

スポーツを「いいとこどり」で観戦するための盛り上がり予測手法

藤本 和[†] 牛尼 剛聡^{††}

[†] 九州大学大学院芸術工学府 〒815-8540 福岡県福岡市南区塩原 4-9-1

^{††} 九州大学大学院芸術工学研究院 〒815-8540 福岡県福岡市南区塩原 4-9-1

E-mail: [†]fujimoto.nodoka.017@s.kyushu-u.ac.jp, ^{††}ushiana@design.kyushu-u.ac.jp

あらまし 我々は、スポーツ中継等のリアルタイムで体験することに価値があるイベントに対して、ユーザにとっての「いいところ」をリアルタイムに体験することを支援するシステムの開発を行っている。このシステムでは、プロ野球中継における実況ツイートを実況ツイートをリアルタイムで分析し、ユーザにとっての「いいところ」の出現を予測し、スマートスピーカを利用してユーザにリアルタイムに通知する。本論文では、実況ツイートの特徴を反映したベクトル化のために、テキストの表記揺れやツイート固有の独特な表現などに対応可能な、文字レベルのオートエンコーダを利用する。そして、深層学習を用いて、リアルタイムに現在の盛り上がりの大きさ、盛り上がりの極性（ポジティブ/ネガティブ）の予測を行う手法を提案する。

キーワード ソーシャルセンサ, Twitter, 実況ツイート, 文字レベル深層学習, オートエンコーダ

1 はじめに

スマートスピーカは、音声によって様々なコンテンツが利用可能なインターフェースであり、日常生活空間での様々な活動をサポートするデバイスとして広まりつつある。スマートスピーカは、スマートフォンなどの画面を操作するデバイスとは違い、作業をしながら操作できるという特徴がある。本研究では、このスマートスピーカを用いて、リアルタイムに配信されるコンテンツについて、ユーザにとっての価値ある状況を知らせるシステムを提案する。

ここで、リアルタイムに配信されるコンテンツとは、テレビの生中継やインターネットで配信される生放送の動画などを指し、ここではこれらをリアルタイムコンテンツと呼ぶ。リアルタイムコンテンツは、ダイジェストやニュースなどでまとめられたものよりも、リアルタイムに楽しむことに価値がある。特にスポーツ中継は、録画ではなく生中継をリアルタイムに楽しむことによって、興奮や感動を味わうことができる。しかし、リアルタイムに楽しむためには、コンテンツを継続して見続ける必要があるが、ユーザが忙しい、あまり盛り上がらないとつまらないときには、視聴者は試合の全てを見ておくのが困難という問題がある。

このような問題を解決するために、我々は、コンピュータがリアルタイムコンテンツの状況を常に観測し、ユーザにとって価値ある部分となる「いいところ」を予測して、ユーザに通知するシステムを開発中である。本システムを使うことによって、ユーザがコンテンツに対して注意を向けていないときでも、ユーザにとって価値がある部分だけを「いいとこどり」し、効率よく体験できることを目指す。本システムの概要を図 1 に示す。スマートスピーカを利用することで、ユーザが忙しい時や、対象とする事象に注意を向けていないときでも、音声によって

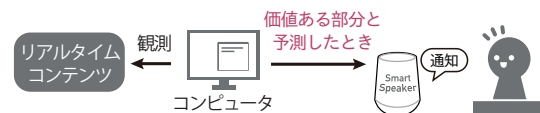


図 1 価値ある場面を予測するシステムの概要図

通知することでリアルタイムに情報を伝えるコンテンツの利点を活かすことができる。

これまでに、スポーツ等の内容を効率的に理解するために、動画コンテンツに対して、動画に対するコメントの特徴を抽出して高度な検索や推薦をしたり、ダイジェストを自動生成する研究が行われてきた [1–3]。これらは、過去の出来事に関するコンテンツの重要な部分や盛り上がった部分を抽出して、ユーザの効率的にコンテンツの体験に役立っている。しかし、本研究で提案するシステムは、リアルタイムに価値ある部分を体験できるという点で異なる。ダイジェストの生成と本システムを比較した図を図 2 に示す。ダイジェストは動画コンテンツの全てが終了してから生成されるものだが、本システムではリアルタイムに「いいところ」を体験することができる。

本システムでは、コンピュータがリアルタイムコンテンツを

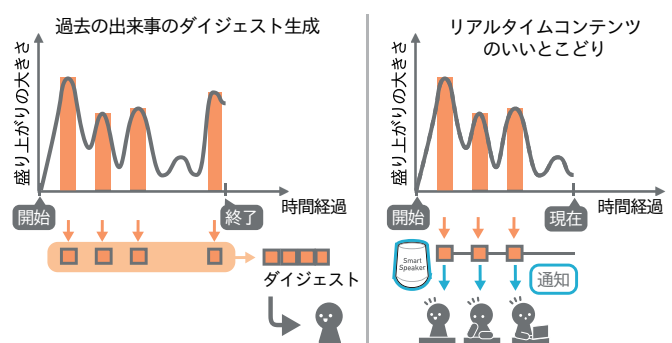


図 2 ダイジェストの生成とリアルタイムの「いいとこどり」の違い

観測するにあたって、SNS 上の情報を用いる。近年の SNS の普及によって、誰でも気軽に情報を発信することができるようになった。SNS の中でも、特に Twitter¹は、既存のメディアよりもリアルタイムな情報を得られるという特徴があり、交通情報や天候、地震に関する状況などを確認することにも役に立つ。このように、実世界のリアルタイムな状況を観測するために SNS が活用されており、物理的なセンサと同様の機能を果たすと考えることができる。この考え方は、「ソーシャルセンサ」などと呼ばれており、イベント推定や流行の抽出などといった研究が多く行われている [4-7]。SNS の利用方法のひとつに、ソーシャルビューイングがあり、これから得られる情報もセンサになりうる一つである。ソーシャルビューイングは、TV 番組やインターネットの動画コンテンツを視聴しながら、リアルタイムにその感想や意見を投稿したり、その投稿を見る形態である。Twitter ではこの投稿を実況ツイートと呼び、同じ番組の視聴者とその感想や意見を共有して盛り上がるができる。

本論文では、リアルタイムに体験することに価値があるスポーツ中継の中でも、プロ野球中継を対象に提案するシステムの開発を行う。ユーザにとっての「いいところ」を予測するために、野球中継に対してのソーシャルビューイングで、Twitter に投稿される実況ツイートから特徴を抽出し、盛り上がりや極性の予測を行う。

2 野球中継の「いいとこどり」システム

2.1 システムの概要

一般的なスポーツの実況ツイートは、客観的な情報と主観的な情報の両方が含まれている。客観的な情報とは、試合の中で実際に起きた出来事であり、主観的な情報とは、客観的な出来事に対するファンの盛り上がりを示している。実況ツイートに基づいて、ユーザにとっての「いいところ」を予測し、ユーザに通知を行う。このシステムの概要を図 3 に示す。

ユーザによって、本システムで通知を希望する「いいところ」は異なる。例えば、盛り上がっていても、応援するチームに点が入ったときはポジティブな盛り上がりであり、チャンスで点が入らなかったときはネガティブな盛り上がりとなる。我々はユーザにとっての「いいところ」を判定するために図 4 のように、「盛り上がり」と「ポジティブ/ネガティブ」の 2 つの軸を利用する。この 2 つの指標のどの部分がユーザにとっての「いいところ」なのかを事前に指定しておき、その予測値が指定された部分になったときに、スマートスピーカが通知する。

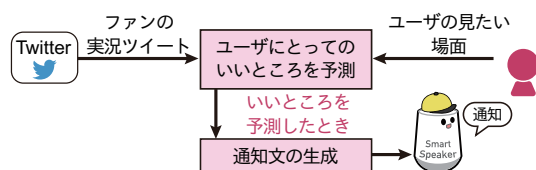


図 3 「いいところ」を予測するシステムの概要図

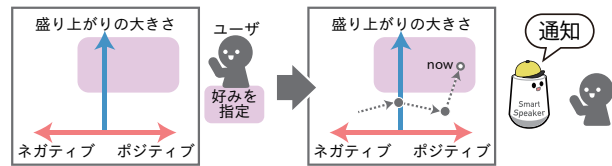


図 4 ユーザに合わせた通知のシステム

2.2 アプローチ

本論文では、ユーザにとっての「いいところ」を予測する手法を提案する。実況ツイートから、リアルタイムに盛り上がりの度合いと、極性（ポジティブ/ネガティブ）の予測を行う。

本研究では、実況ツイートの特徴を量的な側面と質的な面から捉える。スポーツ中継を対象とした実況ツイートでは、盛り上がる場面では、短時間の間に多くの投稿が行われることが多く、これは投稿の量的な特徴であると考えられる。一方テキストの文面から、盛り上がりの種類を想像できる。これは、対象とする場面に投稿された全てのツイートを見るのではなく、投稿されたツイートの一部を見るだけで想像できることが多い。つまり、最新の少量のツイートのテキストの内容を分析するだけで、現在の場面の盛り上がりの度合いを予測できる可能性がある。また、ツイートの取得は TwitterAPI²を利用するが、少量のツイートのみを取得することで、リアルタイムに大量のツイートを取得するというコストを減らすことができる。本研究では、少量のツイートにおける内容的特徴から、盛り上がりを表す直近短時間のツイート投稿数、極性（ポジティブ/ネガティブ）を予測する。

ツイートの内容的特徴の抽出方法については、様々な手法が考えられ、一般的にはツイートに含まれる単語の意味に基づいて、特徴を解析する手法が多い [6, 8]。しかし、スポーツの実況ツイートは、表記揺れやノイズ、Twitter 固有の特殊な表現が多く、一般的な形態素解析を使った単語分割による手法はうまく適用できない場合がある。特に、ツイート投稿数が急激に増える盛り上がる場面では、「！」の記号の連続や顔文字など、ユーザの興奮状態が単語の意味として現れない場合が多い。そのため、文字列のパターンの特徴が重要だと考えられる。そこで、本研究では単語レベルの処理ではなく、文字レベルの処理で特徴を抽出し、機械学習を用いてツイートテキストの埋め込みベクトルを生成する方法を用いる。この処理によって、ツイート投稿者の反応パターンをモデル化する。そして、ツイートの埋め込みベクトルを利用して、盛り上がりを表す単位時間の投稿数、極性（ポジティブ/ネガティブ）を予測する。この学習において、状況のパターンを学習する。この提案手法の概要を図 5 に示す。

3 関連研究

本研究では、動画コンテンツの特徴抽出、ツイート投稿数の予測を行うため、それぞれの関連研究と本研究の位置付けについて述べる。

1 : <https://twitter.com>

2 : <https://developer.twitter.com>

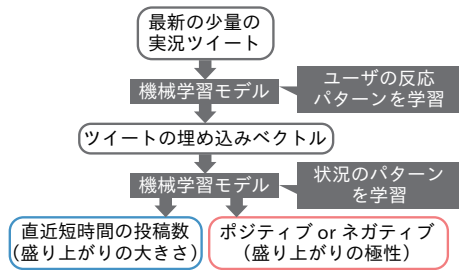


図 5 提案手法の概要図

3.1 動画コンテンツの特徴抽出

本研究では、動画に対する投稿から場面の特徴抽出を行う。コメントなどの投稿の特徴を分析することで動画の特徴を抽出を行う研究は、これまでもいくつか行われている。中澤ら [1] は、コメントの投稿数の変動からテレビ番組の重要シーンを自動的に検出する手法を提案している。山内ら [2] は、コメントの内容を扱う手法として、観点が入れ替わっている部分を特徴的シーンとみなす手法が提案されている。また、佃ら [3] は、感情辞書を用いた登場人物の活躍パターンから動画間の類似度を算出する手法を提案している。これらの手法では、ダイジェストの生成、動画の推薦や検索等を目的としており、対象とするコンテンツの始まりから終わりまで全てを分析している。一方、本研究では、リアルタイムに投稿テキストの特徴を抽出し、盛り上がりを予測することを目的としている。

3.2 ツイート投稿数の予測

Kong ら [9] や Wang ら [10] は、投稿数が急激に増えるバースト状態を予測している。これらは、SNS でトレンドとして話題が拡散されることによって投稿数が増え、しばらくすると徐々に投稿数が減っていく段階を予測している。本研究では、現実起こった事象に対して、体験者にとっての盛り上がりを予測する点で異なる。

4 ツイートの埋め込みベクトル化

4.1 テキストのベクトル化についての先行研究

テキストをベクトル化する方法は、単語レベルの手法 [6, 8] と文字レベルの手法 [11–15] に二分できる。

単語の意味的な情報をベクトル化する単語埋め込みの手法として、代表的な手法に Word2Vec [16] などがある。この手法ではコーパスとなる文章中の単度の出現パターンに基づいて、単語の埋め込みベクトルを生成する。このような単語レベルの処理を行う場合には、日本語のテキストであれば、形態素解析を行って単語分割を行う必要がある。

文字レベルの埋め込みベクトル化を行う方法では、単語分割を行う必要がなく、表記揺れや独特な表現のある文章にも対応できる。Ling ら [11] は、テキストを文字レベルの one-hot ベクトルの系列として表現し、それを単語の埋め込みベクトルに変換させる C2V モデルを提案している。また、Ling らと同じく、文字レベルの one-hot ベクトルの系列を生成し、再帰的ニューラルネットワークや畳み込みニューラルネットワーク等

の深層学習によって、テキストの分類などを行う手法が提案されている [12–14]。一方、Vosoughi ら [15] は、オートエンコーダによってテキストの埋め込みベクトルを生成し、埋め込みベクトルを入力とした機械学習によってテキスト分類等の性能を評価している。オートエンコーダは、エンコーダとデコーダから構成される深層学習モデルを、入力データ自体を正解ラベルとして学習させるものである。オートエンコーダのエンコーダ部分を取り出したモデルによって、重要な特徴を残して次元圧縮することができる。これらの手法では、アルファベットと記号を用いたテキスト、または日本語をアルファベットに変換した、表音文字で構成されたテキストを one-hot ベクトルに変換し、入力としている。

本研究では、Vosoughi らの手法のように、文字レベルの入力による深層学習によって、オートエンコーダで埋め込みベクトルを生成する手法を提案する。日本語のツイートのテキストを対象とし、漢字、ひらがな、カタカナ、絵文字、記号などをそのまま処理するため、アルファベットのテキストよりも文字種の量が格段に多く、また、1 文字に含まれる情報量の大きさに違いがある。

4.2 one-hot ベクトルの生成

本論文で提案するツイートのベクトル化手法について述べる。いま、ツイートに利用される文字種類数を d としたとき、テキストの i 番目の文字を表す one-hot ベクトルは、 $\mathbf{x}_i \in \{0, 1\}^d$ となる。このとき、 n 文字からなるツイートは one-hot ベクトル $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ の行列として表現できる。

Twitter の投稿からリアルタイムの盛り上がりを予測する場合、考察などを行っている長いテキストの投稿よりも、投稿者の突発的な反応が見られる短いテキストの投稿の方が重要だと考えられる。つまり、投稿テキストのうち、先頭のある程度の文字数のみを分析するだけで、盛り上がりが予測できる可能性がある。そこで、リアルタイムに処理可能となるように処理速度を削減するため、投稿テキストの先頭の n 文字のみを使用する。なお、Twitter の最大文字数は 140 であるため、 $n < 140$ である。

日本語は文字の種類が多く、全ての文字種を用いた one-hot ベクトルを作ると次元数が非常に大きくなってしまふ。そのため、出現頻度が上位 d 個の文字を対象とした one-hot ベクトルを使用する。そして、オートエンコーダの入力となる one-hot ベクトルは、 d 個の文字と、文字数 n に合わせるためのパディング (PAD)、未知の文字 (UNK) の 2 つを加えた、 $n \times (d+2)$ 次元の行列とする。

4.3 オートエンコーダによるテキストの埋め込みベクトル化

4.2 で示したツイートの one-hot ベクトルを用いて、ツイート埋め込みベクトルを生成する。オートエンコーダは、図 6 のように、入力データを正解データとしてエンコーダとデコーダを学習させる。学習済みのオートエンコーダのエンコーダ部を利用して、エンコードしたベクトルを埋め込みベクトルとする。Vosoughi らの手法で利用された疑似モデルに加えて、エンコー

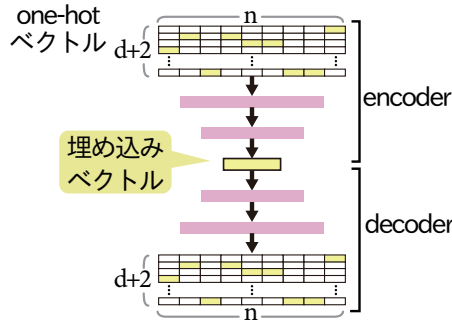


図 6 オートエンコーダによるテキストの埋め込みベクトル化

ダとデコーダに分けて複数のモデルを提案する。

4.3.1 Encoder

Vosoughi らの提案した畳み込み層と再帰的ニューラルネットワークの一つである LSTM [17] を用いる手法から、畳み込み層のフィルターのサイズや数を調整した擬似モデル (model CPCPCCL), GRU [18] を用いた手法 (model GCGFF, GF, bGtCF) を提案する。GRU は、LSTM と同じく再帰的ニューラルネットワークの一つだが、LSTM よりも計算量が少なく高速という特徴がある [19]。提案するモデルの構成を図 7 に示す。図中において、Conv は畳み込み層を表し、 ks はカーネルサイズ、 f はフィルターの数を表す。Pool はプーリング層を表し、最大プーリングを利用し、プーリングサイズは ps とする。また、Flat は 1 次元配列化、FC は全結合層、timeFC は時間単位の全結合層、Bi-GRU は双方向の GRU である。LSTM と GRU において、(seq) は各系列を出力し、(last) は系列の最後を出力する。本研究では、どのモデルでも共通して 300 次元の埋め込みベクトルに圧縮する。

4.3.2 Decoder

本論文で提案するデコーダは、文字数分の繰り返し (Repeat) を行い、GRU、時間単位の全結合層 (time-FC)、活性化関数の softmax 関数で、元の one-hot ベクトルを出力する。提案するエンコーダの構成を図 8 に示す。

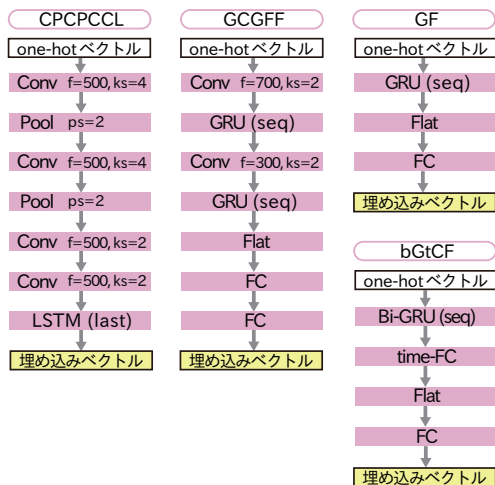


図 7 エンコーダのモデル

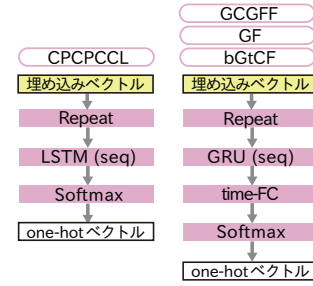


図 8 デコーダのモデル

5 ツイートの投稿数、極性の予測手法

それぞれのツイートを行列 X として表し、エンコーダによって 300 次元の埋め込みベクトル v に変換する。この変換を式で表すと、式 (1) となる。

$$v = \text{Encoder}(X) \quad (1)$$

k 個のツイートから得られた埋め込みベクトル $\{v_1, v_2, \dots, v_k\}$ を入力として、深層学習によって、直近のツイート投稿数、極性 (ポジティブ/ネガティブ) の予測を行う。それぞれの入力と出力、予測モデルの詳細について述べる。

5.1 ツイート投稿数の予測

4 で生成した実況ツイートの埋め込みベクトルから、現在の盛り上がりを示すツイート投稿数の予測を行う。図 9 のように、時系列のツイートに窓関数を適用して、窓幅を 2 分として時系列のツイートを分割する。そして、試合開始から i 分後 $(i+1)$ 分後までに含まれる k 個のツイートを用いて、 i 番目の時間窓に含まれる全体のツイート投稿数 $count_i$ を予測する。ツイート投稿数は、各試合における時間窓ごとの最大ツイート投稿数で割ることによって、正規化を行う。

予測モデルを $estimateC$ とすると、 i 番目の時間窓に含まれるツイートの埋め込みベクトル $v^{(i)}$ を用いて、 $count_i$ は式 (2) で表せる。

また、ツイート投稿数は、試合の時間経過に関連すると考えられる。つまり、試合の始まった直後は視聴者が少なく、ツイート投稿数が増えづらく、時間が経過するごとに視聴者が増える傾向にある。そのため、ツイート投稿数を予測するモデルの入力として、試合経過時間 m (分) を加える。試合経過時間は、0 以上 1 以下にするため、実際の値を 300 (分) で割る。この 300 (分) (=5 時間) という値は、2019 年度のセ・リーグの

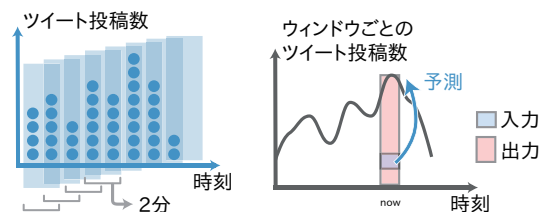


図 9 時系列のツイートの分割とツイート投稿数の予測

試合の最長時間が4時間58分³であったことからそのように定めた。時間窓幅を2分としたため、 $m = i + 1$ となり、予測モデルを $estimateC'$ とすると、 $count_i'$ は、 i 番目の時間窓に含まれるツイート1つの埋め込みベクトル $v^{(i)}$ を用いて、式 (3) で表せる。

$$\hat{count}_i = estimateC(v_1^{(i)}, v_2^{(i)}, \dots, v_k^{(i)}) \quad (2)$$

$$\hat{count}_i' = estimateC'(v_1^{(i)}, v_2^{(i)}, \dots, v_k^{(i)}, m) \quad (3)$$

提案する式 (2) と式 (3) のモデルの構造を図 10 に示す。

5.2 極性の予測

ツイート投稿数の予測と同様に、 k 個の埋め込みベクトルから極性（ポジティブ/ネガティブ）の1または0のバイナリ分類を行う。

試合中に、ユーザが応援するチームに得点が入ったときをポジティブ、対戦相手が点を入れた時はネガティブとし、これを学習データとして、チームごとに予測モデルを学習する。得点および失点シーンの時刻から、1分後までに投稿されたツイートの中の k 個を入力とし、得点シーンであれば出力は1、失点シーンであれば出力は0とする。予測モデルを $estimatePN$ とすると失点き、ポジティブな確率 $p(P) \in \{0, 1\}$ 及びネガティブな確率 $p(N) \in \{0, 1\}$ の予測値を、式 (4) で表す。この予測モデルの構成を図 11 に示す。

$$(p(P), p(N)) = estimatePN(v_1, v_2 \dots v_k) \quad (4)$$

6 実験と評価

6.1 データセットと前処理

TwitterAPI を用いて、プロ野球の試合時間中の実況ツイートを取得した。対象とした試合は、2019/4/14 から 2019/10/10 に開催された日本野球機構（NPB）に所属する全12球団の試合である。実況ツイートの取得に利用したツイートのハッシュタグを表 1 に示す。

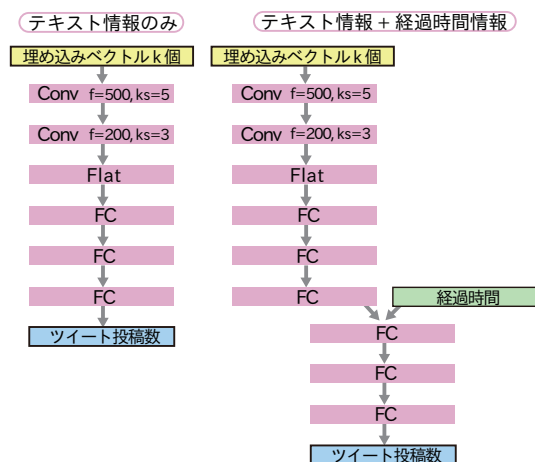


図 10 ツイート投稿数の予測モデルの概略図

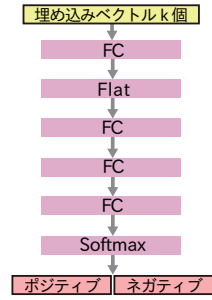


図 11 ツイート投稿数の予測モデルの概略図

表 1 取得ツイートの詳細

球団名	ハッシュタグ
読売ジャイアンツ	#kyojin or #giants
中日ドラゴンズ	#dragons
広島東洋カープ	#carp
東京ヤクルトスワローズ	#swallows or #yakultswallows
阪神タイガース	#hanshin or #tigers
横浜 DeNA ベイスターズ	#baystars
北海道日本ハムファイターズ	#lovefighters
福岡ソフトバンクホークス	#sbhawks
東北楽天ゴールデンイーグルス	#rakuteneagles
埼玉西武ライオンズ	#seibulions
千葉ロッテマリーンズ	#chibalotte
オリックス・バファローズ	#orix_Buffaloes

取得した実況ツイートを、学習データの試合（全718試合/全4,658,002ツイート）とテストデータの試合（全18試合/115,141ツイート）に分割し、これを元に各実験を行った。

ツイートのテキストに対しては、以下の前処理を行った。

- URL, ハッシュタグ, スペース等の除去
- リプライで用いられる@以下の文字の削除
- 数字, アルファベットを半角に変換
- 半角カタカナを全角に変換
- 日本語以外の言語で書かれたツイートの削除

次に、オートエンコーダの入力および出力となる、one-hotベクトルの生成について説明する。学習データにて実況ツイート中の文字種を数えた結果、図 12 のようなヒストグラムとなり、6,669 種類の文字を取得した。この文字種内で出現回数が上位10件のものを表 2 に示す。この中から、上位 d 文字種を使用し、パディング (PAD) と未知文字 (UNK) を加え、先頭から50文字 ($n = 50$) のテキストを、次元数が $(50, d + 2)$ の one-hot ベクトル行列 X に変換した。

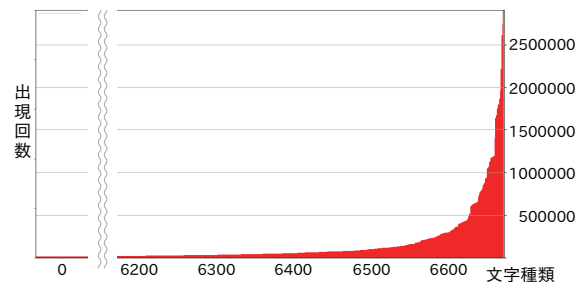


図 12 実況ツイート中の文字種をカウントしたヒストグラム

表 2 文字種の出現頻度（上位 10 件）

文字種	！	い	ー	な	の
出現回数	3062109	2952348	2942625	2490237	2196175
文字種	て	た	っ	か	ん
出現回数	2083158	2009780	2005533	1957574	1849727

6.2 比較手法

ツイート投稿数の予測、ポジティブ/ネガティブの予測を行う実験において、本論文で提案したオートエンコーダを用いたテキストの埋め込み手法と比較する手法について説明する。

比較手法として、テキストの埋め込みに gensim⁴の Word2Vec を用いた。まず、前処理をしたテキストに対して形態素解析を行い、自立語のみ（動詞、形容詞、形容動詞、名詞、副詞、連体詞、接続詞、感動詞）を取り出した。形態素解析は、新語や固有表現に対応できる mecab-ipadic-NEologd 辞書⁵を用いた。自立語のみをコーパスとし、学習データのツイートをを用いて Word2Vec のモデルを学習させた。学習の際、単語の窓サイズは 10、出現回数が 5 未満の単語を取り除き、出力されるベクトルは、オートエンコーダによる埋め込みと同じ 300 次元とした。

6.3 実験手法

6.3.1 オートエンコーダの性能評価

4.3 で提案した、オートエンコーダによるテキストの復元を行った。オートエンコーダのモデル (EmbeddingModel) として CPCPCCL, CGCGFF, GF, bGtCF を用い、それらの性能を比較した。

学習の際には、20%を検証データとして、学習データのツイートを 20,000 個ずつ学習させ、再学習を繰り返した。各学習は、エポック数が 50、バッチサイズ 100、最適化手法は Adam [20]、損失関数は交差エントロピー誤差とした。

最終的に学習されたモデルを、テストデータを使って精度を測り、交差エントロピー誤差で評価した。また、上位 1,000 文字種 ($d = 1,000$) を使用で、EmbeddingModel の CGCGFF, GF, bGtCF に関しては、1500 文字種 ($d = 1,500$) でも実験を行った。

6.3.2 ツイート投稿数の予測

例として、阪神タイガースのファンのツイート投稿数を予測する。ここでは、ツイート学習データのツイートを 10 個 ($k = 10$) ごとに分割し、ツイート 10 個から、そのツイートが含まれる時間窓の投稿数を予測するモデルを学習させる。

ツイート埋め込みベクトルの生成においては、上位 1000 文字種 ($d = 1,000$) を使用した埋め込みベクトルを用いた。学習データにおいてモデルを学習し、学習の際には、20%を検証データとし、各学習はエポック数が 50、バッチサイズ 10、最適化手法は Adam、損失関数を平均二乗誤差とした。オートエンコーダによる EmbeddingModel の CPCPCCL, CGCGFF, GF, bGtCF で生成されたベクトル、Word2Vec による埋め込みベクトル w2v の 5 つの入力において、それぞれツイート量

の予測モデル $estimateC$ と $estimateC'$ で比較を行った。テストデータとした阪神タイガースの 18 試合において、実際のツイート投稿数と予測値との平均平方二乗誤差でそれぞれのモデルを評価した。

6.3.3 極性の予測

本論文では、阪神タイガースのファンのツイートから極性を予測する。得点および失点シーンにおいて、ツイートを 10 個 ($k = 10$) から提案手法で述べた方法でポジティブ/ネガティブをバイナリ分類するモデルを学習し、テストデータで評価を行った。予測されたポジティブの確率が 0.5 以上の場合はポジティブ、0.5 未満の場合はネガティブと判定した。テストデータとした阪神タイガースの 18 試合の得点および失点シーンを対象に、学習データと同様、得点シーンは 1、失点シーンは 0 として、予測された結果を正解率で評価した。

ツイート投稿数の予測精度評価と同様に、ツイート埋め込みベクトルの生成においては、上位 1000 文字種 ($d = 1,000$) を使用した埋め込みベクトルを用いた。学習データにおいてモデルを学習し、学習の際には、ツイート投稿数の予測と同じように、20%を検証データとして、検証データの誤差が低くなった時にモデルを保存し、各学習はエポック数が 50、バッチサイズ 10、最適化手法は Adam、損失関数を交差エントロピー誤差とした。オートエンコーダによる EmbeddingModel の CPCPCCL, CGCGFF, GF, bGtCF で生成されたベクトル、Word2Vec による埋め込みベクトル w2v の 5 つの入力において、モデルの精度を比較をした。

6.3.4 システムの評価

本システムを利用するときは、1 分ごとに k 個のツイートを l セット、計 $k \times l$ 個のツイートを取得し、1 分ごとにそのときの盛り上がりの大きさと極性をリアルタイムに予測する。投稿数の予測値とポジティブの確率は、 l セットの平均で算出する。時間窓のツイートの個数が $k \times l$ 個以下であった場合、投稿数の予測値は正解の値をとり、極性の予測は行わない。

a) 適合率の評価

テストデータとした 18 試合において、ツイート投稿数の予測値に閾値 t_c を設け、予測値が t_c 以上の時間において、極性の予測値が妥当かどうかを評価した。予測されたポジティブの確率を $p(P)$ とするとき、 $p(P) < 0.5$ のときをネガティブ、 $0.5 \leq p(P)$ となるときをポジティブだと予測することとした。ツイート投稿数の予測値が t_c 以上の時間窓に対して、内容を表す出来事のラベルを付与した。この実験において、出来事のラベルは 55 個定義できた。そのラベルの例を表 3 に示す。そして被験者 3 人主観評価によって、ラベルごとにネガティブ、ニュートラル、ポジティブの 3 段階の評価を付け、ラベルが付与された時間窓の正解を定めた。

そして、予測された 2 つの極性について正解との適合率を求めた。本実験では、 $k = 10$ 、 $t_c = 0.45$ として、 l を 1 または 2 で比較した。ツイート投稿数と極性の予測モデルは、先の実験で精度が高かったものを扱った。

b) 再現率の評価

対象とする試合の動画を視聴し、視聴する価値があると考え

4 : <https://radimrehurek.com/gensim/>

5 : <https://github.com/neologd/mecab-ipadic-neologd/>

表 3 試合中の出来事のラベル の例

試合中の出来事のラベル	ラベルに対する極性
攻撃時 代打失敗	ネガティブ
攻撃時 バント成功, チャンスを作る	ニュートラル
攻撃時 盗塁成功, チャンスを作る	ポジティブ
守備時 ホームランを打たれる, 逆転サヨナラ負け	ネガティブ
守備時 リリーフピッチャーに交代	ニュートラル
守備時 ファインプレーの守備	ポジティブ

られる場面を主観評価によって定め、提案手法によって場面が自動的に予測できるのかを検証した。

試合の動画を開始から視聴し始め、視聴する価値があると思われる場面を指定した。対象とした試合は 2019 年 9 月 18 日の阪神タイガース対東京ヤクルトスワローズの試合で、阪神ファンとしての評価を行った。本実験の予測において、 $k = 10$, $l = 2$ とした。そして、この指定した場面それぞれにおいて、本システムを使用することによってその場面を見逃さないようにできるのかの検討を行った。

6.4 実験結果

6.4.1 オートエンコーダの性能評価

オートエンコーダの性能を評価において、結果を表 4 に示す。結果から、 $d = 1,000$, $d = 1,500$ のときの両方において、EmbeddingModel CGCGFF が一番性能が高い結果になった。このモデル ($d = 1,000$) での、テキストの復元例を表 5 に示す (パディング部分は省略)。

6.4.2 ツイート投稿数の予測精度

ツイート投稿数の予測精度は、それぞれの EmbeddingModel において、表 6 の通りとなった。結果、GF での埋め込みベクトルと経過時間情報を入力としたモデルが一番誤差が小さかった。

6.4.3 極性の予測精度

極性 (ポジティブ/ネガティブ) の予測精度 (正解率) は、それぞれの EmbeddingModel において、表 7 の通りとなった。結果、GF が一番正解率が高かった。また、CPCPCCL, w2v に関しては、全ての予測がポジティブと判定してしまっているため、学習が全く進んでいない。

表 4 オートエンコーダの精度 (交差エントロピー誤差)

Embedding Model	文字種類数 d	
	1000	1500
CPCPCCL	6.559	N/A
CGCGFF	0.1074	0.1188
GF	0.1151	0.1351
bGtCF	0.1084	0.6398

表 5 オートエンコーダでのテキストの復元例

入力/出力されたテキスト	
入力	あっぱな一ハラハラしたわ UNK
出力	あっぱな一ハラハラしたわ UNK
入力	おっしゃー初戦とった UNK ナイスゲーム UNKUNK
出力	おっしゃー初さだった UNK ナイス中一ム UNKUNK
入力	ホント UNK によええわ
出力	ホント左にスええわ

表 6 ツイート投稿数の予測精度 (平均平方二乗誤差)

EmbeddingModel	平均平方二乗誤差	
	テキスト情報のみ	テキスト情報+経過時間情報
CPCPCCL	0.3877	0.3877
CGCGFF	0.2118	0.3877
GF	<u>0.2075</u>	<u>0.1982</u>
bGtCF	0.2094	0.3877
w2v	0.2533	0.3877

表 7 ポジティブ/ネガティブの予測精度

EmbeddingModel	正解率
CPCPCCL	0.7937
CGCGFF	0.8586
GF	<u>0.9063</u>
bGtCF	0.8998
w2v	0.7937

6.4.4 システムの評価

評価するシステムのモデルは、ツイート投稿数と極性の予測に使う EmbeddingModel は共に「GF」、ツイート投稿数は「テキスト情報+経過時間情報」を入力とするモデルとした。

a) 適合率の評価

ツイート投稿数の予測値 $t_c \geq 0.45$ の場面の極性予測の結果は、以下の表 8 の通りになった。 l の数値にかかわらず、ポジティブとネガティブの場面に関してはおおよそ正しく予測できた。また、 $l = 1$ のときよりも $l = 2$ の方が精度が高く、10 ツイートからの予測値 1 つだけを用了場合より、予測値 2 つを平均した値の方が有用であった。

b) 再現率の評価

2019 年 9 月 18 日の試合において、試合開始からの時間に対する、ツイート量と予測されたツイート量、予測されたポジティブの確率のグラフを図 13 に示す。予測値に対する閾値を適切に定めることによって、価値ある場面と評価した部分において、開始時間の 1 分後には「いいところ」として通知できるようになっていることがわかる。

6.5 考察

オートエンコーダの性能評価において、EmbeddingModel CPCPCCL の畳み込みと最大プーリングを含んだモデルは、学習中も誤差が下がりづらかった。アルファベットと記号のみのテキストには効果がある可能性はあるが、文字種が多い場合は、最大プーリングのように一部の特徴を強調させる手法は精度をあげることは繋がらないと考えられる。しかし、それぞれのモデルを比較して、オートエンコーダによるテキストの復元精度が高かったとしても、ツイート投稿数や極性 (ポジティブ/ネガティブ) の予測精度が良い結果になるわけではないことが

表 8 ツイート投稿数の予測値 $t_c \geq 0.45$ の場面の極性予測

		主観評価された極性			適合率
		ネガティブ	ニュートラル	ポジティブ	
$l = 1$	ネガティブ	12	1	8	0.5714
	予測 ポジティブ	6	3	66	0.8800
$l = 2$	ネガティブ	11	0	3	0.7857
	予測 ポジティブ	2	2	54	<u>0.9310</u>

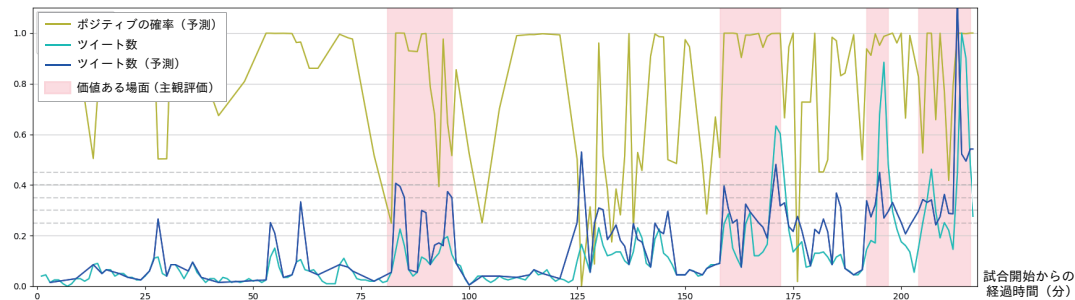


図 13 2019/9/18 の試合の例

わかった。このことから、オートエンコーダの性能が悪くても、本研究の目的である盛り上がりの予測ができる可能性もあると考えられる。

ツイート投稿数の予測に関して、特にツイート投稿数が急激に増える盛り上がりの山となる部分は、かなり正確に予測できているとみられる。これは、ツイート投稿数が多い時間帯は、投稿されるツイートのテキストのパターンが決まっていたり単純なものになるからだと考えられる。また、システムの評価の実験から、ツイート投稿数が多いと予測された場面において、極性の判定はポジティブで約 93%、ネガティブで約 93%と、高精度に予測可能だということがわかった。

7 ま と め

本研究では、プロ野球中継を対象に、ユーザにとっての「いいところ」を通知するシステムの開発を行った。試合に対する実況ツイートから、文字レベルの処理で特徴を抽出し、機械学習を用いて埋め込みベクトルを生成する方法で、ファンの盛り上がりを予測する手法を提案した。

今回の実験では、システムの評価として、一つの試合でしか再現率の評価は行っていないため、さらなる評価が必要である。また、試合の流れやユーザの好みに合わせた、予測値に対する閾値の調整が今後の展望として挙げられる。そして、実際にシステムを使ったユーザ評価を行うことを検討したい。

文 献

- [1] 中澤昌美, 帆足啓一郎, 小野智弘, “Twitter によるテレビ番組重要シーン検出及びラベル付与手法”, 第 73 回全国大会講演論文集, 2011 1 pp.517–519, 2011.
- [2] 山内嶺, 北山大輔, “ダイジェスト映像自動生成のための観点入れ替わりに基づいた特徴的シーン抽出”, DEIM Forum 2014, F4-2, 2014.
- [3] 佃洗撰, 山本岳洋, 中村聡史, 田中克己, “映像に付与されたコメントを用いた登場人物が注目されるシーンの推定”, 情報処理学会論文誌, Vol.52 No.12 pp.3471–3482, 2011.
- [4] Amit Sheth, “Citizen Sensing, Social Signals, and Enriching Human Experience”, IEEE Internet Computing Magazine, 13, 4, pp.87–92, 2009.
- [5] Takeshi Sakaki, Makoto Okazaki and Yutaka Matsuo, “Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors”, WWW2010, 2010.
- [6] Eiji Aramaki, Sachiko Maskawa, Mizuki Morita, “Twitter Catches The Flu: Detecting Influenza Epidemics using Twitter”, Proc. 2011 Conf. on Empirical Methods in Natural Language Processing, pp.1568–1576, 2011.
- [7] 榊剛史, 松尾豊, “ソーシャルセンサとしての Twitter : ソーシャルセンサは物理センサを凌駕するか?”, 人工知能学会誌 (特集) Twitter とソーシャルメディア, pp.67–74, 2012.
- [8] Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md. Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary, “Twitter Trending Topic Classification”, 11th IEEE International Conference on Data Mining Workshops, 2011.
- [9] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye and Zhe Zhao, “Predicting Bursts and Popularity of Hashtags in Real-Time”, SIGIR ’ 14, 2014.
- [10] Senzhang Wang, Zhao Yan, Xia Hu, Philip S. Yu, Zhoujun Li and Biao Wang, “CPB: a classification-based approach for burst time prediction in cascades”, Knowledge and Information Systems, October 2016, Vol. 49, Issue 1, pp.243–271, 2015.
- [11] Wang Ling, Tiago Luis, Luis Marujo, Ramon Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso, “Finding function in form: Compositional character models for open vocabulary word representation”, arXiv preprint arXiv:1508.02096, 2015.
- [12] Xiang Zhang, Junbo Zhao, and Yann LeCun, “Character-level convolutional networks for text classification”, In Advances in Neural Information Processing Systems, pp.649–657, 2015.
- [13] Soroush Vosoughi, Deb Roy, “Tweet Acts: A Speech Act Classifier for Twitter”, Proceedings of the Tenth International AAAI Conference on Web and Social Media (ICWSM 2016), 2016.
- [14] 佐藤孝斗, 折原良平, 清雄一, 田原康之, 大須賀昭彦, “文字レベル深層学習によるテキスト分類と転移学習”, 人工知能学会研究会資料, SIG-FPAI-B505-03, 2016.
- [15] Soroush Vosoughi, Prashanth Vijayaraghavan and Deb Roy, “Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder”, SIGIR ’ 16, 2016.
- [16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, “Distributed Representations of Words and Phrases and their Compositionality”, In Proc. NIPS, pp.3111–3119, 2013.
- [17] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory”, Neural computation, Vol.9 No.8 pp.1735–1780, 1997.
- [18] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches”, arXiv preprint arXiv:1409.1259, 2014.
- [19] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, arXiv preprint arXiv:1412.3555, 2014.
- [20] Kingma, Diederik and Jimmy Ba, “Adam: A method for stochastic optimization”, arXiv preprint, arXiv:1412.6980, 2014.