ドラマ振り返りサーチ・インタフェース

永井 春佳† 牛尼 剛聡‡

†九州大学芸術工学部 〒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: †nagai.haruka.439@s.kyushu-u.ac.jp, ‡ushiama@design.kyushu-u.ac.jp

あらまし 近年，TV ドラマの放送中に番組への感想や考察が Twitter に多く投稿されており，それらは実況ツイートと呼ばれる．ドラマを視聴した後にユーザが興味を持ったシーンについて実況ツイートを検索して，他の視聴者の感想を楽しんだり，別の見方からの考察を深めたりすることができる．しかし，現在の Twitter の検索機能では，基本的に対象とするシーンのキーワードで検索してもキーワードを含むツイートが時間順に提示されるだけであり，ユーザが興味を持つシーンに対する実況ツイートを見つけるのに手間がかかるなどの問題がある．本研究では，ユーザが入力したキーワードに基づいて，ドラマの放送時間中に投稿された実況ツイートを対象として時間ごとのキーワードとの類似度の分布を可視化して提示し，効率的にドラマを振り返ることのできるインタフェースを提案する．

キーワード SNS, Twitter, ソーシャルビューイング，実況ツイート, TV ドラマ，振り返り

1. はじめに

近年，SNS(ソーシャルネットワーキングサービス)が世界的に普及しており，中でも Twitter は、意見の発信やコミュニケーションなど，様々な目的で日常的に利用されている．

そうした中で，TV 番組を視聴しながら実況ツイートを投稿する，ソーシャルビューイングが活発に行われている．実況ツイートとは，TV 番組を視聴しながら投稿された，その番組への感想や意見などのリアルタイムの反応を含むツイートのことである．SNS ユーザは，実況ツイートを投稿することで、普段 TV 番組を視聴しながら家族や友人と感想を言い合ったり，意見交換をするような感覚で，Twitter を介して同じ番組を他のユーザと一緒に盛り上がったり考察を深めたりすることができる．

ソーシャルビューイングでは，実況ツイートを投稿するユーザが楽しいだけではなく，投稿せずに閲覧するだけのユーザも楽しむことができる．その中で本研究では，TV ドラマ視聴後の実況ツイートの閲覧に着目する．TV ドラマ視聴後に視聴者は，印象に残ったシーンや理解ができず疑問を持っているシーンなど，他の人がどのように考えているのか知りたいシーンがある場合がある．そうしたとき，そのシーンに対する他の視聴者の実況ツイートを見て，同様の意見を持っているツイートに共感をしたり，自分とは違う観点のツイートを見て新たな知見を得ることができる．

このように，実況ツイートを閲覧してドラマの内容を振り返り，余韻に浸ることでより深くその番組を楽しむことができるのではないかと考えられる．しかし，TV 番組への実況ツイートは数多く投稿されており，

図 1 実況ツイート閲覧のイメージ

それらの中から目的のシーンに対する実況ツイートを発見する必要がある．本論文では，このようにドラマ視聴後にドラマの内容を振り返るために，特定のシーンに対する実況ツイートを探すことを「ドラマ振り返りサーチ」と呼ぶ．そして本研究では，ドラマ振り返りサーチのためのインタフェースの開発を目的とする．

現在の Twitter の検索機能では，他者が投稿した実況ツイートは，ハッシュタグを用いて取得できる．ハッシュタグとは，投稿を特定のトピックで分類するための「#」から始まるタグである．実況ツイートの多くには対象となる番組のタイトルやその略称がハッシュタグとして付けられているため，そのハッシュタグで検索を行うことで，他の人の実況ツイートを閲覧できる．しかし，この検索機能は，放送されているシーンに対する投稿をリアルタイムに閲覧するには適しているが，TV 番組を視聴した後に序盤のシーンに対する投稿を見たい時や，TV 番組が放映された後に録画視聴する際など，過去の実況ツイートを閲覧する場合には，以

下に示す 3 つの問題がある．

1. TV 番組の実況ツイートは膨大であり，ツイート検索機能によって得られた検索結果をひとつずつ確認し，ユーザが対象とするシーンに対するツイートまで遡ることは多大な労力を必要とする．
2. 実況ツイートの内容は簡潔なものも多く，ただ「すごい！」のような投稿の場合，どのシーンに対してのコメントなのか，そのツイートだけを見ても分かりづらい．
3. ドラマのタイトルや略称のハッシュタグとともに対象のシーンの特徴となる語をキーワードとして検索することでツイートの絞り込みを行うこともできるが，そのキーワードに一致するツイートしか表示されないため，キーワードが曖昧な場合，思うようにツイートを得ることができない．

本研究では，このような問題を踏まえ，効率的にドラマ振り返りサーチのできるインタフェースを提案する．具体的には，ドラマ放送時間に投稿された，その番組に関するハッシュタグのついたツイートを対象として，ユーザが入力したキーワードとの時間ごとの類似度を提示し，その情報をもとにユーザが選んだ時間のツイートを表示するインタフェースを開発する．

2. 関連研究

これまでにも，TV 番組と Twitter の実況ツイートに関する研究は数多く行われている．

中澤ら[1]は，録画したテレビ番組の効率的な視聴を目的として，テレビ番組に関連するツイートから重要シーンを検出し，各シーンでの主要人物とイベントの内容を推定して，シーンを表すラベルとして付与する手法を提案している．大田垣ら[2]は，ソーシャルビューイングのストリームの盛り上がりに着目し，盛り上がった内容をユーザに効果的に提示する集約化手法を提案している．牛島ら[3]は，Twitter を利用した TV ドラマのソーシャルビューイングに着目し，ドラマ放送中に投稿された実況ツイートを用いて，ドラマの時系列でのシーンの特徴の抽出を行うことで,「展開パターン」により TV ドラマの特徴付けを行っている．

また，実況ツイートの感情に着目した研究では，若井ら[4]はテレビで放送されている映画に焦点を当て，ツイートの感情を時系列に抽出して分析を行っており，山内ら[5]は効率的な録画番組視聴の支援を目的として，実況ツイートを用いた視聴者の感情に基づくテレビ番組のインデキシングを提案している．

これらの研究では，実況ツイートに基づいてシーンの特徴やツイートの感情を抽出して可視化している．本研究ではさらに，ユーザが入力したキーワードに対しての盛り上がりを抽出し，時系列順に提示する．

また，佃ら[6]は，ニコニコ動画に投稿されたコメントを用いて，映像に登場する人物が視聴者の注目を集めているシーンの推定と各シーンにおける各登場人物の活躍の度合いの推定を行う手法を提案している．この手法では，登場人物のみに着目し，注目シーンの推定を行なっているが，本研究では人物名に限らず，ユーザが入力したキーワードを対象とする．

他の視聴者の実況ツイートの閲覧を目的とする研究として，松岡ら[7]は番組に関する大量のツイートの中からユーザが共感できる「波長の合う人」が発するツイートのみをユーザに提示することにより，ユーザが快適に満足度の高いソーシャルビューイングを実現する手法を提案している．この研究では，ツイートを投稿するタイミングからユーザの類似度を求めている点で，本研究とは異なる．

3. アプローチ

本研究では，TV ドラマを効率的に振り返るために，ユーザが単純な操作で目的のシーンに関する実況ツイートを閲覧できるインタフェースの開発を目的とする．

TV ドラマの実況ツイートは，対象とするドラマを視聴したユーザのリアルタイムの反応であり，その時間に放送されているシーンの内容が強く反映されていると考えられる．中澤ら[1]や牛島ら[3]の関連研究でも，実況ツイートを利用して視聴者の反応を抽出することで，実際のシーンの特徴を取得している．

そのため，本研究ではキーワードと関連したシーンには，キーワードと類似性が高い実況ツイートが多く投稿されていると仮定し，実況ツイートの内容を用いて，ユーザが入力したキーワードと類似するシーンの検出を行う．具体的には，ドラマ放送時間中に投稿された実況ツイートのタイムラインを一定の時間区間で分割し，それぞれの時間区間内のツイートとキーワードに対する類似度として提示することによって，そのドラマのキーワードに関する盛り上がりの推移を可視化する．システムの概略図と，類似度の可視化の概念図を図 2,3 に示す．

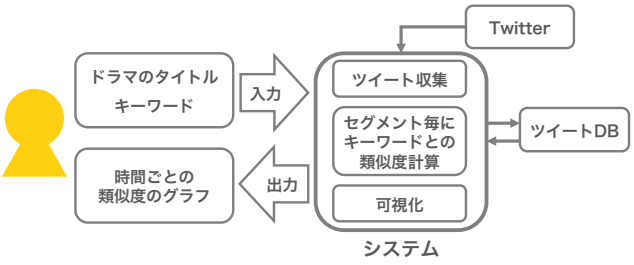

図 2　システムの概略図

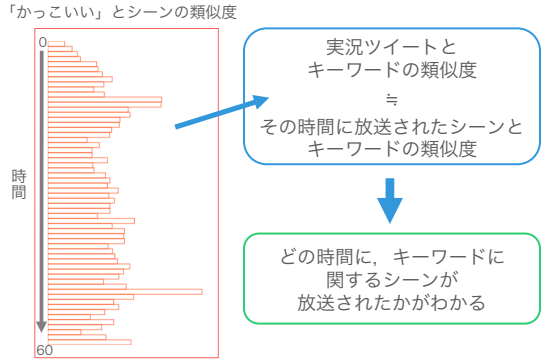

図 3　類似度の可視化の概念図

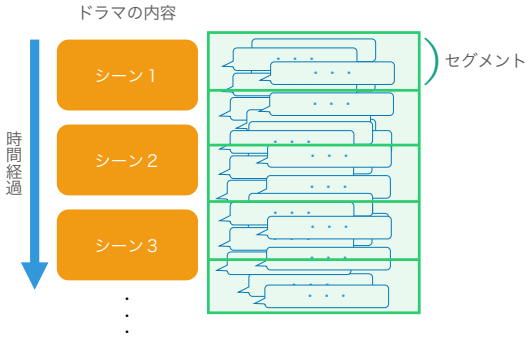

図 4　セグメンテーションの概念図

また，本論文で提案する手法の手順を以下に示す．

1. TwitterAPI を利用して，TV ドラマに対する実況ツイートを収集する．具体的には，対象とするドラマの放送時間中に投稿された，そのドラマのタイトルのハッシュタグを含むツイートを収集し，ツイート DB に保存する．ただし，リツイートとリプライは除外する．
2. ユーザが指定したドラマのツイートをツイート DB から取得し，時間ごとのツイートの特徴を得るために，収集したツイートのタイムラインを時間区間で分割する．分割されたツイートの集合をセグメントと呼ぶ．
3. セグメント内のツイートに対して形態素解析を行う．
4. セグメント内のツイートおよびキーワードのベクトル化を行う．
5. セグメント内のツイートとキーワードから得られたベクトルの cos 類似度を計算し，各セグメントとキーワードの類似度を算出する．
6. セグメントごとの類似度をグラフ化して，ユーザに提示する．

4. 提案手法

4.1. ツイートタイムラインのセグメンテーション

提案手法では，ドラマの進行に合わせた単位時間ごとのキーワードに関する盛り上がりを推定し，可視化して提示することを目的とする．そこで，収集した実況ツイートのタイムラインを一定の時間区間で分割する．分割された時間区間に含まれるツイートの集合をセグメントと呼び，各セグメントには，その時間に放送されたシーンの特徴が強く反映されていると考える．ツイートのセグメンテーションの概念図を図 4 に示す．

4.2. Word2Vec モデルの作成

セグメントとキーワードの類似度を算出するために，Word2Vec[8]を用いてセグメント内のツイートとキーワードのベクトル化を行う．Word2Vec とは，テキスト処理を行うための 2 層からなるニューラルネットワークを利用した単語のベクトル化手法のことである．コーパスを利用してニューラルネットワークの重みを学習させることで単語のベクトル表現を獲得できる．

4.2.1. コーパス

本研究では，Word2Vec 学習用のコーパスとして，日本語 Wikipedia コーパスとドラマの実況ツイートコーパスの 2 種類を使用し，それぞれの性能を比較する．

日本語 Wikipedia コーパスは，Wikipedia が提供している全文データ[9]をダウンロードし，Wikiextractor[10]を利用して XML をパースしたものである．また，ドラマの実況ツイートコーパスは，2019 年 7-9 月期に放送された 16 番組，125 話と 2019 年 10-12 月期に放送された 15 番組，111 話のドラマに対する実況ツイートを集めたものである．具体的には，対象ドラマの放送時間に投稿された，ドラマのタイトルのハッシュタグのついたツイートを，リツイートとリプライを除いて収集した．

このコーパスを使用することで，ドラマの実況ツイート特有の表現や語彙をよく学習し，実況ツイートに対する検索に適した分散表現モデルとなることを期待する．

4.2.2. 前処理

Word2Vec の学習をする前に，Wikipedia コーパス，ドラマの実況ツイートコーパス両方に対して前処理を行う．まず，収集に使用したドラマのタイトルのハッシュタグと URL は，ツイートの特徴を得る際にノイズとなり得るため，テキストから削除する．また，ドラマのタイトル以外のハッシュタグは，その時間に放送されているシーンに出演している役者などの情報を含んでおり，シーンの特徴となり得ると考え，そのまま使用した．そして，オープンソースの日本語形態素解

析エンジンである MeCab を用いて，コーパスの全文書を形態素に分解する．MeCab の辞書には，固有表現や web 上でよく使われる崩れた表記，新語などを幅広くカバーできる mecab-ipadic-NEologd[11]辞書を用いる．

また，分かち書きした単語の中で，名詞，動詞，形容詞，副詞のみを使用し，活用されている語についてはその語の原形を使用する．

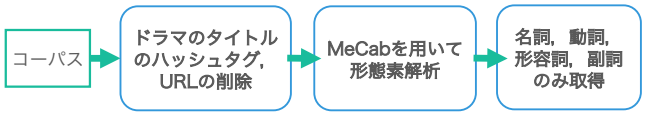

図 5　前処理の流れ

以上の手順により得られたコーパスを用いて，Python のライブラリである gensim[12]を利用して Word2Vec の学習を行う．今回は，2 種類のコーパス共に次元数を 200，ウィンドウサイズを 10 として学習を行った．

4.3. 類似度の算出

セグメント内のツイートとキーワードのベクトル化を行い，セグメントごとの類似度を算出する．提案手法では，単にツイートの数ではなく，そのキーワードに関する盛り上がり，すなわち類似度の高いツイートの密度で盛り上がりを決定していくため，より良い手法を見つけるために 3 つの手法を用いて各セグメントの類似度を算出し，比較を行う．ここで，ツイートのベクトル化については，ツイートを分かち書きし，それぞれの単語のベクトルを求め，それらの平均をとったものをツイートのベクトルとする．

4.3.1 手法 1　ツイートごとの類似度の平均(Word2Vec)

セグメント内のツイートとキーワードを，上記の方法で学習した Word2Vec モデルを用いてベクトル化する．そして，ツイートごとにキーワードとの cos 類似度を計算し，その平均をセグメントの類似度とする．ここで，$\boldsymbol{q}$はキーワードのベクトル，$\boldsymbol{S}$はセグメントのベクトル(ツイートのベクトルの集合)，$\boldsymbol{t}$はツイートのベクトルを表す．

$$\mathrm{sim}_1(\boldsymbol{q},\boldsymbol{S}_i)=\frac{1}{|\boldsymbol{S}_i|}\sum_{t\in S}\mathrm{sim}_t(\boldsymbol{q},\boldsymbol{t}) \tag{1}$$

4.3.2 手法 2　セグメントベクトルの類似度(Word2Vec)

手法 1 と同様にセグメント内のツイートおよびキーワードを Word2Vec モデルでベクトル化する．そして，セグメント内のツイートのベクトルの平均を求め，得られたセグメントベクトルとキーワードのベクトルの cos 類似度をセグメントの類似度とする．

$$\mathrm{sim}_2(\boldsymbol{q},\boldsymbol{S}_i)=\mathrm{sim}_t\left(\boldsymbol{q},\frac{1}{|\boldsymbol{S}_i|}\sum_{t\in\boldsymbol{S}_i}\boldsymbol{t}\right) \tag{2}$$

4.3.3 手法 3　TF-IDF を用いてベクトル化

TF-IDF を用いて，セグメントのベクトル化を行う．TF-IDF とは，文書内に出現する単語について，文書の出現頻度(TF)と単語の逆文書頻度(IDF)から，その単語の重要度を算出する手法である．ある文書d_jに出現する単語t_iについて考える場合，出現回数をfとすると，TF 値は以下の式から算出できる．

$$tf(t_i,d_j)=\frac{f(t_i\in d_j)}{\sum_{t_k\in d_j}f(t_k\in d_j)} \tag{3}$$

また，ある文書集合における単語t_iについて考える場合，総文書数をN，dfを単語t_iが出現する文書数とすると，IDF 値は以下の式から算出できる．

$$idf(t_i)=\log\left(\frac{N}{df(t_i)+1}\right) \tag{4}$$

そして，上記で算出した TF 値と IDF 値を掛け合わせることで，単語ごとの重要度を算出することができる．TF-IDF 値は以下の式から算出できる．

$$tfidf(t_i,d_j)=tf(t_i,d_j)\cdot idf(t_i) \tag{5}$$

この手法では，セグメント内のツイートの集合を 1 文書として，セグメント数分の文書を用いて TF-IDF 値を算出し，ベクトル化を行う．そして，キーワードのベクトルとの cos 類似度を計算し，各セグメントとキーワードの類似度を取得する．

4.4. 時間ごとの類似度の可視化

本手法では，得られたセグメントごとの類似度をグラフとして可視化して表示するユーザインタフェースを提供する．ユーザはグラフを見ることで，どの時間帯にキーワードとの関連のあるシーンが放送されていたのかを知ることができ，グラフ上をマウスで指定すると，その時間に投稿された実況ツイートを閲覧できる．

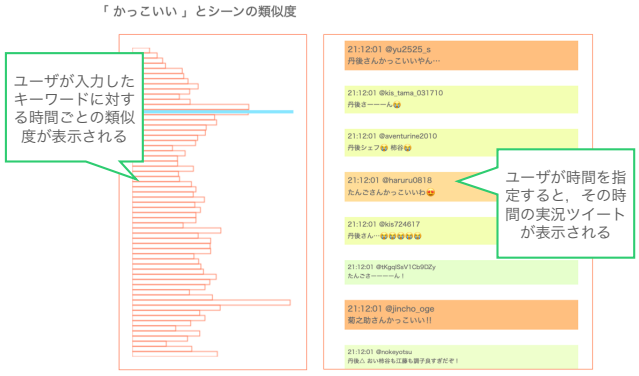

図 6　類似度の可視化のイメージ

5. 使用するコーパスとセグメントの時間区間の決定

提案手法で利用するコーパスやセグメントの時間区間のパラメータを定めるために，以下の 2 種類の実験を行う．

5.1. コーパスの決定

本手法で利用する Word2Vec において，Wikipedia コーパスと，ドラマの実況ツイートコーパスそれぞれで学習したモデルでセグメントとキーワードの類似度を算出し，比較を行った．ここで Wikipedka コーパスで学習したモデルを Wiki_W2V，ドラマの実況ツイートコーパス で学習したモデルを Drama_W2V と表記する．

実験には「あなたの番です-反撃編- 第 11 話」が放送された 2019 年 6 月 30 日 22 時 30 分～23 時 25 分にドラマのタイトルのハッシュタグ(#あなたの番です)をつけて投稿された 17,415 件のツイートを使用した．上記の提案手法に基づいて，収集したツイートを時間区間で分割し，セグメントごとにキーワード「怖い」に対しての類似度を計算した．セグメントを分割する時間幅は 1 分，セグメントの類似度の算出には手法 1 を使用した．2 種類のモデルを用いて得られた，類似度のグラフを図 7 に示す．

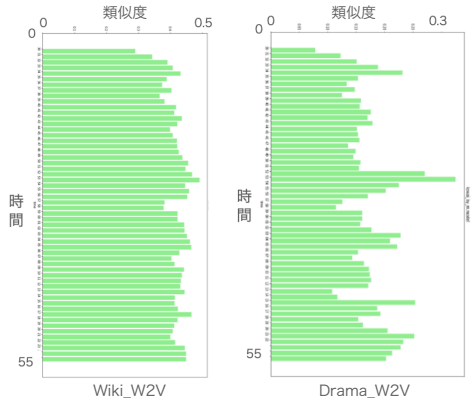

図 7 「あなたの番です-反撃編- 第 11 話」の実況ツイートとキーワード「怖い」の時間ごとの類似度

結果として，Drama_W2V を用いた方が時間によって盛り上がりの差が大きく出ていることがわかる．また,このグラフで山となっている部分を調べたところ,実際にドラマの中でも怖い盛り上がりの部分であり，そのシーンに関するツイートが多く投稿されていた．Wiki_W2V においては，類似度の高いツイートを表示した際に「かわいい」や「凄い」という単語が入ったツイートも上位に分類されていた．Wiki_W2V では実況ツイートでよく使用される形容詞の特徴をあまり学習することができず，どれも似たようなベクトルとなってしまったのではないかと考えられる．

以上より，実況ツイートの特徴抽出には，ドラマの実況ツイートコーパスで学習したモデルを使用する方が適していると考えられる．

5.2. セグメントの時間区間の決定

収集した実況ツイートのタイムラインをセグメントに分割する際の時間区間の適切な間隔について検討する．

各セグメントは，その時間に放送されているドラマのシーンの特徴を表すことを想定している．しかし，セグメントを分割する際の時間区間が長く，同一のセグメント中に異なるシーンに関する実況ツイートが含まれている場合や，逆に時間区間が短く，同一のシーンに関する実況ツイートが複数のセグメントに含まれている場合，適切にシーンの特徴を捉えることができない．そこで，最適な時間区間を決定するために，実際のドラマの各シーンはどれくらいの時間であるのか，調査を行った．

調査に使用したドラマは，2019 年 10-12 月に放送された 3 種類のドラマである．今回は,「ある場所での動作の一区切り」をシーンと定義し，舞台となっている場所が変わったタイミングや，同じ場所であっても，話の流れが変わる何かしらのアクションが起こったタイミングを，シーンの区切りとした．以上の条件でドラマをシーンに分割し，それぞれのシーンが放送されている時間を測定した．その結果を表 1 に示す．

表 1 ドラマのシーン時間測定結果

タイトル	放送時間	シーン数	シーンの平均時間
ドクターX～外科医・大門未知子～ 第9話	2019/12/12 21:00 ~ 21:54	40	1分8秒
モトカレマニア 第9話	2019/12/12 22:00 ~ 22:54	46	0分59秒
グランメゾン東京 第11話	2019/12/29 21:00 ~ 22:24	61	1分11秒
			1分6秒

どのドラマにおいても，シーンの平均時間は 1 分前後となった．そのため，セグメントを分割する時間区間は 1 分が適切であると考えた．

6. 実験・評価

6.1. セグメントの類似度算出手法の比較

上記の提案手法において，セグメントの類似度を求める際の手法 1～3 について,それぞれの手法を用いて類似度を求め，比較を行った．具体的には，3 種類のドラマに対し，合計 22 種類のキーワードについて，実際にセグメントごとに実況ツイートの内容を読んで，内容がどれくらいそのキーワードと類似しているかを

0〜10 の範囲で人手でラベル付けしたものを用意し，正解データとした．そして，この正解データとそれぞれの手法により得られた類似度を，最大値が 1 となるように正規化を行い，誤差を求めた．ここで，手法 1,2 で使用する Word2Vec のコーパスはドラマの実況ツイートコーパスであり，セグメント分割の際の時間区間は 1 分とした．実験に使用するキーワードは，各ドラマの実況ツイートの中でよく使用されている形容詞，名詞からそれぞれ 7〜8 つの語を選定した．比較に使用したドラマの概要と，キーワードを表 2，3 に示す．

表 2　比較に使用したドラマの概要

タイトル	ツイート取得日時	総ツイート数
ドクターX〜外科医・大門未知子〜　第9話	2019/12/12 21:00 ~ 21:54	6,221
モトカレマニア　第9話	2019/12/12 22:00 ~ 22:54	4,058
グランメゾン東京　第11話	2019/12/29 21:00 ~ 21:54	17,550

表 3　比較に使用したキーワード

ドクターX 〜外科医・大門未知子〜 第9話	モトカレマニア 第9話	グランメゾン東京 第11話
辛い	可愛い	かっこいい
面白い	嬉しい	美味しい
怖い	かっこいい	高い
大門	好き	すごい
川谷	さくら	可愛い
原	マコチ	リンダ
痔	ユリカ	尾花
		料理

今回，比較を行った中で最も誤差の平均が小さかったのは「ドクターX〜外科医・大門未知子〜」のキーワード「原」において，手法 3 を用いて類似度を算出した場合である．得られた時間ごとの類似度と正解データとの誤差のグラフを図 8 に示す．

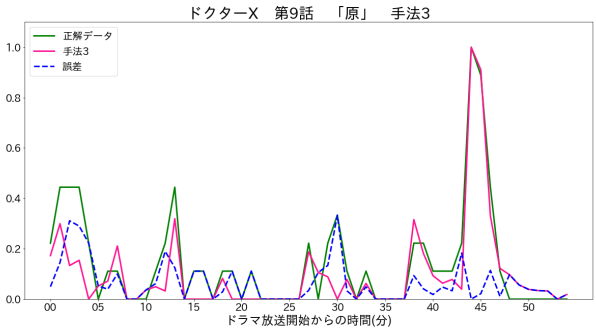

図 8　「ドクターX〜外科医・大門未知子〜」，キーワード「原」における正解データと手法 3 により得られた類似度，それらの誤差のグラフ

キーワードの「原」は主人公ではない登場人物の名前である．主人公と違い，出演するシーンが限られているため，出演したタイミングのみに「原」というワードを含むツイートが多く投稿され，TF-IDF でうまく特徴を捉えることができたのではないかと考えられる．

Word2Vec モデルを用いた手法 1,2 については，「グランメゾン東京」のキーワード「かっこいい」で最も正解データとの誤差の平均が小さくなった．時間ごとの類似度，誤差のグラフを図 9,10 に示す．どちらも，12 分，49 分あたりの盛り上がりを検出することができていた．

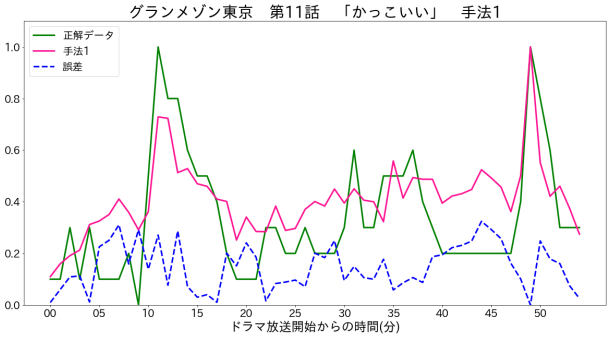

図 9　「グランメゾン東京」，キーワード「かっこいい」における正解データと手法 1 により得られた類似度，それらの誤差のグラフ

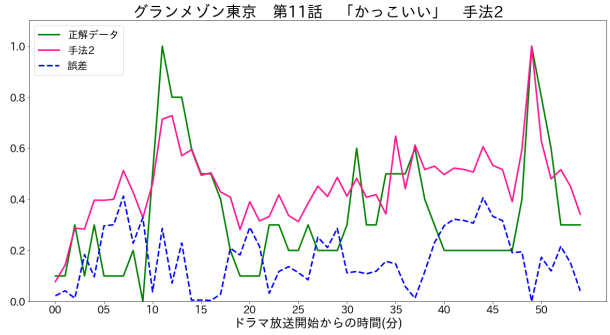

図 10　「グランメゾン東京」，キーワード「かっこいい」における正解データと手法 2 により得られた類似度，それらの誤差のグラフ

全てのキーワードにおける，各手法により得られた類似度と正解データの誤差の平均を表 4 に示す．結果として，手法 3 を用いて類似度を算出した結果が最も正解データとの誤差が小さくなった．

表 4　各手法により得られた類似度と正解データの誤差の平均

手法 1	0.295
手法 2	0.290
手法 3	0.230

手法 1 と 2 はほとんどのキーワードにおいて得られた類似度の差が小さく，セグメントの類似度を求める際，各ツイートのベクトルとキーワードのベクトルの類似度を求め平均する場合と，各ツイートのベクトルを平均してセグメントベクトルとし，得られたセグメントベクトルとキーワードのベクトルの類似度を算出する場合では，結果にほとんど差が見られないことがわかった．

また，手法 1，2 を用いて類似度を算出した際の正解データとの誤差について，実況ツイートへの出現数が少ないキーワードほど誤差が大きくなる傾向があった．多くのツイートに含まれているキーワードはうまく学習することができたが，出現数の少ないキーワードについては，あまり学習ができずに特徴を捉えることができなかったのではなかったのではないかと考えられる．

今回の実験では，手法 3 を用いた場合に最も正解データとの誤差が小さくなるキーワードが多かったが，半分のキーワードについては手法 1 または 2 を用いて類似度を算出した方が誤差が小さくなったため，手法 3 が最も適した手法であるとは断言できない．

Word2Vec モデルを用いた手法 1，2 については，実況ツイートに含まれるキーワードの数が少ないほど，正解データとの誤差が大きくなる傾向も見られたため，もう少し学習データとなる実況ツイートを増やすなど，他の手法を検討する必要があると考えられる．

6.2. インタフェースのユーザビリティの評価

提案手法の有効性を評価するため，20 代の男女 11 名に対し，上記の提案手法により開発した「ドラマ振り返りサーチ・インタフェース」を使用してもらい，アンケートを行った．評価に使用した実況ツイートの対象ドラマは「あなたの番です-反撃編-　第 11 話」,「ドクターX～外科医・大門未知子～　第 9 話」,「グランメゾン東京　第 11 話」である．被験者には好きなドラマに対してキーワードをいくつか入力し，実況ツイートの閲覧を行った後に，アンケートに答えてもらった．セグメントの類似度の算出には手法 1 を用いた．アンケートの設問の内容を以下に示す．

① 本システムを使用してみて，表示されたグラフは，シーンの特徴を表すことができていると思いますか？（キーワードに対するグラフの盛り上がりの度合いが，実際の実況ツイートの内容と合っていると思いますか？）（1：全く思わない，5：とても思う）
② 普段 Twitter で実況ツイートを閲覧する時と比べ，特定のシーンへの実況ツイートを探しやすいと思いましたか？（1：全く思わない，5：とても思う）
③ 使いやすさはどうでしたか？（1：使いづらい,5：使いやすい）
④ 画面の見やすさ(グラフや実況ツイートの表示表方)はどうでしたか？（1：見づらい,5：見やすい）
⑤ 本システムは，ドラマの振り返りに役立つと思いますか？（1：全く思わない，5：とても思う）
⑥ 実際にこのシステムがあった場合，使用してみたいと思いますか？（1：全く思わない，5：とても思う）

得られたアンケート結果を図 11～16 に示す．

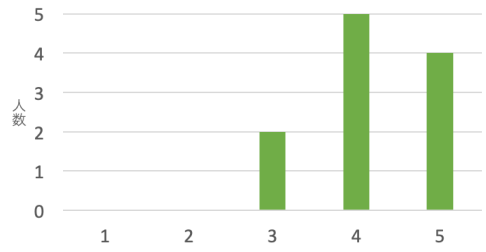

図 11　設問①に対する回答

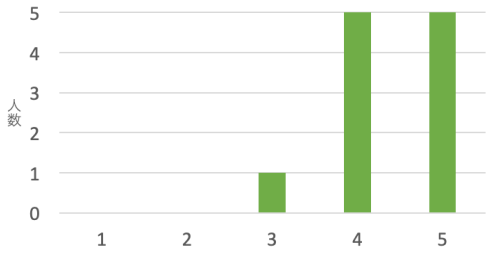

図 12　設問②に対する回答

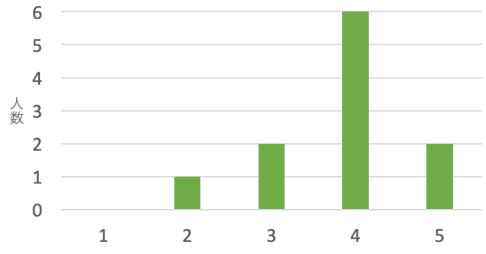

図 13　設問③に対する回答

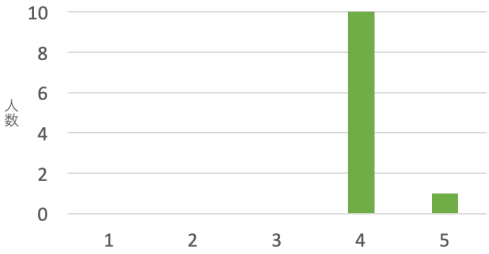

図 13　設問④に対する回答

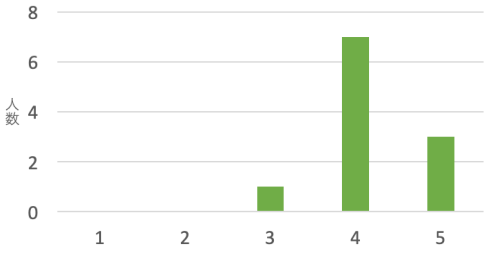

図 15　設問⑤に対する回答

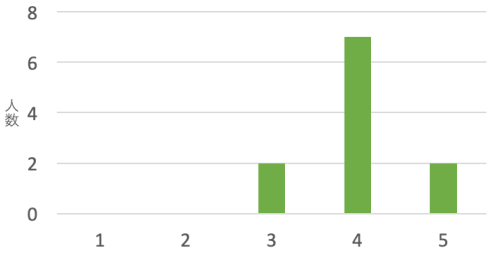

図 16　設問⑥に対する回答

実況ツイートの可視化方法に関する設問①，②では 81.9%以上が「とても思う」または「まあ思う」と回答した．これにより，提案手法により得られた時間ごとのグラフは実際のシーンの特徴を捉えることができており，このインタフェースが特定のシーンへの実況ツイートを探すのに適していることが確認できた．

設問③，④では実際のインタフェースの使いやすさや画面の見やすさについて質問した．多くの被験者が使いやすい，見やすいという回答であったが，中には「少し使いづらい」という意見もあり，さらに使いやすいインタフェースとなるよう，ユーザが実況ツイートを見たい時間をより簡単に指定できるようにするなどの改善の余地がある．

更に，設問⑤，⑥では 81.9%以上が「とても思う」「まあ思う」と回答し，本システムが実際にドラマの振り返りに有用であることがわかった．

7. まとめ

本論文では，TV ドラマを振り返るために，効率的に目的のシーンに対する実況ツイートを閲覧できるインタフェースを提案した．本インタフェースでは，TV ドラマ放送時間中に投稿された実況ツイートを時間区間で分割してセグメントとし，各セグメント内のツイートとキーワードの類似度を算出して，そのドラマのキーワードに関する盛り上がりの推移として可視化を行う．

評価実験の結果から，各セグメントの類似度の算出について，TF-IDF を用いてツイートとキーワードをベクトル化し，セグメントの類似度を算出することで，実際の TV ドラマにおける盛り上がりと近い結果を得ることができた．しかし，キーワードによっては誤差が大きくなるものもあり，より精度の高い類似度算出方法を検討する必要があると考える．

また，ユーザにドラマ振り返りサーチ・インタフェースを実際に使用してもらった結果，本システムは実際のシーンの特徴を捉えることができており，ドラマの振り返りに有効であるということが確認できた．

今回実況ツイートをセグメントに分割する際，時間区間で区切るだけであったが，シーンの放送時間が複数のセグメントにまたがっている場合シーンが分断されてしまい，特徴がセグメントに現れない可能性がある．他にも，TV ドラマの放送時間中には CM の時間が含まれており，その時間に投稿された実況ツイートはシーンの特徴を捉える際にノイズとなり得るが，今回は考慮できていない．これらの問題を踏まえ，今後更に精度を向上させていくことが必要である．

参　考　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”,大 73 回全国大会講演論文集,2011,1,pp.517-519,2011

[2] 大田垣翔,牛尼剛聡,角谷和俊“ソーシャルビューイングにおける盛り上がりの効果的な提示のためのツイート集約化手法“,DEIM Forum 2015 G4-4

[3] 牛島実桜,南大智,牛尼剛聡“視聴者はドラマの「どこ」に「どう」反応しているのか？−実況ツイートを利用したドラマのシーン特徴の抽出−“, DEIM Forum 2017 D6-1

[4] 若井祐樹,山本湧輝,熊本忠彦,灘本明代”映画の実況ツイートにおける時系列毎の感情抽出手法の提案”DEIM Forum 2014 E6-1

[5] 山内崇資,林祐樹,中野有紀子”日本語解析による Twitter の感情分析とシーンインデキシングへの応用”,情報処理学会第 75 回全国大会,6R-6

[6] 佃洸摂，中村聡史，山本岳洋，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 3471-3482 (Dec. 2011)

[7] 松岡藍,牛尼剛聡”共感ビューイング：波長の合う人とのソーシャル TV ビューイング”,DEIM Forum 2014 B7-2

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, “Distributed representations of words and phrases and their compositionality”, Neural Information Processing Systems 2013, 3111-3119, 2013

[9] “Index of/jawiki/,” https://dumps.wikimedia.org/jawiki/, 参照 2020-1-30

[10] “WikiExtractor,” https://github.com/attradi/wikiextractor, 参照 2020-1-30

[11] “mecab-ipadic-NEologd : Neologism dictionary for MeCab”, https://github.com/neologd/mecab-ipadic-neologd/, 参照 2020-1-30

[12] “genism,” https://radimrehurek.com/gensim/, 参照 2020-1-30