



Finding Informative Comments for Video Viewing

Seungwoo Choi¹ · Aviv Segev²

Received: 6 July 2019 / Accepted: 4 November 2019 / Published online: 21 November 2019
© Springer Nature Singapore Pte Ltd 2019

Abstract

Of all the information-sharing methods on the Web, video is a factor with increasing importance and will continue to influence the future Web environment. Various services such as YouTube, Vimeo, and Liveleak are information-sharing platforms that support uploading UGC (user-generated content) to the Web. Users tend to seek related information while or after watching an informative video when they are using these Web services. In this situation, the best way of satisfying information needs of this kind is to find and read the comments on Web services. However, existing services only support sorting by recentness (newest one) or rating (high *LIKES* score). Consequently, the search for related information is limited unless the users read all the comments. Therefore, we suggest a novel method to find informative comments by considering original content and its relevance. We developed a set of methods composed of measuring informativeness priority, which we define as the level of information provided by online users, classifying the intention of the information posted online, and clustering to eliminate duplicate themes. The first method of measuring informativeness priority calculates the extent to which the comments cover all the topics in the original contents. After the informativeness priority calculation, the second method classifies the intention of information posted in comments. Then, the next method picks the most informative comments by applying clustering methods to eliminate duplicate themes using rules. Experiments based on 20 sampled videos with 1000 comments and analysis of 1861 TED talk videos and 380,619 comments show that the suggested methods can find more informative comments compared to existing methods such as sorting by high *LIKES* score.

Keywords Video service · Information sharing · Information needs · Online comments · Informativeness

Introduction

In the past decade, people have become more familiar with accessing information using the Web. As part of the changes of Web 2.0, people are using the Web in various ways. Of all the ways to use the Web, information sharing using video is an important method which has rising influence on the Web environment. For this purpose, there are various services, such as YouTube, Vimeo, and Liveleak, which can support uploading user-generated content (UGC) to the Web. Also, people are able to access massive open online courses

(MOOCs) using video services such as Coursera. Additionally, there is a commonly used conference video sharing service, TED, which supports the sharing and spreading of ideas on the Web. Previous research [15, 46] observed that users tend to seek related information while or after information behavior such as watching an informative video. Users in various services leave comments including opinions [17, 31], discussion [50], additional links [28], and so on. The first step in satisfying these information needs is to find and read the comments on Web services. YouTube provides several ranking methods for comments such as ranking from the video creator, comments generating discussion from the viewers, and comments that have been voted up by the community since November 2013.¹ However, most existing services support only sorting by recentness (newest one) or by high rating (*LIKES* score). Therefore, the search for related information has limitations:

✉ Aviv Segev
segev@southalabama.edu
Seungwoo Choi
sw.choi@kaist.ac.kr

¹ Knowledge Service Engineering, KAIST, 291 Daehak-ro, Daejeon, South Korea

² Department of Computer Science, University of South Alabama, 150 Jaguar Drive, Mobile, AL, USA

¹ View, organize, or delete comments—YouTube Help.
Available: <https://support.google.com/youtube/answer/6000976?hl=en>
(Date last accessed on 4 Oct 2019).

- If a video receives hundreds of comments, then users find it very hard to refer to all the comments when looking for related information. Therefore, users prefer to consider only highly ranked comments.
- In the sorting method based on a high *LIKES* score, comments that are more popular among users tend to be more visible because of their high ranking than comments of higher quality.
- The highly ranked comments can easily remain highly ranked because of their exposure and easy access, since users prefer only highly ranked comments when the number of comments is large.
- Unless users read all the comments, there is no other way for them to identify the informative comments.

We propose a method to find and select informative comments (IC-Finder) to help users understand original contents and supply additional related information. We suggest several methods based on the analysis of user behavior with comments and integrate different components to find the most informative comments for situations where there are a large number of comments. The method classifies the important features identified by the users. For each of these features, an algorithm was developed or integrated to quantify them. The methods are based on the analysis of 1860 videos and 380,619 comments from TED. Previous recommendation approaches are based on recording user behavior [3, 18, 25, 52]. However, our approach is based on the actual information contained in the video, the contents of the comments, and meta-information. The methods are composed of measuring informativeness priority, classifying intention for information, and clustering to eliminate duplicate themes. The method of measuring informativeness priority calculates how much the comment refers to information coverage in the original contents. After the informativeness priority calculation, the method classifies the comment intention for information—whether they trigger or respond to comments. Then, the method selects the most informative comments by applying clustering methods to eliminate duplication of themes using rules.

To verify the method, we randomly selected 20 videos from the TED video service and used human evaluators to judge the data. We compared the informative comments selected by human evaluators to the commonly used method of high rating (*LIKES* score) as a baseline and to the IC-Finder method.

The goal of this research is to find the most informative comments based on content in contrast with the ordinary services which simply use ordered score based on popularity or recentness. The main contributions of our approach are as follows:

- The method supplies users with related informative comments that provide useful information or give a better insight into the original contents.
- The method to select informative comments is based on matching between original contents and the crowd intelligence rather than the crowd numbers.
- The large-scale experiments based on real-world data analyze both the users' preference in selecting the best comments and how to best provide the users with the most informative comments.
- The method shows an extended algorithm for a recommendation system by incorporating user perspective based on commenting behaviors.

The remainder of the article is organized as follows. Section “[Related Work](#)” introduces the related research. In Section “[IC-Finder Method](#)”, we describe our IC-Finder method for finding informative comments. In Section “[Experiments](#)”, we present the experiments and results which evaluate performance compared to human evaluators. Finally, Section “[Discussion](#)” discusses the main contributions, and Section “[Conclusion](#)” presents the conclusion and future works.

Related Work

Specialized Video Services

Specialized video service is broadcast video aimed at a specific purpose like sports, idea sharing, and education. In Korea, there is a famous live sports video service, “Naver Sports TV”. This service supports live chatting during the game. Ko et al. [22] investigated the motives for using this service and the relations between motives and usage. Another video conference service for idea sharing is TED. Multiple studies regarding the use of the TED service have been conducted on topics of lecture recommendation [29], video recommendation [55], automatic quiz material generation for education [19], video skimming [43], and statistical machine translation (SMT) [6, 30] based on the multilingual scripts of videos provided by the service. In addition, researches about humor [13] and distinguishing between native and non-native speakers [26] were conducted. In the last few years, massive online open courses (MOOCs) for education, such as Coursera, have received attention as a new education model. However, education experts disagree with the contention that MOOCs are a breakthrough and replacement for current classroom education [32]. The increase of MOOC qualities is analyzed in various ways, such as dealing with system architecture issues [11, 36] and deriving criteria for better satisfaction [42, 51]. Recommendation by characterizing personal behavior has been used for

course recommendation [1], intelligent feedback management [40] based on text mining, and thread [50] and question recommendation [49] based on characterizing personal behavior. Previous research focused on thread and question recommendations and not on the quality of the answers.

Comments Analysis

Some researches focus on comment content. The common criteria for evaluating comments are quality, usefulness, and helpfulness. All criteria can be viewed as having similar meaning, but were analyzed for different purposes. Figueiredo et al. [16] investigated the quality of textual features including comments, title, description, and tags in Web service. The criterion of helpfulness is usually used in online commerce. Zhang and Tran [53] proposed a method to predict the helpfulness of reviews in e-commerce Web service based on entropy-based scoring. Xiong and Litman [48] proposed a summarization method that employs a review of helpfulness ratings for content selection. Momeni et al. [27] suggested a usefulness classifier based on features such as surface level, syntactic, semantic, and topic. A topic-focused trust model [54] measured credibility of users and tweets by applying heterogeneous contextual properties. Comment analysis was used for crisis knowledge representation [38] and humanitarian assistance in crisis response [20]. Emotion classification [8] and sentiment analysis [4] on YouTube videos were performed by utilizing the video comments. Ghose and Ipeirotis [17] suggested a ranking mechanism in two aspects, a consumer-oriented aspect and a manufacturer-oriented aspect. These studies revealed that ranking the comments or contents depends on adding or applying the features or factors but did not describe how to evaluate them. Additionally, these research studies revealed the lack of the study of the human perspective.

Community on Question and Answering(CQA)

In CQA service, questions are expressed explicitly with the user interface. Researches in CQA systems have also focused on the answer. Some of the researches have tried to discover which features influence the quality of the answers [5, 21, 39]. Another approach for finding the related video answer is based on the user's question [24]. Since there are considerable differences between the user interfaces, our research is based on recognizing the user's information intention.

Many of the research works try to match between question and answer. Cong et al. [10] proposed to detect the answers to questions based on graph propagation. Wang et al. [45] suggested a method to match between new question and answer pairs based on link prediction. Another method is based on semantic relevance using deep learning [44]. Prior work deals with multi-topic environments. However, question and answer

comments on social video service environments usually focus on a single topic.

IC-Finder Method

Overview

In our previous study [9], we investigated why users leave comments and which features users are more satisfied with based on the TED video service. We chose some significant features to be used for designing the method based on previous results:

- Users tend to prefer comments referring to well-balanced information.
- There are two information intentions, which are *Trigger* and *Respond*.
- There can be various themes in the comments based on each user intention because of several reasons such as personal experience or opinion.

A *Trigger* comment is defined as a comment whose purpose is to get information from the other users by triggering comments. A *Respond* comment is defined as a comment whose purpose is to answer a question from others by supplying information.

The measure of connection between videos and comments is made by comparing textual similarity in our method. In recent research by Krishnamoorthy et al. [23], they suggested a technique that automatically generates natural language descriptions for videos using text-mined knowledge. Therefore, we assumed there are existing solutions of voice to text.

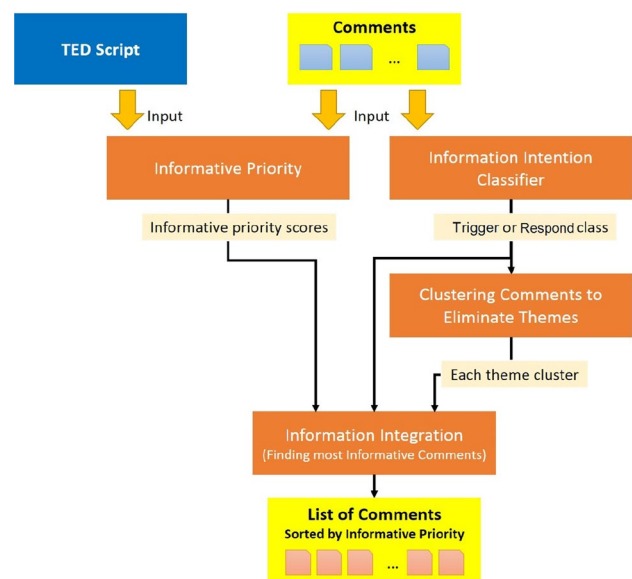


Fig. 1 IC-Finder method

Figure 1 shows the IC-Finder method diagram for finding informative comments. First, we considered an information coverage approach, named *Semantic Entropy*, for calculating the informative priority score. Second, we developed an information classifier algorithm, which recognizes the intention of each comment. We applied the algorithm in our method and evaluated the algorithm in the “Experiments” Section. Third, for each intention, *Trigger* and *Respond*, we clustered the comments into themes. Despite the large number of themes, we can eliminate duplicate themes using the cluster approach. Last, we prioritized the selected comments using rules based on the informativeness score. Algorithm 1 shows the pseudo-code overview of the IC-Finder method.

Algorithm 1: Psuedo-Code for ICFinder Method

Data: C, C denote each comment set in specific video
Result: Selected comments
 initialization \leftarrow Split script & load the Intention Classifier
begin
 $S \leftarrow \text{LoadingSplitScript}(\text{SpecificVideoInfo})$
 $\text{WeightMatrix} \leftarrow \text{GetSimilarities}(C, S)$
 $\text{InformativePriority} \leftarrow \text{GetInformativenessScore}(C, S, \text{WeightMatrix})$
 $\text{InformationIntentionClass} \leftarrow \text{GetClass}(C, \text{Classifier})$
 $\text{EachCluster} \leftarrow \text{GetEachClassCluster}(C, \text{InformationIntentionClass})$
 $\text{SelectedComments} \leftarrow \text{Integrating}(C, \text{EachCluster}, \text{InformativePriority}, \text{InformationIntentionClass})$
end

Informative Priority Based on Semantic Entropy

Based on the qualitative study [9], information coverage in comments is one of the most important factors for identifying comment informativeness. If the comments are too detailed in comparison to the original contents, then the user feels dissatisfaction. To avoid choosing comments which have a concentration of detailed information in the script, we first parse the target video script, for which we seek additional information, into sentences. Then, we generate a weight matrix relating each sentence and each comment with variables representing a weight score using latent semantic analysis (LSA) [14]. Algorithm 2 shows how we generate the weight matrix.

Algorithm 2: GetSimilarities

Data: C, S
Result: WeightMatrix
begin
 $\text{WeightMatrix}[C.\text{length}][S.\text{length}] \leftarrow 0$
 for $c \in C$ **do**
 for $s \in S$ **do**
 $\text{WeightMatrix}[c][s] \leftarrow \text{SimilarityScorebyLSA}(c, s)$
 end
 end
 Return WeightMatrix
end

Next, we borrow the concept of entropy from information theory [2]. In information theory, entropy represents how much information is balanced in each event, message, category, or class. Entropy is also best known as a measurement of uncertainty. Uncertainty means the

tendency of probability distribution in the event. If one of the events occurs more than the others, then this situation looks less informative when the occurring event is observed. Conversely, other events occur rarely. In this case, entropy (uncertainty) is lower than in other cases. Therefore, entropy will be close to zero when only one certain event is expected. Equation 1 shows the traditional entropy formula.

$$\text{Entropy} = \sum_i -P_i \log P_i, \quad (1)$$

where P is the occurrence probability of each event

With the weight matrix we can easily calculate the entropy for each comment. Equation 2 shows how we calculate the probability.

$$P_i = \frac{W_i}{\sum_{i \in S} W_i}, \quad i \in S \quad (2)$$

where S is a parsed sentence, W is each weight relating each sentence and comment, i is a specific identifier of each comment.

Then, we organize entropy sets for each comment. Algorithm 3 shows the entire procedure for calculating the entropy.

Algorithm 3: GetInformativenessScore Algorithm

Data: $C, S, \text{WeightMatrix}$
Result: $\text{InformativePriority}$
begin
 $\text{EntropySet} \leftarrow 0$
 for $c \in C$ **do**
 $\text{SUM} \leftarrow \text{SummationAllWeight}(c)$
 $\text{CommentSemanticProbabilities} \leftarrow \text{WeightMatrix}(c) / \text{SUM}$
 $\text{Entropy} \leftarrow 0$
 for $p \in \text{CommentSemanticProbabilities}$ **do**
 $\text{Entropy} = \text{plog}(p)$
 end
 $\text{EntropySet} += \text{Entropy}$
 end
 $\text{InformativePriority} \leftarrow \text{SortedEntropySetByDesc}$
 Return $\text{InformativePriority}$
end

Information Intention Classifier

It is important to classify the comment intention, which will allow us to use different intention approach methods, such as Q&A, based on the qualitative study [9]. To organize the information intention classifier, we picked comments randomly and made an answer set using a human evaluator. Then, we trained the classifier using the answer set. Finally, we applied the classifier to judge the comment intention. We assume that the classifier should get a high accuracy rate, and we verify this assumption empirically in Section “Experiments”. Algorithm 4 shows the overall classification procedure pseudo-code.

Algorithm 4: GetClass Algorithm

Data: $C, \text{Classifier}$
Result: $\text{InformationIntentionClassSet}$
begin
 $\text{InformationIntentionClassSet} \leftarrow 0$
 for $c \in C$ **do**
 $\text{result} \leftarrow \text{Classifier}(c)$
 $\text{ClassSet} \leftarrow \text{result}$
 end
 Return $\text{InformationIntentionClassSet}$
end

Clustering Comments to Eliminate Themes

Despite the successful information intention classification, we still have to deal with various themes in the comments. More processing is needed because most of the comments are talking about similar topics, which could trigger an error. To eliminate the duplication of similar themes, we used a clustering method to find the duplicated groups. Before clustering, we preprocessed all comments on TED for each specific video. We used the vector space model [34]. This approach extracts features (terms) from documents as a vector. Each feature corresponds to a unique term weight. A well-known term-weighting approach in information retrieval is TF-IDF [33], which consists of term frequency (TF) and inverse document frequency (IDF). This approach considers the use not only of term frequency but also term importance, whether the term is common or uncommon in the entire document set, using IDF. Here is the procedure for extracting vectors from TED comments.

1. We eliminated stopwords and stemmed all comments. In our method, the rest of the words are mentioned as *STOPSTEM* words.
2. We treated *STOPSTEM* words in each comment as a document and calculated the TF-IDF weight set for each document.
3. We sorted each TF-IDF weight set by descending order in each document.
4. We picked the top five *STOPSTEM* words with the highest TF-IDF score in each document. If there are not enough words for picking five *STOPSTEM* words, we only picked top K *STOPSTEM* words.
5. We gathered all features from each *STOPSTEM* word and generated the data set using these features and TF-IDF score pair.

After we obtained the data set for making a cluster, we used WEKA² API for the expectation-maximization (EM) [12] clustering algorithm. The EM clustering algorithm is

² Weka 3: Data Mining Software in Java, Expectation-Maximization API. Available: <http://weka.sourceforge.net/doc.dev/weka/clusters/EM.html> (Date last accessed on 4 Oct 2019).

well-known and extensively used for clustering. EM uses a probability distribution to assign the words belonging to each of the clusters. Also, EM can choose how many clusters to generate using cross-validation. To verify the number of clusters, we set the number of folds to 100 and the maximum iteration number to 500. Algorithm 5 shows the entire procedure for making the cluster.

Algorithm 5: GetEachClassCluster Algorithm

Data: $C, \text{InformationIntentionClass}$
Result: EachClusterInfo
begin
 $\text{EachClusterInfo} \leftarrow 0$
 for $\text{class} \in \text{IntentionClass}$ **do**
 $\text{LoadEachClass} \leftarrow \text{class}$
 $\text{VectorSpaceModel} \leftarrow \text{ModelGeneratorTFIDF}(C, \text{class})$
 $\text{EachClusterInfo} \leftarrow \text{BuildingCluster}(\text{class}, \text{VectorSpaceModel})$
 end
 Return EachClusterInfo
end

Information Integration (Finding Most Informative Comments)

Last, we integrated all results from previous steps. We ordered comments by their informative priority, their information intention, and the number of clusters generated for each information intention. Then we selected the top five comments by integrating rules.

1. We consider each portion of the class *Trigger* and *Respond* to analyze the information intention by volume. We assume that the intention class with the greater volume is more important. If one class accounts for more than 80% of the total comments, we picked four comments in this class. However, if the difference of class volume is lower, then we picked three comments in the bigger class. In each class, the method picks one comment from the largest cluster volume until its entire class portion is finished.
2. In each cluster the method picks one comment having the highest informative priority. Therefore, we can select the best comment that covers all video information in various and duplicate themes.
3. If the comment is classified as *Trigger* class, then the method should find its answer comments, which are classified as *Respond* class, and unite the *Trigger* and *Respond* class.

We design the IC-Finder method based on the considerations from the qualitative study. Algorithm 6 illustrates the procedure in pseudo-code.

Algorithm 6: Integrating Algorithm

Data: C , $EachCluster$, $InformativePriority$, $InformationIntentionClassSet$
Result: $EachCluster$

```

begin
  SelectedComments ← Null
  for class ∈  $IntentionClass$  do
    PortionOfClass ←  $GettingPortion(class)$ 
    NextCluster ← Null
    while PortionOfClass != 0 do
      if NextCluster is Null then
        | NextCluster will be chosen by largest volume
      else
        | NextCluster ←  $GetNextVolumnCluster(NextCluster)$ 
      end
      InformativeComment ←
         $MostInformativeCommentInCluster(C, NextCluster, InformativePriority)$ 
      AnswerComment ← Null
      if InformativeComment = Trigger then
        | AnswerComment =  $FindAnswer(InformativeComment)$ 
      else
        | AnswerComment = empty
      end
      SelectedComments ←  $InformativeComment + AnswerComment$ 
      PortionOfClass --
    end
  end
  Return SelectedComments
end

```

Algorithm 7 is the method for choosing the proper answer for *Trigger* comments, which we defined as asking other users questions. This method also uses the LSA similarity API. To explain our method, we should describe the thread environments in the comments. In the TED service, the system supports comments and their reply as a thread. The thread means comments with their answer directly input by the user interface. If an informative comment from the IC-Finder method is in the thread, which means a set of messages grouped, then IC-Finder will find the answer comment in the thread. In this situation, the answer comment has to be *Respond* class and obtain highest similarity with the original *Trigger* comment. However, IC-Finder tries to search all comments if the original *Trigger* comment does not have a thread. In this situation, IC-Finder should find comments with two conditions. The first condition is to have a highest similarity score by LSA API. The second condition is to be posted later than the original *Trigger* comment.

Algorithm 7: Find Answer Algorithm

Data: $InformativeComment$
Result: $AnswerComment$

```

begin
  if InformativeComment is in Thread then
    | Return  $MostHighSimilarityAnswerinThread$ 
  else
    for RespondComment ∈ Respond do
      SimilarityScore ←
         $SemanticSimilarity(InformativeComment, RespondComment)$ 
      if MAX(SimilarityScore) & Later than This(RespondComment) then
        | Return RespondComment
      end
    end
  end
end

```

Experiments

As described in Section “[IC-Finder Method](#)”, IC-Finder consists of three elements: measuring informativeness priority, classifying intention for information, and clustering to eliminate duplicate themes. First, we analyze classification

accuracy for information intention. Second, we evaluate the effectiveness of the method in applying and organizing clusters. Last, we deepen the understanding of the results to observe which characteristics can be found in comments.

To analyze the performance, we crawled 1861 videos and 380,619 comments until October 31, 2014, and sampled 20 videos from TED which have at least 200 comments and are at least 10 minutes long. Our IC-Finder method is based on sufficiency of comments. Otherwise, we can encounter the cold start problem [35], which appears when a system has insufficient information.

To show how much the user interacts in the TED service, we analyzed the crawled dataset. The average length of all videos’ script is 11,701 characters (min: 81, max: 34,781, SD: 5,569) and 2109 words (min: 14, max: 6,432, SD: 1,010) except for non-script videos. The average number of comments for each video is 205 (min: 8, max: 6,447, SD: 288). Using the language detection library for Java,³ we found that 367,672 comments (96.6%) are in English. We analyzed the length of the English comments: the average length of comments for all videos with comments in English is 493 characters (min: 2, max: 19,777, SD: 473) and 85 words (min: 1, max: 3158, SD: 81).

The average length of 20 sampled videos’ script is 16,245 characters (min: 9483, max: 23,980, SD: 3491) and 2,938 words (min: 1653, max: 4,443, SD: 654). In 20 sampled videos, each video expects to have 512 comments as the average number (min: 216, max: 1192, SD: 255). In all comments of 20 sampled videos, the average length of comments is 575 characters (min: 1, max: 1286, SD: 493) and 97 words (min: 1, max: 2007, SD: 84). Comparing the sampled videos to the entire data set described, we can see that the sampled set accurately represents TED talks.

The TED.com website provides a link to other videos with similar topics. The main topics include *Culture*, *Science*, *Children*, and *Technology* in addition to topics suggested also by TED such as *Statistics* and *Economics*.

Classification Accuracy for Information Intention

Experiment Setting

For classification of information intention, we randomly extracted 50 comments from each of the previously selected 20 videos. Two evaluators divided the 1000 comments into 2 sets and coded each set separately. To evaluate human agreement, we selected 2 videos and calculated Kappa inter-rater agreement score. Overall Kappa score is 0.74.

³ *Language Detection Library for Java*. Available: <https://code.google.com/archive/p/language-detection/> (Date last accessed on 4 Oct 2019).

Explanation for Features

In the preliminary study [9], we extracted some features from the user study. We added more features to improve the accuracy and enable empirical evaluation. Figure 2 shows the overview of all features. We chose features from the user study that we can extract computationally. The features include: **URL**, **New Information**, **Additional Opinion or Thought**, **Length** and **Explicit Information Source**.

1. URL

- How many times *URL* appears in the comment using regular expressions.

2. New Information

- How many *Questions* appear in the comment using regular expressions.

Question form 5W1H (When, Where, What, Who, Why, How) and modal verbs (Can, May, Could, Might, Should, and so on).

Implicit question form like “I have a question”, “right?”, “I guess ...”, “Anyone who ...”, “I wonder ...”, “My question ...” and “One question ...”.

- How many times *Explicit Answers* appear in the comment using regular expressions.

Answer form like “That’s good question”, “The answer ...”, “I answer ...”, “I agree ...” and “I disagree ...”. Yes and no answer words which appear in the beginning of the sentence.

- How many times *Email* appears in the comment using regular expressions.

3. Additional Opinion or Thought

- How many subjects *I* appear in the comment using Stanford POS Tagger.⁴
- How many Subjects *You* appear in the comment using Stanford POS Tagger.
- Identify *opinion* in the online review using *Syntactic Patterns* [41].
- How many *Greetings* appear in the comment using Regular Expressions. Basic greeting starters are “Hi”, “Hello”, and “Dear”. Simple compliment form like “Thanks” and “Thank you”.

4. Length

- *Length* of comment.
- *Sum of all parsed sentences length* in comment.
- How many *sentences* in comment.

- How many *token words* appear in the comment after processing with Stanford POS tagger.

5. Explicit information source

- How many *Quotations* appear in the comment using Regular Expressions.
- How many *Parentheses* appear in the comment using regular expressions.
- How many *Colons* appear in the comment using regular expressions.
- How many *Named Entities* appear in the comment using Stanford Named Entity Recognizer.

These are **Basic Features** from the qualitative study. Other quantitative features were added.

1. Language Feature

- How many *Nouns* appear in the comment using Stanford POS tagger.
- How many *Adjectives* appear in the comment using Stanford POS tagger.
- How many *Adverbs* appear in the comment using Stanford POS tagger.
- *Sum of each sentiment score for parsed sentences* in the comment using Stanford CoreNLP.

2. Comments Meta-Feature

- Whether there is a *parent comment* for the target comment in the thread.
- *User level* value.
- *LIKES* score value.
- How many comment *Replies* appear in the thread.
- Boolean value for *Deleted* option.

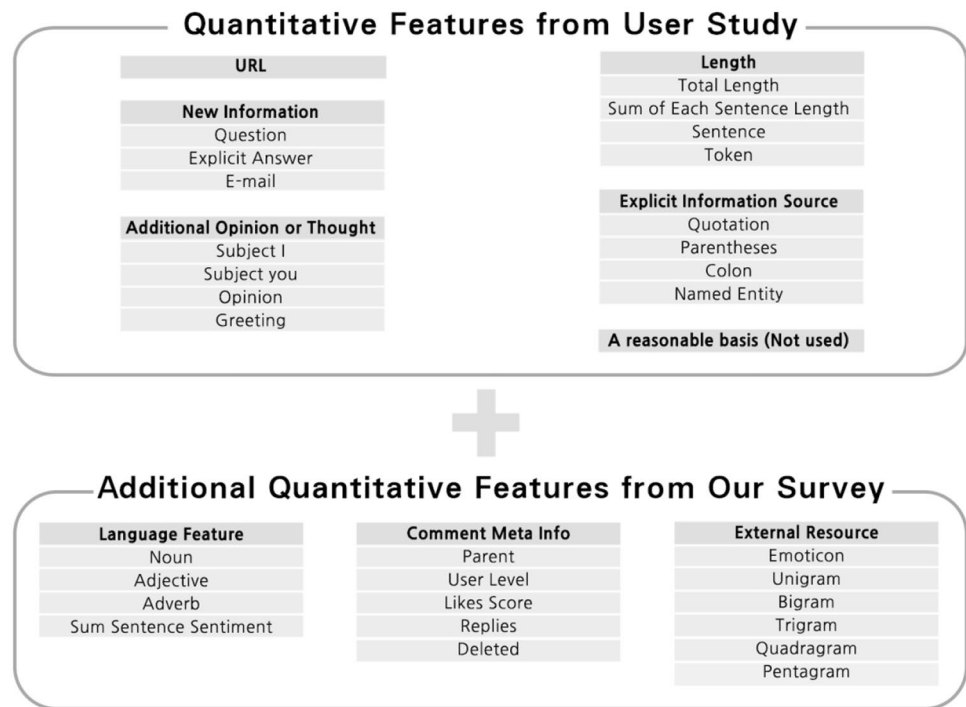
3. External Resource Feature

- How many *Emoticons* appear in the comment. We used the emoticon data sets⁵ which provide analysis of emoticons in over 96 million Tweets from Twitter APIs. The tokenized processing shows 2241 emoticons.
- How many times *N-gram words* appear in the comment. N-gram words contain the 1 million most frequent 2, 3, 4, and 5 gram⁶ and provide over 2.7 million 1-gram (i.e., unique words) words from 400 million words of text from 1810 to 2009.

⁴ Software—The Stanford Natural Language Processing Group. Available: <http://nlp.stanford.edu/software/index.shtml> (Date last accessed on 4 Oct 2019).

⁵ Emoticon Analysis. Available: <http://www.datagenetics.com/blog/october52012/index.html> (Date last accessed on 4 Oct 2019).

⁶ N-grams: based on 520 million word COCA corpus. Available: <http://www.ngrams.info/> (Data last accessed on 4 Oct 2019).

Fig. 2 All features for classification

Classification Result

Decision Tree [47] and LibSVM [7] were used as classifiers based on their popularity and performance. The volume difference in the classification of intention from the human evaluators was between *Trigger* (25.8%) and *Respond* (74.2%). To overcome this difference, we generated the baseline using the ZeroR algorithm and the Decision Tree algorithm (J4.5) with WEKA library⁷. LibSVM was used with kernel-type set as radial basis function, degree in kernel function of 3, and $\gamma = 1/(\text{number of features})$. Experiment results conducted using the basic feature and the additional features with Decision Tree are displayed in Table 1 and experiment results with LibSVM appear in Table 2. The results show that the Decision Tree outperformed the LibSVM. Therefore, Decision Tree was used in the following classification experiments.

The goal of the experiment was to check discernment power of suggested features for classification tasks and show preliminary results. We used both algorithms, Decision Tree and LibSVM, to show that we can achieve high accuracy without using more complex algorithms such as RandomForest and XGBoost. The parameter tuning was based on features originated from a qualitative study [9].

The results show over 92% accuracy. However, all of the additional features are ineffective in classification

improvement. To analyze which basic feature contributes to accuracy, we analyzed the results with attribute selection using *InfoGain* (Eq. 3) and *GainRatio* (Eq. 4) up to the top 19 elements, not including 0.

$$\text{InfoGain}(\text{Class}, \text{Attribute}) = H(\text{Class}) - H(\text{Class}|\text{Attribute}) \quad (3)$$

$$\text{GainRatio}(\text{Class}, \text{Attribute}) = (H(\text{Class}) - H(\text{Class}|\text{Attribute})) / H(\text{Attribute}) \quad (4)$$

Table 3 shows the attribute selection score based on *InfoGain* and *GainRatio* [2]. It shows that a large portion of the classification ability depends on the **Count Question** attribute which counts how many times a question appears in the comment.

Effectiveness of the Methods

We used the 20 sampled videos to show the effectiveness of the methods. The baseline used is the sorted comments *LIKES* score in TED.com. If the *LIKES* scores were the same, then we picked the more recent one.

Settings

We picked the top five ranked comments according to the *LIKES* scores and according to our IC-Finder method. If the comment appeared in both the baseline method and our method, then we discarded the comment from both methods

⁷ Weka 3: Data Mining Software in Java. Available: <http://www.cs.waikato.ac.nz/ml/weka> (Date last accessed on 4 Oct 2019).

Table 1 Classification accuracy with Decision Tree

	Precision (%)	Recall (%)	<i>F</i> measure (%)	Accuracy (%)
Algorithm with ZeroR as baseline				
Baseline	55.1	74.2	63.2	74.2
Features				
Basic feature	92.6	91.6	91.8	91.6
+ Language feature (Lang)	92.5	91.6	91.8	91.6
+ Meta-feature (Meta)	91.1	90.7	90.8	90.7
+ External source feature (Ext)	92.2	91.4	91.6	91.4
+ Lang, Meta-features	91.2	91.1	91.1	91.1
+ Meta, Ext features	91.2	91.0	91.1	91%
+ Lang, Ext features	92	91.2	91.4	91.2
+ All features	89.4	89.5	89.5	89.5
Algorithm with attribute selection				
Decision Tree	93.7	92.3	92.6	92.3

Table 2 Classification accuracy with LibSVM

	Precision (%)	Recall (%)	<i>F</i> Measure (%)	Accuracy (%)
Algorithm with ZeroR as baseline				
Baseline	55.1	74.2	63.2	74.2
Features				
Basic feature	83.9	83.1	80.7	83.1
+ Language feature (Lang)	82.6	81.5	78.2	81.5
+ Meta-feature (Meta)	82.3	81	77.4	81
+ External source feature (Ext)	81.9	80.7	76.9	80.7
+ Lang, Meta-features	82.8	80.5	76.2	80.5
+ Meta, Ext features	83.2	80.4	75.9	80.4
+ Lang, Ext features	83.3	80.8	76.6	80.8
+ All features	82.8	79.9	75.1	79.9
Algorithm with attribute selection				
LibSVM	90.1	90.2	90.1	90.2

and tried to pick an alternative comment. We excluded the common comments since we wanted to measure only the effectiveness of our method by eliminating the high *LIKES* score effects. Each video has ten comments for experiments. Then, we generated an interview form with the randomly ordered comments with a Likert-type scale of 4 levels ranging from 0 (Not Useful) to 3 (Very Useful). For the experiment, we recruited five evaluators in addition to the researcher and trained them using the findings from the qualitative study. The overall agreement of the five evaluators and one researcher based on Kappa score was 0.697. We used only six evaluators to evaluate the machine performance. Nonetheless, the results indicate that our methodology can achieve results comparable to the human perspective. Next, we tried to distribute randomly selected videos with similar playtime for each participant. After watching each video, they filled out their interview form.

Experiment Metric

For the qualitative analysis between the baseline method and the IC-Finder method, we used *Cumulative Gain* (CG) and *Discounted Cumulative Gain* (DCG). CG shows the absolute number of scores evaluating the impact of comments for both the baseline method and our method. DCG is used to measure the ranking quality based on multiple aspects and absolute informativeness score. We tried to use another sorting strategy, which consists of two criteria, sort by cluster volume and sort by *Semantic Entropy* score, in our method.

Results

We used 200 comments in 20 videos. The results show the method's advantages of incorporating user perspectives for recommending comments in watching video. Figure 3a shows the result of sum of CG. As can be seen, the absolute informativeness score of the IC-Finder method is higher than

that of the baseline by 11.56%. Figure 3b shows the result of the sum of DCG between baseline and two trials by different sorting strategies in our method. As seen in the graph, the method of sorting by *Semantic Entropy* outperforms *Cluster Volume* and baseline by 3.22% and 6.84% respectively.

We analyzed the rank position @*K* in the graph. @ represents the top positions and *K* the number of positions evaluated. Each ordinate contains the sum of CG and DCG scores for the baseline and for our method up to position (@*K*). Figure 4a displays the results for CG and Fig. 4b for DCG. The results of the top positions such as first (@1) or second (@2) show slightly better performance. However, the results of low positions such as fourth (@4) and fifth (@5) outperform the baseline. The sorting method with semantic entropy especially shows more effectiveness than cluster volume does.

Exploratory Case Study

We analyzed the results of the performance difference between our method and the baseline. We divided the results in each video using the CG score. We use ▲ for our method having a higher score, ▼ for the baseline having a higher *LIKES* score, and = for the same score. The number of ▲ cases is ten, and the number of ▼ cases is seven. The same score cases (=) is three.

We analyzed the similar topic terms for this case study. We picked the topic terms by the number of occurrences of the selected top five ▲ cases and top five ▼ cases. As we mentioned, the subjects in ▲ cases have more objective topic terms such as *Science*, *Biology*, and *Technology*. However, the subjects in ▼ cases have more subjective topic terms such as *Education*, *Children*, and *Health*. We assumed that both cases might show different results in information intention. Therefore, we applied the classifier from previous experiment to this experiment, and we calculated the number of comments and their ratio in each intention, *Trigger* and *Respond*.

As in Section “Results”, we already used information intention for 1000 sample comments in Section “Classification Accuracy for Information Intention”. The number of *Trigger* intention is only 258 and the number of *Respond* intention is 742. If the number of *Trigger* intention is divided by the number of *Respond*, then we obtain 0.3478. We used this number as a baseline for evaluating the number difference between ▲ and ▼ cases. Table 4 shows the results. All of *Trigger* and *Respond* rate results in ▲ cases show above the baseline score. However, two of results in ▼ cases show below the baseline score. It means that there is a possibility that our algorithm performs less well due to the lack of cases of *Trigger* information class.

Discussion

In Section “Classification Accuracy for Information Intention”, we discussed the information intention classifier based on identifying the *Trigger* and *Respond* intention by combining quantitative features from the user study and additional features from our survey. The results in Section “Classification Accuracy for Information Intention” show maximum classification accuracy of 92.3% using a decision tree. However, eliminating most of the features using the attribute selection method improves the precision, recall, and F1-measure. To quantify importance of the features, we evaluate *InfoGain* and *GainRatio* score for each feature. As a limitation, the result of feature importance reveals that the classifier for the information intention depends greatly on the value of *Question Count* feature. We assume that features related to the definition of information intention could influence the results. The discovery of these features is left for future work.

Analysis of Figs. 3b and 4a shows the advantage of the methods proposed. Our methods showed better results for top *K* up to lower limit of top 5 comments. From top 1 to top 4 comments, the difference of CG and DCG score is lower than the score difference in the top 5 comments. This means that the method of sorting high *LIKES* scores might be capable of showing informative comments. However, it is not valid for low top *K* values. Conversely, our method is able to find informative comments.

Table 3 Result with *InfoGain* and *GainRatio* at top 19

Attribute	Scores	
	<i>InfoGain</i>	<i>GainRatio</i>
Question	0.5423	0.44
Sentence	0.0411	0.047
Subject <i>You</i>	0.0327	0.0391
Unigram	0.0304	0.0304
Token	0.03	0.0301
Total length	0.0297	0.0297
Trigram	0.0291	0.0293
Sum of each sentence length	0.029	0.029
Bigram	0.0283	0.0284
Sum of sentence sentiment	0.0252	0.0257
Quadragram	0.0244	0.0256
Opinion	0.0241	0.0385
Adverb	0.0219	0.0281
Noun	0.0217	0.0219
Explicit answer	0.0211	0.0211
Pentagram	0.0197	0.0197
Adjective	0.0171	0.027
Quotation	0.013	0.0264
<i>LIKES</i> score	0.0123	0.0684

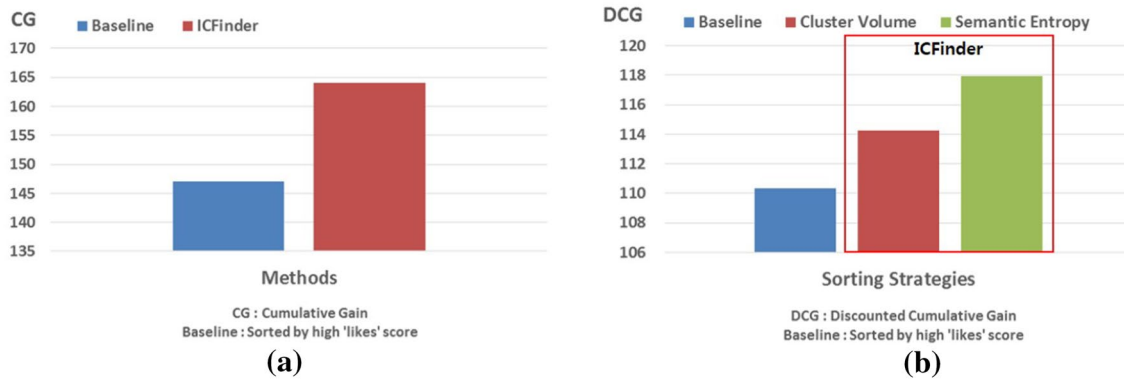


Fig. 3 **a** Sum of cumulative gain, **b** sum of discounted cumulative gain

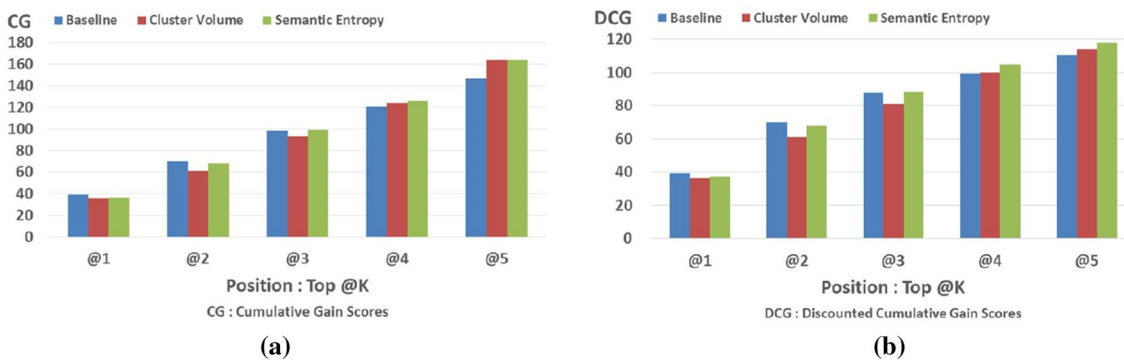


Fig. 4 **a** Cumulative gain @K, **b** discounted cumulative gain @K

Although overall the results of our method show better performance, not all cases show ideal results. Therefore, we conducted an exploratory case study to examine the difference between the methods. The ▼ represents those cases in which our method shows lower performance than the method of sorting by number of *LIKES* scores. The ▼ represents the cases in which the number of *Trigger* information intention class scores might be lower than average. We assume that these results are connected with our user study about highlighting the importance of the question. The cases in which users provide their information, opinions, and feelings in comments on a platform such as YouTube [37] might not be suitable for applying our method. Overall, we can conclude that, despite these limitations, our method can find more informative comments than can methods based on a high *LIKES* score.

Conclusion

In this paper, we present the IC-Finder method to find informative comments that assist in understanding video viewing. The method proposes specific solutions to deal

with each type of feature. Our approach for finding informativeness comments shows promising results compared to existing methods based on sorting high *LIKES* scores.

Future work can improve accuracy and case coverage. One possible approach can be identifying additional useful features of informative comments and combining them with our classifier to improve accuracy. Future work also includes identifying more specific classes. As we have seen in the experiments, there is a possibility of classification based on the expression of user feelings. Such work would require conducting more case studies for *Information Needs* in video viewing. Additional work can be aimed at tuning the algorithm to find more fitting question and answer sets.

Another direction of research can be based on experiments with user interface about when and how the informative comments can be displayed. A specialized user interface which tracks the users' comment behavior could supply more personalized data. We assume that this data would include informative comments which are also satisfactory from the user's personal perspective.

It would be interesting to utilize informative comments for related information search. The use of the IC-Finder method for the extraction of informative comments provides

Table 4 Difference in information intention comments between ▲ and ▼ cases

Unit: number of comments				
Case	Speaker	Trigger	Respond	Trigger/Respond
▲	Bart Weetjens	69	128	0.5391
▲	Bill Gates & Melinda Gates	76	157	0.4841
▲	Paul Root Wolpe	202	332	0.6084
▲	Stefano Mancuso	277	512	0.5410
▲	Taylor Wilson	102	283	0.3604
Case	Speaker	Trigger	Respond	Trigger/Respond
▼	Andrew Solomon	75	323	0.2322
▼	Bill Gates	136	275	0.4945
▼	Glenn Greenwald	117	183	0.6393
▼	Hans Rosling	87	343	0.2536
▼	Sugata Mitra	139	358	0.3883

the users with more relevant and useful information. These results can provide a more satisfactory experience to users who are watching videos for information needs or educational purposes and not for general or entertainment purposes. To extract more information, we need to investigate the connections between extracted information and user information needs.

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Apaza RG, Cervantes EV, Quispe LC, Luna JO. Online courses recommendation based on LDA. In: 1st Symposium on information management and big data, pp. 42–48. CEUR Workshop 2014.
2. Arndt C. Information measures: Information and its description in science and engineering. With 64 figures. New York: Springer Science & Business Media; 2001.
3. Benevenuto F, Rodrigues T, Cha M, Almeida V. Characterizing user behavior in online social networks. In: Proceedings of the 9th ACM SIGCOMM internet measurement conference, IMC '09, pp. 49–62. ACM, New York, NY, USA 2009. <https://doi.org/10.1145/1644893.1644900>.
4. Bhuiyan H, Ara J, Bardhan R, Islam DMR. Retrieving YouTube video by sentiment analysis on user comment. In: Proceedings of the IEEE International conference on signal and image processing applications (ICSIPA), pp. 474–478 2017. <https://doi.org/10.1109/ICSIPA.2017.8120658>.
5. Blooma MJ, Chua AY, Goh DH. Selection of the best answer in CQA services. In: 2010 Seventh International Conference on information technology: new generations (ITNG), pp. 534–539, 2010. <https://doi.org/10.1109/ITNG.2010.127>.
6. Cettolo M, Girardi C, Federico M. WIT³: Web inventory of transcribed and translated talks. In: Proceedings of the 16th Conference of the European Association for machine translation (EAMT), pp. 261–268. Trento, Italy 2012.
7. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST). 2011;2(3):27.
8. Chen YL, Chang CL, Yeh CS. Emotion classification of YouTube videos. Decis Support Syst. 2017;101:40–50.
9. Choi S, Segev A. Finding informative comments for video viewing. In: 2016 IEEE International Conference on big data workshop, application of big data for computational social science, IEEE Big Data '16, pp. 2457–2465. IEEE Computer Society 2016. <https://doi.org/10.1109/BigData.2016.7840882>.
10. Cong G, Wang L, Lin C, Song Y, Sun Y. Finding question-answer pairs from online forums. In: Proceedings of the 31st Annual International ACM SIGIR Conference on research and development in information retrieval, SIGIR '08, pp. 467–474. ACM, New York, NY, USA 2008. <https://doi.org/10.1145/1390334.1390415>.
11. Daradoumis T, Bassi R, Xhafa F, Caballe S. A review on massive e-learning (MOOC) design, delivery and assessment. In: 2013 Eighth International Conference on P2P, parallel, grid, cloud and internet computing (3PGCIC), pp. 208–213 2013. <https://doi.org/10.1109/3PGCIC.2013.37>.
12. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological). 1977; 39(1): 1–38.
13. Di Carlo GS. Humour in popularization: analysis of humour-related laughter in TED talks. Eur J Humour Res. 2014; 1(4):81–93.
14. Dumais ST. Latent semantic analysis. Annu Rev Inf Sci Technol. 2004;38(1):188–230. <https://doi.org/10.1002/aris.1440380105>.
15. Ellis D. A behavioural approach to information retrieval system design. J Doc. 1989;45(3):171–212. <https://doi.org/10.1108/eb026843>.
16. Figueiredo F, Belém F, Pinto H, Almeida J, Gonçalves M, Fernandes D, Moura E, Cristo M. Evidence of quality of textual features on the Web 2.0. In: Proceedings of the 18th ACM Conference on information and knowledge management, CIKM '09, pp. 909–918. ACM, New York, NY, USA 2009. <https://doi.org/10.1145/1645953.1646070>.
17. Ghose A, Ipeirotis PG. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In: Proceedings of the Ninth International Conference on electronic commerce, ICEC '07, pp. 303–310. ACM, New York, NY, USA 2007. <https://doi.org/10.1145/1282100.1282158>.
18. Gündüz c, Özsu MT. A web page prediction model based on click-stream tree representation of user behavior. In: Proceedings of the Ninth ACM SIGKDD International Conference on knowledge discovery and data mining, KDD '03, pp. 535–540. ACM, New York, NY, USA 2003. <https://doi.org/10.1145/956750.956815>.
19. Huang Y, Tseng Y, Sun YS, Chen MC. TEDquiz: Automatic quiz generation for TED talks video clips to assess listening comprehension. In: 2014 IEEE 14th International Conference on advanced learning technologies (ICALT), pp. 350–354 2014. <https://doi.org/10.1109/ICALT.2014.105>.
20. Jihan SH, Segev A. Context ontology for humanitarian assistance in crisis response. In: Proceedings of the International Conference on information systems for crisis response and management (ISCRAM), pp. 526–535 (2013)
21. John BM, Chua AY, Goh DH. What makes a high-quality user-generated answer? IEEE Internet Comput. 2011;15(1):66–71. <https://doi.org/10.1109/MIC.2011.23>.
22. Ko M, Choi S, Lee J, Yang S, Lee U, Segev A, Song J. Motives for mass interactions in online sports viewing. In: Proceedings of the Companion Publication of the 23rd International Conference

- on World Wide Web Companion, WWW Companion '14, pp. 329–330. International World Wide Web Conferences Steering Committee. 2014. <https://doi.org/10.1145/2567948.2577340>.
23. Krishnamoorthy N, Malkarnenkar G, Mooney RJ, Saenko K, Guadarrama S. Generating natural-language video descriptions using text-mined knowledge. In: Proceedings of the Twenty-Seventh AAAI Conference on artificial intelligence 2013, AAAI '13. AAAI Press, 2013. <http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6454>.
24. Li G, Ming Z, Li H, Chua T. Video reference: question answering on YouTube. In: Proceedings of the 17th ACM International Conference on multimedia, MM '09, pp. 773–776. ACM, New York, NY, USA, 2009. <https://doi.org/10.1145/1631272.1631411>.
25. Liu J, Dolan P, Pedersen ER. Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on intelligent user interfaces, IUI '10, pp. 31–40. ACM, New York, NY, USA, 2010. <https://doi.org/10.1145/1719970.1719976>.
26. Lopes J, Trancoso I, Abad A. A nativeness classifier for ted talks. In: Acoustics, speech and signal processing (ICASSP), 2011 IEEE International Conference on, pp. 5672–5675, 2011. <https://doi.org/10.1109/ICASSP.2011.5947647>.
27. Momeni E, Cardie C, Ott M. Properties, prediction, and prevalence of useful user-generated comments for descriptive annotation of social media objects. In: Proceedings of the Seventh International Conference on weblogs and social media 2013, ICWSM '13. AAAI Press, 2013.
28. Momeni E, Sageder G. An empirical analysis of characteristics of useful comments in social media. In: Proceedings of the 5th Annual ACM Web Science Conference, WebSci '13, pp. 258–261. ACM, New York, NY, USA, 2013. <https://doi.org/10.1145/2464464.2464490>.
29. Pappas N, Popescu-Belis A. Combining content with user preferences for ted lecture recommendation. In: Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on, pp. 47–52, 2013. <https://doi.org/10.1109/CBMI.2013.6576551>.
30. Paul M, Federico M, Stüker S. Overview of the IWSLT 2010 evaluation campaign. In: Proceedings of the 7th International Workshop on spoken language translation (IWSLT), vol. 10, pp. 3–27, 2010.
31. Potthast M, Becker S. Opinion summarization of Web comments. In: Gurrin C, He Y, Kazai G, Kruschwitz U, Little S, Rüger S, van Rijsbergen K, editors. Advances in information retrieval, lecture notes in computer science, vol. 5993. Berlin: Springer Berlin Heidelberg; 2010. p. 668–9. https://doi.org/10.1007/978-3-642-12275-0_73.
32. Russell DM, Klemmer S, Fox A, Latulipe C, Duneier M, Losh E. Will massive online open courses (MOOCs) change education? In: CHI '13 Extended Abstracts on human factors in computing systems, CHI EA '13, pp. 2395–2398. ACM, New York, NY, USA, 2013. <https://doi.org/10.1145/2468356.2468783>.
33. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Inf Process Manag.* 1988;24(5):513–23. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
34. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *Commun ACM.* 1975;18(11):613–20. <https://doi.org/10.1145/361219.361220>.
35. Schein AI, Popescul A, Ungar LH, Pennock DM. Methods and metrics for cold-start recommendations. In: Proceedings of the 25th Annual International ACM SIGIR Conference on research and development in information retrieval, SIGIR '02, pp. 253–260. ACM, New York, NY, USA, 2002. <https://doi.org/10.1145/564376.564421>.
36. Schmidt DC, McCormick Z. Producing and delivering a Coursera MOOC on pattern-oriented software architecture for concurrent and networked software. In: Proceedings of the 2013 Companion Publication for Conference on systems, programming, & applications: software for humanity, SPLASH '13, pp. 167–176. ACM, New York, NY, USA, 2013. <https://doi.org/10.1145/2508075.2508465>.
37. Schultes P, Dorner V, Lehner F. Leave a comment! an in-depth analysis of user comments on YouTube. In: Tagungsbände der Wirtschaftsinformatik, pp. 42 2013.
38. Segev A. Adaptive ontology use for crisis knowledge representation. *Int J Inf Syst Crisis Response Manag (IJISCRAM).* 2009;1(2):16–30. <https://doi.org/10.4018/jiscrm.2009040102>.
39. Shah C, Pomerantz J. Evaluating and predicting answer quality in community qa. In: Proceedings of the 33rd International ACM SIGIR Conference on research and development in information retrieval, SIGIR '10, pp. 411–418. ACM, New York, NY, USA, 2010. <https://doi.org/10.1145/1835449.1835518>.
40. Shatnawi S, Gaber MM, Cocea M. Text stream mining for massive open online courses: review and perspectives. *Syst Sci Control Eng.* 2014;2(1):664–76. <https://doi.org/10.1080/21642583.2014.970732>.
41. Turney PD. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for computational linguistics, ACL '02, pp. 417–424. Association for Computational Linguistics, Stroudsburg, PA, USA, 2002. <https://doi.org/10.3115/1073083.1073153>.
42. Vihavainen A, Luukkainen M, Kurhila J. Multi-faceted support for MOOC in programming. In: Proceedings of the 13th Annual Conference on information technology education, SIGITE '12, pp. 171–176. ACM, New York, NY, USA, 2012. <https://doi.org/10.1145/2380552.2380603>.
43. Vivekraj VK, Debashis S, Balasubramanian R. Video skimming: taxonomy and comprehensive survey. *ACM Comput Surv.* 2019;52(5):106:1–38. <https://doi.org/10.1145/3347712>.
44. Wang B, Wang X, Sun C, Liu B, Sun L. Modeling semantic relevance for question-answer pairs in web social communities. In: Proceedings of the 48th Annual Meeting of the Association for computational linguistics, ACL '10, pp. 1230–1238. Association for Computational Linguistics, Stroudsburg, PA, USA, 2010. <http://dl.acm.org/citation.cfm?id=1858681.1858806>.
45. Wang X, Tu X, Feng D, Zhang L. Ranking community answers by modeling question-answer relationships via analogical reasoning. In: Proceedings of the 32Nd International ACM SIGIR Conference on research and development in information retrieval, SIGIR '09, pp. 179–186. ACM, New York, NY, USA, 2009. <https://doi.org/10.1145/1571941.1571974>.
46. Wilson T. Models in information behaviour research. *J Doc.* 1999;55(3):249–70. <https://doi.org/10.1108/EUM0000000007145>.
47. Witten IH, Frank E. Data Mining: Practical machine learning tools and techniques. Burlington: Morgan Kaufmann; 2005.
48. Xiong W, Litman D. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In: Proceedings of COLING 2014, the 25th International Conference on computational linguistics, pp. 1985–1995. Dublin, Ireland 2014.
49. Yang D, Adamson D, Rosé CP. Question recommendation with constraints for massive open online courses. In: Proceedings of the 8th ACM Conference on recommender systems, RecSys '14, pp. 49–56. ACM, New York, NY, USA, 2014. <https://doi.org/10.1145/2645710.2645748>.
50. Yang D, Piergallini M, Howley I, Rose C. Forum thread recommendation for massive open online courses. Proceedings of the 7th International Conference on educational data mining pp. 257–260, 2014.
51. Yousef AMF, Chatti MA, Schroeder U, Wosnitza M. What drives a successful MOOC? an empirical examination of criteria to assure design quality of MOOCs. In: Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on, pp. 44–48, 2014. <https://doi.org/10.1109/ICALT.2014.23>.
52. Yu H, Zheng D, Zhao BY, Zheng W. Understanding user behavior in large-scale video-on-demand systems. In: Proceedings of the 1st

- ACM SIGOPS/EuroSys European Conference on computer systems 2006, EuroSys '06, pp. 333–344. ACM, New York, NY, USA, 2006. <https://doi.org/10.1145/1217935.1217968>.
53. Zhang R, Tran T. An entropy-based model for discovering the usefulness of online product reviews. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, volume 01, WI-IAT '08, pp. 759–762. IEEE Computer Society, Washington, DC, USA, 2008. <https://doi.org/10.1109/WIIAT.2008.149>.
54. Zhao L, Hua T, Lu CT, Chen R. A topic-focused trust model for twitter. *Comput Commun.* 2016;76:1–11.
55. Zhao Z, Hong L, Wei L, Chen J, Nath A, Andrews S, Kumthekar A, Sathiamoorthy M, Yi X, Chi E. Recommending what video to watch next: A multitask ranking system. In: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, pp. 43–51. ACM, New York, NY, USA 2019. <https://doi.org/10.1145/3298689.3346997>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.