

# Doc2Vec 手法による 映画レビューサイトからのクエリと意味的に類似した評価表現の発見

栗原 光祐<sup>†</sup> 莊司 慶行<sup>†</sup> 藤田 澄男<sup>††</sup> Martin J. Dürst<sup>†</sup>

<sup>†</sup> 青山学院大学理工学部情報テクノロジー学科 〒252-5258 神奈川県相模原市中央区淵野辺 5-10-1  
<sup>††</sup> ヤフー株式会社 〒102-8282 東京都千代田区紀尾井町 1-3 東京ガーデンテラス紀尾井町 紀尾井タワー  
 E-mail: †kurihara@sw.it.aoyama.ac.jp, ††{shoji,duerst}@it.aoyama.ac.jp, ††sufujita@yahoo-corp.jp

あらまし 本研究では、Doc2Vec を用いて、映画レビュー文からクエリ語の語意をよく言い表した多様な表現を得る手法を提案する。現在、「みんなが『驚いた』と言っている映画を探す」というように、レビューからの評判に基づいて映画を検索する方法は確立されていない。レビュー文には、似た意味を表す様々な表現が含まれる。簡潔なクエリからレビュー文を検索するためには、検索クエリと対応するレビュー中の表現をできるだけ多く抽出する必要がある。提案手法では、レビューを映画ごとにまとめて Doc2Vec を学習し、それをを用いて各レビュー文を事前にベクトル化する。辞書を用いて拡張したクエリをベクトル化し、それぞれの文のベクトルと比較することで、クエリと意味的に近いレビュー中の文をランキングする。Yahoo!映画の実際の約 129 万件の作品ユーザーレビューと、クラウドソーシングを用いた大規模被験者実験により、提案手法の有用性を示した。

キーワード 映画レビュー, Doc2Vec, 印象検索, 言い換え

## 1. はじめに

映画の評判情報は、鑑賞する映画を選定する際には非常に重要な情報源である。映画レビューサイトには、ユーザから投稿されたレビューが数多く掲載されており、自由に閲覧することができる。しかしながら、投稿レビューをもとに、ある評判を得ている映画を検索する方法は確立されていない。例えば現在、映画レビューサイトで、「みんなが『驚いた』と言っている映画」を検索することはできない。そのため、現状では、映画タイトルからたどって各映画に投稿されたレビュー文を 1 つずつ読み、見る映画を決定しなければならない。しかし、Web 上の評判情報を手作業で調べる作業は、藤村ら [13] も述べているように非常に長い時間を要する。そのうえ、ストーリーの根幹にかかわる重要な情報 (ネタバレ) を見てしまう可能性がある。そのため、多数のレビュー文を読むということはリスクをとまう。

Web 上の評判情報は多数の人が投稿するため、似た意味を表す異なる表現が多数出現する。Web 上の評判情報を単純なクエリで検索するためには、検索クエリの語の類似表現や、言い換えとして機能する表現をできるだけ多く発見することが必要である。

このような、文中の短い表現への関連度計算のニーズは高まってきている。というのも、従来、Web 上の文書は個人サイトやブログなど、1 人あるいは数人によって書かれたまとまりのある文書が大半であった。しかし近年のユーザ投稿型サイトや SNS の台頭をうけ、非常に多くの人々が大量の断片的な文書を投稿するようになった。現在の Web 検索エンジンが用いる検索手法はまとまりのある文書を主なターゲットとしているため、断片的な文書が多く集まるような Web サイトに対してはうまく検索できない場合が多い。現在の高精度な Web 検索

エンジンであっても、得られる検索結果はページ単位であり、文単位や表現単位ではない。断片的な文書の文脈情報を得るためには、他の多くの断片的文書との関係を解析する必要があると考えられる。

本論文では、Word2Vec [8] や Doc2Vec [7] を用いることによって、簡潔なクエリを用いて映画レビュー文を検索し、クエリと意味的に類似した多様な表現を得る手法を提案する。提案には主体となる手法が 1 つと、補助的な手法が 1 つ含まれる。

主体となる手法は、映画ごとの文脈情報を用いたレビュー文のベクトル化である。この手法では、映画レビューは映画ごとに語彙的、あるいは文脈的な特徴をもっているという仮定をおく。Doc2Vec による学習時に、同じ映画に対する他のレビューを文脈情報として用いることで、その仮定をモデルに反映させる。これにより、映画ごとのレビューの特徴を反映したベクトル化が期待される。

補助的な手法は、クエリ拡張である。検索の際、クエリによってはそのクエリの語意が狭く、多様な表現の検索に適していないことが考えられる。それを解消するために、検索時に Word2Vec を用いてクエリの類義語を抽出し、クエリが表す意味の範囲を拡張する。

提案手法により、あらかじめ人手で観点を用意せずに語や文の類似度が計算可能になり、自由なクエリを用いてクエリに関連する多様な表現を検索可能になると考えられる。検証のため、Yahoo!映画の実データを用い、クラウドソーシングによる大規模な評価実験を行った。実際の作品ユーザーレビュー約 129 万件から、10 個のクエリについて 2 つの提案手法と 2 つの派生手法、2 つのベースライン手法を用いて、それぞれのクエリに対応するレビュー文を検索した。その結果得られた文と、検索クエリとの適合度について、クラウドソーシングによる大規模

被験者実験によってラベル付けを行うことで、提案手法の有用性を示した。また、各手法での検索結果上位のレビュー文に対して出現する語彙数を数えることによって、手法ごとの検索結果に含まれる表現の多様性を明らかにした。

## 2. 関連研究

本研究は、語の分散表現を用いた Web 上の評判情報検索に関する研究である。Web 上には多数のユーザ投稿型サービスが存在し、関連する研究として、評判情報検索、レビューの感情分類、類似アイテムの検索などの研究が行われている。

### 2.1 評判情報検索

あるアイテムを購入あるいは利用した人は、そのアイテムに対して何らかの意見や感想を Web 上に公開している場合がある。この公開された意見や感想を評判情報という。評判情報は、そのアイテムを購入あるいは利用することを検討している人にとっては、意思決定に役立つものである。そのため、あるアイテムに対する評判情報を Web 上から発見する研究が数多く行われている。

立石ら [11] は、評価表現辞書と近接演算処理によって商品に対する意見を自動で抽出している。そして、同じ評価軸上の肯定意見と否定意見の割合をもとに意見を要約している。この研究は、アイテムから評判情報を見つける研究である。これは、購入あるいは利用する具体的なアイテムの候補をもっているユーザにとっては有用である。しかし、アイテムの候補をもたないユーザにとっては、アイテムごとに評判情報を見なければならぬため、時間的なコストが大きい。

杉木ら [14] は、評判情報を任意の自然言語クエリで検索可能にするために、レビュー文中の係り受けを解析して意見を抽出するという手法を提案している。杉木らは論文内にて、自然言語クエリで評判情報を検索可能であるが、提案手法ではクエリに完全に一致しない意見、すなわちクエリと意味的に近い異なる表現を検索できなかったとしている。杉木らの手法では、自然言語で記述されたクエリを受け付けることで、ユーザのあらゆる要求に応えようと試みている。しかし、クエリと異なる表現であってもクエリと同じような意味を持つ表現が検索できていない。そこで本研究では、映画レビューを用いて、意見や感想を表す検索クエリから、そのクエリと同じような意味を持つ多様な表現を発見する。

### 2.2 投稿レビューの感情分類

映画レビューをはじめとした投稿レビューは、ユーザがそのアイテムに対する意見や感想を自由に記述するものである。そのため、ユーザごとに観点、観点ごとの評価、用いる表現などあらゆる基準が異なっている。そこで、各レビューにどのような意見や感想が書かれているかを分類する研究が行われている。

Singh ら [9] は、映画レビューに対する感情分類手法を提案している。SentiWordNet を用い、語の品詞やある語の周辺に存在する語などに着目しながら感情表現やその極性について分類している。

Jo ら [4] は、投稿レビューから様々な観点と、観点に対する感情の組み合わせを自動的に発見する手法を提案している。論

文では、単一文中的所有の単語が一つの観点から生成されるという仮定のもと、確率的生成モデルを提案し、それを感情と統合したモデルに拡張することで観点に対する感情をモデル化している。この手法の重要な利点は、レビューの感情ラベルをあらかじめ用意せずにモデル化可能である点である。

これらの研究は、レビューに書かれた感情に強く着目した研究である。しかし、鑑賞する映画を選定する際には、その映画を鑑賞してどのような感情を抱くかのみならず、その映画が描く情景や主題、その映画を鑑賞するのに適したシチュエーションなどで検索を行うことが考えられる。そこで本研究では、語や文の分散表現を用いることで、それらの幅広い検索要求に応える。

### 2.3 分散表現を用いたアイテム検索

本研究では、語や文の分散表現を用いたアイテム検索手法を提案している。分散表現を用いると、語や文同士の類似度を簡単に計算できる。これに着目したアイテム検索の研究は以前より行われている。

Trieu ら [10] は、Twitter 上に投稿されたニュース情報をタグ付け分類し、類似したニュースを検索する手法を提案している。42 万以上のニュース記事を収集して予め Doc2Vec を学習させ、そのモデルでニュース記事をベクトル化し、ラベル付けを行う。その際の候補の選出に語の分散表現を用いた自動クエリ拡張の手法 [5] を使用している。

本研究では、仮定とする映画ごとの文脈を学習時に反映させるために、学習はレビュー文ごとではなく、映画ごとに行う必要がある。そのため、通常の Doc2Vec の利用法、すなわち文単位で学習を行い、文単位でベクトル化する方法では対応できない。そこで、本研究ではベクトル化する単位よりも大きな単位での文脈情報の考慮が可能な新たな手法を提案する。

## 3. 手法の提案

本章では、簡潔なクエリを用いて、映画レビューに含まれる文の中から、クエリ語の語意を言い表す多様な表現を発見する手法について述べる。手法の要点を流れ図の形式で表したものを図 1 に示す。提案手法は、Doc2Vec でレビュー文を学習する際に、同じ映画に対する他のレビュー文を文脈情報として利用する。また、検索性能向上のため、ベクトル化前にクエリ拡張を用いるオプションを提案する。これらの詳細について、各節で説明する。

### 3.1 データセットの前処理

一般的に、映画レビューサイトの構造はサイト、映画、レビューの 3 つの階層からなる。1 つの映画レビューサイト内には、多数の映画の情報が掲載されている。それぞれの映画には、映画の詳細情報とならんで多数のレビューが投稿されている。それぞれのレビューには、評価点、本文、タグなどの情報が含まれる。本研究では、映画レビューは映画ごとに語彙的、あるいは文脈的な関連をもっているという仮定をおく。その仮定を Doc2Vec に反映させるため、学習の際に、それぞれのレビューがどの映画に対して書かれたかの情報 (映画 ID) を利用する。

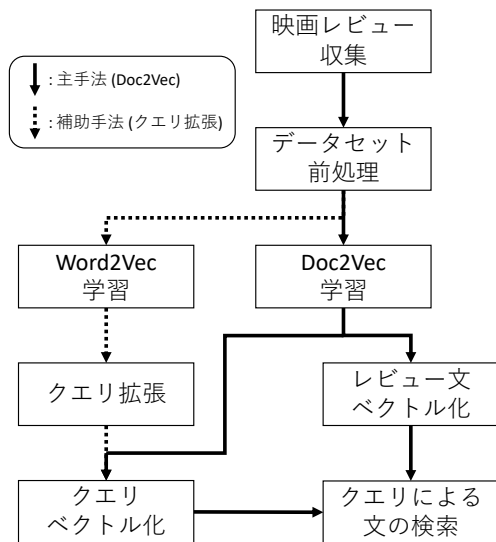


図1 提案手法の手順流れ図

はじめに、Word2Vec と Doc2Vec の学習のための前処理と、レビュー本文の文単位の分割を行う。

まず、Word2Vec や Doc2Vec で日本語の文書を学習させるために、文書を単語単位に分ち書きする。Word2Vec や Doc2Vec を使用する場合、名詞のみを使用して学習を行う場合が多い。しかし、本研究では映画を観た際の感情や感想、その映画を観るのに適した状況などの情報も重要となる。そのため、本研究では名詞以外の品詞の単語も全て使用する。

名詞以外の品詞も使用することにしたため、語彙数の増大と、それに伴う学習効率および精度の低下が懸念される。これを避けるために、語彙数を削減する3つの処理を行った。1つ目は、活用語の終止形化である。日本語の活用は語幹がもつ意味を大きく変えるようなものではないため、異なる活用形を異なる語として扱う必要はないと考えられる。そこで、活用されている語は全て終止形に統一した。2つ目は、終止形がひらがな1字の単語の除去である。日本語において、ひらがな1字の単語は多くが助詞や助動詞であり、重要な意味を持つことはまれである。そのため、これらの語はクエリとレビュー文との比較の際には不要であると考えた。3つ目は、数値の除去である。これもクエリとレビュー文との比較の際には不要であると考えられるため、単位等もついておらず、数字のみで成り立っている語はすべて除去した。

次に、制御文字や余分な空白など、不要な文字の除去を行った。空白文字、水平タブ文字、書式送り文字は半角スペースに置換した。これらが連続する場合はまとめて1つの半角スペースに置換した。連続する改行文字は2つ目以降を除去した。その他の制御文字 (BOM や垂直タブ文字など) は全て除去した。

さらに、レビュー本文の文単位分割も行った。自然言語の文章を自動処理で文単位に正確に分割することは難しいため、人為的に決定したルールに基づき分割することとした。本研究では、改行と句点の他、エクスクラメーションマークとクエスチョンマーク (全半角とも) を文の区切り文字と設定し、レビュー

本文を分割した。

### 3.2 Doc2Vec を用いたレビュー文ごとのベクトル化

あるクエリの意味をよく言い表すレビュー文を発見するためには、クエリとレビュー文の類似度が計算可能でなければならない。そのため、本節では Doc2Vec を用いてレビューを文単位でベクトル化する手法について述べる。

本研究では、映画レビューで用いられる表現には映画ごとに語彙的、あるいは文脈的な特徴が存在すると仮定する。例えば、あるユーザによって「泣ける」というレビューが投稿されている映画は、そうでない映画に比べて他の観た人も「泣ける」という感想をもつ可能性が高い。そのため、その映画に対して「泣ける」という内容の異なる表現によるレビューが投稿されている可能性が高いと考えられる。本手法では、Doc2Vec による学習に仮定を反映させるため、同じ映画に対する他のレビュー文を文脈情報として用いる。具体的には、レビュー文を映画ごとに1つの文書とする。すなわち、1本の映画に対して投稿された全てのレビュー本文を1つに連結し、1文書とする。これを、学習対象の全映画分用意し、学習用データセットとする。これを Doc2Vec に学習させることで、映画単位でのレビュー文の特徴を利用したモデルを得ることが期待できる。

得られた Doc2Vec モデルを用いて、各レビュー文をベクトル化する。Doc2Vec では、学習済みのネットワークに新しい文を入力し、その時の中間層の実際の値をとることで、そのモデル上での新しい文の分散表現を得ることができる。

これにより、任意の次元のベクトルが文の数だけ得られる。検索の際に実際にクエリとの比較に用いるため、あらかじめデータセットに含まれる全ての文に対してベクトル化を行い、元の文と対応付けた。

### 3.3 Word2Vec を用いたクエリ拡張

本研究の補助的なもう1つの提案は、クエリ拡張である。クエリ拡張とは、ユーザが入力したクエリの関連語を抽出し、自動的にクエリに加えることによって検索精度を向上させる方法である。本研究では、ユーザが入力するクエリは一単語から数単語程度を想定している。しかし、少ない語数からなるクエリをベクトル化した場合表現力に乏しく、クエリ語の周囲の限られた範囲の意味しか表すことができないと考えられる。本研究の目的である多様なレビュー表現の検索のために、クエリ拡張によってクエリの表現力を強化することが有用であると考えられる。

クエリ拡張には、関連語の抽出に外部の語彙情報から構築されたシソーラス辞書を使用する手法 [12] や、トピックに合ったデータセットを学習して得られる語の分散表現を使用する手法 [2] などが存在する。Diaz ら [2] によれば、Wikipedia など外部の網羅的なデータセットから構築された辞書や分散表現モデルを使用したクエリ拡張では、過剰に一般的な関連語が抽出されることが多い。目的に応じたデータセットを用意して学習することで、目的に応じた関連語を正しく抽出できるとしている。このことから、本研究ではデータセット内の映画レビュー文全てを用いて Word2Vec を学習し、得られた分散表現モデルを用いてクエリを拡張する。全ての映画レビュー文を学習する

ことで、映画レビューでよく用いられる表現や語彙などをよく学習し、映画レビュー文に対する検索に適した分散表現モデルを得ることが期待できる。

### 3.4 クエリのベクトル化

ベクトル化されたレビュー文に対して類似度による検索を行うために、レビュー文のベクトル化時と同一の分散表現モデルを用いてクエリをベクトル化する。まず、レビュー文と同様に、クエリに対しても節 3.1 で述べた前処理を行う。その後の手順は、クエリ拡張を適用する場合としない場合で以下の 2 通りに分かれる。

#### 3.4.1 クエリ拡張なし

前処理で得られた単語列を 1 つの文として、節 3.2 で学習した Doc2Vec モデルに与えることでクエリをベクトル化する。クエリが 1 単語である場合でも、その 1 単語を 1 文としてベクトル化する。

#### 3.4.2 クエリ拡張あり

前処理で得られた単語列に含まれる用言と名詞全てに対して、Word2Vec による類義語抽出を行う。それらの類義語でもとの語を置き換えた単語列を 9 系列生成し、元の単語列の後ろに連結する。これを 1 つの文として、節 3.2 で学習した Doc2Vec モデルに与えることでクエリをベクトル化する。

### 3.5 検 索

節 3.2 で文ごとのベクトルが、節 3.4 でクエリのベクトルが得られた。これらは同一のモデルによってベクトル化されているため、数学的な類似度の計算を意味的な類似度の計算として用いることができる。クエリのベクトルと各文のベクトルのコサイン類似度を計算し、類似度の高い文ほどクエリによく適合している文であるとみなす。

## 4. 実データを用いた大規模クラウド評価

本章では、提案手法の有用性を示すために用いたデータセットと、そのデータセットを用いて行った評価実験の詳細について述べる。

評価実験として、用意した 10 個のクエリについて、提案手法を含めた 6 手法でレビュー文の検索を行い、それぞれのランキングで上位にランクされたレビュー文について、クラウドソーシングによるアンケートでクエリとの適合度を測った。また、上位にランクされたレビュー文を集めて語彙数を分析することにより、多様な表現が検索できているかを評価した。

### 4.1 データセットの概要

提案手法の評価のために、ヤフー株式会社が運営する総合映画情報サイトである Yahoo!映画に投稿された作品ユーザーレビューデータを用いた。Yahoo!映画では、約 63,000 本の映画の情報、およびそれらの映画に対して投稿された合計約 524 万件の作品ユーザーレビューを掲載している。そのうち大多数の映画はレビュー件数が 10 件に満たず、1,000 件以上のレビューが投稿されている映画は 1,159 本であった。映画の本数をレビュー数を基準に集計したものを表 1 に示す。

表 1 レビュー数と映画本数の関係

レビュー数	映画本数
0 件	18,696 本
1 件以上 10 件未満	23,318 本
10 件以上 100 件未満	13,702 本
100 件以上 1,000 件未満	5,412 本
1,000 件以上 10,000 件未満	1,141 本
10,000 件以上	18 本

本評価では、このレビューデータのうち 300 件以上のレビューが投稿されている 3,245 本の映画に投稿された作品ユーザーレビューを対象とした。対象となるレビューのレビュー文に対しては、節 3.1 で述べた前処理を適用した。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、全レビュー文を形態素に分解した。MeCab の辞書には、Web 上でよく使用される用語 (スラング)、新語や一般の辞書には掲載されないような語も幅広くカバーできる mecab-ipadic-NEologd<sup>(注1)</sup> を用いた。また、活用語の語形の統一には、MeCab の機能の一つである語の標準形の出力を用い、数値の除去も MeCab による判定を基準にした。

対象となる 3,245 本の映画には、合計で約 129 万件のレビューが投稿されており、それは合計で約 1,190 万文からなる。

### 4.2 評価対象の手法

提案手法の有用性を評価するために、2 つの提案手法の他に提案手法の派生手法 2 つと、Doc2Vec を用いないベースライン手法 2 つの、計 6 手法を比較した。本節では、評価対象とする各手法の詳細について述べる。

Doc2Vec は Python のトピックモデリングライブラリである gensim<sup>(注2)</sup> による実装を用いた。主要な学習パラメータを表 2 に示す。表に示されていないパラメータは gensim のデフォルト値を用いた。gensim では、Doc2Vec モデルの infer\_vector メソッドに文を与えることで、そのモデル上で文の分散表現を推論させることができる。

クエリ拡張を適用する際に必要な Word2Vec は、Mikolov 自身らによる C 言語での実装<sup>(注3)</sup> を用いた。主要な学習パラメータを表 3 に示す。表に示されていないパラメータは実装のデフォルト値を用いた。

表 2 映画単位文書による Doc2Vec 学習時のパラメータ

パラメータ	値
次元数	200
ウィンドウ幅	7
$\alpha$ (学習率)	0.025
学習モデル	PV-DM

表 3 Word2Vec 学習時のパラメータ

パラメータ	値
次元数	300
ウィンドウ幅	11
学習モデル	Skip-Gram

(注1) : <https://github.com/neologd/mecab-ipadic-neologd>

(注2) : <https://radimrehurek.com/gensim/>

(注3) : <https://github.com/tmikolov/word2vec>

### 4.3 提案手法

#### 4.3.1 映画単位文書による学習 (movie-D2V)

節 3.2 で述べた映画単位文書による学習を用いた手法である。Doc2Vec の学習は 1 つの映画に対して投稿されたレビュー全文を 1 つに結合し、それを 1 文書として扱う。評価のために用いたデータセットは計 3,245 本の映画に投稿されたレビューからなるため、学習時は 3,245 の文書を用いて Doc2Vec を学習する。学習したモデルを用いてデータセット内の全文 (約 1,190 万文) をベクトル化する。クエリ拡張は適用しない

#### 4.3.2 クエリ拡張の適用 (plain-D2V+クエリ拡張)

節 3.3 で述べたクエリ拡張を用いた手法である。Doc2Vec の学習はレビュー文 1 文を 1 文書として行う。ただし、計算量の関係でデータセットからランダムに 10% 抽出したものを学習させた。学習したモデルを用いてデータセット内の全文をベクトル化する。クエリに対しては節 3.3 のクエリ拡張を適用する。

### 4.4 派生手法

#### 4.4.1 複手法 (movie-D2V+クエリ拡張)

節 3.2 で述べた映画単位文書による学習と、節 3.3 で述べたクエリ拡張の両方を用いた手法である。提案手法を同時に適用することでの相乗効果の有無や、逆に悪影響の有無などを評価することを目的としている。

#### 4.4.2 文単位での学習 (plain-D2V)

何も工夫をしない Doc2Vec による手法である。Doc2Vec の学習はレビュー文 1 文を 1 文書として行う。ただし、計算量の関係でデータセットからランダムに 10% 抽出したものを学習させた。学習したモデルを用いてデータセット内の全文をベクトル化する。クエリ拡張は適用しない。

### 4.5 ベースライン

#### 4.5.1 LSI によるベクトル化 (LSI)

提案手法および派生手法と比較するために、一般的な文のベクトル化手法である LSI (Latent Semantic Indexing, 潜在意味インデキシング) を用いる [1]。

LSI は、ベクトル空間モデルを用いた自然言語処理手法の 1 つで、文書群中の単語を任意の数のトピックに分類でき、また各文書に対する任意の次元のベクトル表現を得ることができる。自然言語処理や情報検索の分野でよく用いられている手法である。

計算量の関係でデータセットからランダムに 10% 抽出したものに LSI を適用した。LSI モデルおよび TF-IDF は、gensim による実装を用いた。対象の文書群を TF-TDF コーパスに変換したのち、LSI を適用した。トピック数は Doc2Vec を用いた手法と同じく 200 とした。これにより、それぞれの文に対して、200 次元のベクトルが得られた。

gensim の LSI の実装では、既に構築された LSI モデルに対して後から文書を追加することが可能である。クエリを 1 つの文書として、構築済み LSI モデルに追加することでクエリをベクトル化する。

#### 4.5.2 ランダム抽出 (random)

この手法では、いかなるクエリに対しても、データセットからランダムに文を抽出して提示する。この手法で得られる検索

結果のランキングには、クエリに適合した文書が一様に分布していることが期待される。そのため、この手法による検索結果に対して適合度評価を行うことで、データセット中にそのクエリに適合した文書がどの程度の割合で含まれているかを推測することができる。

### 4.6 各手法によるレビュー文検索

評価実験を行うために、実際にレビュー文を検索するためのクエリを 10 個選定した。映画レビューサイトにおいてよく利用されている映画の特徴を表すタグ、Web 上の映画情報サイトのカテゴリ区分や「○○な映画」のような特集記事などから、実際に需要があると思われるクエリを人手で選定した。選定したクエリとその特徴を表 4 に表す。各クエリは、映画が描く情景などを表すもの、映画を観た際に抱く感情や感想を表すもの、映画を観るのに適した状況などを表すものの 3 種類に大別できる。また、1 語のクエリは 7 つ、複数語からなるクエリは 3 つである。評価対象の 6 手法それぞれに対して選定した 10 個のクエリを与え、類似度の高いレビュー文を検索した。

表 4 評価対象に用いたクエリ一覧

クエリ	クエリの特徴	クエリの語数
シュール どんでん返し 家族愛 近未来	映画が描く情景などを表す	1 語
ほっこり 泣ける 懐かしい	映画を観た際の感情や感想を表す	複数語
旅行に行きたくなる デートにぴったり 何度観ても飽きない	映画を観る状況などを表す	

### 4.7 クラウドソーシングによる評価実験

各手法を用いて得られたランキングの評価のために、クラウドソーシングによる大規模な被験者実験を行った。実験はアンケート形式で、Yahoo!クラウドソーシングを利用した。

被験者には 1 つのクエリと、そのクエリで検索した結果得られたレビュー文のうち 1 文を提示し、そのクエリと文の適合度を 4 段階で回答させた。設問は各クエリ、各手法ごとに 100 問ずつ計 6,000 問用意し、それぞれの設問に対して異なる 4 人の回答を収集した。設問は用意した 6,000 問からランダムに出題され、1 タスクあたり 10 問 (チェック設問を加えて 11 問)、1 人の回答者につき 20 タスク (200 問) を上限とした。

本研究は、クエリと意味的に類似した多様な評価表現の発見を目的としている。もしクエリをそのまま含む文を抽出したいのであれば、レビュー文に対して全文検索を行えば十分である。このことから、クエリ語をそのまま含んでいるレビュー文は本評価の際には不要であると判断した。そのため、ランダム抽出以外の各検索結果から、検索に用いたクエリと同一の表現を含むレビュー文は全て除去した。除去の対象は、表 5 に示すクエリと同一であるとみなすことができる語と、クエリの表記ゆれと考えられる語である。

また、限られた設問数でランキング下位までの傾向を評価するために、それぞれのランキングから一部を間引いた。抽出の

基準を表 6 に示す。設問に採用するレビュー文は、クエリ語を含む文を除去したランキングの上位 30 位までの全て、31 位から 100 位までの 70 件からランダムに抽出した 30 件、および 101 位から 500 位までの 400 件からランダムに抽出した 40 件の合計 100 件とした。

クエリ	除外対象語
シユール	シユール
どんでん返し	どんでん返し ドンデン返し
家族愛	家族愛
近未来	近未来
ほっこり	ほっこり ホッコリ
泣ける	泣ける 泣けた
懐かしい	懐かし なつかし
旅行に行きたくなる	旅行に行きたくなる 旅行に行きたくなった
デートにぴったり	デートにぴったり デートにピッタリ
何度観ても飽きない	何度観ても飽きない

表 6 設問に使用するレビュー文の抽出基準

ランキング順位	抽出件数
1 位から 30 位まで	30
31 位から 100 位まで	30
101 位から 500 位まで	40

#### 4.8 タスクの実施

本クラウドソーシング被験者には、タスクの実施前に必ずタスクの説明を見せた。このタスクがキーワード一つと文一つのペアを見て、文がキーワードをどのくらいよく言い表しているか、言い換えているかを回答するものであること、選択肢が 4 段階であること、および回答基準について説明した。設問の選択肢は、『表している』、『やや表している』、『あまり表していない』、『表していない』の 4 段階とし、それぞれ 4 点、3 点、2 点、1 点として扱った。回答基準は、提示された文がクエリをどの程度よく言い表しているかである。「笑える」の例の場合では、笑いの程度については評点に反映させず、笑っているシーンなどを明確に表していれば、「大笑い」でも「クスクス」でも『表している』を選択させる。また、「笑える」に言及しているものの、それを否定する表現である場合や、「笑い」に言及しているが「笑えるかどうか」について言及していないものは『あまり表していない』とした。

## 5. 実験結果

クラウドソーシングによる実験では、用意した設問 6,000 問に対して各 4 回答、計 24,000 件の回答を得た。回答者間で回

答した設問数に 10 問から 200 問までの差があり、回答者数は 298 人であった。本章では、この回答データをもとに、各手法を比較する。

### 5.1 実験結果の信頼性評価

アンケート結果を分析するにあたって、まずそのアンケート結果がどの程度信頼できるものなのかを測る必要がある。アンケートが正しく行われたとするならば、ある設問における各回答者の回答には、誤差や偶然ではない程度の一致や一貫性がみられるはずである [15]。本節では、 $\kappa$  係数と呼ばれる一致度評価指標を用いて、アンケート結果の信頼性を測る。

1971 年に Fleiss [3] によって提案された Fleiss の  $\kappa$  係数では、3 人以上の評価者間の一致度を計算できる。ただし、Fleiss の  $\kappa$  係数は順序尺度による評定には用いることができないため、本評価の 4 段階の評定のうち 1 と 2 は「表していない」、3 と 4 は「表している」であるとして二値化して用いた。その結果、Fleiss の  $\kappa = 0.283$  であった。

Landis ら [6] によれば、 $\kappa$  係数が 0.21 から 0.40 である場合には、評定者間には一応の一致 (fair agreement) があるとされる。このことから、今回のアンケート結果には偶然以外の理由による評定者間の多少の一致が存在すると判断できる。

### 5.2 適合率

本評価での適合率は、各手法によって得られた検索結果のレビュー文のうち、クエリの語意をよく言い表している和被験者が判定したものの割合である。各レビュー文に対する 4 つの回答の平均評点が 2.75 点以上であった文を、本評価における適合文書とする。

ランダム抽出手法で検索した結果の平均適合率は、0.072 であった。クエリによって多少の差はあるものの、あるクエリに適合する文は、レビュー文全体の 7% 前後であると推測できる。

ランダム抽出以外の各手法のクエリごとの適合率を表 7 に示す。また、各手法のランキング上位 100 件および上位 500 件の 10 クエリ平均適合率を表 8 に示す。平均的には提案手法である movie-D2V が最もよく、他の手法より 0.1 程度高いという結果であった。さらに、クエリによって適合率が高いものと低いものが存在した。クエリ拡張は、plain-D2V に対して適用した場合には適合率に大きな変化はなく、movie-D2V に対して適用した場合には適合率が低下した。

### 5.3 nDCG

nDCG (normalized Discounted Cumulative Gain) を用いて、各手法によるランキング結果を評価した。nDCG は、アイテム群を何らかのシステムで順位付けして並べた際の DCG を、ランキングを実際の評点が高い順に並び変えて求めた DCG で割った値である。 $N$  件のランキングの第  $i$  位のアイテムに対する実際の評点を  $r_i$  としたとき、DCG は、

$$DCG = r_1 + \sum_{i=2}^N \frac{r_i}{\log_2(i+1)} \quad (1)$$

である。実際の評点が高い順に正確にアイテムを並べることができる最良のシステムにおいては、nDCG は 1.0 となる。

各手法の検索結果ランキングから nDCG を求めたものを表 8

表 7 各手法での検索結果の適合率

クエリ	movie-D2V	plain-D2V	movie-D2V	plain-D2V	LSI
		+クエリ拡張	+クエリ拡張		
シュール	<b>0.28</b>	0.18	0.24	0.17	0.16
どんでん返し	0.20	0.20	0.24	0.19	<b>0.40</b>
家族愛	0.25	0.36	<b>0.45</b>	0.10	0.28
近未来	0.38	0.42	<b>0.48</b>	0.37	0.18
ほっこり	<b>0.69</b>	0.53	0.28	0.51	0.37
懐かしい	<b>0.48</b>	0.29	0.44	0.43	0.46
泣ける	<b>0.78</b>	0.67	0.24	0.52	0.77
デートにびったり	<b>0.36</b>	0.12	0.11	0.17	0.05
何度観ても飽きない	<b>0.50</b>	0.34	0.27	0.39	0.23
旅行に行きたくなる	0.30	0.06	0.09	0.06	<b>0.33</b>
10 クエリ平均	<b>0.43</b>	0.32	0.28	0.29	0.32

表 8 各手法の適合率およびランキング評価 (全クエリ平均)

	p@100	p@500	nDCG
movie-D2V	<b>0.45</b>	<b>0.43</b>	<b>0.74</b>
plain-D2V +クエリ拡張	0.34	0.32	0.67
movie-D2V +クエリ拡張	0.30	0.28	0.65
plain-D2V	0.31	0.29	0.65
LSI	0.36	0.32	0.67
random	0.08	0.07	0.51

表 9 クエリ「泣ける」の movie-D2V と LSI の検索結果上位各 5 件

手法	順位	本文	平均評点
movie-D2V	1	涙もろい人、犬が好きな人は ハンカチ必須です。	<b>3.75</b>
	2	泣かず映画でした。	<b>3.25</b>
	3	号泣しました。	<b>3.75</b>
	4	ハンカチは必須です。	<b>3.5</b>
	5	ラスト涙した方が結構いる みたいですが、理解出来ません。	2.25
LSI	1	泣ければいいんじゃない。	<b>3.25</b>
	2	泣けて笑えて暖かい。	<b>3</b>
	3	泣け!!	<b>2</b>
	4	泣いた	<b>4</b>
	5	ルミナスで泣いた	<b>3.5</b>

に示す。評価実験で得られた各アイテムに対する 4 人の評点の平均値を、nDCG を求める際の各アイテムの評点として用いた。提案手法である movie-D2V の nDCG が、他の手法と比べて高い値を示した。

#### 5.4 表現の多様性

movie-D2V と LSI において、適合率が同等であったクエリ「泣ける」の検索結果上位 5 件ずつを表 9 に示す。LSI では、検索結果の本文が「泣く」「泣ける」のような表現にまともな表現にのみ対し、movie-D2V では、「ハンカチ必須」「泣かず」「号泣」など複数の表現が出現していることがわかる。

各手法における検索結果順位と、その順位までに含まれる語彙数の関係を、図 2 に示す。ここで語彙数とは、対象の文中に

含まれる名詞、動詞、形容詞、形容動詞とし、活用語は終止形へ統一したうえでユニークな単語数である。語彙数は 10 クエリの平均の値である。どの手法でも、文が多くなればなるほど語彙数は増加するが、語彙数の増加速度は減速することがわかる。また、クエリ拡張を行うことによって語彙数および語彙数の増加速度が大きくなることがわかる。提案手法である movie-D2V と、ベースラインである LSI を比べると、300 位まではほぼ同等の語彙数であるが、500 位まで、1000 位までと下位にゆくにづれ、両手法間に語彙数の差が広がり、LSI の方が語彙数が多くなった。

## 6. 考察

実験の結果、提案手法である movie-D2V が他の手法より平均的に高い適合率、および nDCG を示した。他の手法では利用しなかった映画ごとのレビューの文脈情報 (映画 ID) を利用したことにより、本研究で仮定したレビュー文の映画ごとの特徴を学習することができたためと考えられる。加えて、学習時の 1 文書の文の長さが必然的に長くなったことによって、学習の精度が向上したことも要因ではないかと考えられる。

movie-D2V と LSI の違いについて考察する。表 9 に示したように、両手法の検索結果の間には特徴的な違いがあった。movie-D2V では「涙」、「号泣」、「ハンカチは必須」など、「泣ける」をうまく言い換えているであろう表現が発見出来たのに対し、LSI ではほぼ単一とみなせる表現が上位に固まって出現した。LSI は、クエリ語そのものや異形、およびデータセットにおいてクエリ語とよく共起する語を重要視する。対して Doc2Vec は、周辺語 (共起語) が似ている語同士を類義語として扱うモデルを基礎としているため、共起語よりは文脈を重視しているといえる。加えて、movie-D2V 手法にて、同じ映画に対するレビューかどうかの情報を用いることによって、さらに文脈情報を考慮した検索が可能になったと考えられる。

クエリ拡張は、適合率を大きく上げるような効果は見られなかった。しかし、図 2 に示したように、検索結果中の語彙数を増加させた。このことから、検索結果の再現率 (recall) を重視する場合には、クエリ拡張は有用であると考えられる。

plain-D2V と LSI では、モデル構築時にデータセットから



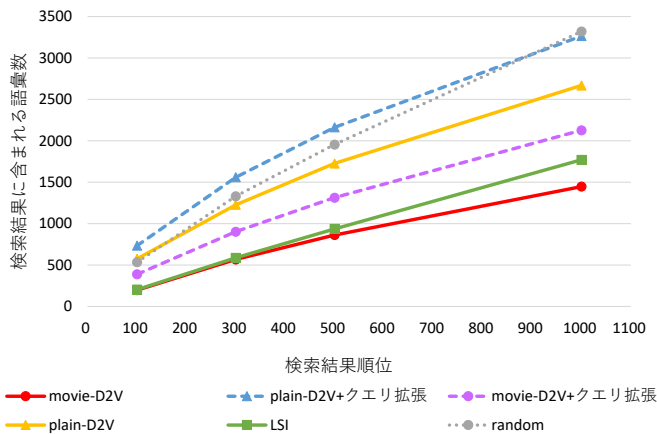


図 2 各手法における検索結果順位とその順位までの語彙数の関係

10% 抽出して用いたが、データセット全体を用いて学習した movie-D2V のモデル構築に比べ 2 倍から 5 倍の時間を要した。plain-D2V と LSI は、データセットのレビュー 1 文を 1 文書として用いるため、データセットの文の数がそのまま文書数となる。対して、movie-D2V は、映画ごとにレビューを結合して 1 文書とするため、データセットの映画本数が文書数となる。その結果、movie-D2V は文脈情報を補うと同時に、文書数を削減することができる。このことから、提案した movie-D2V 手法は、他の手法より高精度でありながら、高効率でもあり考えられる。

## 7. おわりに

### 7.1 まとめ

本研究では、簡潔なクエリを用いて映画レビュー文を検索し、クエリと意味的に類似した多様な評価表現を発見する手法を提案した。Doc2Vec でレビュー文を学習する際、同じ映画への他のレビュー文を文脈情報として用いることで、学習精度の向上を図った。Yahoo!映画の実際の作品ユーザーレビューに対し、提案手法と、その派生手法、およびベースライン手法を適用し、クラウドソーシングによる大規模な被験者実験で比較することで、提案手法を評価した。

評価実験の結果、提案手法である movie-D2V 手法の適合率が、他の手法と比較して最も高く、本研究の有用性を確認することができた。また、movie-D2V は他の手法よりも多様な表現を発見できる可能性が高いことが、実際の検索結果からわかった。実験結果の分析から、クエリ拡張には検索結果の語彙数を増加させる効果があることがわかった。同時に、本研究のクエリ拡張手法で検索結果の語彙数を増加させても、適合率の向上には寄与しにくいことも判明した。

### 7.2 今後の課題

映画レビュー以外でも、それぞれの文が言及している対象について明確であるようなデータ群の場合、提案手法が応用できると考えられる。例えば、商品レビューであれば商品 ID ごと、SNS の投稿であればハッシュタグごとのような単位で文をまとめることで、短文の集合からそれぞれの文脈を学習することが

可能ではないかと考えられる。

また今後、評価表現の集計手法や、程度の判定手法が必要である。例えば 1,000 人中 500 人が「泣いた」と表現した映画と、100 人中 40 人が「ハンカチ必須」と表現した映画のどちらをより「泣ける」映画とするかという問題がある。このような課題を解決することで、簡潔なクエリによって映画の評判情報から映画を推薦するという、本研究の背景に存在する要求に応えられるようになると思われる。

## 謝 辞

本研究は JSPS 科研費 18K18161 (代表: 荘司慶行), 18H03243 (代表: 田中克己) の助成を受けたものです。ここに記して謝意を表します。

## 文 献

- [1] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JASIST*, Vol. 41, No. 6, pp. 391–407, 1990.
- [2] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proc. of the 54th Annual Meeting of ACL*, Vol. 1, pp. 367–377, 2016.
- [3] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, Vol. 76, No. 5, p. 378, 1971.
- [4] Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proc. of the 4th ACM WSDM*, pp. 815–824. ACM, 2011.
- [5] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proc. of the 25th ACM CIKM*, pp. 1929–1932. ACM, 2016.
- [6] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, Vol. 33, No. 1, pp. 159–174, 1977.
- [7] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. of the 31st ICML - Vol. 32*, pp. II-1188–II-1196, 2014.
- [8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proc. of the 26th NIPS - Vol. 2*, pp. 3111–3119, 2013.
- [9] Vivek Kumar Singh, Rajesh Piryani, Ashraf Uddin, and Pranav Waia. Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In *Proc. of iMac4s 2013*, pp. 712–717. IEEE, 2013.
- [10] Lap Q. Trieu, Huy Q. Tran, and Minh-Triet Tran. News classification from social media using Twitter-based Doc2Vec model and automatic query expansion. In *Proc. of the 8th SoICT*, pp. 460–467. ACM, 2017.
- [11] 立石健二, 石黒義英, 福島俊一. インターネットからの評判情報検索. *情報処理学会研究報告 (NL)*, Vol. 2001, No. 69 (2001-NL-144), pp. 75–82, 2001.
- [12] 中山浩太郎, 原隆浩, 西章治郎. Wikipedia マイニングによるソーラス辞書の構築手法. *情報処理学会論文誌*, Vol. 47, No. 10, pp. 2917–2928, 2006.
- [13] 藤村滋, 豊田正史, 喜連川優. Web からの評判および評価表現抽出に関する一考察. *情報処理学会研究報告 (DBS)*, Vol. 2004, No. 72 (2004-DBS-134), pp. 461–468, 2004.
- [14] 杉木健二, 松原茂樹. 消費者の意見に基づく商品検索. *情報処理学会論文誌*, Vol. 49, No. 7, pp. 2598–2603, 2008.
- [15] 酒井哲也. *情報アクセス評価方法論: 検索エンジンの進歩のために*. コロナ社, 2015.