Journal of Internet Services and Applications

RESEARCH Open Access

Is it possible to describe television series from online comments?

Túlio C. Loures*, Pedro O. S. Vaz de Melo and Adriano A. Veloso

Abstract

Due to the omnipresence of the Internet and Social Media in current society, it has become easy to find groups or communities of people discussing the most varied subjects in discussion forums, social network interactions, or comments on web pages. In this paper, we try to answer the question about whether, even when nothing is explicitly known about the entity referred to in the discussion, it is possible to formulate a general and brief idea of its characteristics when reading comments about it. To study this problem, we characterize a collection of online discussions about television series episodes, investigate the potential that comments have to describe these series, implement several different summarization methods, and finally evaluate these different methods and analyze the results obtained from them. Results reveal that a small set of comments can describe the corresponding episodes and, when taken together, the series as a whole.

Keywords: Online discussion, Comments, Summarization

1 Introduction

The Internet consists of millions of devices in which each of them is responsible for the generation, storage and transmission of countless data. In this context, automatic learning algorithms have been widely used to process and extract valuable information from all of this data. To do this, algorithms need formal representations for the characteristics of the entities they want to learn about, which is usually a challenging task [1, 2]. This is an even more difficult task if these entities do not have any explicitly structured information associated with them. Consider, for example, the problem of describing the content of a personal video posted on Facebook, or an event associated with a Twitter *hashtag*.

As Social Media technologies encourage user input rather than simply providing information, more and more people are able to freely comment on different types of entities. Although it does come with its fair share of challenges for those interested in studying it, this boom in user participation provides an opportunity for exploring a new source of data. Thus, in this article we investigate the following question: is it possible to explain and describe entities using only online discussions associated with them as a source of knowledge?

Using comments rather than explicitly structured information as data source has several advantages. One clear advantage is that comments are ubiquitous in the Internet. They are associated with the most varied objects, such as movies, products, personal videos, social media posts, episodes of TV shows, restaurants, touristic attractions, hotels, online news and so on. Also, comments are cheap, i.e., ideally, no one is getting paid to join discussions and share their ideas through online comments. More important, comments are spontaneously generated by people when there is something they judge as useful that should be shared with others. From a rational perspective, if the user who posted the comment thinks she/he is not gaining anything by posting her/his thoughts, she/he would not have made the effort to write and post them. Finally, comments may serve as a natural filter for entities that should be indexed and processed. If an old Youtube video has not received any comment, maybe this is an indicator of low quality. Thus, there is some value associated with each comment, even if it is not clearly stated, and even if many view comment sections as harmful[1]. In short, we see comments about a particular entity as *crowdsourcing*, similar to what is done in the ESP game [3], an online game licensed by Google that is used to improve the accuracy of

*Correspondence: loures.tc@dcc.ufmg.br
Department of Computer Science, Federal University of Minas Gerais, Av. Pres. Antônio Carlos, 6627 - Pampulha, 31270-901 Belo Horizonte - MG, Brazil

its image search engine. With this, we expect to generate summaries for entities and domains without anyone ever having to explicitly explain what they are.

In addition to a thorough exploratory analysis, this article also investigates how much can be learned about entities solely from general conversations about them. It is known that extracting relevant information from comments is a very challenging task [4–6]. Online comments are generally short, and it is common for users to use informal and unstructured texts to express themselves, for example, through acronyms and word shortening. Another difficulty lies in the fact that comments allow people to initiate and permeate conversations which are often about matters quite different from the entity itself. Thus, any method for learning features of entities from comments should be able to disregard (or filter) this type of conversation.

To study this problem, we look at the potential that comments have to describe television series. A television series is a type of TV show with a predefined number of episodes per season. From comment sequences posted in online forums, we investigate whether they can (and how they can) be used to automatically generate summaries that describe the television series associated with them. More specifically, in this article we will discuss the following questions:

RQ1 Are online discussions a good source of data to learn about television series and episodes?
RQ2 Given a sequence of comments associated with an episode and a manually created summary of it, how much of this summary can such a sequence automatically generate?
RQ3 How can we extract a relatively small amount of text from these comments that is able to accurately describe the television series they are associated with?

In this work, in order to answer these questions, we formally define the problem, present and characterize a data set, show that comments have, in their textual content, enough discriminatory information for classification, and finally propose and evaluate different summarization methods that use only information found in online discussions. In short, the contributions of this work can be summed up in the following points:

C1 Definition of the problem of summarization of a domain based on the discussions associated with its entities, as well as the necessary definitions for comment sequences and entity sets.
C2 Analysis of online discussions, specially concerning how well they can be used to identify and describe their associated entities. This includes a classification task that verifies whether comments can be correctly classified according to their respective series and episodes, based only on their words.
C3 Proposal and evaluation of two summarization methods based on comment selection: *TKW-AF*, which uses all comments that have at least some words considered as important, and *TKW-MSC*, which tries to select the minimum number of comments so that all important words are part of the generated summary. These methods are compared to the *TextRank* algorithm, using the same discussion text as input.

The remainder of this article is organized as follows. The related works are described in Section 2. The problem definition, along with the notation used in this article, are given in Section 3. In Section 4, we describe the data set. An analysis of the potential that comments have to explain television series is shown in Section 5, and in Section 6 we explore that potential further with a simple classification task. In Section 7 the summarization methods used in this work are described. In Section 8 we analyze and discuss the summaries obtained. Finally, in Section 9, we describe the conclusions taken from this work.

2 Related work

In recent years, there have been a large number of studies aimed at generating and evaluating text summaries. To the best of our knowledge, most of the work in this area focuses on summarizing full texts, such as books and news [7–10], or summarizing opinions from user-written reviews [11–14]. However, in our case, the use of comments as a source of data raises many challenges that, individually, serve as inspiration for this study.

In fact, several studies have tried to analyze and characterize online interactions such as online discussions. In [5], for example, the authors analyzed comments posted on different social media sites, mainly studying how their content and rating can be related. The work of [15], in turn, develops a formal approach to the modeling of "activity bursts", which can be applied to comment streams. Choi et al. [16] characterized conversations collected from Reddit in terms of volume, responsiveness, and virality.

Methods to extract relevant information from comments have also been proposed in recent works. Khabiri et al. [17] proposed a clustering-based approach to identify groups of correlated comments in YouTube videos. Yang et al. [18] presented a method for generating Web page summaries that also considers the comments associated with them and the social network among users who commented. In [19], the authors proposed the use of vector representations of sentences as a metric of similarity in the process of extractive summarization. Liu et al.

[20] model the problem of summarizing comments as a clustering problem, in which the number of topics covered by the comments must be known a priori. Chua and Asur [21] proposed a temporal correlation-based topic model to identify the most relevant tweets of a given query for the summarization task. Unlike the contexts used in these works, using online discussions as a source of information for describing an entity has several new problems. Comment sequences represent conversations, which may derail to different subjects or may contain only assertive information, such as "I agree with what you said", or even a simple "Me too". These types of comments do not add relevant information to the description of the entity (in our case, a TV series) in question, so considering a way to deal with these cases is an important aspect of our article.

Considering our previous works in the area, in [22], a model was proposed to capture the dynamics of communication activities on the Web. Following this work, in [23], universal and distinct communication patterns were described considering the technology used as means of communication. In [24], a parsimonious model was proposed to characterize the burstiness of general random series of events in the Web. In [25] we selected comments from online discussions in order to best summarize the conversation itself. Although these studies presented relevant contributions in understanding the dynamics of online discussions, none of them proposed methods to use the content of the comments as a source of information to describe and summarize the entities they refer to.

In [26], the paper this work is based on, we first explored the task of selecting comments from online discussions for entity description, studying if and how comment sequences can be used to represent entities. In [27] we used the results found in our previous work to generate and evaluate entity representations from online discussions.

This article differs from the aforementioned studies in three main aspects. First, this work focuses on the characterization of the potential of comments as a single data source for generating formal entity representations to be used by machine learning algorithms. Second, through this characterization, we investigate and quantify the differences between the language used in formal and manually generated summaries and the language used by users in comments, specially in those comments evaluated as "non-descriptive". Finally, we compare different extractive summarization methods, and evaluate them by summary size and by how much of a human-written summary they are able to describe.

3 Problem definition

For the remainder of this article, we define by c a comment submitted to an online discussion. For the purposes of this work, we assume that each comment is composed only of its textual content and its timestamp, which is used as a sorting key. Thus, a comment sequence $\mathcal{C}$ is an ordered set of n comments $\{c_1, c_2, \ldots, c_n\}$ taken from a common context, for example, all comments of a given YouTube video.

We also define by t an entity responsible for generating an online discussion, e.g. a YouTube video or a text on a news portal. Thus, each entity t is associated with a comment sequence $\mathcal{C}^t$, where

$$\mathcal{C}^t = \{\mathcal{C}_1^t, \mathcal{C}_2^t, \ldots, \mathcal{C}_{n^t}^t\}$$

is the sequence of n^t comments generated from that entity. Henceforth this sequence of comments will also be referred to simply as "entity's discussion", or "entity's comments". A sequence of entities t is an ordered set of m entities $\{t_1, t_2, \ldots, t_m\}$ taken from a common context.

A domain d consists of a set of m^d entities

$$\mathcal{T}^d = \left\{t_1^d, t_2^d, \ldots, t_{m^d}^d\right\}.$$

In addition, we can also define a comment sequence of a domain d as $\mathcal{C}^d$, where

$$\mathcal{C}^d = \left\{\mathcal{C}^{t_1^d}\mathcal{C}^{t_2^d}\ldots\mathcal{C}^{t_{m^d}^d}\right\}$$

is the concatenation of the comment sequences $\mathcal{C}^{t_i^d}$ of each entity t_i^d belonging to the domain d. A domain can be, for example, all Youtube videos from the band *Aerosmith* or all Reddit discussion topics on the *Game of Thrones* series.

The diagram in Fig. 1 illustrates the relationship between a domain *Dom*, its set of entities $\mathcal{T}^d = \left\{\mathcal{T}_1^d, \mathcal{T}_2^d, \mathcal{T}_3^d\right\}$, and the comment sequences of each of these entities: $\mathcal{C}^{t_1^d}$, $\mathcal{C}^{t_2^d}$, and $\mathcal{C}^{t_3^d}$.

Finally, we define by R^t the human-written summary associated with the entity t. Similarly, we also define by R^d the human-written summary associated with the domain d. In our case, R^t can be a human generated summary of an episode of Game of Thrones, for example, available on IMDB[2] or Wikipedia[3]. Moreover, R^d will be, in this case, the concatenation of the summaries for all episodes $t \in \mathcal{T}^d$ from this particular TV series d.

Given these definitions, our goal is to find a concise portion of text (e.g., a subset of comments, or a selection of sentences) that is able to describe a context well. For this article, we will be focusing on the case of comment sequences referring to television series episodes. Having established the definitions of the above concepts, we can write this task as: automatically generating a piece of text r^d relating to series d, using its discussion $\mathcal{C}^d$ as source of information, with a small number of words describing each episode t, so that R^d is well explained by r^d.

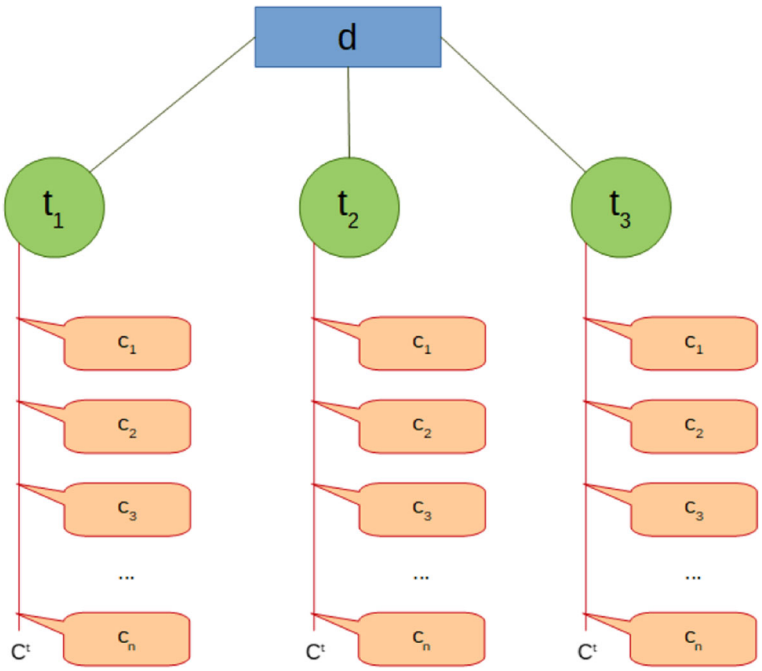


Fig. 1 Domain, entities and comment sequences. Diagram illustrating the relationship between a domain d, a set of entities $\mathcal{T} = \{t_1^d, t_2^d, t_3^d\}$, and sets of comments $\mathcal{C}^{t_1^d}$, $\mathcal{C}^{t_2^d}$, and $\mathcal{C}^{t_3^d}$

4 Data set

For this work, we collected all the comments directly associated with episodes of several different animated series from the *MyAnimeList.net* website[4]. *MyAnimeList.net*, also referred to as *MAL*, is a website that, as its core functionality, allows users to list animated series according to how much of it they've watched. In addition to that, *MAL* also gives the option for users to rate, review, and recommend shows, and also provides varied information regarding each series, from its cast and crew to the release date. More importantly for this work, *MAL* also has an online forum, with sub-forums for each animated series, usually containing a separate discussion for each episode from that series. In [27], we have used much of the data provided for each series from our collected data set, and they can also be used in the future to expand and improve the work presented in this article.

In addition to the comment data, human-written summaries R^t were also collected, when available, for each episode t. These summaries were obtained from Wikipedia pages dedicated to listing and describing episodes of each series [5]. The summary R^d of each series d is simply the concatenation of the summaries of all episodes of the series in question.

As the number of comments and episodes varies significantly from series to series, three criteria were used to select the series we analyzed in this article. First, we tried to choose those series with a similar number of comments, so that the popularity of all series is similar, in general, allowing us to more easily compare the different series without taking the discussion size too much into consideration. The second determining factor for the choice was the existence of an easily accessible human-written summary (collected from Wikipedia) for each episode of the series. Finally, we chose a set of series with a number of episodes in the order of tens and comments in the order of thousands. Table 1 shows the series that were chosen for analysis, along with the number of episodes and comments associated with them[6].

Once the texts described above were collected, both the comments and the summaries, a sequence of operations was performed on them in order to transform the original texts into workable data. First, quotes from other comments present in the texts were removed. Keeping the quoted comments as repeated text could lead to interesting results, as that piece of text would be considered more "relevant" by some algorithms, but a new study would be needed to verify that. Then, the HTML formatting of the texts was removed, leaving only the unformatted comment texts. Doing this discards useful information in the form of text formatting, such as boldfaced words, but the methods used in this article do not take those into consideration, making them a pointless complication. Also, non-printable characters were excluded. Finally, the *stop words* in the texts were removed, and all letters were converted to their lower case forms. For this last step, we kept

Table 1 Number of episodes and number of comments for each series from the data set

Series ID	# Episodes	# Comments
1	26	2675
19	50	3701
30	26	3540
205	26	1812
226	13	1459
356	24	2065
457	26	2833
777	10	1663
790	23	2037
820	50	3630
877	47	2906
934	26	3529
13599	22	3643
Total	369	35493
Average	28.38	2730.23

a different version of the texts with and without applying the *stop word* filter and lower case transformation. This was done because some of the methods used in this article require actual natural language sentences, while others work best with only a list of informative words. Unless otherwise specified, all text used as data in this article has *stop words* filtered out and has been completely converted to lower case.

5 Comments analysis and characterization

As mentioned in Section 2, comments (and comment sequences) are composed of texts with very distinct characteristics, which can vary significantly in size, form and content. Thus, to better understand the collected data described in Section 4, in this section a series of analysis on the characteristics of the comment sequences were carried out. Such analysis will serve as a basis for the methodology presented in the following sections.

5.1 Basic analysis

As an initial analysis, general characteristics of the collected comments were evaluated. Figure 2a shows the histogram for the size (number of words) of the comments in our data set. Note that the vast majority of comments have a small number of words, while a few comments are very long, with more than 500 words. This result evidences the difficulty of extracting relevant comments in comment sequences, since the great majority is composed of small comments, most probably with little informative content.

In order to investigate if there is a discrepancy in user participation per episode, we show in Fig. 2b the histogram of the number of comments posted per episode. Note that many episodes have a small number of comments associated with them, i.e., most episodes have between 20 and 100 comments. Similarly to what was shown in Fig. 2a, there are also episodes with a large amount of comments. Thus, a method for identifying relevant and descriptive comments will often face the problem of having either too little or too much information available, depending on the episode and series.

In short, Fig. 2a suggests that most of the comments do not have useful information given their relatively small sizes, but Fig. 2b indicates that most episodes will have a sufficient amount of comments so that we can infer a description from them.

5.2 Word analysis

In order to determine which comments would be most descriptive of the human-written summary, we first made a feature analysis that could be used to define and represent the different terms present in the data set. We used the t-SNE[7] [28] dimensionality reduction method to interpret the quality of these representations. This visualization technique aims to capture and illustrate the local structure of high-dimensionality data while also indicating its global structures. After considering some different options for word-related characteristics, we determined a set of features which presents sufficient and relevant information regarding the usefulness of each word to discriminate between different series and episodes. Defining $\mathcal{V}^X$ as the set of words used in context X (e.g. $\mathcal{V}^d$ as the set of words used in the discussion for series d), the following features were used:

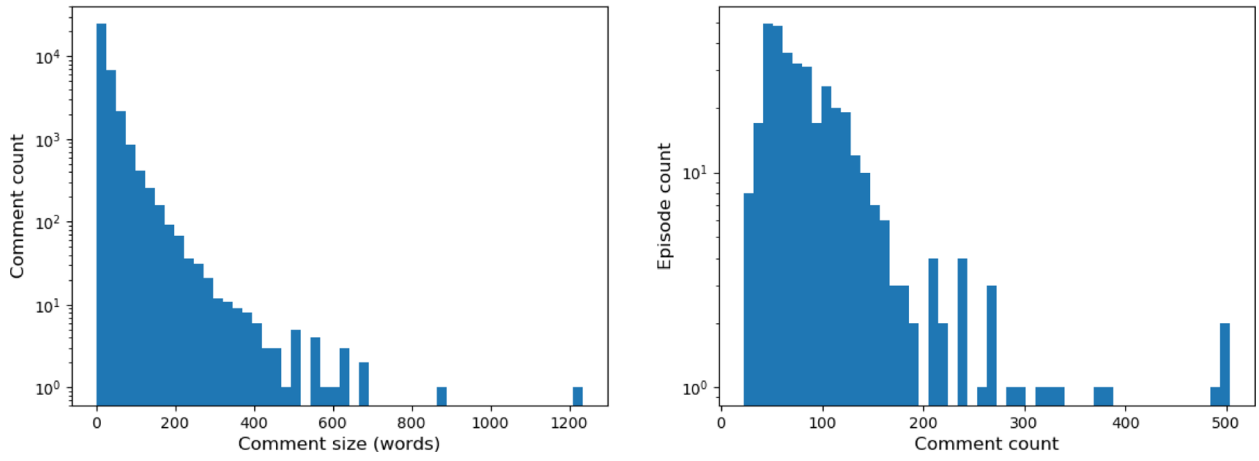
- Word probability in each series: the probability of the word v occurring in each series' discussion in comparison to it occurring in other series' discussions, generating one feature for each series.

$$P\left(v \in \mathcal{V}^d\right) = \frac{tf(v,d)}{tf(v,\mathcal{D})}$$

  In which $tf(v,d)$ is the count of occurrences of word v in series d, and $tf(v,\mathcal{D})$ is the count of occurrences of word v in the entire data set.
- Shannon entropy of the word among all series: how much information is produced by the word v occurring in a series' discussion.

$$H_v^{\mathcal{D}} = -\sum_{d\in\mathcal{D}} P\left(v \in \mathcal{V}^d\right) log_2 P\left(v \in \mathcal{V}^d\right)$$

- Shannon entropy of the word among all episodes of the data set: how much information is produced by the word v occurring in an episode's discussion.



(a) Comment sizes (b) Comments per episode

Fig. 2 Basic comment analysis. **a** shows the size distribution of the comment texts, in *log* scale. **b** shows the distribution of the number of comments per episode, in *log* scale

$$H_v^{\mathcal{T}} = -\sum_{t\in\mathcal{T}} P\left(v \in \mathcal{V}^t\right) log_2 P\left(v \in \mathcal{V}^t\right)$$

- Shannon entropy of the word among the episodes for each series: how much information is produced by the word v occurring in an episode's discussion, generating one feature for each series.

$$H_v^{\mathcal{T}^d} = -\sum_{t\in\mathcal{T}^d} P\left(v \in \mathcal{V}^t | v \in \mathcal{V}^d\right) log_2 P\left(v \in \mathcal{V}^t | v \in \mathcal{V}^d\right)$$

- TF-IDF of the word for each series: *term frequency – inverse document frequency* of the word v, generating one feature for each series. The TF-IDF is calculated by multiplying the count of occurrences of word v in series d, $tf(v,d)$, by the inverse document frequency $idf(v,d)$:

$$idf(v,\mathcal{D}) = log\frac{|\mathcal{D}|}{\left|\left\{d \in \mathcal{D} : v \in \mathcal{V}^d\right\}\right|}.$$

Thus, each word is represented by a feature vector of size $2 + 3 * |\mathcal{D}|$ (particularly, $2 + 3 * 13 = 41$) that indicates how this particular word is used through the data set, and how those uses are distributed across the TV series.

Figure 3 shows the visualization generated through the t-SNE dimensionality reduction method for all words in the data set, which are represented by this set of features. In the image, each word is drawn as a point, colored according to the series in which it appears the most, i.e. the domain d for which the word has the highest occurrence count $tf(v,d)$. It can be noted that, although most words are located in a single large cluster ("main body" with mixed points of different colors), there are some smaller clusters ("small tendrils" with points of a single color each) that contain words associated with the same series. Since the features used to represent the words, and then generate the t-SNE visualization, reflect the presence of the word in the analyzed series, it was verified that these small clusters contain those few words that occur predominantly in a single series (character names, and other specific terms from the series). Thus, we can conclude that, in fact, the number of words that have their use highly concentrated in a single series is small. This means that if a comment contains one of these words, we are probably able to tell which series it is associated with. This hypothesis will be explored in the following sections of this article.

5.3 Summarization potential

Another fundamental analysis to understand the potential of extracting relevant comments from discussions is to investigate the ability of each comment to explain the manually generated summary of the episode to which it is associated. To evaluate this, we used the ROUGE-N metric [29]. This metric informs the proportion of N-grams from the reference text (S) that are also present in the candidate text s:

$$ROUGE-N = \frac{count\left((grams_N \in S) \cap (grams_N \in s)\right)}{count\left(grams_N \in S\right)}$$

In other words, we can consider that this recall-oriented metric measures how much of the reference text is covered by another text of interest. As such, a higher ROUGE-N value informs that the candidate text s covers more of the reference text S with respect to their textual contents, indicating that s contains more information from S. It is worth noting that, despite its simplicity, Lin reports good correlation values between ROUGE-N metrics (specially

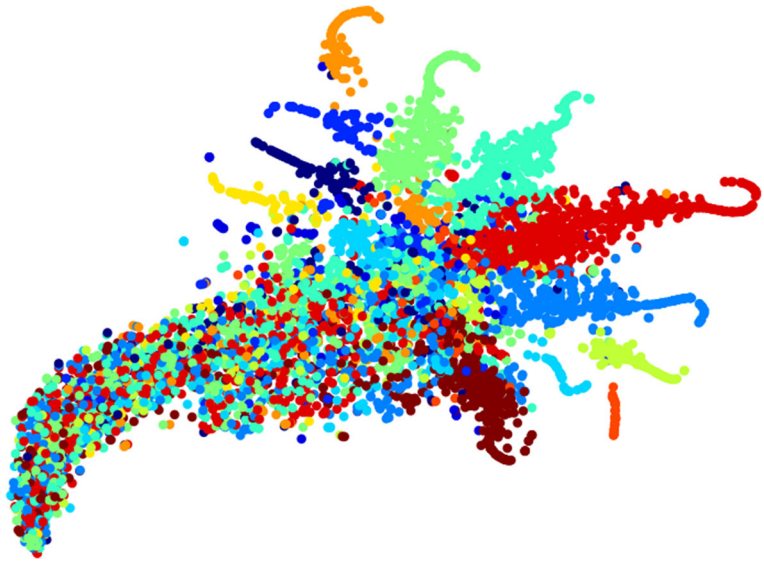


Fig. 3 2D representation of words from the data set. 2D representation of the words from the data set, generated by the t-SNE tool, and colored according to the series in which they appear the most

for low values of N) and human judgment for summary evaluations.

As such, in this article we evaluated text coverage with the ROUGE-1 metric, i.e., considering only unigram overlaps. Tests with the ROUGE-2 metric, i.e., considering bigram overlaps, revealed similar results.

Figure 4 shows the distribution of the coverages of the human-written summaries R^t (reference text) of each episode t by their respective comment sequences $\mathcal{C}^t$ (candidate text), as measured by the ROUGE-1 metric. Note that, on average, only 40% of the words from an episode's summary are found in the comments associated with it.

In order to analyze this result in greater depth, we show in Fig. 5 the cumulative coverage (ROUGE-1) of an episode's human-written summary $R^{d_{457}}$, taken as an example, over the course of its discussion $\mathcal{C}^{d_{457}}$, with comments ordered according to their timestamps. As more comments are added to the discussion, the summary coverage increases, until the full discussion reaches a ROUGE-1 score of approximately 0.8 (80% summary coverage). Note that much of the summary $R^{d_{457}}$ is explained

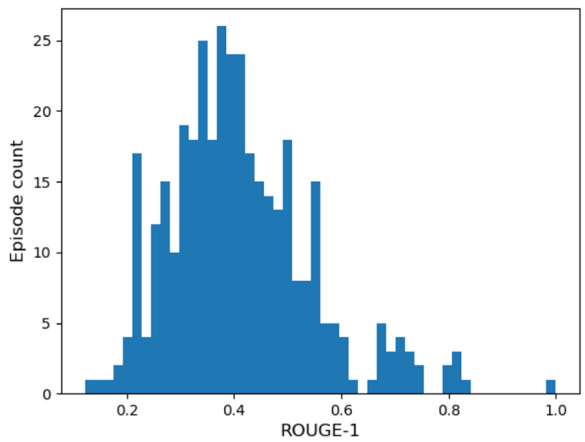


Fig. 4 Episode summary covered by comments. Distribution of how much of the human-written summary of each episode is covered by its respective comments, as measured by the ROUGE-1 metric

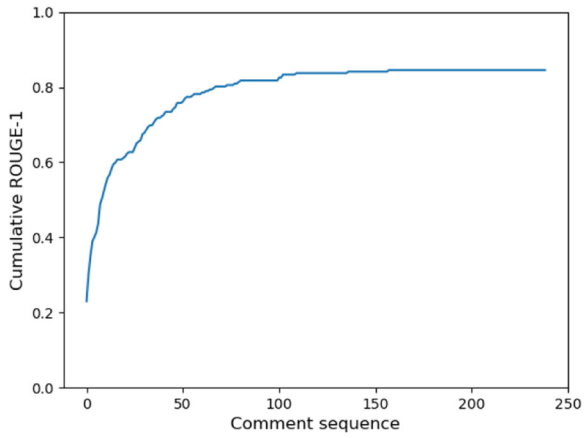


Fig. 5 Summary coverage over the course of discussion. Cumulative coverage (ROUGE-1 score) of the human-written summary $R^{d_{457}}$, of series d_{457} taken as example, over the course of its discussion $\mathcal{C}^{d_{457}}$

in the first few comments from the series' discussion, with little relevant information being added after about the hundredth comment. This general behavior, disregarding the specific ROUGE-1 values of the considered example, can also be seen for all series in our data set.

With these last analyses, we verified that there are episodes for which comments have a higher descriptive value than others, with respect to how much of its human-written summary they are able to explain. In addition, it is noticeable that the gain in summary coverage from adding more comments decays as more comments are considered at the same time.

6 Classifying comments

As a way to better verify our hypothesis that online discussions can be used as a source of information to describe the entities and domains they belong to, we developed a simple classification task that checks how easily comments can be attributed to their source. This serves as a "sanity check" for our work, as the goal of this task is to determine if comments have discriminative information with respect to their associated entities and domains.

6.1 Comment representation

In order to perform this classification task on comments, we first need structured representations for them. A simple way to do this is by means of a *bag-of-words* style representation, i.e., each comment c is represented by a boolean vector:

$$\mathbf{c} = \left[w_1^c, \ldots, w_{|\mathcal{V}|}^c\right],$$

where $|\mathcal{V}|$ is the size of the dictionary $\mathcal{V}$ (number of words), and w_j^c indicates whether the word v_j of the dictionary appears in comment c or not, $w_j^c = 1$ if so and $w_j^c = 0$ otherwise.

A general problem with this approach is that the vector $\mathbf{c}$ can become very large and sparse, especially if $\mathcal{V}$ is the set of all words present in the data set [30]. In our case, however, from the analysis done in Section 5, we know that the proportion of words capable of discriminating one series from the other is small (see Fig. 3). Therefore, the idea is to leverage this observation in the process of generating comment representations, in which only the most informative K words of each entity are used for the vector $\vec{c}$. To quantify the importance each word has for a series or episode, the TF-IDF metric will be used, which is calculated as follows:

$$tfidf(v, p, \mathcal{P}) = tf(v, p) * idf(v, \mathcal{P}).$$

The term $tf(v, p)$ is the *Term Frequency*, that is, the number of times that the word v appears in document p. The term $idf(v, \mathcal{P})$ is the *Inverse Document Frequency*, that is, how common or rare the word v is between the set of documents $\mathcal{P}$, being calculated by the formula:

$$idf(v, \mathcal{P}) = log \frac{|\mathcal{P}|}{\left|\{p \in \mathcal{P} : v \in \mathcal{V}^p\}\right|}.$$

In the case of this work, we calculate the TF-IDF metric considering two different contexts: words used in series, and words used in episodes of a given series. For the first one, we use the concatenation of the comments from each sequence $\mathcal{C}^d$ as a document p, with all series $d \in \mathcal{D}$ from the data set being considered in $\mathcal{P}$ for the IDF calculation. For the second, each document p is constructed from the concatenation of the comments of each sequence $\mathcal{C}^t$ belonging to an episode $t \in \mathcal{T}^{\mathcal{D}}$, for a given series $\mathcal{D}$. The set of documents $\mathcal{P}$ considered for it is built only from the episodes of this single series d.

Thus, we define by $\mathcal{V}^{K_p}$ the set of words in document p (either an entity $\mathcal{T}$ or a domain d) given by this process as the K most important, hereafter also referred to as "*top-K* words" of the document. From the *top-K* words of each of the $|\mathcal{P}|$ relevant documents (relative to the TF-IDF values), we defined the set of $|\mathcal{V}^K|$ relevant words among all the documents as:

$$\mathcal{V}^K = \bigcup_{p \in \mathcal{P}} \mathcal{V}^{K_p}.$$

Thus, the vector representing each comment is now given by:

$$\mathbf{c} = \left[w_1^c, \ldots, w_{|\mathcal{V}^K|}^c\right],$$

where $w_i^c = 1$ if the word $v_i \in \mathcal{V}^{K_p}$ appears in comment c or 0 otherwise. Note that $|\mathcal{V}^K|$ is at most equal to $K \times |\mathcal{P}|$.

With this, each comment is represented by a Boolean vector indicating, for the *top-K* words of each episode or series, whether that word is present or not in the comment.

6.2 Comment filter

Through preliminary analysis, we noticed that a large number of comments do not contain even a single word from the set of *top-K* words that discriminate their series or episode. For example, series d_{457} contains 1719 comments (of a total of 2833) that do not contain any word from the *top-K*, for $K = 10$, which equates to approximately 60% of the published comments about the series. Our hypothesis is that such comments are less relevant and descriptive to the series or episode with which they are associated, in comparison to comments containing words from the set of *top-K* words. Therefore, we considered a second parameter, α, which indicates the minimum number of relevant words (i.e., words from the *top-K* set) that a comment c must have so that it is not discarded. In other words, we discard all comments for which:

$$\sum_{j}^{|\mathcal{V}^K|} w_j^c < \alpha.$$

Thus, given determinate values for the K and α parameters, we wish to know how easily the selected comments can be identified as being associated with a certain episode or series. By doing this, we want to determine a way to select a relevant and descriptive subset of comments for the series or episode in question. Being able to select such a subset would be useful in finding good comments to explain the summaries for those entities.

As K increases, the greater the number of words considered to be relevant for the comment representations. A longer vector representation increases the amount of information about each comment, but the usefulness of this information varies, depending on how discriminative the words used as features for the vector are. Greater values of K also facilitate the inclusion of more comments in the selected subset due to how the the K and α parameters interact, i.e., a longer vector is more likely to have its norm be non-zero. If we take, for example, $K = \mathcal{V}$, that is, all words in the data set as the set of *top-K* words, all comments of size at least equal to α would be selected as "relevant", for any value of α.

On the other hand, as α increases, more comments are considered "irrelevant" regardless of the document. Comments with few words from the *top-K* set would be considered to have low descriptive utility for an entity or domain. It also becomes harder for shorter comments to meet this parameter's requirements, in general, likely increasing the average length of the selected comments.

Thus, our goal in this section is to find a representation that can serve as input for classification algorithms so that they can accurately identify those comments that are clearly associated with their series or episode and, consequently, good candidates to describe them.

6.3 Comment classification

For this purpose, we define two classification tasks to identify how well comments represent a given series and episode, respectively, as K and α vary:

- In the first task, the comments are grouped by series and the objective is to classify each comment to the series with which it is associated, among the $|\mathcal{D}|$ series of our data set. For the TF-IDF calculation, a document p is the collection of comments $\mathcal{C}^d$ associated with each series d.
- In the second task, the comments of a given series d are grouped by episode and the objective is to classify each comment to the episode with which it is associated, among the $|\mathcal{T}^d|$ episodes of that series. In this second case, for the TF-IDF calculation, a document p is the collection of comments $\mathcal{C}^t$ associated with each episode t.

To perform such tasks, we used the Naive Bayes [31] classifier from the Weka tool collection[8] [32]. Other classifiers were also tested, and the results were similar. The standard parameters values for the Naive Bayes implementation in Weka were used, and the results were obtained using 10-fold cross-validation. Thus, our goal is to find values of K and α which present a good compromise between the accuracy of the classification and the number of comments correctly classified in each of the tasks. Remember that if the value of K is too large, many words will be used in the classification task and probably many of them will be less discriminative. At the same time, if the value of α is too large, few comments will be considered in the classification task.

Results of the classification tasks are illustrated in Fig. 6. First, observe that identifying which series each comment refers to is very easy. Second, note that considering the comments with no words from the *top-K* set present in the representation causes a considerable loss of accuracy. This follows the expected behavior, since comments of this type do not have any discriminative information in their representations (their representation vectors being entirely comprised of 0s, regardless of domain).

When we begin to classify the comments by episode, given a series, the classification accuracy decreases significantly. This is expected, as the number of possible episodes a comment can belong to is far greater than the number of possible series, as seen in Table 1 (13 series, and 369 episodes in total). Note in Fig. 7 that as we consider more words (greater value of K), it becomes more difficult to identify which episode a comment refers to. In addition, disregarding comments with low informational value (relative to the number of relevant words) has a significant positive impact on the results, especially for lower values of K, although removing too many comments also worsens the accuracy.

On the other hand, in Fig. 8, we can verify that the number of comments considered in the evaluation increases with greater values of K, and decreases with greater values of α. This follows the expected behavior, and shows that we can get a more concise set of comments and with better explanatory capacity for the episode if we choose an appropriate parameter configuration.

Specifically for the data set used in this work, it may be noted that selecting a value of 10 for the parameter K not only considers a smaller number of comments (due to the interaction with α) but also creates more discriminative representations for the comments (higher classification accuracy). However, the parameter α, for $K = 10$, has the best accuracy result for entity classification with $\alpha = 2$, with $\alpha = 3$ achieving an accuracy value less than 1% lower.

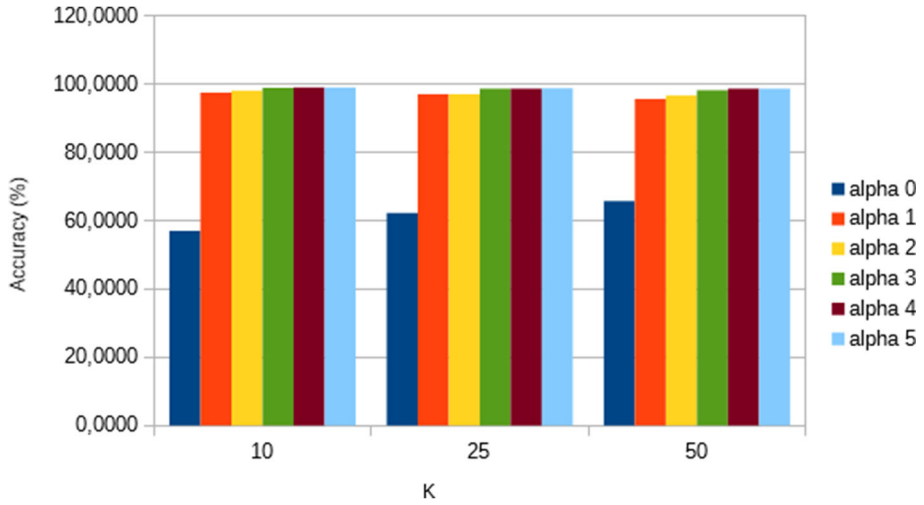


Fig. 6 Correctly classified comments according to domains. Percentages of correctly classified comments in their respective series, according to values of K and α

As for the domain classification task, the classification accuracy is high for any value of α greater than 0, with only insignificant improvements as α increases beyond that. This generates a trade-off in the selection of comments between descriptive potential (we want comments that identify well their series or episode) and succinctness (we want few comments).

7 Summarization methods

In this section we describe the summarization methods used in this work, through which we automatically generate the summaries of each series from the data set. We first explain two word relevance-based comment selection methods, *TKW-AF* and *TKW-MSC*, and then describe another extractive summarization algorithm used for comparison, *TextRank*.

Section 7.1 describes the *TKW-AF* and *TKW-MSC* summarization methods as well as the intuition behind them. Section 7.2 provides a brief explanation for the *TextRank* method.

7.1 Selecting descriptive comments

Since we want to select comments from each domain to act as a summary for the series it is associated with, we need to first find a way to estimate their descriptiveness. In Section 5, we identified the possibility that only few words are easily associated with a single series. In Section 6, we verified that comments which contain

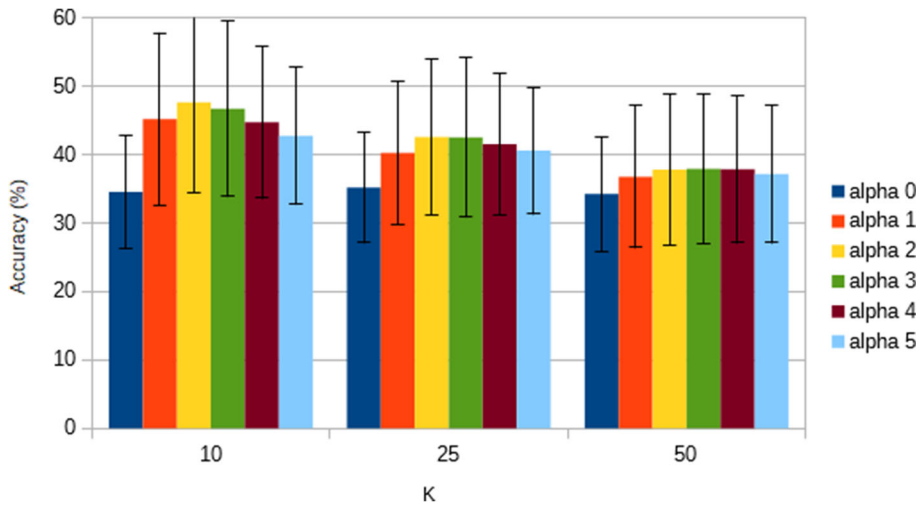


Fig. 7 Correctly classified comments according to entities. Average percentage of comments correctly classified to their respective episodes, according to values of K and α

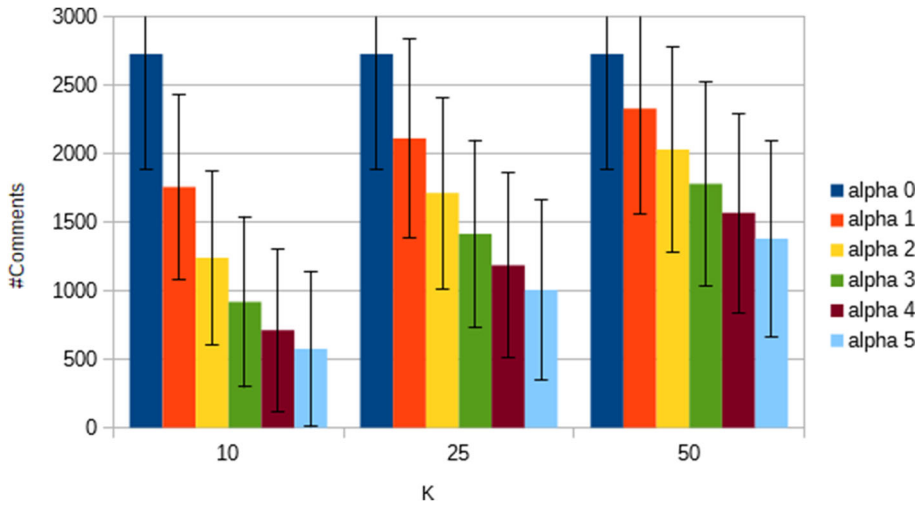


Fig. 8 Classified comments by entity. Average number of comments considered in the classification by episode, according to values of K and α

at least a small number of relevant words are easier to classify according to series and episodes. Taking these conclusions into consideration, we developed comment selection methods based on the *top-K* words concept.

Thus, we determine the K most relevant words $\mathcal{V}^{K_p}$ in document p (either an entity t or a domain d) as described in Section 6. Given these *top-K* words, we are then able to select, with one of the methods described later in this subsection, subsets of comments $\mathcal{C}'^d \subseteq \mathcal{C}^d$ and $\mathcal{C}'^t \subseteq \mathcal{C}^t$, considered as being descriptive of the series and episode, respectively. With this, we consider

$$\mathcal{C}^{\star d} = \mathcal{C}'^d \cup \left(\bigcup_{t \in \mathcal{T}^{\mathcal{D}}} \mathcal{C}'^t \right)$$

as the subset of comments selected as summary for the series d.

Therefore, we propose two methods for selecting descriptive comments based on the *top-K* words, described as follows.

7.1.1 Top-K words - alpha-based filter

The first method for selecting the "most descriptive" comments from a document p given the set of *top-K* words and the resulting comment representations $\mathbf{c}$ is based on the hypothesis that comments with a low number of words from the set of *top-K* words are less likely to describe the series or episode in question. In general, it is a comment filter based on the parameter α, which tells the minimum number of words from the *top-K* set a comment is required to have in order to be selected. This method is referred to as *Top-K Words - Alpha-based Filter* (or *TKW-AF*, for short).

Thus, with each comment c being represented by a vector

$$\mathbf{c} = \left[w_1^c, \ldots, w_{|\mathcal{V}^K|}^c \right],$$

as defined in Section 6, the method *TKW-AF* selects as relevant comments to compose the summary r^d, for a given series d, all comments $c \in \mathcal{C}^d$ for which:

$$\sum_{j}^{|\mathcal{V}^K|} w_j^c \geq \alpha$$

7.1.2 Top-K words - minimum set cover

The second method for selecting the "most descriptive" comments tries to find the least number of comments that, together, are able to cover all the most relevant words for the series or episode. In general terms, it selects as few comments as possible so that all words in the *top-K* set are present in the selected set. This method is referred to as *Top-K Words - Minimum Set Cover* (or *TKW-MSC*, for short).

This can be seen as a particular case of the *Minimum Set Cover* (MSC) problem [33]. The MSC problem can be solved with good results with a simple greedy algorithm. Applying this greedy algorithm to our case, at each step we select the comment with the highest number of words not yet present in the coverage of the set, until all words $\mathcal{V}^K$ are present in the selected comment set.

7.2 TextRank

In order to compare the *top-K* words-based extractive summaries described in Section 7.1 to another method, we also obtained summaries from another extractive summarization method, namely *TextRank* [34]. In short,

TextRank extracts sentences from a document according to its rankings in a graph-based model, calculated similarly to other graph-based ranking algorithms such as PageRank [35].

The *TextRank* algorithm uses a parameter *ratio*, a number between 0 and 1, that determines what proportion of sentences from the input text will be selected for the summary. For most of this article, and unless otherwise specified, we used a ratio of 0.1, since it produces summaries of length close to the average between the ones generated by the *TKW-AF* and *TKW-MSC* methods.

Similarly to what is done with the *top-K* words-based methods, we first generate a *TextRank* summary r^t for each episode t's discussion, and then take all summaries r^t, for $t \in \mathcal{T}^d$, together as the summary r^d of the series d as a whole. We do this so that we guarantee that each episode will have at least a few sentences related to it. This is necessary since our human-written summaries $R^{\mathcal{D}}$ are composed of summaries of each episode R^t, as described in Section 4. As the *TextRank* algorithm takes unprocessed texts as input, we consider the concatenation of the comments (without removing stop words) from $\mathcal{C}^t$ as the input document to be summarized for each episode t.

8 Results

8.1 Summary generation

As described in Section 7, each summarization method generated a summary r^t for each episode $t \in \mathcal{T}^d$ of series d. The concatenation of these summaries for a given series results in the summary r^d of series d as a whole. This summary r^d is then evaluated against the human-written summary R^d of the same series.

For the methods *TKW-AF* and *TKW-MSC*, the parameter K was set to 10, while the value for α was fixed at 3. This was done following the results found at the end of Section 6. We used the *Gensim*[9] implementation of the *TextRank* algorithm, with ratio of 0.1 defined as the default value, as described in Section 7.2.

8.2 Summary evaluation metric

In order to evaluate how well the generated summaries r^d are able to describe each series d, we compare them to the human-written summaries R^d. For this, we use the ROUGE-1 metric [29], as described in Section 5.3. We used this metric to inform how much of the human-written summary R^d (reference text) is covered by the generated summary r^d (candidate text).

8.3 Summarization results

Table 2 shows the results obtained from the summarization methods described in Section 7. For each of the summaries r^d generated by those methods, as well as for the full text of the series discussion $\mathcal{C}^d$, the table indicates the text length, in word count, and the ROUGE-1 score in relation to the human-written summary R^d. The table also shows in parenthesis how well those values for each summarization method compare to the ones obtained if we used the entire discussion text as a "summary". These same results can be seen in Fig. 9, which shows a scatter plot of the ROUGE-1 and text length for each domain in the data set.

Note that larger summaries have better ROUGE-1 values, as expected. However, methods that generate summaries of relatively small sizes (for example, average text length of approximately 10% of the total words from the original discussion, when using *TKW-MSC*) are able to achieve a ROUGE-1 score of at least 50% of that obtained when using the text from all comments in the domain. Overall, the *TKW-MSC* summaries had the lowest values for both text length and ROUGE-1 score, followed by *TextRank* (ratio 0.1), and finally *TKW-AF*.

To verify this even further, we plotted a second scatter plot including multiple variations of *TextRank* summaries, generated with different values of ratio (0.01, 0.1, and 0.9). This can be seen in Fig. 10a. This graph also has a curve indicating the exponential regression obtained from the data, shown in black. We conclude that in order to increase the ROUGE-1 score of the summary linearly, it's necessary to increase the size of the text exponentially. This conclusion can also be reached when considering the results achieved with different values for the *TextRank* ratios. Even with a ratio of 0.9, i.e., using 90% of the sentences for each episode's discussion as summary, it is not possible to achieve a ROUGE-1 value much greater than the results obtained by the *TKW-AF* method, which only uses approximately 40% of the comments' text.

We can also take into consideration the graph from Fig. 10b, which shows what percentage the values from Fig. 10a represent from the maximum achievable with all comments $\mathcal{C}^d$ for each series d. The graph shows that a small portion of a discussion's text is enough to cover a moderate portion of its respective series summary. However, a increasingly higher proportion of words is needed to improve the ROUGE-1 score, to the point that we would need almost all of the series' discussion text to cover as much as possible words from the human-generated summary R^d.

It's also worth noting that for no domain there was a summary (*TKW-AF*, *TKW-MSC*, or *TextRank*) that was Pareto dominated by another, i.e. had both a lower text length and a higher ROUGE-1 score. This defines a clear trade-off between succinctness and descriptiveness for the studied methods, as the analysis from Section 6 indicated.

8.4 Selected comments

In order to get a better grasp at what the contents of the selected comments indicate, and what constitutes a

Table 2 Results table

Series ID	Concatenated $\mathcal{C}^d$ text		*TKW-AF*		*TKW-MSC*		*TextRank*	
	Text length	ROUGE-1	Text length	ROUGE-1	Text length	ROUGE-1	Text length	ROUGE-1
1	48800	0.6642	15444 (31.65%)	0.5087 (76.59%)	4202 (8.61%)	0.3750 (56.46%)	9545 (19.56%)	0.4201 (63.24%)
19	98412	0.7800	52210 (53.05%)	0.7152 (91.70%)	13186 (13.40%)	0.5657 (72.53%)	19912 (20.23%)	0.5705 (73.14%)
30	99388	0.7821	37417 (37.65%)	0.6893 (88.13%)	6877 (6.92%)	0.4578 (58.53%)	19580 (19.70%)	0.5683 (72.67%)
205	32171	0.6834	11446 (35.58%)	0.5620 (82.24%)	3270 (10.16%)	0.4011 (58.69%)	6297 (19.57%)	0.4433 (64.86%)
226	36146	0.7500	14074 (38.94%)	0.6496 (86.61%)	2470 (6.83%)	0.3675 (49.00%)	7692 (21.28%)	0.5406 (72.08%)
356	59593	0.7852	27185 (45.62%)	0.6984 (88.95%)	6194 (10.39%)	0.4552 (57.97%)	13529 (22.70%)	0.5661 (72.10%)
457	56062	0.8452	15958 (28.46%)	0.7619 (90.14%)	4809 (8.58%)	0.6111 (72.30%)	11044 (19.70%)	0.6786 (80.28%)
777	43624	0.6724	15165 (34.76%)	0.5666 (84.26%)	3547 (8.13%)	0.4061 (60.41%)	9234 (21.17%)	0.4642 (69.04%)
790	50300	0.7566	21334 (42.41%)	0.6675 (88.22%)	6855 (13.63%)	0.5084 (67.20%)	9908 (19.70%)	0.5229 (69.11%)
820	114110	0.8073	50503 (44.26%)	0.7205 (89.25%)	11416 (10.00%)	0.5088 (63.02%)	22596 (19.80%)	0.5645 (69.92%)
877	67725	0.7962	34445 (50.86%)	0.7208 (90.52%)	7960 (11.75%)	0.5491 (68.96%)	13406 (19.79%)	0.5849 (73.46%)
934	82331	0.7458	28013 (34.02%)	0.6325 (84.81%)	6091 (7.40%)	0.4305 (57.73%)	17586 (21.36%)	0.5527 (74.11%)
13599	99245	0.7856	27932 (28.14%)	0.6689 (85.15%)	6200 (6.25%)	0.4106 (52.27%)	21323 (21.49%)	0.5674 (72.23%)
Average	68300.54	0.7580	27009.69 (39.55%)	0.6586 (86.89%)	6390.54 (9.36%)	0.4651 (61.36%)	13973.23 (20.46%)	0.5418 (71.48%)

The table shows for each series: the length (number of words) of the concatenated texts of all comments $c \in \mathcal{C}^d$, the ROUGE-1 score of the this concatenated text in respect to the human-written summary R^d, and the lengths and ROUGE-1 scores for the summaries generated with methods *TKW-AF*, *TKW-MSC*, and *TextRank*, with how much of the respective values from the full text of $\mathcal{C}^d$ they represent indicated in parenthesis

descriptive comment, we analyzed some of the comment sets defined by the *top-K* words-based methods.

Figure 11a shows, for a particular series (Dom_{457}), a word cloud generated from comments that had no words from those in the *top-K* words set. We can immediately notice that none of the words presented as most frequent seem to be specific to the series. In fact, the most used words from these "discarded comments" appear to be quite simple and generic, specially taking in consideration words that would be used for any domain in the data set, such as "*episode*" and "*ending*". Another point of interest that can be noticed from this word cloud is that several of the words in it express some sentiment, like "*good*", "*nice*", and "*sad*". This indicates a tendency for non-descriptive or

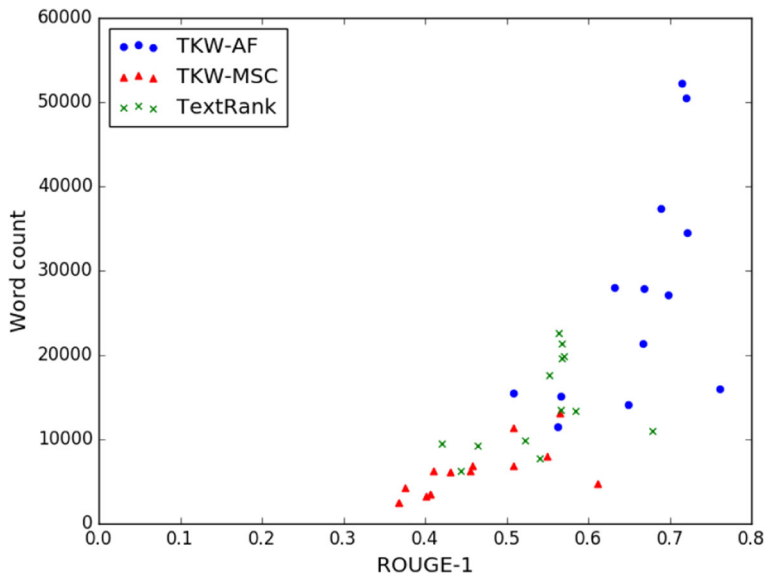
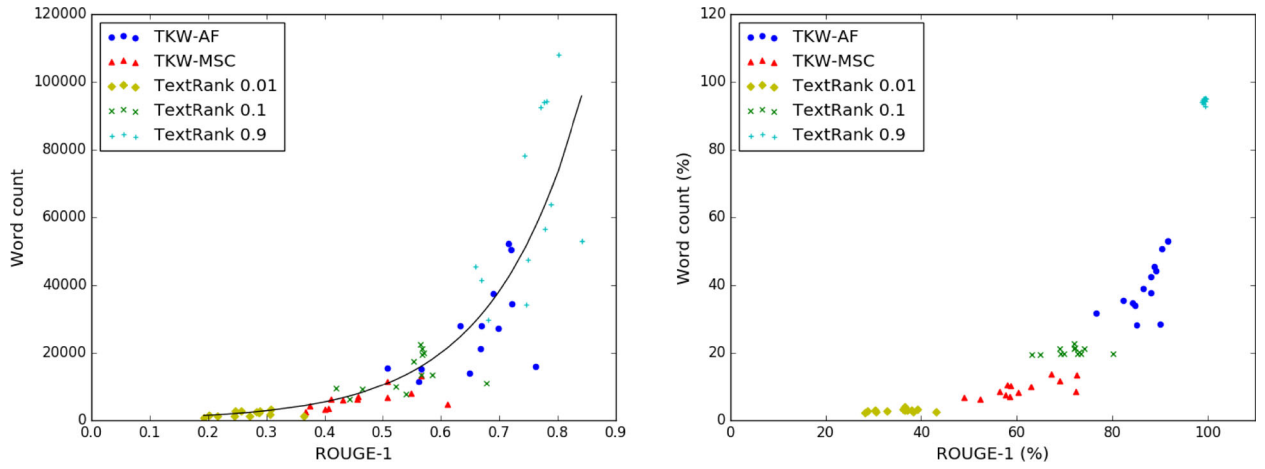


Fig. 9 Size and ROUGE-1 of summarization methods. Size (number of words) and ROUGE-1 of the different summarization methods, for each domain



(a) Word Count X ROUGE-1 (b) Word Count (%) X ROUGE-1 (%)

Fig. 10 Size and ROUGE-1 of summarization methods, with varied *TextRank* parameters. **a** shows the word count and ROUGE-1 score for the summaries obtained from each summarization method, for each series in the data set. **b** shows the same results with the axis indicating the percentage of the respective value obtained with the text from all comments $\mathcal{C}^d$ as summary

"irrelevant" comments to have higher positive sentiment values.

In comparison to that, Fig. 11b shows a word cloud, for the same series Dom_{457}, generated from comments that had at least one word from those in the *top-K* words set. It may be noted that, while there are some generic prevalent words, such as "*think*", "*really*" and "*episode*", several of them are specific to the series, or at least closely related to it, e.g. "*Ginko*", "*Mushi*", "*time*". Words that express sentiment also appear much smaller in this word cloud than in Fig. 11a, further pointing towards a higher positive sentiment value for "irrelevant" comments.

To verify this, we compared the intensity of sentiment polarities (negative, neutral, and positive) from the discarded comments (no words from the *top-K* words set) to the selected comments (at least one word from the *top-K* words set). These sentiment polarity values for each comment were obtained with the *Vader* model [36], which, in short, incorporates the sentiment intensity of lexical features into a rule-based model that evaluates the sentiment of a given text.

Figure 12 shows the average polarity of negative, neuter, and positive sentiments for both the discarded and the selected comments. We notice that, in fact, discarded comments tend to have a higher positive sentiment polarity than selected comments, while selected comments tend to have higher neuter sentiment polarity than discarded comments. We also verified that same relationship is found when considering comments from each series separately, so we can say that this is a common behavior among all domains from our data set. The average negative sentiment polarity is also higher for the selected

(a) (b)

Fig. 11 Comment word clouds, according to presence or absence of relevant words. Figure 10a shows the word cloud generated from comments of series Dom_{457} that had no words from those in the *top-K* words set. Figure 10b shows the word cloud generated from comments of series Dom_{457} that had at least one word from those in the *top-K* words set. **a** Discarded Comments. **b** Selected Comments

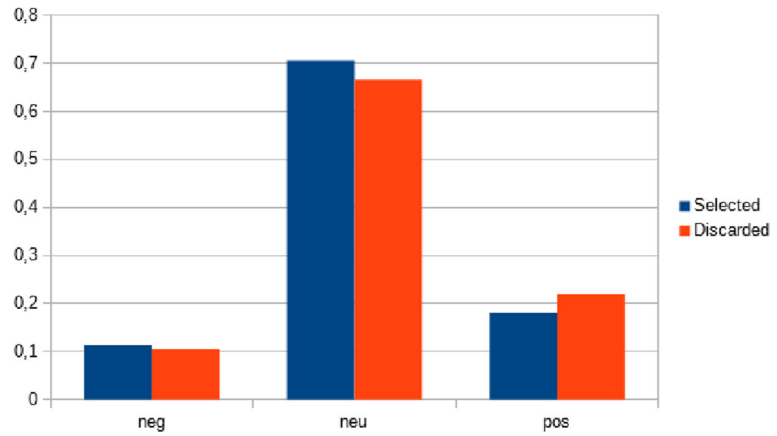


Fig. 12 Sentiment polarities for selected and discarded comments. Negative, neuter and positive sentiment polarities for Selected and Discarded comments

comments, although this isn't true for all domains, when considering comments from each of them separately.

Considering that the *Vader* model evaluated a large amount of comments as having high neuter sentiment, a deeper analysis may be required to better understand how comment sentiment polarity and descriptiveness are related, perhaps using other sentiment analysis methods, such as the ones used in the *iFeel 2.0* benchmarking system [37].

9 Conclusions and future works

This paper presented a characterization of the potential of comments as a single data source for explaining and describing entities. We analyzed characteristics of online discussions of TV series episodes, such as the size of the comments, relevance of the words, and the ability of comments to explain a human-written summary extracted from Wikipedia.

Based on this characterization, we implemented a series of extractive summarization methods. The first two, *TKW-AF* and *TKW-MSC*, used a set of discriminative words, also referred to as "*top-K* words", to select highly descriptive comments for the episodes, and then for the series. The third method, using the *TextRank* algorithm, selected relevant sentences from each episode's discussion following a graph-based ranking algorithm.

All the tested methods were able to achieve a relatively good coverage of the human-written summary while using only a small fraction of the original text. However, each summarization method obtained results with a different focus: *TKW-AF* got the best ROUGE-1 scores for its summary, while having a slightly larger text; *TKW-MSC* sacrificed part of its summary expressiveness in favor of smaller texts; finally, the *TextRank* algorithm manages to pick a higher ROUGE-1 score or a lower text length, according to its parameter's value.

With the characterization studies we performed, including a classification task, we verified that online discussions have textual content that make it so that we can easily identify which series they are referring to, and to a lesser degree, the episode they are referring to. With the results obtained in the summarization task, we were able to find small subsets of comments that had enough descriptive content to cover a good part of the summary for each series. Considering these results, we conclude that we can, in fact, use online comments to describe television series.

In future works, we plan to generate abstractive summaries, which try to reflect more closely what people naturally do when they write a summary about a given entity. In an abstractive summarizer, after extracting the knowledge from the comments, the most informative content is selected and expressed in natural language. As pointed out in [38], abstractive summarization is usually a much more complex and challenging task than the extractive one, since it requires a natural language generation module and also a domain dependent component to process and rank the extracted knowledge.

Another aspect we analyzed was difference in vocabulary between types of comments, and the tendency comments considered to be particularly "non-informative" have to express more positive sentiments than those considered to have at least some useful information (that is, comments that have at least one of the discriminative words from the *top-K* set). This discrepancy in sentiment values indicates that it's possible to design a method for identifying relevant/irrelevant comments for a determinate topic, or even entire comment sequences, based on sentiment polarity and word usage. Such a method could be used to automatically find out-of-topic comments, spam, or flamers. For that, a deeper study on how these sentiment values differ according to the type

of comment would be required, properly identifying how these behaviors change according to different contexts.

Endnotes

[1] https://www.theguardian.com/science/brain-flapping/2014/sep/12/comment-sections-toxic-moderation

[2] http://www.imdb.com/title/tt2178784/plotsummary?ref_=tt_stry_pl

[3] https://en.wikipedia.org/wiki/The_Rains_of_Castamere

[4] https://myanimelist.net/

[5] An example of one of these pages can be found at https://en.wikipedia.org/wiki/List_of_Mushishi_episodes.

[6] Each series can be accessed at *https://myanimelist.net/anime/<ID>*, replacing <ID> with the values given in the table.

[7] https://lvdmaaten.github.io/tsne/

[8] Available at http://www.cs.waikato.ac.nz/ml/weka/index.html

[9] https://radimrehurek.com/gensim/summarization/summariser.html

Abbreviations

MAL: MyAnimeList.net; TF-IDF: term frequency – inverse document frequency; TK-AF: Top-K Words - Alpha-based Filter; TK-MSC: Top-K Words - Minimum Set Cover.

Acknowledgements

The authors would like to thank the *Google Research Awards for Latin America* program for the financial support it provided.

Availability of data and materials

Please contact author for data requests.

Authors' contributions

TCL is a Master's student at DCC-UFMG, and has come up with the initial idea for the research, and worked on most of the implementation. PVM is the advisor for TCL, and helped in several problem definitions, besides proposing several analyses, classification, and summarization methods. AAV is the co-advisor for TCL, and helped direct the work with his background in Machine Learning and Natural Language Processing. In addition, all authors participated in discussions regarding the research and had some input in the writing phase of the final document. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 6 March 2018 Accepted: 2 October 2018
Published online: 15 December 2018

References

1. Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35(8):1798–1828.
2. Ji Y, Eisenstein J. Representation Learning for Text-level Discourse Parsing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Baltimore: Association for Computational Linguistics; 2014. p. 13–24.
3. von Ahn L, Dabbish L. Labeling Images with a Computer Game. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '04. New York: ACM; 2004. p. 319–326.
4. Hsu C-F, Khabiri E, Caverlee J. Ranking Comments on the Social Web. In: Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04. CSE '09. Washington, DC: IEEE Computer Society; 2009. p. 90–97.
5. Siersdorfer S, Chelaru S, Pedro JS, Altingovde IS, Nejdl W. Analyzing and Mining Comments and Comment Ratings on the Social Web. ACM Trans Web. 2014;8(3):17:1–17:39.
6. Cheng J, Danescu-Niculescu-Mizil C, Leskovec J. Antisocial Behavior in Online Discussion Communities. In: Cha M, Mascolo C, Sandvig C, editors. ICWSM. Oxford: AAAI Press; 2015.
7. Radev DR, Jing H, Budzikowska M. Centroid-based summarization of multiple documents. In: NAACL-ANLP 2000 Workshop on Automatic Summarization -, vol. 4. Morristown: Association for Computational Linguistics; 2000. p. 21–30. http://portal.acm.org/citation.cfm?doid=1117575.1117578.
8. Liu F, Flanigan J, Thomson S, Sadeh NM, Smith NA. Toward Abstractive Summarization Using Semantic Representations. In: Mihalcea R, Chai JY, Sarkar A, editors. HLT-NAACL. Denver: The Association for Computational Linguistics; 2015. p. 1077–1086. http://dblp.uni-trier.de/db/conf/naacl/naacl2015.html#0004FTSS15.
9. Moratanch N, Chitrakala S. A survey on abstractive text summarization. In: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). Nagercoil: IEEE; 2016. p. 1–7. http://ieeexplore.ieee.org/document/7530193/.
10. Gambhir M, Gupta V. Recent automatic text summarization techniques: a survey. Artif Intell Rev. 2017;47(1):1–66.
11. Lu Y, Zhai C, Sundaresan N. Rated aspect summarization of short comments. In: Proceedings of the 18th International Conference on World Wide Web - WWW '09. New York: ACM Press; 2009. p. 131. http://portal.acm.org/citation.cfm?doid=1526709.1526728.
12. Ganesan K, Zhai C, Han J. Opinosis : A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. In: Proceedings of the 23rd International Conference on Computational Linguistics. COLING '10. Stroudsburg: Association for Computational Linguistics; 2010. p. 340–348. http://dl.acm.org/citation.cfm?id=1873781.1873820.
13. Potthast M, Becker S. Opinion Summarization of Web Comments. Berlin: Springer-Verlag; 2010. p. 668–669. http://link.springer.com/10.1007/978-3-642-12275-0_73.
14. Ganesan K, Zhai C, Viegas E. Micropinion generation. In: Proceedings of the 21st International Conference on World Wide Web - WWW '12. New York: ACM Press; 2012. p. 869. http://dl.acm.org/citation.cfm?doid=2187836.2187954.
15. Kleinberg J. Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '02. New York: ACM; 2002. p. 91–101.
16. Choi D, Han J, Chung T, Ahn Y-Y, Chun B-G, Kwon TT. Characterizing Conversation Patterns in Reddit: From the Perspectives of Content Properties and User Participation Behaviors. In: Proceedings of the 2015 ACM on Conference on Online Social Networks. COSN '15. New York: ACM; 2015. p. 233–243.
17. Khabiri E, Caverlee J, Hsu C-F. Summarizing User-Contributed Comments. In: ICWSM. Barcelona: The AAAI Press; 2011.
18. Yang Z, Cai K, Tang J, Zhang L, Su Z, Li J. Social context summarization. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information - SIGIR '11. New York: ACM Press; 2011. p. 255. http://portal.acm.org/citation.cfm?doid=2009916.2009954.
19. Kågebäck M, Mogren O, Tahmasebi N, Dubhashi D. Extractive Summarization using Continuous Vector Space Models. In: Proceedings of the 2nd Workshop on Continuous Vector Space Models and Their Compositionality (CVSC). Gothenburg: Association for Computational Linguistics; 2014. p. 31–39. http://www.aclweb.org/anthology/W14-1504.

20. Liu C-Y, Chen M-S, Tseng C-Y. IncreSTS: Towards Real-Time Incremental Short Text Summarization on Comment Streams from Social Network Services. IEEE Trans Knowl Data Eng. 2015;27(11):2986–3000.
21. Chua FCT, Asur S. Automatic Summarization of Events from Social Media. In: Kiciman E, Ellison NB, Hogan B, Resnick P, Soboroff I, editors. ICWSM. Boston: The AAAI Press; 2013.
22. Vaz de Melo POS, Faloutsos C, Assuncao R, Loureiro AAF. The Self-feeding Process: A Unifying Model for Communication Dynamics in the Web. In: Proceedings of the 22Nd International Conference on World Wide Web. WWW '13. New York: ACM; 2013. p. 1319–1330.
23. Vaz de Melo POS, Faloutsos C, Assunção R, Alves R, Loureiro AAF. Universal and Distinct Properties of Communication Dynamics: How to Generate Realistic Inter-event Times. ACM Trans Knowl Discov Data. 2015;9(3):24:1–24:31.
24. Alves R, Assunção R, Vaz de Melo POS. Burstiness Scale: A Parsimonious Model for Characterizing Random Series of Events. In: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. New York: ACM; 2016. p. 1405–1414.
25. Zuin GL, Magalhães LFG, Loures TC. MAL-FITT: MyAnimeList Forum Interpreter Through Text. In: XIII Encontro Nacional de Inteligência Artificial e Computacional (SBC ENIAC-2016). Recife-PE: SBC; 2016. p. 205–216.
26. Loures TC, Vaz de Melo POS, Veloso A. É possível descrever episódios de séries de televisão a partir de comentários online? In: Brazilian Workshop on Social Network Analysis and Mining (BraSNAM). São Paulo: SBC; 2017.
27. Loures TC, Vaz de Melo POS, Veloso A. Generating entity representation from online discussions: Challenges and an evaluation framework. In: Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web. WebMedia '17. New York: ACM; 2017. p. 197–204. http://doi.acm.org/10.1145/3126858.3126882.
28. van der Maaten LJP, Hinton GE. Visualizing high-dimensional data using t-sne. J Mach Learn Res. 2008;9:2579–605.
29. Lin C-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In: Moens M-F, Szpakowicz S, editors. Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. Barcelona: Association for Computational Linguistics; 2004. p. 74–81.
30. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. CoRR. 2013;abs/1301.3781.
31. Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn. 1997;29(2–3):131–63.
32. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The weka data mining software: An update. SIGKDD Explor Newsl. 2009;11(1):10–8.
33. Chvatal V. A greedy heuristic for the set-covering problem. Math Oper Res. 1979;4(3):233–5.
34. Mihalcea R, Tarau P. TextRank: Bringing Order into Texts. In: Lin D, Wu D, editors. Proceedings of EMNLP. Barcelona: Association for Computational Linguistics; 2004. p. 404–411.
35. Page L, Brin S, Motwani R, Winograd T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999–66, Stanford InfoLab (November 1999). Previous number = SIDL-WP-1999-0120. http://ilpubs.stanford.edu:8090/422/.
36. Hutto CJ, Gilbert E. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In: Adar E, Resnick P, Choudhury MD, Hogan B, Oh AH, editors. ICWSM. Ann Arbor: The AAAI Press; 2014.
37. Araújo M, Diniz JP, Bastos L, Soares E, Júnior M, Ferreira M, Ribeiro F, Benevenuto F. iFeel 2.0: A Multilingual Benchmarking System for Sentence-Level Sentiment Analysis. In: Proceedings of the International AAAI Conference on Web-Blogs and Social Media. Cologne: AAAI Press; 2016. p. 758–759.
38. Carenini G, Murray G, Ng R. Methods for Mining and Summarizing Text Conversations. Synth Lect Data Manag. 2011;3(3):1–130.