Journal of Data Intelligence, Vol. 3, No. 4 (2022) 441–459
© Rinton Press

DOC2VEC-BASED APPROACH FOR EXTRACTING DIVERSE EVALUATION EXPRESSIONS FROM ONLINE REVIEW DATA

KOSUKE KURIHARA[a]
Yahoo Japan Corporation.
Ciyoda-ku, Tokyo 102 – 8282, Japan
kurihara@sw.it.aoyama.ac.jp

YOSHIYUKI SHOJI
College of Science and Engineering, Aoyama Gakuin University.
Sagamihara, Kanagawa 252-5258, Japan.
shoji@it.aoyama.ac.jp

SUMIO FUJITA
Yahoo Japan Corporation.
Ciyoda-ku, Tokyo 102 – 8282, Japan
sufujita@yahoo-corp.jp

MARTIN J. DÜRST
College of Science and Engineering, Aoyama Gakuin University.
Sagamihara, Kanagawa 252-5258, Japan.
duerst@it.aoyama.ac.jp

This paper proposes a method for extracting diverse expressions from online movie review texts for a given keyword query. When people watch a movie that makes them cry, they generally do not say “I cried.” Instead, they use such euphemistic language as “I needed a handkerchief” or “My makeup was running.” To enable information retrieval based on audience reactions such as “movies that make me cry” using review texts, various paraphrased expressions must be collected for arbitrary queries. Our proposed method extracts such expressions from review datasets by applying two extensions to Doc2Vec: 1) it changes the granularity of the training sentences to mitigate a lack of context, and 2) it applies query expansion for similarity calculation in advance. We conducted a large-scale crowdsourcing experiment with 1.29 million actual sentences taken from Yahoo! Movies, Japan. The experimental result revealed that changing the training data granularity and adding the query expansion effectively collect more diverse expressions that have a meaning similar to the given query.

Keywords: Online Review, Doc2Vec, Euphemism

1. Introduction

As internet broadband and smart device proliferation deepen, people are streaming and watching more and more movies online. Searching for movies to stream has become a part of our lives. Therefore, finding a movie that matches our information needs is a difficult information retrieval problem.

For example, if a person wants to watch a tear-jerker movie that makes him/her cry, using a text search at a movie information site is a simple way. Such text search systems

[a] Kosuke Kurihara contributed to this research while at Aoyama Gakuin University until March 2021.

look for movies whose titles or descriptions contain the term related to the query keyword. However, a movie that is causing crying rarely contains the word “tear-jerker” in its title. The descriptions are mostly comprised of synopses; the word “tear-jerker” rarely appears in the synopsis. A simple text search is probably insufficient to find a movie to watch.

One cause is the problem that the metadata of a movie might not represent its audience’s impressions. Some movie information sites assign tags to solve this problem. Tags are concise and probably reflect the impressions of audiences. However, the coverage of tags is inadequate to meet all information needs.

Users often want movies that match trivial and personal information needs. For instance, for such unusual information needs as “movies that make you want to go on a trip”, no corresponding tag exists. The information granularity of tags is rougher than keyword queries.

Recommendations are another approach for identifying a movie. Most online video streaming sites recommend movies based on personal viewing history. However, users cannot directly input their interests or desires in such recommendation systems. Viewing history is too vague to reflect complex needs.

Online review sites often compensate for such deficiencies. Reviews on such sites contain information about movies written by many different people. Reviews include how they felt after watching a particular movie as well as its notable features. Since they contain more information about a movie than metadata or tags, they can be easily matched with queries because of their flexible descriptions, high coverage, and fine granularity.

Unfortunately, at the moment, such reviews are being underutilized. To learn about movies based on reviews, we have to look individually at each review. Reading each and every review is time-consuming and increases the risk of being exposed to spoilers. In the current situation, although reviews are suitable for determining whether a certain movie will make one cry, they cannot be used to find a list of movies that will make you cry. Thus, a method must be established that allows users to search for a sad movie by inputting the keyword query “cry”.

Writing flexibility is another factor that restricts such searches. People sometimes use euphemisms and metaphors to describe their opinions. When people watch a tear-jerker movie that makes them cry, they might not clearly admit “it is a tear-jerker movie.” Instead, they use such euphemistic language as “I needed a handkerchief” or “My makeup was running.” A search algorithm needs to match a query with a corresponding evaluation expression.

To solve this problem, we propose a method that matches a certain keyword query with expressions in reviews using Doc2Vec [1], which is a very popular method that vectorizes sentences and calculates the similarity between sentences and queries. However, Doc2Vec performs with low accuracy when learning short sentences because they do not provide enough context. Each evaluation expression included in the review is short. Therefore, for example, it is impossible to compare the keyword query “nostalgic” with such short phrases as “brings back memories” or “reminds me of my childhood” with sufficient accuracy.

Our method, which ingeniously uses Doc2Vec to extract more diverse evaluation expressions from the real data of movie reviews, contains two improvements:

- it changes the granularity of the training sentences to compensate for the lack of context, and

- it applies query expansion for similarity calculation in advance.

First, when training the Doc2Vec model, the method modifies the data granularity with the idea of **target-topics**. This expansion supplements the lack of context during the training phase of Doc2Vec when learning short sentences. We introduce the term "target-topic" for the objects or the referents of social media posts. Posts on social sites (*e.g.*, movie review sites, and online forums) generally have a target-topic. For instance, a social media post often contains one or more hashtags, a review in an online review site has a target item, and a comment in an online discussion forum has a news article as the topic. We modify Doc2Vec using target-topics as an additional context for training Doc2Vec networks. We expect that this step will allow the similarity to be exploited between sentences that are related to the same target-topic and improve search accuracy.

For the second improvement, the method uses query expansion techniques before vectorization for similarity calculation. The algorithm calculates the similarity between the query vectors and the respective evaluation expressions during the actual search. The query and the evaluation representation differ in length and information content. Therefore, before vectorizing the query, we use Word2Vec for query expansion. A keyword query consisting of only one or a few words is made into a short bag-of-words consisting of synonyms, aligning the granularity of the query and the evaluation expression to improve the vectorization accuracy.

We conducted a large-scale evaluation experiment to confirm the effect of both extensions. We compared the accuracy of the search result rankings of the six methods (*i.e.*, four variant methods comprised of a combination of the two proposed improvements and two baselines) using a large review dataset consisting of over 60,000 movies on an actual movie review site. Each search result was labeled by crowdsourcing through over 20,000 tasks.

This paper is an advanced version of the work presented at iiWAS2021 [2]. This paper is structured as follows. In this section, we explained the motivation and the goal of this research. Section introduces existing research on reputation mining, review analysis, and information retrieval using distributed representation. In Section , we describe the details of our method proposed in this study. Section shows the settings for the experimental evaluation and its results. Section discusses the results obtained through the experiments, and Section concludes with experimental results and explains future work.

2. Related Work

This section introduces related work from the viewpoint of our proposed method and application. This research uses distributed representation to extract reputations from online review sites and make them searchable. We discuss research on reputation mining, review information, and research on information retrieval using distributed representation.

2.1. *Opinion and Reputation Mining*

Opinion mining or reputation extraction, which estimates an item's reputation from social sites, has been widely studied [3]. Electronic Word-of-Mouth (eWOM) is a critical information source to change people's purchasing behavior. The most classic reputation analysis methods extracted the overall sentiment of articles about a certain product. The extracted sentiments can help users choose based on more positive or negative articles about a particular product.

As a more advanced method, extracting aspects from documents about an item and estimating the polarity for each aspect is becoming more common [4]. For example, it is possible to learn the features of a product for individual aspects, such as "screen size, rated high" and "picture quality, rated low" for a certain television [5]. In particular, many studies have extracted sentiment and polarity toward products [6]. In recent years, machine learning methods have been widely used for these purposes [7, 8]. For example, Titov *et al.* [9] proposed a method to extract and summarize word-of-mouth perspectives using a Bayesian model.

2.2. *Online Review Analysis*

Online review analysis is the second research field strongly related to our proposal. Online reviews are a powerful information resource for item retrieval [10], recommendations [11], decision support [12, 13], and so on. Singh *et al.* [14] and Bader *et al.* [15] focused on expressions and sentiments in reviews. Jo *et al.* [16] proposed a method that automatically detects a combination of various aspects and polarities in reviews. Tan *et al.* [17] proposed another way to find short sentences that have a similar sentiment. To search for movies with arbitrary keywords, the polarity must be computed for an infinite number of aspects. For example, it is difficult to predict and calculate in advance the aspect of the "degree to which viewing makes you hungry." Therefore, we need a text search-like method that can execute on demand. One of the benefits of our research is that it can also search for sentences with any aspect, not only sentiments but also story patterns or genres.

In recent years, machine learning [18] and ontologies [19] have become more common as methods for handling movie review information. Such technologies need to prepare a sentiment label dictionary in advance. It is difficult to make movies searchable by arbitrary keywords using these techniques.

Another common approach is to summarize movie reviews to help users choose suitable movies to watch next. Zhuang *et al.* [12] proposed a method for summarizing movie reputations by applying classical opinion analysis techniques to movie reviews. Liu *et al.* [20] summarized movie reviews for mobile devices using Latent Semantic Analysis (LSA). Although these methods can determine a movies reputation, they cannot search for movies by keywords.

2.3. *Information Retrieval Using Distributed Expression*

Our method is an example of information retrieval with distributed expressions. Many methods use Word2Vec or Doc2Vec to find information from social sites. Gysel *et al.* [21] also used Doc2Vec models for short sentences on social sites. This active research field tackles short sentences on social sites. Trieu *et al.* [22] proposed a method for tagging and classifying news information posted on Twitter and searching for similar news. Neither uses target-topics to improve search accuracy. Zuo [23] *et al.* used external information to vectorize short sentences in social sites for probabilistic topic models. Our method also uses the external context for Doc2Vec-based vectorization. The main difference is that we use target-topics as external information; our method does not need ontologies or dictionaries.

Many studies use distributed representation for applications similar to our study. Barkan *et al.* [24] proposed Item2Vec, a distributed representation of products in e-commerce sites. Item2Vec applies the Skip-Gram model to infer items based on a set of simultaneously pur-

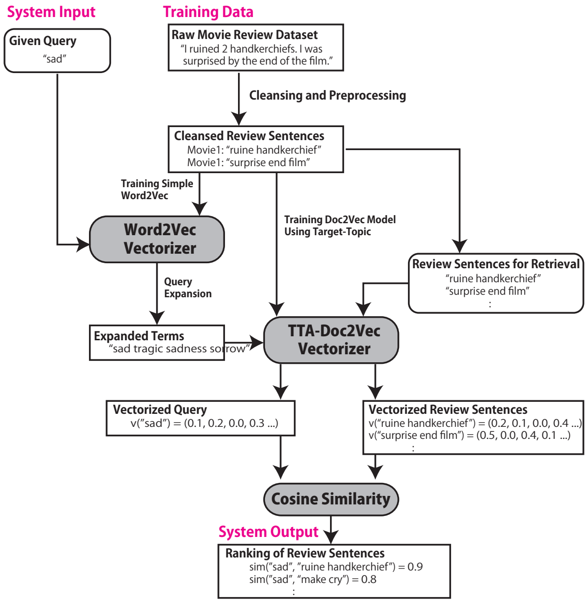
System Input
Training Data
Given Query
"sad"
Raw Movie Review Dataset
"I ruined 2 handkerchiefs. I was surprised by the end of the film."
Cleansing and Preprocessing
Cleansed Review Sentences
Movie1: "ruine handkerchief"
Movie1: "surprise end film"
Training Simple Word2Vec
Word2Vec Vectorizer
Training Doc2Vec Model Using Target-Topic
Review Sentences for Retrieval
"ruine handkerchief"
"surprise end film"
:
Query Expansion
Expanded Terms
"sad tragic sadness sorrow"
TTA-Doc2Vec Vectorizer
Vectorized Query
v("sad") = (0.1, 0.2, 0.0, 0.3 ...)
Vectorized Review Sentences
v("ruine handkerchief") = (0.2, 0.1, 0.0, 0.4 ...)
v("surprise end film") = (0.5, 0.0, 0.4, 0.1 ...)
:
Cosine Similarity
System Output
Ranking of Review Sentences
sim("sad", "ruine handkerchief") = 0.9
sim("sad", "make cry") = 0.8
:

Fig. 1. Overall flow diagram of the proposed method. The input is a query and the output is a ranking of sentences. A large review dataset is used as training data.

chased items as one item. This makes it possible to discover similar items. Phi *et al.* [25] also vectorized products in e-commerce sites for collaborative filtering. Their research treats a user's purchase history as a document for Doc2Vec learning. In this way, both users and items can be represented in a distributed representation. In research on the distributed representation of movie information for recommendations, Liu *et al.* [26] used Doc2Vec to recommend movies. Our work similarly vectorizes movies from their surrounding documents, although the purpose is different from their research, since it extracts evaluative expressions from reviews.

3. Method

This section describes our method for discovering various evaluation expressions for arbitrary keywords from a large dataset in practice. The method consists of three main parts: preprocessing, vectorization of the representation using Doc2Vec, and similarity calculation using query expansion.

The actual overall flow of the method is shown in Figure 1. The system accepts a query keyword as its input, and outputs a ranking of short sentences that paraphrase it. For this purpose, the system uses a large-scale review dataset to train Doc2Vec and Word2Vec models in advance. When training Doc2Vec, the system takes our proposed target-topic into account. During retrieval, the received query is expanded into a short sentence with concatenated synonyms using Word2Vec. Finally, it generates a ranking by taking the cosine similarity between this short sentence and each of the review sentences.

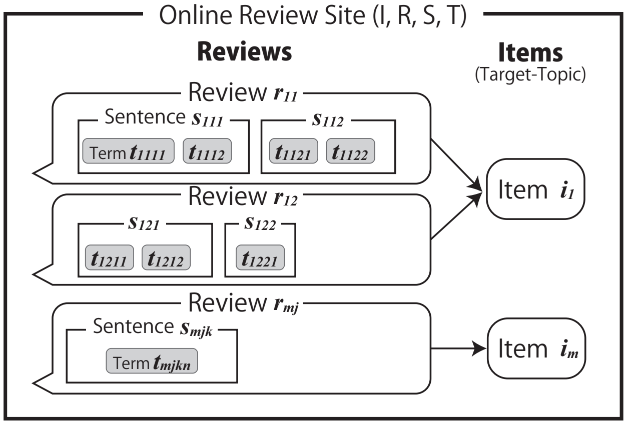

Fig. 2. Site structure of a typical online review site. Every sentence points to one item (target-topic).

3.1. *Preprocessing*

A dataset taken from an online movie review site is preprocessed for actual computation. Since the experiment uses actual large-scale review data written in Japanese, we also discuss problems specific to it.

The information on a review site generally consists of three layers: site, movie, and review (Figure 2). One site is associated with multiple movies, and one movie is associated with multiple reviews. Each element has a variety of metadata attached to it. A movie is given such information as its title, director, actors, and year of release. Its review is generally accompanied by such information as an author, a grade, a tag, and a review date.

In this study, we vectorize the expressions in the review and perform the learning on a sentence-by-sentence basis in each review (*i.e.*, s_{mjk} in Figure 2). Since we assume that the reviews for the same movie are contextually related, the sites entire information is used during learning.

First, a sentence written in a natural language is split into words. In the documents in Japanese, words are not divided by spaces. Therefore, we used a morphological analyzer to split them into morphemes.

Next, we filtered the terms by word classes and rule-based cleansing. The method used nouns, adjectives, and verbs, unlike other general methods using Word2Vec, which often extract only nouns. In this application, users are likely to input how they feel and how they do after watching the movie, and adjectives in their queries. Other useful watching information includes emotions and impressions when watching a movie and the suitable situations for watching it. Therefore, we added most words to the training data, excluding particles. Since the method uses words other than nouns, the total amount of words and noises increased. The method uses language patterns and rules to remove unnecessary words. First, it normalized the sentence; it conjugated all verbs into standard forms and changed all nouns to their singular forms. All words were lemmatized, leaving just the stem. Next, we removed words with extremely few letters because words consisting of only one or two characters in specific character types are probably noise. Most are fragments of colloquialisms that could not be correctly morphologically analyzed. Therefore, such words were removed. Numbers and symbols were also removed.

Similarly, control characters, special symbols, and low-frequency words were removed. On online review sites, a variety of users post reviews from various devices. As a result, posts sometimes contain words in different languages or special characters, depending on their devices and environments. For example, some posts that have incorrect line feed codes contain too much white space to make the review look better. Therefore, we replaced such white spaces as consecutive spaces or tab characters with a single white space. Control characters and line breaks were also removed.

We treated each sentence as a single evaluation expression for each of the preprocessed sentences. In other words, each sentence was removed with symbols to indicate the end of the sentence (*i.e.*, period, exclamation mark, *etc.*).

3.2. *Vectorizing Sentences Using Target-Topic Aware Doc2Vec*

Using Doc2Vec, this method next vectorizes each sentence, represented as a series of valid words obtained in the preprocessing.

Overall, our proposed method is a variant of the Paragraph Vector Distributed Memory (PV-DM) model of Doc2Vec. First, a two-layer network is trained by estimating the next term in a sentence from its context. In the original PV-DM model, a separate context is used for each sentence. We modify this approach using the target-topic itself as the context for all the sentences about it. Second, each sentence is vectorized using the trained network. This step is identical as in the original model; the granularity is different for the training and vectorization steps.

We used an online review site as an example to explain the details of our new method; it is one of the most typical examples of a CGM (Consumer Generated Media) site. Online review sites have a common structure, as shown in Figure 2. In this example, we call a reviews target-topic an "item" for clarity. A review site consists of items (target-topics) I, reviews R, sentences S, and terms T. Each item i_m is discussed by many reviews. Each review r_{mj} for item i_m consists of a number of sentences. Each sentence s_{mjk} is a sequence of terms denoted by t_{mjkn}. The goal of our method is to vectorize s_{mjk} accurately. For that purpose, our method uses item i_m as the context of s_{mjk}.

As shown in Figure 3, we modified the input vector for training as follows:

$$v(t_{mjkn}) = \left(v_{\text{onehot}}(i_m), \text{w2v}(t_{mjk(n-w)}), \cdots, \text{w2v}(t_{mjk(n-1)})\right), \tag{1}$$

where w is the window size, $\text{w2v}(t)$ is a distributed expression of term t obtained using Word2Vec, and $v_{\text{onehot}}(i_m)$ is a one-hot vector for item i_m. The p-th dimension of $v_{\text{onehot}}(i_m)$ is defined as follows:

$$v_{\text{onehot}}(i_m)_p = \begin{cases} 1 & \text{if} \quad p = m, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

It uses a one-hot vector about the item instead of about the sentence.

Next, our method vectorizes all the sentences in the dataset using the trained network. The vectorization procedures are identical as in the previous Doc2Vec. The middle layers values are used as the vector of a sentence in the vectorization phase. When used for a

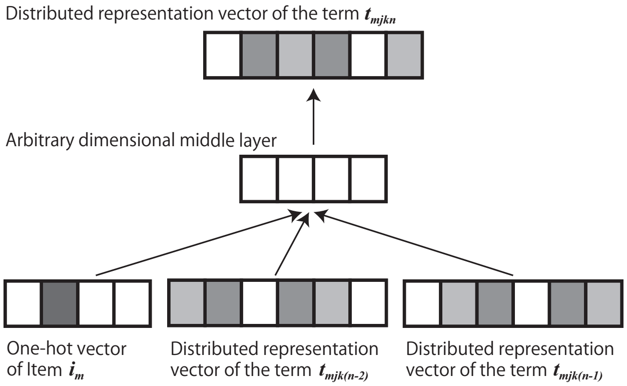
Distributed representation vector of the term t_{mjkn}
Arbitrary dimensional middle layer
One-hot vector of Item i_m
Distributed representation vector of the term $t_{mjk(n-2)}$
Distributed representation vector of the term $t_{mjk(n-1)}$

Fig. 3. Input and output of training with window size 2. The color depth of each cell reflects the value of each dimension.

search, a query is vectorized with the same model. To vectorize a particular short text, it uses input vector $v(t_{mjkn})$ that contains m-dimensional zero vector instead of a one-hot vector. The vectorizations of the sentences in the dataset and the query are then used for similarity calculation (*e.g.*, cosine similarity).

3.3. *Similarity Calculation Using Query Expansion*

The vectorization of the keyword query and each sentence in the review texts allows for similarity calculation. Then the method can rank the evaluation expressions corresponding to the query. Cosine similarity is one of the most general methods to compare vectors. However, queries and sentences have a different volume of information. A keyword query consisting of only one or two words has a smaller range of expressions and fewer types than an evaluation expression of multiple words. Therefore, we increased the information content of the keyword query using the traditional query expansion method before comparing it with each sentence.

When a given query was q, the method enumerates its lexical synonyms. Then it concatenates the synonyms into one short sentence form with Word2Vec by adding the top five synonyms to the original query. For instance, when q is a single term, “sad”, sentence q' becomes: “sad tragic sadness sorrow sorrowful regretful ...”. In this case, any corpus can be used to calculate the synonyms to the Word2Vec calculation, and another thesaurus can also be used for this purpose. For our experiment, we just used the movie dataset itself.

This expansion provides two benefits. First, if a query consisting of words is too narrowly defined, many results related to one specific word will appear at the top of the search results. By extending the words at the lexical and conceptual levels, we can gather more diverse expressions. Second, the length of the sentence itself affects the Doc2Vec calculation. If we vectorize sentences and words as they are, there are only a few types of vectors that a word can take. To improve the recall rate at the expense of the match rate, the query should be extended in advance.

After extending the query, we vectorize the extended sentences using the above model. Then the method calculates the cosine similarity with all the sentences in the vectorized review and outputs the results from the top.

4. Evaluation

We experimentally verified the usefulness of the proposed method using movie reviews, which are typical applications that can benefit. We compared our new method against three baseline methods. Since our final goal is to make a comfortable item search system based on reviewer opinions, we evaluated the methods with the metrics and measurements used in information retrieval research. We retrieved sentences from actual movie reviews for ten queries prepared in advance whose meanings resemble the queries. After ranking the sentences, a crowdsourcing questionnaire evaluated the degree of matching between the query and a selection of sentences. We also qualitatively evaluated the variability of the expressions in the actual search results.

4.1. *Dataset*

We used entire movie reviews posted on Yahoo! Movies, one of Japan's biggest online movie review sites. Yahoo! Movies provides information on about 63,000 movies and more than 5 million reviews for those movies. We reduced the data size by selecting only movies with 300 or more reviews, resulting in 3,000 movies, approximately 1.3 million reviews, and approximately 12 million sentences.

We preprocessed the data to make them suitable for Doc2Vec. All the sentences were separated into words using a morphological analyzer called MeCab. This step is essential because Japanese sentences are written without spaces between words.

Since it was impossible to calculate plain-**D2V** and **LSI** on the whole dataset because of high memory requirements, we created a sampled dataset by reducing 90 percent of the data. Each sentence in the 300 movie reviews was ranked by calculating the relevancy to the given query.

4.2. *Evaluated Methods*

Six methods were prepared for evaluation. Their details are as follows:

- **TTA-D2V+QE**: Target-Topic Aware Doc2Vec with Query Expansion is the proposed method described in Section . It evaluates the effectiveness of both the granularity manipulation of the training data in Doc2vec and the query expansion.

- **TTA-D2V**: Target-Topic-Aware Doc2Vec is a variant method to evaluate the effects caused by the target-topic alone.

- **D2V + QE**: Doc2Vec [1] with query expansion is another variant method to evaluate the effectiveness of query expansion alone. It combines query expansion and the original Doc2Vec.

- **D2V**: Doc2Vec is a baseline method using the original Doc2Vec without any changes.

- **LSI**: Latent Semantic Indexing [27] is another baseline method that uses topic modeling. We selected it because LSI is considered more suitable for short sentences than probabilistic topic models (*e.g.*, pLSA and LDA).

- **random**: Random Extracting ranks sentences randomly from the dataset for any query.

Table 1. Queries for movie review search task and their nDCG scores (translated)

Query	TTA-D2V+QE	TTA-D2V	D2V+QE	D2V	LSI	random	Type
Surrealistic	**0.85**	0.73	0.70	0.70	0.69	0.64	Movie content
Surprise ending	0.58	**0.86**	0.62	0.62	0.74	0.53	
Familial love	0.59	**0.89**	0.69	0.55	0.61	0.49	
Near-futuristic	0.70	**0.74**	0.69	0.69	0.63	0.49	
Relaxing	0.72	**0.75**	0.71	0.74	0.63	0.49	Viewer sentiment
Nostalgic	**0.79**	0.71	0.65	0.69	0.73	0.48	
Tear-jerker	0.71	**0.88**	0.76	0.66	0.78	0.46	
Makes me want to go on a trip	0.69	**0.78**	0.52	0.53	0.72	0.45	
Suitable for a date	0.75	**0.80**	0.65	0.64	0.56	0.57	Situation
Rewatchable	0.54	**0.93**	0.67	0.70	0.63	0.56	
average	0.69	**0.81**	0.67	0.65	0.67	0.51	

As a Doc2Vec implementation, we used gensim[b]for the proposed methods and **D2V**. The vector size is 200, and the window size is 7 in the Doc2Vec and TTA-Doc2Vec calculations. All other learning parameters are the default values of gensim. The number of topics (vector size) for **LSI** is 200, equal to the methods using Doc2Vec.

4.3. *Queries*

For the evaluation experiment, we derived ten queries from tags on movie review sites, the categories of movie information sites, and feature articles about movies. Table 1 shows the selected queries and their features. These queries were used with each method, and review sentences with high similarity were retrieved.

4.4. *Relevance Labeling with Crowdsourcing*

We used a well-known crowdsourcing service for labeling the search results. The participants scored the similarity between the shown queries and the sentences on a scale from 1 to 4: completely different, slightly different, slightly similar, and identical.

The number of questions was 100 for each of ten queries and four methods, resulting in 4,000 questions. Because the method aims to find expressions different from the query, sentences containing the query term itself were removed from the search results.

Sentences used for the questions were sampled from the 500 top-ranked results for each of the four methods. The sampling rate was set high in the top part of the ranking and lower further down. In addition to all of the top 30 sentences, 30 sentences from ranks 31 to 100 and 40 sentences from ranks 101 to 500 were randomly selected.

Dummy questions were interspersed in the questionnaire to weed out dishonest workers. They were occasionally asked to perform simple arithmetic calculations, such as whether a certain number was even. Dishonest workers were removed, and the same questionnaire was reassigned to another worker.

4.5. *Evaluation Metrics*

We proposed an algorithm that, given an arbitrary keyword query, ranks short sentences that contain evaluative expressions that paraphrase it. This algorithm can be evaluated as an

[b] https://radimrehurek.com/gensim/

information retrieval algorithm. Therefore, we first consider the rate of conformity of our top search results using P@k (Precision at k). Then, to check the accuracy of the created ranking, we calculate nDCG. Finally, we discuss the diversity of the results obtained. For this purpose, we analyzed the number of unique words actually included up to the k-th item in the ranking, and the number of words included in each sentence.

5. Results

This section describes the experimental results from the viewpoints of precision, ranking, and expression diversity. We collected 24,000 answers; four answers each prepared 4,000 questions.

To check the validity of the crowdsourcing results, we calculated the degree of agreement among the raters. Since the number of subjects per task was four, we used Fleiss's Kappa coefficient. Since $\kappa = 0.28$ was between 0.21 to 0.40, these opinions have fair agreement [28]. In order to further analyze the inter-cloud worker agreement, we calculated the ICC (Intraclass Correlation Coefficients). There are three types of ICCs: ICC (1, X) for intra-rater reliability, ICC (2, X) for inter-rater reliability, and ICC (3, X) for inter-rater reliability in relative agreement [29]. Each worked assessed the evaluation entity once in our task, so ICC (2,1) was used to analyze the degree of agreement among the three workers. The four responses to each question were considered an independent series, and 95 percent confidence intervals were calculated. The value of ICC (2,1) in this case was $0.343 <$ ICC (2, 1) < 0.370; This result indicates that there is fair agreement among the evaluators.

We next compared the precision of the rankings retrieved by each method. A sentence with an average score of 2.75 points or more for the four answers is defined as relevant. Table 2 shows the precision at 10, at 100, and at 500 of each method. A precision of 0.07 for the **random** extraction shows the difficulty of this search task. The dataset contains only seven percent correct answers.

TTA-D2V performed the most accurately among the proposed methods. The methods using the idea of target-topic and **LSI** achieved higher precision than the others. The **TTA-D2V** method showed significantly higher precision than **LSI** ($p = 0.00$ on Welch's t test). In contrast, the accuracy of method **TTA-D2V+QE**, which included all the extensions, did not differ from LSI. Figure 4 shows the precision in each section of the rankings. For methods other than **random**, the precision steadily declined. The **TTA-D2V** method had the highest precision in the top ranks and among the totals.

The ranking accuracy of each method was evaluated using normalized Discounted Cumulative Gain (nDCG). As Table 2 shows, the proposed method has a higher nDCG score than the other methods.

The expression diversity of the sentences in the results is another important factor. The number of unique terms in sentences in a ranking section is shown in Fig 5, as a simple indicator of how the method could find diverse evaluation expressions. Note that this result is just the number of words obtained, so it does not necessarily mean correctly paraphrasing the query. The number of terms can be used only to compare methods with similar accuracy. More diverse expressions can be found between the two methods with the same accuracy if the number of unique words is high. Depending on the method, the length of sentences with high similarity to the query is different. For example, in the method without query expansion, words and short sentences are compared for similarity, so more short sentences are likely to

Table 2. Precision and nDCG (average of ten queries)

	p@10	p@100	p@500	nDCG
TTA-D2V+QE	0.44	**0.56**	0.29	0.69
TTA-D2V	**0.72**	0.36	**0.51**	**0.81**
D2V+QE	0.32	0.34	0.32	0.67
D2V	0.32	0.31	0.29	0.65
LSI	0.46	0.31	0.32	0.67
random	0.11	0.36	0.07	0.51

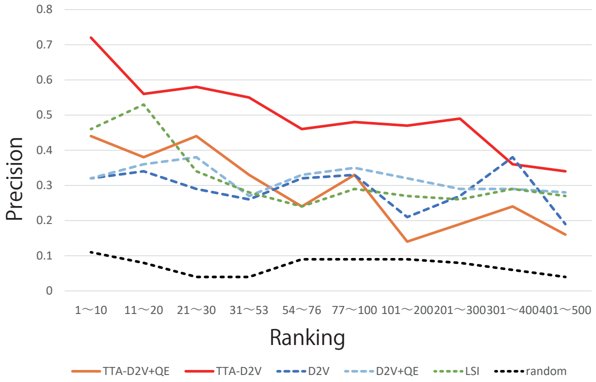

Fig. 4. Precision for each method and ranking section

be extracted. For each method and query, the average sentence length of the search results is shown in Table 5. As expected, the number of unique terms is high when the sentences are extracted randomly. Comparison of the results with and without query expansion shows that the number of words increased for both Doc2Vec-based methods when query expansion was used.

As an example, Table 3 shows the top five search results of our new method and **LSI** for the query "tear-jerker." The precision of both methods is almost the same for this query, although our method found diverse expressions.

6. Discussion

Our experimental results show that the **TTA-D2V** method is significantly more accurate than the others. Our method may be more precise because it uses target-topics as contexts. Since method **TTA-D2V+QE** using target-topics also outperformed the plain **Doc2Vec**, considering the target-topic seems effective to vectorize short sentences.

The diversity of search results will also be discussed in more detail. When query expansion was applied, the precision and nDCG tended to decrease. On the other hand, an increase in the number of acquired terms is also observed (*e.g.*, see the gap between orange and red lines in Figure 5). This indicates that acquiring more diverse representations provides a trade-off with accuracy.

Although the accuracy fell, the methods using query expansion extracted some evaluation expressions that could not be obtained with the conventional methods. As Table 3 shows,

Table 3. Top five results for proposed methods (TTA-D2V and TTa-D2V+QE) and LSI for query "Tear-jerker" (translated).

Method	Rank	Sentence	Relevance
TTA-D2V+QE	1	I couldn't stop my tears during the last scene.	**4.00**
	2	It was a heart-warming movie.	2.25
	3	Although I didn't want to cry, I couldnt help myself during the last scene.	**3.00**
	4	I was deeply moved by the leading actor.	**3.75**
	5	I was scared by that impression.	1.25
TTA-D2V	1	The last scene made me cry. I sobbed!	**4.00**
	2	This is a movie that make me cry.	**4.00**
	3	I can't help but cry now.	**3.50**
	4	I cried!	**3.75**
	5	Tearful!	**3.00**
LSI	1	Let's all just cry together.	**3.25**
	2	I cried, laughed, and was impressed.	**3.00**
	3	Crying!	2.00
	4	I cried.	**4.00**
	5	I cried during the song.	**3.50**

the query expansion excavated such diverse expressions as focusing on tears, handkerchiefs, and hearts for the query "tear-jerker". Much onomatopoeic language (*e.g.*, boohoo, phew) was included in the results of methods using query expansion. In contrast, **LSI** found many simple synonyms for queries.

Table 1 shows the nDCG for each query. **TTA-D2V** earned the highest scores throughout, but for some queries, the query expansion increased their accuracy. For the queries related to viewer content, both our methods and **LSI** were highly accurate. In these tasks, the viewer impressions are simply included in the review. Therefore, when the method extracted common synonyms, the crowdworkers judged them relevant to the query. However, the proposed method extracted various sentences that were not just synonyms concerning the actually obtained expressions.

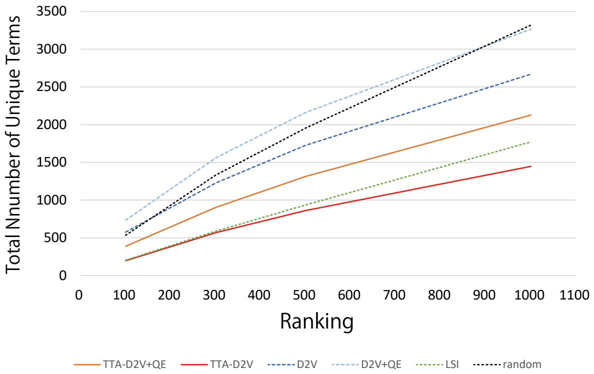

Fig. 5. The number of unique terms that appear in search results up to a ranking section.

For another queried related to the movie content, our methods outperformed the other existing methods. In these search tasks, the methods had to extract specific descriptions or scenes from the movies. **TTA-D2V** obtained many relevant expressions. On the other hand, the method's accuracy was reduced using query expansion. Since the range of expressions became too divergent in these methods, too many irrelevant expressions were included in the results.

Query expansion was particularly effective when the queries were "makes me nostalgic" and "surrealistic". These ambiguous queries tended to be highly diverse expressions. The query "nostalgic" was ambiguous because the query itself has multiple meanings. Young people feel nostalgic about movies they saw in their childhood, and older people long for older movies. Even a new movie might seem nostalgic if it is set in a previous era. A similar trend was found in the query "surrealistic". The perception of the surreal is highly subjective. What people label as surrealistic differs from person to person. Query expansion increased the accuracy of such ambiguous queries by giving them various meanings.

The accuracy of the overall search results will also be discussed. Based on the results of the **random** labeling, these data sets contain roughly seven percent of the sentences relevant to the queries. The search target used in this experiment was 1.2 million sentences; it means that it should contain about 80,000 correct answers. We sampled and evaluated the top 1,000 sentences in the ranking in this experiment. Ideally, the methods would have been able to fill all the 1,000 search results with relevant sentences. Therefore, there is still room for improvement in all the methods using Doc2Vec and LSI.

On the other hand, the number of relevant sentences tended to decrease with a lower ranking for all methods except **random**. Such a phenomenon is commonly seen in the results of information retrieval. The proposed method is based on the distributed representation of queries and sentences, and ranking is by calculating cosine similarity. Vectors do not always reflect everything in a query and sentences. For example, information such as writing style and word order are lost in vectorization. Even if the reviewer writes the review in the wrong word order, interspersed with brief sobs to describe the movie as a "tear-jerker", our approach cannot detect it. Therefore, these sentences could appear throughout the ranking. We need to use a more advanced language model such as BERT (Bidirectional Encoder Representations from Transformers [30]) to find such sentences correctly.

We use examples of actual results to discuss the properties of each method. Table 4 shows the search results for "nostalgic" for which the proposed method worked well. In this query, many methods found many correct sentences.

Since "nostalgic" is an ambiguous query, both sentences referring to the content of the movie and sentences referring to the reviewer's personal experience were labeled by the worker as relevant. Each method using Doc2Vec correctly calculated the relevance of the reviewer's experience, such as "remembering", and the content of the movie, such as "economic growth" and "youth". In the method using target-topic, the model was trained using different reviewers' sentences of the same movie. Therefore, even expressions that do not co-occur with "nostalgic" got high relevance scores.

It is not easy to analyze the elements of a vector using a neural network in detail, but this is possibly caused by the bias of the values in dimensions of the vector. A one-word query might have values concentrated in some dimensions when vectorized. In these cases, shorter sentences are likely to be more relevant. Suppose one reviewer simply described a movie as “nostalgic”, while another reviewer briefly described it as “reminds me of economic growth.” In that case, **TTA-D2V** may have correctly represented these words in a similar vector. Since these trends are query-dependent, an additional detailed analysis comparing vectors will be needed in the future.

7. Conclusion

This paper proposed a Doc2Vec-based method to find actual expressions related to a given keyword query. This research contains two ideas: focusing on target-topics and using query expansion before similarity calculation. Our method produced better ranking quality and more diverse results through a large-scale evaluation with actual movie review data.

Although the proposed method can obtain many evaluative expressions, it does not sufficiently consider the depth of impressions. A practical application is also essential. At the moment, this method is limited to matching the expressions in reviews with queries. In practice, movies must be ranked from queries based on the obtained expressions. To achieve this goal, we need to aggregate the review information for each movie, summarize multiple reviews of different degrees, and sort them by their relevance to the query.

Based on the results in this paper, we are working on a method actually to rank movies [31]. In order to rank movies, it is necessary to aggregate the reviews of many people; it is not enough to simply rank the movies with many reviews that contain relevant evaluation expressions. We also have to determine how to correctly aggregate a movie that ten people described as “sad” and a movie that one person labeled “very sad”. For this reason, superficial cosine similarity is not sufficient. Although we use the learning to rank method, it is still insufficient, and we need to develop the method further.

Table 4. Top five results for all comparison methods for query "nostalgic" (translated)

Method	Rank	Sentence	Relevance
TTA-D2V+QE	1	A sweet and sour memory of school days.	**4.00**
	2	Moist, sad, and soft.	2.00
	3	The cityscape of Sasebo reminds us of our distant past youth.	**4.00**
	4	Nostalgic, reflecting the downtown of Japan during its period of rapid economic growth.	**4.00**
	5	It is hot, sad, and heartwarming work.	1.75
TTA-D2V	1	These adults used to be children.	**2.75**
	2	It reminded me of those days.	**4.00**
	3	I felt a "Godfather" and "Once Upon a Time in America" atmosphere.	**3.25**
	4	Maybe it reminded me of the old days when I was young.	**3.25**
	5	Watching it reminded me of my own childhood.	**3.25**
D2V+QE	1	I liked the ending song, which was very youthful.	2.25
	2	The scenery and trains of that time were recreated, which made me laugh, cry, and warm my heart.	**3.50**
	3	After writing the review, I remembered and cried again.	2.00
	4	A high school student writes the scenario, so there are many embarrassing scenes, but it reminded me of my youth.	**3.50**
	5	I love flamenco songs, so I felt nostalgic for Bolver.	**3.25**
D2V	1	The scene of the song reminded me of the video.	**3.75**
	2	I haven't seen such a hot movie in a long time.	**3.00**
	3	Watching this movie made me want to drive around listening to 60's rock music.	**3.75**
	4	I recently noticed that the main theme was used in a commercial.	2.25
	5	The scene of crossing the bridge with the theme song in the background was very moving.	**4.00**
LSI	1	The time was when I was born.	**3.75**
	2	The time was during the Pacific War.	**3.00**
	3	It reminded me of my childhood.	**3.75**
	4	The character's childhood scene was cute.	2.25
	5	It reminded me of old youth drama.	**4.00**

Table 5. Average number of words per sentence for each query.

Query	TTA-D2V+QE	TTA-D2V	D2V+QE	D2V	LSI	random
Surrealistic	5.00	4.57	9.39	11.06	1.88	13.11
Surprise ending	4.79	3.54	11.86	7.11	2.38	8.25
Familial love	5.38	6.96	8.86	12.90	2.68	9.00
Near-futuristic	7.77	8.11	12.98	12.41	4.61	5.33
Relaxing	3.54	1.94	6.94	4.45	2.54	12.20
Nostalgic	6.52	6.04	7.41	5.88	2.52	7.50
Tear-jerker	4.08	1.38	5.52	3.21	1.58	7.75
Makes me want to go on a trip	8.00	3.60	7.50	12.50	1.58	6.33
Suitable for a date	5.27	3.67	16.67	9.06	4.40	4.71
Rewatchable	3.59	1.82	7.94	6.97	1.43	5.25
Average	5.39	4.16	9.51	8.55	2.56	7.94

Acknowledgements

This work was supported by JSPS KAKENHI Grants Number 22H03905, 21H03774, 21H03775, 18K18161, 18H03243, and ROIS NII Open Collaborative Research 2022 Grant Number 22S1001.

References

1. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.
2. Kosuke Kurihara, Yoshiyuki Shoji, Sumio Fujita, and Martin J. Dürst. Doc2vec-based approach for extracting diverse evaluation expressions from online review data. iiWAS2021, page 1118, New York, NY, USA, 2021. ACM.
3. Arti Buche, M. B. Chandak, and Akshay Zadgaonkar. Opinion mining and analysis: A survey. *International Journal on Natural Language Computing*, 2(3), 39 – 48.
4. Fatemeh Hemmatian and Mohammad Karim Sohrabi. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, 52(3):1495–1545, 2019.
5. Vijay B Raut and DD Londhe. Opinion mining and summarization of hotel reviews. In *2014 International Conference on Computational Intelligence and Communication Networks*, pages 556–559. IEEE, 2014.
6. Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.
7. Jayashri Khairnar and Mayura Kinikar. Machine learning algorithms for opinion mining and sentiment classification. *International Journal of Scientific and Research Publications*, 3(6):1–6, 2013.
8. Sumbal Riaz, Mehvish Fatima, Muhammad Kamran, and M Wasif Nisar. Opinion mining on large scale data using sentiment analysis and k-means clustering. *Cluster Computing*, 22(3):7149–7164, 2019.
9. Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *proceedings of ACL-08: HLT*, pages 308–316, 2008.
10. Kenji Sugiki and Shigeki Matsubara. A product retrieval system robust to subjective queries. In *2007 2nd International Conference on Digital Information Management*, volume 1, pages 351–356, Oct 2007.
11. Libing Wu, Cong Quan, Chenliang Li, and Donghong Ji. Parl: Let strangers speak out what you like. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 677–686, New York, NY, USA, 2018. ACM.
12. Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM

'06, pages 43–50, New York, NY, USA, 2006. ACM.

13. Nan Hu, Paul A. Pavlou, and Jennifer Zhang. Can online reviews reveal a product's true quality?: Empirical findings and analytical modeling of online word-of-mouth communication. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 324–330, New York, NY, USA, 2006. ACM.
14. Vivek Kumar Singh, Rajesh Piryani, Ashraf Uddin, and Pranav Waila. Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pages 712–717, March 2013.
15. Nadeem Bader, Osnat Mokryn, and Joel Lanir. Exploring emotions in online movie reviews for online browsing. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces Companion*, IUI '17 Companion, pages 35–38, New York, NY, USA, 2017. ACM.
16. Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA, 2011. ACM.
17. Jiaxing Tan, Alexander Kotov, Rojiar Pir Mohammadiani, and Yumei Huo. Sentence retrieval with sentiment-specific topical anchoring for review summarization. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 2323–2326, New York, NY, USA, 2017. ACM.
18. Abd Samad Hasan Basari, Burairah Hussin, I Gede Pramudya Ananta, and Junta Zeniarja. Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, 53:453–462, 2013.
19. Lili Zhao and Chunping Li. Ontology based opinion mining for movie reviews. In *International Conference on Knowledge Science, Engineering and Management*, pages 204–214. Springer, 2009.
20. Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou. Movie rating and review summarization in mobile environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(3):397–407, 2011.
21. Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. Mix 'n match: Integrating text matching and product substitutability within product search. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 1373–1382, New York, NY, USA, 2018. ACM.
22. 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 *Proceedings of the Eighth International Symposium on Information and Communication Technology*, SoICT 2017, pages 460–467, New York, NY, USA, 2017. ACM.
23. Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 2105–2114, New York, NY, USA, 2016. ACM.
24. Oren Barkan and Noam Koenigstein. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.
25. Van-Thuy Phi, Liu Chen, and Yu Hirate. Distributed representation based recommender systems in e-commerce. In *DEIM Forum*, 2016.
26. Gaojun Liu and Xingyu Wu. Using collaborative filtering algorithms combined with doc2vec for movie recommendation. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 1461–1464. IEEE, 2019.
27. Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
28. J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

29. Patrick E. Shrout and Joseph L. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420, 1979.
30. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
31. Kosuke Kurihara, Yoshiyuki Shoji, Sumio Fujita, and Martin J. Dürst. Learning to rank-based approach for movie search by keyword query and example query. In *The 23rd International Conference on Information Integration and Web Intelligence*, iiWAS2021, page 137145, New York, NY, USA, 2021. ACM.