

実時間と動画時間から面白い動画コメントを抽出する手法とその適用事例

早川 卓弥^{†,a} 土方 嘉徳^{†,a} 西田 正吾^{†,a}

† 大阪大学大学院基礎工学研究科

a) (Family name)@nishilab.sys.es.osaka-u.ac.jp

概要 ニコニコ動画を始めとして、動画の再生に同期した動画コメントの投稿・再生が可能な動画共有サービスが広がっている。本研究では、動画コメントから有益なコメントを抽出する手法についての提案を行う。本稿では有益なコメントを“ユーザが面白い、笑えると感じるコメント”と定義し、これを抽出するためにコメントを動画時間と実時間という2つの時間軸から分析する。分析の結果、有益なコメントが存在する場合に特徴的な、コメントの時間的分布の傾向がわかった。そこで本研究では、有益なコメント周辺における時間的なコメント量の変化についての特徴に基づく手法を提案する。提案手法を適用した結果を、コメントの評価機能と評価者を用いて評価を行った。結果として、時間情報及び簡単なテキスト情報から有益なコメントを抽出する可能性を示唆する結果が得られた。

キーワード 動画投稿サイト、ソーシャルメディア、推薦システム

1 はじめに

近年、YouTube やニコニコ動画、FC2 動画に代表される動画共有サービスが広く利用されるようになっていく。中でもニコニコ動画やFC2 動画においては、視聴者は動画の再生中にコメントを投稿することができる（本稿では同期型動画サイトと呼ぶ）。投稿されたコメントは投稿された動画内の時間位置が記録され、そのタイミングで表示されるようになる。他の視聴者は、動画の再生に同期して表示された、これらのコメントを読むことができる。しかし、いずれの動画投稿サイトにも、コメントの検索や抽出を行う機能は存在しない。そのため、ユーザに対して有益なコメントを抽出して提供する必要がある。

本研究では、動画に同期したコメントから有益な動画コメントを抽出する手法について提案する。ここで、有益な動画コメントを“視聴者が面白い、笑えると感じるコメント”と定義する。動画自体の面白さに加え、面白いコメントと一緒に視聴することで、動画視聴の楽しみが増すと考えられる。

我々は、コメントに付与された2種類の時間情報を用いて、有益なコメントを抽出することにした。具体的には、コメントの投稿日時を表す実時間情報と、コメントの動画中における時間位置を表す動画時間情報である。面白さを生むきっかけとなったコメントは、それに反応する他のユーザのコメントを多く伴うと考えられる。そこで、上記の時間情報を利用して、コメントの集中している箇所を検出する。

本稿の構成は以下のようになっている。2章で、2種

類の時間情報を用いることにした背景について詳しく述べる。3章で、面白いコメントに対して、他のユーザから多くのコメントが出現しているかどうかについて調査する。4章で、本研究の提案手法と比較手法について述べる。5章では、提案手法を評価する方法について述べ、6章でその結果を述べる。7章で、本研究と関連のある研究について述べる。

2 アプローチ

コメントの有益さを評価するには、コメントのテキストそのものを評価することが一般的である。しかし、コメントは通常非常に短文であり、また、コメントの中には、2バイトの記号文字を用いた装飾的なものも多い（このようなコメントをする人は、一般にコメント職人と呼ばれている）。これらのことから、コメントのテキストの内容を分析して、その有益さを評価するのは難しいと思われる。そこで我々はコメントを評価するのに、そのテキストではなく、時間情報を用いることとした。一般に、動画時間で見た時に、動画中の面白い部分やハイライトとなる部分には多くのコメントが集中することがわかっている [1][3]。この現象は、動画中の面白いコメントに対しても、起こると考えられる。よって、コメントの集中している箇所を検出することは、有益なコメントを抽出するための手がかりになると考えられる。

しかし、コメントの集中を検出するだけでは、その集中が動画の面白さによるものかコメントの面白さによるものかの区別はできない。そこで、動画内容によるコメントの集中と面白いコメントによるコメントの集中について、それぞれの時間的分布の特徴について考える。動画内容による面白さは時間（実時間）が経過しても不変

のものである。従って、動画内容に対してコメントの集中が発生している部分は、時間が経過しても変化しないと考えられる。

反対に、面白いコメントに対するコメントの集中は、そのきっかけとなる面白いコメントが投稿されてから現れるものである。特に、元々の動画の内容では注目される場面ではなかった部分に対して、面白いコメントが投稿された場合を考える。この場合、動画の面白い部分ではないため、動画が投稿された直後にはあまりコメントは多く投稿されない。しかし、ある時点できっかけとなる面白いコメントが投稿されることで、それに対する反応として多くのコメントが投稿されるようになる。つまり、その部分には実時間の経過に従って、コメント量の大きな変化が現れることになる。そこで我々は、実時間情報と動画時間情報の2種類を用い、実時間で見た時に、各コメントの前後でコメントの量が変わるかどうかを見ることにより、有益なコメントを抽出する。

ここで、本稿で用いる用語を次のように定義する。

トリガーコメント：面白さを発生させるきっかけとなったコメント

レスポンスコメント：トリガーコメントに対して反応するコメント

クライマックス：動画本来の面白さから注目されコメントが集中する部分

コメントバースト：トリガーコメントによってレスポンスコメントが集中するようになった部分

3 ニコニコ動画におけるコメントの投稿傾向についての調査

ニコニコ動画におけるコメント投稿について調査した。2014年8月1日から3日の間にニコニコ動画総合ランキング(24時間)に掲載された動画の上位100件を調査の対象とした。ただし、動画のコメント数が2000件未満のものは取得しないこととした。

3.1 実時間軸を用いた場合の調査

コメント投稿の時間的な傾向を調査するために、ある動画に付与されたコメントを、動画時間と実時間の2軸からなる散布図にマッピングした。しかし、この散布図の多くは実時間方向のコメント量に大きな偏りが存在した。これは、動画投稿から時間が経ち視聴者が減少しコメントの投稿頻度が減ったことや、逆に動画ランキングや他のメディアに掲載され急激に視聴者数が増加したことが理由と考えられる。これらのバイアスが存在すると、コメントバーストを発見することが難しくなる。このようなイベントは実時間にリンクして発生しているため、実時間ではなくコメントの投稿順序の番号(コメント番号)を軸にして、同様の散布図を得ることとした。

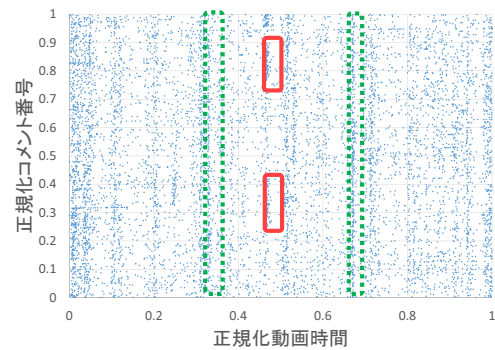


図1 コメント番号と動画時間を用いたコメントの散布図

3.2 コメント番号軸を用いた場合の調査

図は図と同じ動画のコメントを、動画時間とコメント番号の2つの軸からなる散布図にプロットしたものである。まず、図2中の緑色の点線で囲まれた部分を見ると、この動画部分には常にコメントが集中している。ここには常に注目が集まっているため、この注目の原因は動画に本来ある面白さによるものであると考えられる。

次に、図中の黄色の実線で囲まれた部分を見るとこの部分にもコメントは集中している。しかし、動画が投稿された直後は、この動画部分にはあまりコメントが集中していない。この動画部分は、何らかのコメントをきっかけに新たに注目されるようになったと考えられる。従って、ここにコメントバーストが存在すると考えることができる。

なお、実際の動画を見て緑色の点線の動画部分と黄色の実線の動画部分について確認したところ、緑色部分は動画のクライマックスであり、黄色部分はあるコメントがきっかけとなって発生したコメントバーストであることがわかった。

以上からクライマックスとコメントバーストでは、コメントの時間的分布に違いがあることがわかった。この知見を用いて次の章では、コメントバーストを検出して有益なコメントを抽出する手法について提案する。

4 提案手法

提案手法は、コメント番号と動画時間の二次元平面において、コメント番号の軸に沿って急激にコメント数が増えた部分を探す。そのために上記二次元平面において、あるコメントが投稿される前の周辺領域と後の周辺領域をウィンドウにより切り取る。投稿後のコメント数が大きく増加していれば、対象のコメントをきっかけにコメントバーストが発生した可能性がある。

アルゴリズム1に、提案手法のアルゴリズムを示す。アルゴリズムは、動画の全コメント及び窓の大きさ、及び後述する変化量計算の繰り返し回数と有益なコメント

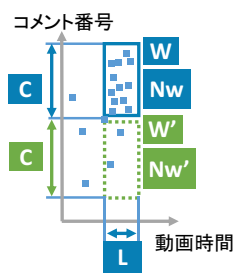


図2 ウィンドウの様式図

の抽出数を入力として、有益なコメントと考えられるコメントを出力する。CommentList はある動画の全てのコメントを格納した配列である。この配列の要素はコメントの情報を格納した構造体であり、コメント番号、動画位置がコメント本文が記憶されており、さらに計算したコメント数の変化量を記憶するデータ型を持つ。L はウィンドウの動画時間軸に対する幅である。CList は C の大きさを格納した配列であり、コメントの変化量を N 回繰り返し計算する際に繰り返し回数に応じて C の大きさを変化させる。これについては後述する。K は最終的に有益なコメントとして抽出するコメントの数を決定する変数である。

ここでウィンドウの定義について、図2を用いて説明する。図2はウィンドウの様式図である。

基準となるコメントを A とする。A を基準として2つのウィンドウ W と W' を作る。W と W' はそれぞれ、A の投稿前と投稿後の周辺領域を切り取るためのウィンドウである。W と W' の動画時間軸に対する幅はどちらも L 秒である。今回の実装では L=5 秒とする。W と W' のコメント番号軸に対する幅はどちらも C である。

C と L で定められるウィンドウを元に、関数 CalculateCommentDensity ではコメントごとにウィンドウ前後のコメント数の差を計算する。関数内では、引数にとったコメント群に含まれるコメントの一つ一つについて次のような処理を行う。まず、対象とするコメント A を基準として、ウィンドウ W と W' 内に含まれるコメントの数をそれぞれ計算する。CountComment は、C と L、そして対象とするコメントの配列とウィンドウの起点となるコメントを入力として受け取り、ウィンドウ内のコメント数を数える関数である。次に、得られたコメントの数を C で除して正規化する。本稿では、ここで得られた値をコメント密度と呼ぶ。コメントごとに、W と W' 内のコメント密度の差を計算する。

CalculateCommentDensity の処理を、配列 CList に記録された C の値ごとに繰り返し、それぞれのコメント密度の差は、コメント情報の密度差の変数に加算し記録しておく。繰り返しを行う理由は、これは、コメント

Algorithm 1 提案手法のアルゴリズム (Main)

```

1: arguments(CommentList, CList, L, K, N)
2: for i = 0 to N do
3:   C = CList[i]
4:   CalculateCommentDensity(CommentList, C, L)
5: end for
6: ValuableCommentList =
7:   Extract(CommentList, K)
8:
9: 関数 CalculateCommentDensity
10: arguments(CommentList, C, L)
11: Length = CommentList の要素数
12: for i = 0 to Length do
13:   A = CommentList[i]
14:   countW = CountComment(CommentList, C, L, A)
15:   countW' = CountComment(CommentList, -C, L, A)
16:   densityW = countW/C
17:   densityW' = countW'/C
18:   CommentList[i].density+ = densityW - densityW'
19: end for
20: return

```

量の急峻な変化点の中でも、その変化が持続するものを抽出するためである。直近のコメント量の変化のみに注目すると、偶然起きたコメントの増減によってノイズが多くなる。そこで、窓の大きさを大きくしながら毎回の密度差を加算していくことにより、コメント量の変化が持続している部分の密度差がより大きくなるようにしている。本実装では CList = 100, 250, 500, 750, 1000 とした。

最後に Extract 関数で、コメント密度の差が大きいコメントから上位 K% を取り出す。

5 評価手法

この章では、提案手法を評価する方法について述べる。次節以降ではまず、比較に用いる手法について説明する。続いて、評価方法を説明する。評価には、ユーザによるコメントの評価機能（ニコニコ動画の”ニコる”）を用いる方法と評価者を用いる方法を採用する。

5.1 比較手法

5.1.1 特定文字を用いた手法

コメントの本文を詳細に解析することは難しいものの、簡単なテキスト内容の利用を行うことは可能である。そこで、面白いコメントへのレスポンスコメントの中に現れやすい特定の文字を利用してコメントを抽出する。具体的には”w”と矢印(”↑”, ”↓”, ”←”, ”→”)を

Algorithm 2 特定文字を用いたアルゴリズム

```

1:  $arguments(CommentList, C, L, K)$ 
2:  $CountSpecificWord(CommentList, C, L)$ 
3:  $ValuableCommentList =$ 
4:  $Extract(CommentList, K)$ 

```

用いる。"w"は笑いの感情を表す文字である。また、矢印はレスポンス先のコメントの位置を示すものである。

アルゴリズム2が特定文字を用いた手法のアルゴリズムである。CountSpecificWordはそれぞれのコメントについて、C, Lで表されるウィンドウWに含まれる特定文字の数を数える関数である。数えた特定文字の数は、提案手法同様コメントの構造体が持つ変数に記録される。なお、矢印の数と"w"の数を加算する際、それぞれに重みをかけて加算を行っている。この時の式は、重みをaとして、(特定文字の数) = $a \times (w \text{ の数}) + (1 - a) \times (\text{矢印の数})$ で表される。Aは0以上1未満である。特定文字の数の多いもの上位K%をExtractValuableComment関数で抽出する。本実装では、Cは1000件とした。

5.1.2 提案手法と特定文字を組み合わせた手法

提案手法のあるコメントの投稿前後のコメント数を比較するという考え方で、特定文字を使う考え方を組み合わせる。具体的には、ウィンドウWとW'に含まれる特定文字の数を数え、その差を計算する。差の大きいコメントの上位K%を抽出することとする。この手法により、有益なコメントが存在しやすと考えられる笑いの感情を表すコメントが多い場面の中でも、特にコメントの増加率が多い部分を特定することができると考えられる。あるいは、矢印を含むコメント量の変化を検出することで、レスポンスコメントの増加をより精度よく検出できると考えられる。本実装では、C = 500とし、2つの窓を合わせて1000件となるようにした。

5.2 評価方法**5.2.1 コメントの評価機能を用いた評価方法**

動画共有サービスには、ユーザによるコメントの評価機能を備えているものが存在する。具体的には、ニコニコ動画の"ニコる"機能である。これは、ユーザが良いと思ったコメントに対して投票出来る機能であり、コメントごとに何回"ニコる"がされたかが記録されている。この評価の高いコメントを有益なコメントの正解データとして、抽出したコメントの精度を評価する。

"ニコる"を用いた具体的な評価方法について説明する。コメントのうち、"ニコる"が3回以上付いたコメントを有益なコメントとすることにした。本来の精度の評価方法を取るならば、抽出したコメントに対して、それが有益であったかどうかを検証することで行われる。し

かし、本タスクでは、正解コメントの全コメント数に対する割合は、平均0.01程度と非常に少ない。

そこで、抽出されたコメントの周辺に正解コメントが存在すればそのコメントを正解とする。実際の運用を考えた場合、ユーザは抽出したコメントに加え動画時間及びコメント番号で近接したコメントも一緒に視聴することになる。今回は、動画時間で前後2.5秒以内、コメント番号で前後500件以内に正解コメントが含まれていれば正解とした。この方法を用いた結果を緩和精度と呼ぶこととする。

"ニコる"を用いた方法は、機械的に多くの動画に対して評価が行えるという利点を持つものの、"ニコる"が付けられるコメントの性質から欠点も持っている。"ニコる"が付けられるコメントは、必ずしも面白いコメントに対して付けられるとは限らない。例えば、ユーザが共感できると感じた感想のコメントや、動画に対して補足説明を加えるコメントなどである。この性質から、精度に関して必ずしも正確な値が算出できるとは限らない。従って、より正確に精度を算出するため、次に述べる評価者を用いた評価を行う。

5.2.2 評価者を用いた評価方法

抽出したコメントを評価者に見てもらい、面白いコメントであったかどうかを評価してもらう。また、面白いかどうかという基準は人によって異なることが考えられるため、評価者の間で多数決をとって精度を計算する。この方法では、実際に人間が見て面白いかどうかを判定すること、そして多数決を取ることから前節の"ニコる"を用いた評価よりも正確に精度を算出できると考えられる。

ただし、評価者はコメントの本文だけでなくその内容を理解するため、動画も一緒に視聴する必要がある。そのため、多くの動画とコメントについて評価してもらうことは被験者の負担が大きくなるため、評価できるコメントは少数となる。

20代の男性5人に評価者となってもらい、3つの動画について、まずコメント無しでその動画をすべて視聴してもらった。その後、それぞれの提案手法によって抽出したコメントと特定文字を用いて抽出したコメントを評価してもらった。なお、評価してもらうコメントは前節と同じく抽出したコメントから一定範囲内(今回は動画時間で前後5秒以内、コメント番号で前後500件以内)にあるコメントである。評価者にはコメントとそれが現れる部分の動画(今回は抽出したコメントを中心に前後5秒間)を一緒に視聴してもらった。評価者には視聴した部分ごとに、その中に面白いコメントが存在したかどうかを尋ねた。また、評価してもらう動画数が少ないため、有益なコメントがそもそも無い動画が評価対象に含

まれていた場合の影響が大きい。そのため、視聴してもらった動画は、視聴者数コメント数共に多く、コメントによる動画への参加が活発な動画カテゴリであるアニメカテゴリから選ぶこととした。

6 評価結果

6.1 コメントの評価機能を用いた評価結果

6.1.1 提案手法の評価

手法により抽出するコメントの割合 $K(\%)$ を (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 5.0, 10.0, 15.0, 20.0) と変化させて、それぞれで緩和精度の評価を行った。また、4章で述べた繰り返し回数 N を 1 から 5 まで変化させて、結果を比較した。

結果を図??左上に示す。いずれも右肩下がりのグラフになった。 K が 0.1 の時に精度が約 0.2 ($N=2$ の場合) となっている。タスクそのものが難しく、高い精度を得ることができなかった。少しでも高い精度でユーザに提示する方が良いと思われるため、 K は 0.1 が望ましいと思われる。また、 $N=2$ の場合が最も高い結果となった。ここから、直近のコメントの変化量を見るだけでなく、さらに長いスパンでのコメント量の変化を見ることで、精度が上がるのがわかる。これは、偶然のコメントの増減によるノイズを除去できたことによると考えられる。

6.1.2 特定文字を用いた手法の評価結果

特定文字を用いた手法の評価結果を図 3 右上に示す。前節と同じように K を変化させ評価を行った。さらに、コメントごとの "w" の数と矢印の数を加算する際の "w" への重みを (0, 0.5, 1) と変化させ、それぞれの場合の比較を行った。

結果から、"w" の重み=0 つまり矢印のみを用いた場合に最も良い結果となることがわかった。矢印はあるコメントへのレスポンスを行っていることを示す最も典型的な文字である。従って、周辺に矢印が多いコメントは他のコメントからのレスポンスを多く受けていると考えられ、有益なコメントである確率が高いと考えられる。また、精度は、 $K=0.1$ で約 0.3 となっている。これも高い値とは言えず、少しでも高い精度でユーザに提示するには、 K は 0.1 が望ましいと思われる。

6.1.3 提案手法と特定文字を組み合わせた手法の評価結果

提案手法と特定文字を組み合わせた手法の評価結果を図 3 左下に示す。 K は 6.1.1 節と同じ変化である。また、"w" の重みを (0, 0.5, 1) と変化させその結果を比較した。

"w" の重み=0、つまり矢印のみを用いた場合がもっともよい結果となった。これは前節での議論と同じ理由によると考えられる。

表 1 評価者による評価結果

	動画 1	動画 2	動画 3	平均
提案手法	0.118	0.333	0.476	0.309
特定文字	0.056	0	0.050	0.035

6.1.4 各手法間の比較

各手法で最も高い緩和精度を出したものを比較する。また、比較対象として $K\%$ のコメントをランダムに抽出した場合の結果を用意した。結果を図 3 右下に示す。

いずれの手法も $K \leq 1\%$ の範囲ではランダムに比べて t 検定を用いて有意水準 5% で有意な向上が見られた。また、提案手法に比べて特定文字を用いた場合と組み合わせた手法は有意な向上が見られたが、特定文字の手法と組み合わせの手法の間に有意な差はなかった。以上より、この評価方法においては特定文字、特に矢印の存在が有益なコメントの抽出に非常に大きな影響を持っていることがわかる。以上より、残念ながら提案手法は、特定文字を用いた手法よりも、有益なコメントを抽出する性能が低いことが分かった。有益なコメントの抽出には、矢印の存在が大きな影響を持っていることがわかった。

6.2 評価者による評価結果

表 1 に評価者を用いて評価した時の、提案手法と特定文字を用いた手法の精度を示す。いずれも抽出するコメント数は $K=0.1\%$ とし、特定文字については前節で最も精度の高かった矢印のみを用いた。なお、抽出したコメントが面白いコメントであったかどうかは、被験者の間で多数決を採って正解クラスとしている。

表から、絶対値は低いものの、いずれの動画でも提案手法のほうが精度が高い結果が見て取れる。これは、レスポンスが多いコメントを特定する性能の違いであると考えられる。提案手法では、時間的なコメント量の変化から相対的にレスポンスコメントの量が多いと考えられる部分を抽出している。これはテキスト内容には依存しない方法である。反対に、特定文字を用いた方法では矢印を含むコメントの数からレスポンスコメントの特定を試みている。しかし、この方法では、矢印を用いていないレスポンスコメントは特定できない。さらに、矢印によるレスポンスは面白いコメント以外に対しても行われるため、視聴者同士の単純なコメントのやりとりまで抽出されていた。以上の理由から、提案手法のほうが、レスポンスを多く伴う面白いコメントを抽出できていたと考えられる。

また、この結果は"ニコる"を用いた評価とは逆の結果である。この原因は、今回の評価方法によると考えられる。今回の評価では、有益なコメントを正確に抽出することが難しいことから、抽出したコメントから時間的に

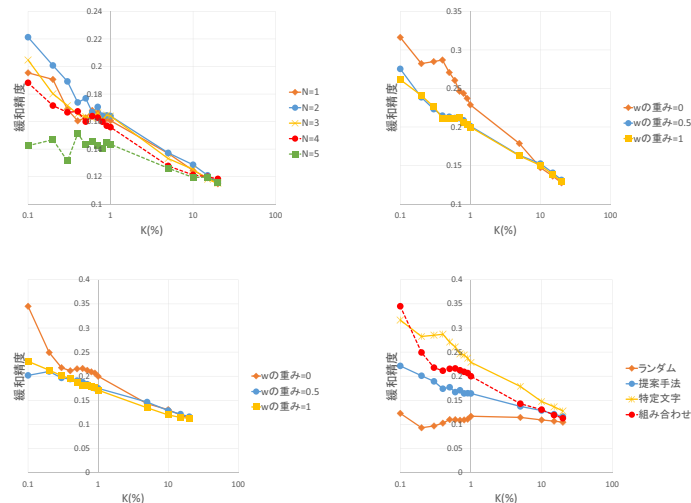


図3 コメントの評価機能による評価結果。

一定範囲内に正解コメントが存在すれば正解とした。この方法では、ある一つの正解コメントに対して複数の正解が得られることになる。経験的に、面白いコメントに対し矢印を用いたレスポンスが行われる場合、その一箇所に集中して矢印を含むコメントが現れる。この時、特定文字を用いた手法はその部分に集中してコメントを抽出する。この時、それらのコメントがすべて正解となるため精度が高めに算出されることになる。このため、前節では特定文字を用いた手法が提案手法を上回ったと考えられる。

7 関連研究

動画共有サイトで動画に対して投稿されたコメントについて行われた研究について紹介する。Thelwallらは、Youtubeにおいて、Youtubeのレスポンス機能を用いることで、コメント群中の議論を抽出している [2]。しかし、コメントの時間的な関係は考慮されていない。青木らは、ニコニコ動画の動画の中で、コメントが集中して投稿される部分が動画の重要な箇所であると考え、映像要約やサビの検出に利用している [1]。佃らは、ニコニコ動画に投稿されたコメントから、動画中の登場人物の名前が含まれたコメントに注目し、動画のインデックス付けを行っている [3]。

本研究では、コメントが動画とコメントどちらに対して付けられたものであるかの判定と、コメントから発生したコメントの集中の検出を初めて試みたものである。

8 おわりに

本研究では、同期型動画サイトのコメントから時間情報のみを用いて有益なコメントを抽出する手法を提案した。

コメントの評価機能を用いた評価では、矢印を用いた手法が提案手法を上回る精度を出したものの、評価者を用いた評価では、提案手法が上回るという結果を示した。ただし、いずれの場合も必ずしも高い精度とは言えない。しかし、これは非常に多くのコメントの中から、時間情報あるいは簡単なテキスト情報のみを用いて面白いコメントを抽出するという、タスクの難しさが大きな原因と考えられる。今回の結果から、本稿で用いた情報のみによる有益なコメントの特定は容易ではないことがわかったものの、被験者による評価で一定の結果が出たこと、さらにその後にとったアンケートで、面白い”空耳”のコメントがあったという回答が得られたことから、時間情報を用いた有益なコメントの抽出には一定の効果があることも分かった。

今後は、有益なコメントの持つ時間的あるいはテキスト特徴をさらに詳しく分析し、得られた特徴を利用した精度の向上を目指す。

参考文献

- [1] 青木 秀憲, 宮下 芳明, “ニコニコ動画における映像要約とサビ検出の試み”, 情報処理学会研究報告. HCI, ヒューマンコンピュータインタラクション研究会報告 2008(50), 37-42, 2008.
- [2] M. Thelwall, P. Sud, “Commenting on YouTube Videos: From Guatemalan Rock to El Big Bang”, Journal of the American Society for Information Science and Technology Volume 63, Issue 3, pages 616-629, March 2012.
- [3] 佃 洸撰, 中村 聡史, 田中 克己, “視聴者のコメントに基づく動画検索および推薦システムの提案”, WISS2011.