# 時間的相関性を考慮したTwitterに対するエンティティリンキング

長城　沙樹†　　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
E-mail: †s.nagaki@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

**あらまし**　Twitterでは，多くのユーザによってリアルタイムに大量の投稿がなされており，それらの投稿（ツイート）を活用した技術が数多く研究・開発されている．自然言語処理の基盤技術として，文章に含まれるエンティティを示す語句と，知識ベース中のエンティティを対応づけるエンティティリンキングという技術がある．多くの既存手法は，同一文章内に出現するエンティティ同士の知識ベース内の関連度を利用している．しかしながらTwitterはリアルタイム性が特徴として存在し，1ツイートに含まれるエンティティ同士が既存の知識ベース上では関連度が大きくない場合でも，特定のイベントに関しては共起しやすいといった場合がある．そこで，本研究ではTwitterにおける時間的相関性を考慮したエンティティリンキング手法を提案する．

**キーワード**　エンティティリンキング，Twitter，知識ベース，Wikipedia

## 1. 序　　論

「トランプ大統領の就任演説聞いた？」「明日トランプ持ってきて！」この2つの投稿に共通して含まれる「トランプ」という語句は概念的に違うものを示しているが，機械がその概念を区別するにはどのようにすればよいだろうか．

本論文では，Twitter (注1) に投稿されたツイートに対するエンティティリンキング手法を提案する．エンティティリンキングは自然言語処理技術の1つであり，文章に含まれるエンティティを示す語句と，知識ベース中のエンティティを対応づける技術である．これにより，自然言語で書かれた文章を知識ベースにおけるエンティティ，つまり機械理解可能な概念に変換することができる．

Twitterでは，多くのユーザによって大量のツイートが投稿されており，そのツイートを人出で処理するには限界がある．これらのツイートを機械理解可能な概念に紐付けることで，様々な技術に応用できる．例えば，従来は文章に含まれる単語を元に開発されたトレンド抽出手法やイベント抽出手法などを,「概念」を元にすることで，その概念の背景知識などを考慮した手法に拡張することができる．

しかしながら，従来のエンティティリンキング手法はツイートに対しては精度が低くなる [1] という問題がある．この問題は，Twitterに投稿されるツイートの以下の特徴に起因する．

**Shortness**　ツイートは140字以内という制限がある．

**Ambiguity**　ツイートに含まれている語は表記ゆれが大きい．

**Ungrammaticality**　ツイートは正しい文法で記載されているとは限らない．

特に，一般的なエンティティリンキング手法では，同じ文章の内容には一貫性があると仮定し，同じ文章内に出てくる語同士の関連度を利用する．そのため，ツイートの短さ，つまり，1つのツイート内に出てくる語句が少ないということは重要な課題となる．また語同士の関連度は，多くの既存手法において知識ベース上の情報を用いて計算されるが，同じツイート内に出てくる語同士が，知識ベース上で関連度が大きいとは限らない．

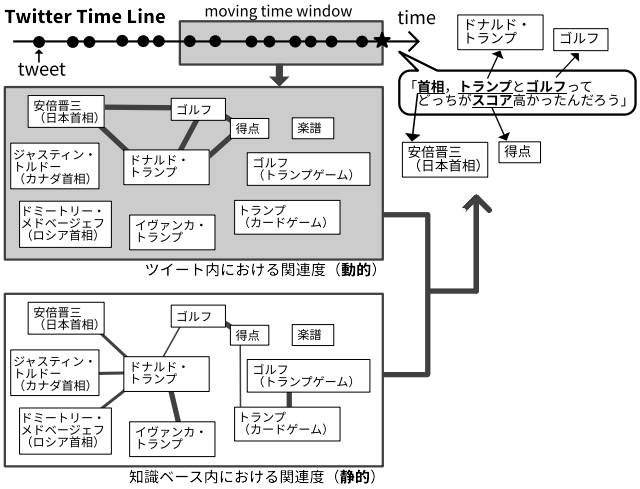



図 1: 提案手法 TAGME+概要. 矩形で囲まれた語句（「ドナルド・トランプ」,「ゴルフ」など）はエンティティを示す．エンティティ同士を結ぶ線は，エンティティ間の関連度を示し，線が太いほど関連が強いことを意味する．

本研究では，これらの問題に対処するため，Twitterにおけるリアルタイム性を利用する．そのために，Twitterにおける時間的相関性を利用した **TEAM** (**T**emporal **E**ntity **A**ssociation **M**easure) という尺度を導入する．TEAMは, ある時間ウィンドウにおけるエンティティ同士の関連度を表し，ツイートに対するエンティティリンキングに有用であると考えられる．

続いて，既存手法をTEAMを用いて拡張したTAGME+を提案する．提案手法TAGME+の流れを図1に示す．この図は,「首相，トランプとゴルフってどっちがスコア高かったんだろ

(注1)：https://twitter.com/

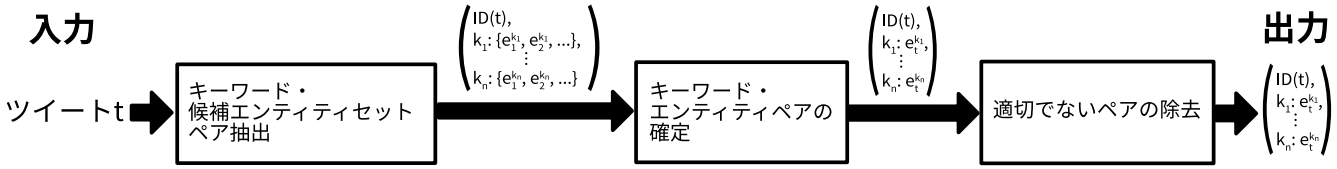



図 2: エンティティリンキングフレームワーク．各入力文章（ツイート）を入力とし，入力文章に含まれる（キーワード，対応するエンティティ）ペアのセットを出力とする．フレームワークは，(1) キーワード・候補エンティティセットの抽出，(2) キーワードに割り当てるエンティティの確定，(3) 本文と関連度が低いキーワード・エンティティペアの除去，の 3 処理によって構成される．

う」というツイートに対するエンティティリンキングを考えたものである．このツイートのみでは，それぞれの「トランプ」,「首相」,「ゴルフ」という語が，どのエンティティを意味するのかを知識ベースに含まれる情報のみを用いて判断することは難しい．知識ベース中には,「首相」を表すエンティティが複数存在したり，カードゲームのトランプの 1 つのゲームとして「ゴルフ」というゲームが存在するからである．しかしながら，直前に「ドナルド・トランプ」アメリカ大統領が「安倍晋三」首相と「ゴルフ」をしたというツイートが多くある時，3 つのエンティティ「ドナルド・トランプ」「安倍晋三」「ゴルフ」に関して，互いの関連度が一時的に高くなる．この一時的な関連度を利用することで,「首相，トランプとゴルフってどっちがスコア高かったんだろう」というツイートに含まれるそれぞれのキーワードが正しいエンティティに割り当てられる．

本研究の貢献は以下の通りである．

- Twitter に投稿されたツイートを用いて，時間的なエンティティ間の関連度である TEAM を定義した．
- 実データを用いた評価実験によって，定義した TEAM が，Twitter に対するエンティティリンキングに有用であることを示した．

評価実験において，TEAM を用いた提案手法と，それを用いない既存手法の両方を用いて，エンティティリンキングを行った．その結果，TEAM の重みを変化させることで，再現率・適合率曲線の AUC (Area Under the Curve) 値で最大 60% の精度向上を達成した．また，提案手法で用いるタイムウィンドウ幅を変化させ，精度が変化することを示した．これは，ツイート内の時間的相関性が，エンティティリンキングにおいて有効であることを示している．

## 2. エンティティリンキングフレームワーク

本節では，提案手法で用いるエンティティリンキングフレームワークについて記す．本フレームワークは，入力を各ツイート，出力を入力ツイートに含まれる ⟨ キーワード，対応するエンティティ⟩ ペアのセットとする．

本フレームワークは，一般的なエンティティリンキング手法と同様，(1) キーワード・候補エンティティセットの抽出，(2) キーワードに割り当てるエンティティの確定，(3) 本文と関連度が低いキーワード・エンティティペアの除去，の 3 処理によって構成され，図 2 のような流れとなる．各処理の詳細を「首相，トランプとゴルフってどっちがスコア高かったんだろう」というツイートに対する処理の例とともに以下で示す．

### 2.1 キーワード・候補エンティティセットの抽出

まず，入力した文章に対して，エンティティを割り当てる対象となる語句（以下，キーワードとする）を抽出する．また，本モジュールではキーワードに割り当てる候補となるエンティティ集合も同時に抽出する．この処理によって，1 つの入力文章に対して，文章内に存在するキーワードと，各キーワードに割り当てる候補となるエンティティ集合のペア集合が抽出される．

「首相，トランプとゴルフってどっちがスコア高かったんだろう」というツイートに対しては，エンティティに対応するキーワードとして,「首相」,「トランプ」,「ゴルフ」,「スコア」が抽出できるとする．それぞれのキーワードに対して，候補エンティティ集合が抽出される．キーワードに対する候補エンティティ集合を抽出するには，あらかじめキーワードをキー，候補エンティティ集合をバリューとする辞書を作成しておき，この辞書を利用して抽出する手法が一般的である [2]．例えば，キーワード「首相」に対しては,「ジャスティン・トルドー（カナダ首相）」,「安倍晋三（日本首相）」,「ドミートリー・メドヴェージェフ（ロシア首相）」が候補エンティティ集合として抽出できる．ここで，キーワード「トランプ」に対しては,「ドナルド・トランプ」「イヴァンカ・トランプ」「トランプ（カードゲーム）」，キーワード「ゴルフ」に対しては,「ゴルフ」「ゴルフ（トランプゲーム）」，キーワード「スコア」に対しては,「得点」「楽譜」が候補エンティティ集合としてそれぞれ抽出されるとする．

### 2.2 キーワードに割り当てるエンティティの確定

次に，抽出した各キーワードについて，その割り当てる候補となるエンティティ集合をスコアリングし，割り当てるエンティティを確定する．本処理によって，入力文章に対して，文章内に存在するキーワードと，割り当てるエンティティのペア集合が抽出される．

第 2.1 節で抽出したペアセットに対して当該モジュールの処理を行った結果，キーワード「首相」に対しては「安倍晋三（日本首相）」，キーワード「トランプ」に対しては「ドナルド・トランプ」，キーワード「ゴルフ」に対しては「ゴルフ」，キーワード「スコア」に対しては「得点」が割り当てられる．

### 2.3 本文と関連度が低いペアの除去

最後に，2.2 節で抽出された ⟨ キーワード，エンティティ⟩ ペアから，入力文章と関係の強くないペアを除去する．このようにして，入力文章に対して，その文章に含まれるキーワードと割り当てるエンティティペアの集合を取得できる．

キーワード「スコア」，エンティティ「得点」のペアは本文中の他のキーワード，エンティティペアとの関連が弱いと考えると，このペアが本処理で除去される．最終的に，入力ツイート「首相，トランプとゴルフってどっちがスコア高かったんだろう」について，⟨ キーワード，エンティティ⟩ ペアとして，⟨ 首相，安倍晋三（日本首相）⟩，⟨ トランプ，ドナルド・トランプ ⟩，⟨ ゴルフ，ゴルフ ⟩ が出力される．

## 3. TAGME

本研究では，Twitter における時間的相関性をエンティティリンキングに利用するための，基となるアルゴリズムとして，TAGME [3] を利用する．TAGME は，Wikipedia の記事を 1 つのエンティティとみなし，入力された短い文章に対して，その文章内に含まれる語句を Wikipedia の記事に紐付ける．本研究の基となる手法として TAGME を採用した理由としては，2010 年に Google Faculty Award を受賞した手法であり (注2)，非常に引用数が多い (注3) 以外にも以下が挙げられる．

（1） 品詞分類や機械学習を必要としない

他の手法の多くは，先に知識ベースと紐付ける対象となる語句を抽出する必要がある．日本語の場合，形態素解析器を利用した語句抽出が考えられるが，エンティティリンキングの精度が形態素解析結果に依存する可能性が高くなる．英語のようなスペース区切りの限語に関しても，対象となる語句が 1 単語とは限らない．また，事前に他の固有表現抽出手法を用いることも考えられるが，本研究でリンキングする対象は固有表現に限らないため，この方法は利用できない．

（2） 高速に処理が可能

Twitter には非常に多くのツイートがリアルタイムに投稿されている．本研究では，直前に投稿されたツイートの情報をエンティティリンキングに用いるため，高速に処理されることがのぞましい．また，高速に処理することが可能になれば，トピック抽出やイベント抽出などの Twitter におけるリアルタイム性を活かした応用例にも適用ができると考えられる．

TAGME は，第 2. 節で示したフレームワークの通り，(1) キーワードの抽出，(2) 候補となる記事のスコアリング，(3) 本文と関連度が低いキーワード・記事ペアの除去，の 3 処理によって構成される．以降の節で，それぞれの処理の流れを記す．

### 3.1 キーワード・候補エンティティセットの抽出

本節では，エンティティを紐付ける語句（以下，キーワードとする．）の抽出方法について述べる．まず，入力文章から Wikipedia のアンカーテキストとして用いられている語句を全て抽出し，キーワードの候補とする．次に，抽出された 2 つの語句 $k_1$, $k_2$ について，$k_1$ が $k_2$ の部分文字列である時，キーワードとしてどちらが適切かを，Wikipedia 内で各語句がアンカーテキストとして使われる割合

$$lp(k) = \frac{link(k)}{freq(k)}, \tag{1}$$

を用いて判断する．ここで，$k$ は抽出されたキーワードの候補を示し，$link(k)$ は語句 $k$ が Wikipedia 内でアンカーテキストとして出現している回数，$freq(k)$ は語句 $k$ が Wikipedia 内で出現している回数である．ある語句 $k_1$ が，別の語句 $k_2$ の部分文字列である時，$lp(k_1) < lp(k_2)$ であれば，$k_2$ のみをキーワードとして抽出し，$lp(k_1) \geqq lp(k_2)$ であれば，$k_1$，$k_2$ のいずれもキーワードとして抽出する．

### 3.2 キーワードに割り当てるエンティティの確定

本節では，第 3.1 節で抽出したキーワードに対して，対応する候補となる Wikipedia の記事集合のスコアリング手法を示す．キーワード $k \in K$ について，対応する候補となる記事集合を，そのキーワードによってリンクされる Wikipedia 記事集合 $Pages(k)$ とする．TAGME では, ひとつの文章に含まれるエンティティの関連度は高いという仮定をし，文章内に出現する他のエンティティとの関連度を用いて，各記事 $e_k \in Pages(k)$ をスコアリングする．

エンティティ間の関連度は，Wikipedia Link-based Measure（以下，WLM とする.）[4] を利用する．WLM は，Wikipedia における記事同士のリンクを利用した，記事の関連度を測る尺度であり，TAGME 以外にも多くのエンティティリンキング手法で用いられている [2]．WLM では，Wikipedia 内の記事 $a$, $b$ の関連度を以下と定義する．

$$WLM(a,b) = 1 - \frac{\log\max(|A|,|B|) - \log|A \cap B|}{\log|All| - \log\min(|A|,|B|)}. \tag{2}$$

ここで，$|A|$, $|B|$ はそれぞれ記事 $a$, $b$ にリンクしている記事集合を示し，$All$ は Wikipedia における全記事集合を示す．

WLM を用いて，文書中のあるキーワード $k$ が記事 $e_k$ に紐づく可能性を表すスコアを以下のように計算する．

$$rel_k(e_k) = \sum_{k' \in K \setminus \{k\}} \frac{\sum_{p_{k'} \in Pages(k')} WLM(p_k, e_k) \cdot Pr(p_{k'} \mid k')}{|Pages(k')|}, \tag{3}$$

ここで，$Pr(e_k \mid k)$ は語句 $k$ がアンカーテキストとして出現する回数に占める，記事 $e_k$ にリンクする割合である．式 3 は，記事 $e_k$ が文書中に出現する他のキーワードと紐づく記事 $p_{k'}$ と関連度が強いほど，高いスコアになる．TAGME では，$rel_k(e_k)$ のスコア上位 $\epsilon\%$ の記事のうち，$Pr(e_k \mid k)$ が最も高い記事 $e_k$ をキーワード $k$ が示す記事とする．

### 3.3 本文と関連度が低いペアの除去

第 3.2 節の処理によって紐付いたキーワード・記事ペア集合には，入力文章と内容的に関連のない組み合わせが含まれている可能性がある．そのような組み合わせを排除するため，各ペア $k, e_k$ に対して，

$$coherence(k, e_k) = \frac{1}{|S| - 1} \sum_{e'_k \in S \setminus \{e_k\}} WLM(e_k, p_{k'}), \tag{4}$$

を計算する．ここで，$S$ は今候補となっている全てのペア内の記事集合とする．最終的に，

$$\frac{1}{2}(lp(k) + coherence(k, e_k)) > \rho, \tag{5}$$

を満たすペアを，文書中のキーワードとそのキーワードに紐づく記事とする．

(注2)：https://TAGME.d4science.org/TAGME/TAGME_help.html
(注3)：引用数 283.（2016 年 11 月 8 日現在．Google Scholar 参照.）

## 4. 提 案 手 法

本研究の提案手法は，エンティティリンキングを行う対象となるツイートより前に投稿されたツイート集合における，エンティティの共起率を用いる．第 4.1 節において，あるタイムウィンドウ $T$ におけるエンティティ同士の関連度を表す指標である **TEAM** (**T**emporal **E**ntity **A**ssociation **M**easure) について定義する．続いて，第 4.2 節において，定義した TEAM を用いて TAGME を拡張した手法 TAGME+について示す．

### 4.1 TEAM

本節では，Twitter における時間的なエンティティの関連度 **TEAM** (**T**emporal **E**ntity **A**ssociation **M**easure) を提案する．TEAM は,「ある時間におけるエンティティ同士の関連度」を意味し，以下で定義する．

**定義（Temporal Entity Association Measure）**
あるタイムウィンドウ $W$ に投稿されたエンティティ $a, b$ を含むツイート集合を，それぞれ $A, B$ とし，タイムウィンドウ $W$ に投稿された全ツイート数を $|All|$ とする．また，$p(a,b)$ を，タイムウィンドウ $W$ 内で投稿された全ツイートに対する $A \cap B$ の割合とし，$\frac{|A \cap B|}{|W|}$ で計算する．同様に，$p(a)$, $p(b)$ を，タイムウィンドウ $W$ 内で投稿された全ツイートに対する $A$, $B$ の割合とし，それぞれ $\frac{|A|}{|W|}$, $\frac{|B|}{|W|}$ で求める．

この時，エンティティ $a, b$ のタイムウィンドウ $W$ における Temporal Entity Association Measure を以下で定義する．

**TEAM-jaccard**

$$TEAM_{jac}(a,b,W) = \frac{|A \cap B|}{|A \cup B|}. \tag{6}$$

**TEAM-dice**

$$TEAM_{dice}(a,b,W) = \frac{2 \cdot |A \cap B|}{|A| + |B|}. \tag{7}$$

**TEAM-npmi**

$$TEAM_{npmi}(a,b,W) = \frac{\ln \frac{p(a,b)}{p(a)p(b)}}{-\ln p(a,b)}. \tag{8}$$

**TEAM-gsd**

$$TEAM_{gsd}(a,b,W) = 1 - \frac{\log \max(|A|,|B|) - \log |A \cap B|}{\log |All| - \log \min(|A|,|B|)}. \tag{9}$$

ただし，式 9 の値が 0 未満の場合，$TEAM_{gsd}(a,b,W) = 0$ とする． □

TEAM-jaccard は，2 つの集合に対する類似度を計算する指標である Jaccard 係数を利用したものである．2 つの集合 $X$, $Y$ の Jaccard 係数は以下の式で計算される．

$$Jac(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}. \tag{10}$$

式 10 の値域は $0 \leqq Jac(X,Y) \leqq 1$ である．ゆえに，TEAM-jaccard の値が高い（1 に近い）ほど，エンティティ $a, b$ を含むツイートはタイムウィンドウ $W$ 内で投稿されたツイート内で共起しやすいことを示している．

TEAM-dice は，2 つの語に対する類似度を計算する指標である Dice 係数を利用したものである．2 つの語 $x$, $y$ の Dice 係数は以下の式で計算される．

$$Dice(x,y) = \frac{2 \cdot |X \cap Y|}{|X| + |Y|}. \tag{11}$$

式 11 の値域は $0 \leqq Dice(x,y) \leqq 1$ である．ゆえに TEAM-jaccard と同様，TEAM-dice の値が高い（1 に近い）ほど，エンティティ $a, b$ を含むツイートはタイムウィンドウ $W$ 内で投稿されたツイート内で共起しやすいことを示しているが，Jaccard 係数とくらべて共起の割合を重視する．

TEAM-pmi は，自己相互情報量（pointwise mutual information (PMI)）を利用したものである．PMI は 2 つの確率分布 $X, Y$ に対して，それぞれの事象 $x, y$ が同時に起こる確率を示す指標であり，以下の式で計算される．

$$PMI(X=x, Y=y) = \ln \frac{P(X=x, Y=y)}{P(X=x)P(Y=y)}. \tag{12}$$

PMI の値域は $-\infty < PMI(X=x, Y=y) < \infty$ であるため，TEAM-pmi には PMI を正規化した Normalized PMI [5] を利用した．Normalized PMI は，以下の式で計算される．

$$NPMI(X=x, Y=y) = \frac{\ln \frac{P(X=x,Y=y)}{P(X=x)P(Y=y)}}{-\ln P(X=x, Y=y)}. \tag{13}$$

多くの手法で，Jaccard 係数，Dice 係数及び PMI はいずれも語句の意味的な類似度を測るために利用されている [6] [7] [8]．

TEAM-gsd は，Google Similarity Distance (GSD) [9] を利用したものである．GSD は，2 つのキーワードの類似度を Google 検索を用いて計算するものであり，2 つのキーワード $x$, $y$ の GSD は以下の式で計算される．

$$GSD(x,y) = 1 - \frac{\log \max(|X|,|Y|) - \log |X \cap Y|}{\log |All| - \log \min(|X|,|Y|)}. \tag{14}$$

ここで，$X$, $Y$ は，キーワード $x$, $y$ で Google 検索をしたときの結果として返ってくるページ集合を示す．また $|All|$ は，Google によって検索される全ての記事数を示す．GSD の値域は，$-\infty < GSD(x,y) \leqq 1$ であるため，式 14 の計算結果が 0 未満となるとき，$TEAM_{gsd} = 0$ とする．この条件により，$TEAM_{gsd}$ の値域は $0 \leqq TEAM_{gsd} \leqq 1$ となる．

ツイートに対するエンティティリンキングにどの TEAM が有効かを，第 5. 節の評価実験にて示す．

### 4.2 TAGME+

第 3. 節で示した通り，TAGME では静的な指標である WLM を用いている．本節では，動的な指標である TEAM を用いて，TAGME を拡張した手法 TAGME+を示す．

提案手法の基本アイデアは，動的な指標である TEAM と静的な指標である WLM を組み合わせてエンティティ同士の関連度を利用することである．エンティティリンキングを行う対象のツイートが投稿された時刻を $\tau$ とする．タイムウィンドウ幅を $w$ とした時，$\tau$ の直前の時間 $w$ をタイムウィンドウ $W$ とする．この時，動的な指標である TEAM と静的な指標である WLM を組み合わせた以下の指標を定義する．

**Algorithm 1:** TAGME+

**Input:** ツイートタイムライン $TL$, タイムウィンドウ幅 $w$, キーワード・候補エンティティ集合辞書 $D$, パラメータ $\rho, \lambda, \epsilon$
**Output:** $EL$

```
for tweet T from TS do
    i ← id(T), τ ← time(T), t ← text(T)
    EW ← shaveWindow(τ, w)
    K ← extractKeywords(T)
    for k from K do
        for e_k from Candidates(k) do
            rel_{e_k} ← calcRel(k, e_k, K, EW, λ)
            Rel_{e_k} ← Rel_{e_k} ∪ {(e_k, rel_{e_k})}
        end
        E_k ← topRel(Rel_{e_k}, ε)
        e_k ← arg max_{e_k ∈ E_k} Pr(e_k|k)
        S.enqueue((k, e_k))
    end
    for (k, e_k) from S do
        coh_{k,e_k} ← calcCoherence(k, e_k)
        score ← 1/2 (kp(k) + coh(k, e_k))
        if score > ρ then
            EL ← EL ∪ {(i, k, e_k)}
            EW ← updateWindow(i, τ, k, e_k)
        end
    end
end
return EL
```

$$rel_\tau(a,b) = (1-\lambda)\cdot WLM(a,b) + \lambda \cdot TEAM(a,b,W). \quad (15)$$

ここで，$\lambda$ は, WLM と TEAM の重みを調整するパラメータである．

TAGME において，WLM を用いている式 3 及び式 4 について，WLM の代わりに式 15 を用いる．そのようにすることで，静的な関連度だけでなく，エンティティ間の動的な関連度も考慮することができる．

提案手法 TAGME+のアルゴリズムをアルゴリズム 1 に示す．TAGME+は，入力をツイートストリーム $TS$，タイムウィンドウ幅 $w$，パラメータ $\rho, \lambda, \epsilon$ とし，出力を各ツイートの ID，キーワード，リンキングされたエンティティの 3 タプルから構成されるリストとする．TAGME+は，ツイートストリームに含まれるツイートに対して，以下の処理を行う．まず，ツイートの投稿時間から，タイムウィンドウを移動させる（3 行目）．ここで $EW$ は，タイムウィンドウ内に存在するツイートの (1) ツイート ID, (2) ツイートに含まれるキーワード，(3)(2) に紐づくエンティティの 3 タプルから構成されるリストを示す．関数 $shaveWindow(\tau, w)$ は，投稿時間とタイムウィンドウ幅を用いて，$EW$ から $\tau - w$ 以前に投稿されたツイートの情報を除く関数である．次に，第 3.1 節で述べた通り，ツイート本文からキーワードを抽出する（4 行目）．TAGME では，式 1 を用いていたが，中村ら [10] が拡張した手法を採用した．この手法は，式 1 の代わりに Mihalcea ら [11] の *Keyphraseness* を用いる．続いて，各キーワードに対して，そのキーワードが紐づく候補となる記事集合から，式 3 及び式 15 を用いて，キーワードと各記事の関連度を計算し，キーワードと紐づく記事を決定する（6–13 行目）．最後に，第 3.3 節で述べたようにツイート本文に含まれる他のキーワード，記事ペアを元に，本文との関連度が低いペアを排除する．式 5 に関してもキーワード抽出と同様，*Keyphraseness* を用いる．

## 5. 評価実験

本章では，第 4.1 節で定義した TEAM が，エンティティリンキングに対して有効か否かを，実際のツイートデータを用いて評価する．本実験での評価項目は以下の通りである．

（1） 各 TEAM のエンティティリンキングに対する影響度はどのようになるか.（第 5.2 節）

（2） TEAM は，ツイートに対するエンティティリンキングに対して有効であるか.（第 5.3 節）

（3） 提案手法のタイムウィンドウ幅 $w$ が精度にどの程度影響するか.（第 5.4 節）

第 5.1 節にて実験設定を，第 5.2, 5.3, 5.4 節にてそれぞれの実験について述べる．最後に，第 5.5 節にて本実験について議論する．

### 5.1 実験設定

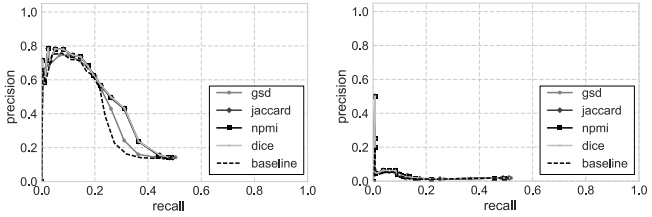
本実験で用いるデータは，WWW2016 の併設ワークショップ #Microposts2016 (注4) にて開催された，NEEL(Named Entity rEcognition and Linking) Challenge のデータを用いて作成した．このデータセットは，2011 年及び 2015 年に投稿されたツイートを Twitter Firehose を用いて収集したものから作成したものであり，開発データ，訓練データ，学習データとしてそれぞれ 100 件，3,164 件，6,025 件の計 9,289 件のツイートから構成される．提案手法は教師なし学習であるため，学習データを必要としないため，今回の評価実験では用意されている 9,289 件のツイートのうち，2016 年 12 月 26 日現在 Twitter REST API (注5) を用いて収集できる 1,486 件のツイートを使用した．この評価データの正解ラベルには DBpedia (注6) を用いている．そのため，SPARQL1.1 (注7) を介して，DBpedia のエンティティを Wikipedia の記事に変換し，正解ラベルとした．Wikipedia データは，2016 年 12 月 1 日に公開されたバージョンのダンプデータ (注8) を利用した．

提案手法は，Twitter タイムライン上のツイートに対してエンティティリンキングを行うため，上記のデータセットを 2011 年，2015 年に投稿されたものの 2 種類に分割し，2 種類のデータセット *tweets2011*，*tweets2015* を作成した．このデータセットはいずれも，(1) ツイート本文，(2) ツイートの投稿日時，(3) ツイート本文で言及されるエンティティセット，から構成される．本データセットの概要を表 1 に示す．

(注4)：http://microposts2016.seas.upenn.edu/index.html
(注5)：https://dev.twitter.com/rest/public
(注6)：http://wiki.dbpedia.org/
(注7)：https://www.w3.org/TR/sparql11-query/
(注8)：https://dumps.wikimedia.org/enwiki/

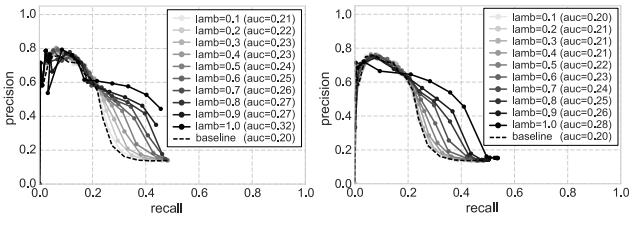
表 1: 評価実験データセット概要

| | | *tweets2011* | *tweets2015* |
|---|---|---|---|
| ツイート数 | | 689 ツイート | 797 ツイート |
| エンティティを含むツイート数 | | 636 ツイート | 70 ツイート |
| エンティティ数/ツイート | 平均値 | 1.97 個/ツイート | 0.16 個/ツイート |
| | 最大値 | 9 個/ツイート | 5 個/ツイート |
| | 最小値 | 0 個/ツイート | 0 個/ツイート |
| 平均単語数 | | 17.42 語 | 13.03 語 |
| 投稿日時 | | 2011 年 7 月 16 日から 8 月 15 日 | 2015 年 12 月 15 日から 12 月 16 日 |

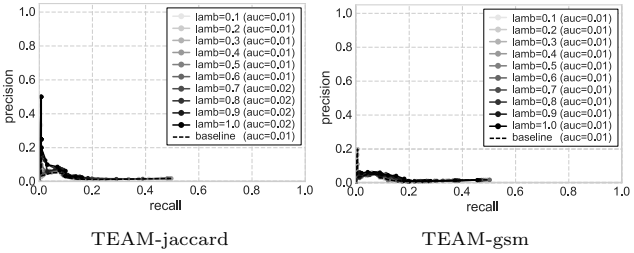



(a) *tweets2011* (b) *tweets2015*

図 3: 各 TEAM による影響度の差．$x$ 軸は再現率，$y$ 軸は適合率を示す．この図から，*tweets2011* については，指標による差がみられるが，*tweets2015* では差がないことがわかる．*tweets2011* では，TEAM-gsm のみが他の指標と傾向が異なることがわかる．




(a) *tweets2011*




(b) *tweets2015*

図 4: TEAM の有効性．$x$ 軸は再現率，$y$ 軸は適合率を示す．式 15 で用いられているパラメータ $\lambda$ を変化させ，WLM と TEAM の重みを変化させた結果を示す．TEAM を全く使用しない場合よりも，TEAM を使用した方が精度は高く，TEAM がツイートに対するエンティティリンキングに有効といえる．

ベースラインには中村らが拡張した TAGME-kp [10] を用いる．提案手法，ベースラインに共通するパラメータ $\epsilon$ は TAGME [3] で用いられている $\epsilon = 30(\%)$ を採用した．

評価のため，各データに対して式 5 のパラメータ $\rho$ を 0 から 1 まで 0.5 刻みで変化させ，再現率・適合率曲線（以下，PR 曲線）を描画した．

### 5.2 各 TEAM の影響度の差

本節では,「各 TEAM のエンティティリンキングに対する影響度はどのようになるか.」という点について評価する．そのため，第 4.1 節で提案した TEAM-jaccard, TEAM-dice, TEAM-npmi, TEAM-gsm を用いてツイートデータに対してエンティティリンキングを行い，PR 曲線を描画した．その他のパラメータについては，$\lambda = 0.5$, $w = 6hours$ とした．

結果を図 3 に示す．$x$ 軸は再現率，$y$ 軸は適合率を示す．この図から，*tweets2011* については，指標による差がみられるが，*tweets2015* では差がないことがわかる．*tweets2015* は，1 ツイートあたりに含まれるエンティティ数が少ないため，エンティティ間の関連度による影響が少ないと考えられる．そのため，エンティティ間の関連度を変更しても精度に差がでにくいと考えられる．

また，*tweets2011* では，TEAM-GSM のみが他の指標と傾向が異なることがわかる．これは，エンティティ $a, b$ がタイムウィンドウ内のツイートで共起しないとき，TEAM-jaccard, TEAM-dice, TEAM-npmi はいずれも 0 をとるが，TEAM-gsm のみ異なる値をとるからだと考えられる．そのため，この後の評価実験では，TEAM-jaccard 及び TEAM-gsm の 2 指標を用いて実験を行う．

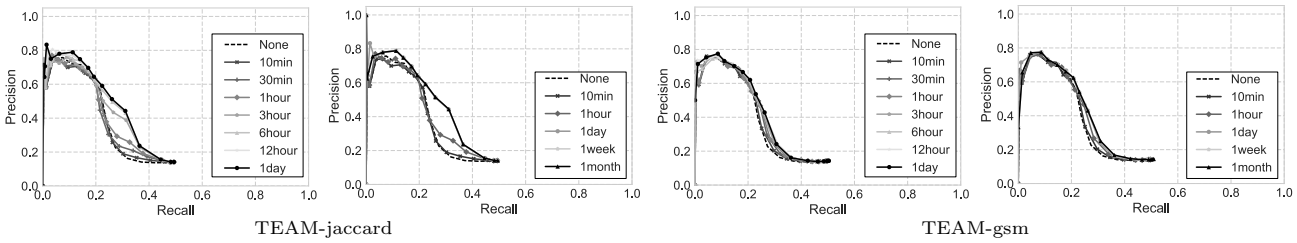
### 5.3 TEAM の有効性

本節では,「TEAM は，ツイートに対するエンティティリンキングに対して有効であるか.」という点について評価する．そのため，提案手法を用いて，式 15 にて用いられているパラメータ $\lambda$ を 0 から 1 まで 0.1 刻みで変化させ，ツイートデータに対してエンティティリンキングを行い，PR 曲線を描画した．ベースラインは，$\lambda = 0$ の時と同等である．その他のパラメータについては，$w = 6hours$ とした．

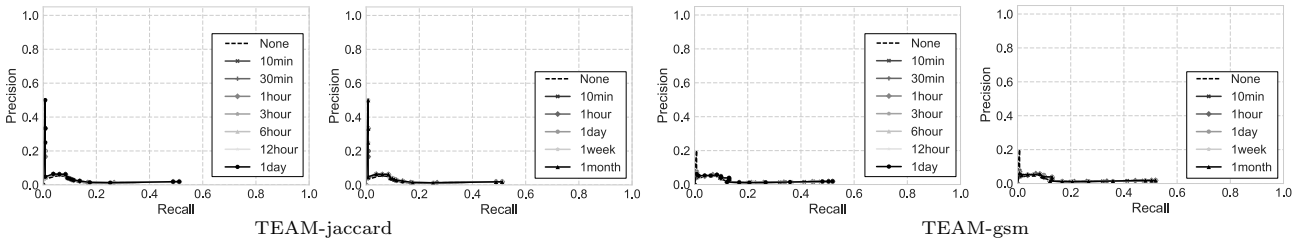
結果を図 4 に示す．式 15 で用いられているパラメータ $\lambda$ を変化させ，WLM と TEAM の影響度合いを変化させた結果を示し，$x$ 軸が再現率，$y$ 軸が適合率を示す．

データセット *tweets2011* に関して，ベースライン（$\lambda = 0$）は，PR-AUC は 0.20 である．それと比較して $\lambda = 1$ の時，TEAM-jaccard を用いると PR-AUC は 0.32，TEAM-gsm を用いると PR-AUC は 0.28 となった．これは，既存手法と比べて提案手法によって PR-AUC 値で最大 60%精度が上昇したことを示す．よって，TEAM がツイートに対するエンティティリンキングに有効であるといえる．

また，データセット *tweets2011* では，TEAM-jaccard, TEAM-gsm のいずれも $\lambda = 1.0$ の時が最も PR-AUC 値は高くなるが，PR 曲線を見ると，再現率が低いときの適合率は $\lambda \neq 1.0$ の方が高いことがわかる．これは，式 15 の値が高いときの適合率は，TEAM のみを用いてエンティティリンキングを行うよりも，WLM と TEAM を融合させた方が高くなることを示している．エンティティリンキングは，その特性として，適合率を重視することが多いため，応用手法によって $\lambda$ 値を調整する必要があると考えられる．




(a) *tweets2011*




(b) *tweets2015*

図 5: タイムウィンドウによる影響度の差．$x$ 軸は再現率，$y$ 軸は適合率を示す．データセット *tweets2011* については，いずれも $w = 1day$ が最も精度が良く，そこからタイムウィンドウ幅を広げると，$\rho$ を低くした時の適合率が低くなる．

データセット *tweets2015* に関しては，第 5.2 節での実験と同様，差はほとんど見られなかった．これは，第 5.2 節と同様，1 ツイート内で共起するエンティティが少なく，エンティティ間の関連度を変更しても精度に差がでにくいからだと考えられる．

### 5.4 タイムウィンドウ幅の影響

本節では，「提案手法のタイムウィンドウ幅 $w$ が精度にどの程度影響するか.」という点について評価する．そのため，提案手法について $w$ を変化させ，ツイートデータに対してエンティティリンキングを行った．$w$ の変化方法については，0 から $1day$ まで変化させたものと，0 から $1month$ まで変化させたものを 2 種類で変化させた．$w = 0$ (None) の時，他の評価実験で用いているベースラインと同様になる．WLM と TEAM の重みパラメータは $\lambda = 0.5$ とした．

結果を図 5 に示す．データセット *tweets2015* については，他評価実験と同様に差が見られなかった．データセット *tweets2011* については，TEAM-jaccard，TEAM-gsm ともに似た傾向を示している．いずれも，$w = 1day$ が最も精度が良く，そこからタイムウィンドウ幅を広げると，PR 曲線の立ち上がりが遅くなっている．これは再現率が低い時，式 15 の値が高い時に，影響を与えている TEAM に余分な情報が入り，正解以外のエンティティを割り当ててしまうために，適合率が低くなっていると考えられる．ゆえに，タイムウィンドウ幅 $w$ は $1day$ が適切であるといえる．これは，Twitter のリアルタイム性により，タイムウィンドウが長すぎると精度が落ちるためと考えられる．

### 5.5 議　　論

評価実験により，データセット *tweets2011* について，TEAM がツイートに対するエンティティリンキングに有効であることが示された．特に第 5.3 節の実験では，ツイートに対するエンティティリンキングについては，提案した TEAM のみをエンティティ間の関連度として用いる手法が全体的には精度が最も高いことが示された．しかしながら，応用手法によって，$\lambda$ 値は適切に調整する必要があると考えられる．また，タイムウィンドウ幅は $w = 1day$ が適切であると示された．これは，Twitter にはリアルタイム性が存在するために，タイムウィンドウが長すぎると精度が落ちるという直感にも従う．いずれの実験についても，TEAM-jaccard を用いた TAGME+は，TEAM-gsm を用いた TAGME+やベースラインとくらべて，適合率が高いときも再現率が下がりにくいという特徴がある．これは多くのツイートについて，TEAM-jaccard を用いた TAGME+は適切にエンティティリンキングができることを示している．

データセット *tweets2015* については，TEAM を組み合わせても精度が変化しなかった．これは，*tweets2015* は，1 ツイートに含まれるエンティティ数が少なく，エンティティ間の関連度による影響が少ないと考えられる．そのため，エンティティ間の関連度を変更しても精度に差がでにくいと考えられる．

## 6. 関 連 研 究

ツイートに対するエンティティリンキングは，(1)140 字の文字制限，(2) 語の表記ゆれ，(3) 文法が曖昧という課題のもと取り組まれてきた．特に一般的な手法は同じ文章内に現れる語句が示すエンティティとの関連度を利用するが，Twitter の文字制限はこの手法に対する，大きな課題となる．最も単純な手法としては，ひとりのユーザが投稿したツイートをつなぎ合わせて，ひとつの文章とみなすという方法が考えられるが，これはひとつの文章内のトピックは一貫しているという仮定に反する．

そのような問題に対処するため，ツイートに対する効果的なエンティティリンキング手法が考案されてきた．Liu ら [12] は，ツイートに対して，キーワード同士，キーワードとエンティティ，リンキング候補となるエンティティ同士の関連度を測る複数の尺度を提示し，それらを組み合わせることによって，ツイートに対するエンティティリンキングの精度がどのように向上するかを検証した．その結果，ツイートを投稿したユーザ情

報やハッシュタグが，エンティティリンキングに有効であることを示した．また，Shen ら [13] は，ユーザの投稿した全ツイートを用いてそのユーザの興味モデルを作成し，それを用いたエンティティリンキングシステム KAURI を考案した．KAURI は知識ベースを利用して作成したエンティティ同士の関連度を示すグラフに対して，ユーザの興味を示すスコアを埋め込むことで，高い精度でのエンティティリンキングを実現した．しかしながら，KAURI はユーザの興味モデルを構築するために，そのユーザが今までに投稿した全ツイートが必要となる．

Twitter の特徴として，リアルタイム性があげられる．これを利用してエンティティリンキングを行う手法が考案されてきた．Fang ら [14] は，既存のテキスト情報を利用した手法に加え，Twitter に含まれる投稿時刻と位置場所の情報を利用して，ツイートに対するエンティティリンキング手法を考案した．しかしながら，GPS を用いた位置情報付きツイートは全体の 1.6%しか存在せず [15]，Fang らが利用したユーザのプロフィール情報を用いた手法を用いても全体の 25%ほどしか位置情報を推定できない．そのため，Fang らの手法は全てのツイートに対して適用できるわけではない．また，Hua ら [16] は，(1) ユーザのフォロー・フォロワー関連と，(2) エンティティの一時的な人気度を用いた手法を考案した．Hua らの手法では，Twitter におけるエンティティの人気度を，そのエンティティと知識ベース上で関連の強いエンティティに伝搬させ，エンティティについてのスコアリングを行う．しかしながら，Wikipedia に存在しない関連については適応できていない．本研究は，エンティティ同士の一時的な関連度をツイートのリアルタイム性を用いて発見するため，Hua らの研究とは異なる．

## 7. 結　　論

本研究では，リアルタイムに投稿されるツイートを用いて，動的なエンティティ間の関連度を示す尺度である TEAM を定義した．そして，知識ベースの情報のみを用いて計算される静的なエンティティ間の関連度に加えて，定義した TEAM を考慮したエンティティリンキング手法を提案した．実際のツイートデータを用いた評価実験により，TEAM が Twitter に対するエンティティリンキングに有用であることを示した．

今後は，ツイート本文と割り当てられたエンティティの分析を行う．また現在は，第 2. 節での「キーワード・エンティティペアの確定」にて，キーワードと対応するエンティティを**１つのみ**抽出しているが，これを上位 $k$ 件に変化させることにより，精度がどのように変化するかを確かめる．また，データセット *tweets2015* 内のツイートのように，1 ツイート内にエンティティが共起しない場合，非常に精度が低くなってしまう．そのようなツイートにも対応した手法を考案してゆきたい．

## 謝　　辞

本研究の一部は，NICT 高度通信・放送研究開発委託研究「欧州との連携による公共ビッグデータの利活用基盤に関する研究開発」による．また，既存手法のソースコードを提供してくださった大阪大学の中村様に深く感謝する．

## 文　　献

[1] G. Rizzo, M. van Erp, and R. Troncy. Benchmarking the extraction and disambiguation of named entities on the semantic web. In *Proc. 9th International Conference on Language Resources and Evaluation*, LREC '14, Reykjavik, Iceland, 2014.

[2] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 2, pp. 443–460, 2015.

[3] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, Vol. 1, No. 29, pp. 70–75, 2012.

[4] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press*, pp. 25–30, Chicago, USA, 2008.

[5] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proc. the Biennial GSCL Conference*, Vol. 156, 2009.

[6] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proc. the 16th International Conference on World Wide Web*, WWW '07, pp. 757–766, New York, USA, 2007.

[7] D. Bollegala, Y. Matsuo, and M. Ishizuka. A web search engine-based approach to measure semantic similarity between words. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 7, pp. 977–990, 2011.

[8] G. Lu, P. Huang, L. He, C. Cu, and X. Li. A new semantic similarity measuring method based on web search engines. *W. Trans. on Comp.*, Vol. 9, No. 1, pp. 1–10, 2010.

[9] R. L Cilibrasi and P. MB Vitanyi. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, Vol. 19, No. 3, 2007.

[10] 中村達哉, 白川真澄, 原隆浩, 西尾章治郎. ソーシャルメディアからの言語横断的な話題抽出に向けたエンティティリンキング手法. 第 7 回データ工学と情報マネジメントに関するフォーラム, 2015.

[11] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proc. the 16th ACM conference on information and knowledge management*, pp. 233–242, 2007.

[12] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *ACL (1)*, pp. 1304–1311, 2013.

[13] W. Shen, J. Wang, P. Luo, and M. Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proc. the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 68–76, 2013.

[14] Y. Fang and M. Chang. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics*, Vol. 2, pp. 259–272, 2014.

[15] K. Leetaru, S. Wang, G. Cao, A. Padmanabhan, and E. Shook. Mapping the global twitter heartbeat: The geography of twitter. *First Monday*, Vol. 18, No. 5, 2013.

[16] W. Hua, K. Zheng, and X. Zhou. Microblog entity linking with social temporal context. In *Proc. the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1761–1775, 2015.