the Workshop on Languages in Social Media, LSM 2011, pp.30–38, Portland, Oregon, 2011.*

[7] Z. Luo, M. Osborne, and T. Wang, “Opinion retrieval in twitter,” in *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012.*

[8] G. Amati, M. Bianchi, and G. Marcone, “Sentiment estimation on twitter,” in *Proceedings of the 5th Italian Information Retrieval Workshop, pp.39–50, Roma, Italy, January 20-21, 2014.*

[9] B. Sharifi, M. Hutton, and J. K. Kalita, “Experiments in microblog summarization,” in *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010, pp.49–56,*

Minneapolis, Minnesota, USA, August 20-22, 2010.

[10] D. Chakrabarti and K. Punera, "Event summarization using tweets," in *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011.*

[11] J. Nichols, J. Mahmud, and C. Drews, "Summarizing sporting events using twitter," in *17th International Conference on Intelligent User Interfaces, IUI 2012, pp.189–198, Lisbon, Portugal, February 14-17, 2012.*

[12] S. V. Canneyt, M. Feys, S. Schockaert, T. Demeester, C. Develder, and B. Dhoedt, "Detecting newsworthy topics in twitter," in *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), pp.25–32, Seoul, Korea, April 8, 2014.*

[13] M. Osborne and M. Dredze, "Facebook, twitter and google plus for breaking news: Is there a winner?," in *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*

[14] B. Shi, G. Ifrim, and N. Hurley, "Be in the know: Connecting news articles to relevant twitter conversations," *CoRR*, vol. abs/1405.3117, 2014.

[15] W. Guo, H. Li, H. Ji, and M. T. Diab, "Linking tweets to news: A framework to enrich short text data in social media," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, pp.239–249, Sofia, Bulgaria, Volume1: Long Papers, August 4-9, 2013.*

[16] T. Stajner, B. Thomee, A. Popescu, M. Pennacchiotti, and A. Jaimes, "Automatic selection of social media responses to news," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, pp.50–58, Chicago, IL, USA, August 11-14, 2013.*

[17] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops, pp.120–123, Toronto, Canada, August 31 - September 3, 2010.*

[18] H. Turtle and J. Flood, "Query evaluation: Strategies and optimizations," *Inf. Process. Manage.*, vol. 31, pp. 831–850, Nov. 1995.

[19] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien, "Efficient query evaluation using a two-level retrieval process," in *Proceedings of the Twelfth International Conference on Information and Knowledge Management, CIKM 2003, pp.426–434, New Orleans, LA, USA, 2003.*