



UNIVERSITÀ POLITECNICA DELLE MARCHE  
Repository ISTITUZIONALE

Representation, detection and usage of the content semantics of comments in a social platform

This is the peer reviewed version of the following article:

*Original*

Representation, detection and usage of the content semantics of comments in a social platform / Bonifazi, G.; Causeruccio, F.; Corradini, E.; Marchetti, M.; Terracina, G.; Ursino, D.; Virgili, L. - In: JOURNAL OF INFORMATION SCIENCE. - ISSN 1741-6485. - 50:2(2024), pp. 317-341. [10.1177/01655515221087663]

*Availability:*

This version is available at: 11566/296426 since: 2024-05-07T13:07:18Z

*Publisher:*

*Published*

DOI:10.1177/01655515221087663

*Terms of use:*

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. The use of copyrighted works requires the consent of the rights' holder (author or publisher). Works made available under a Creative Commons license or a Publisher's custom-made license can be used according to the terms and conditions contained therein. See editor's website for further information and terms and conditions.

This item was downloaded from IRIS Università Politecnica delle Marche (<https://iris.univpm.it>). When citing, please refer to the published version.

note finali coverpage

(Article begins on next page)

# Representation, detection and usage of the content semantics of comments in a social platform

Gianluca Bonifazi<sup>1</sup>, Francesco Cauteruccio<sup>1</sup>, Enrico Corradini<sup>1</sup>, Michele Marchetti<sup>1</sup>, Giorgio Terracina<sup>2</sup>, Domenico Ursino<sup>1</sup>, and Luca Virgili<sup>1</sup>

<sup>1</sup> DII, Polytechnic University of Marche

<sup>1</sup> DEMACS, University of Calabria

\* Contact Author

[g.bonifazi@univpm.it](mailto:g.bonifazi@univpm.it); [f.cauteruccio@univpm.it](mailto:f.cauteruccio@univpm.it); [e.corradini@pm.univpm.it](mailto:e.corradini@pm.univpm.it);  
[michele.marchetti97@hotmail.it](mailto:michele.marchetti97@hotmail.it); [terracina@mat.unical.it](mailto:terracina@mat.unical.it); [d.ursino@univpm.it](mailto:d.ursino@univpm.it);  
[l.virgili@pm.univpm.it](mailto:l.virgili@pm.univpm.it)

## Abstract

The analysis of people’s comments in social platforms is a widely investigated topic because comments are the place where people show their spontaneity most clearly. In this paper, we present a network-based data structure and a related approach to represent and manage the underlying semantics of a set of comments. Our approach is based on the extraction of text patterns that take into account not only the frequency but also the utility of the analyzed comments. Our data structure and approach are “multi-dimensional” and “holistic”, in the sense that they can simultaneously handle content semantics from multiple perspectives. They are also easily extensible, because additional content semantics perspectives can be easily added to them. Furthermore, our approach is able to evaluate the semantic similarity of two sets of comments. In this paper, we also illustrate the results of several tests we conducted on Reddit comments, even if our approach can be applied to any social platform. Finally, we provide an overview of some possible applications of this research.

**Keywords:** Comment analysis; Social Network Analysis; Text Pattern Mining; Semantic Similarity; Reddit

## 1 Introduction

In recent years, content analysis of people’s comments on social media has received an increasing boost [6, 9, 53, 14] as part of a trend that has affected a wide variety of contexts somehow related to data and process analysis (see, for instance, [23, 22]). In fact, comments on social media represent one of the places where a person expresses her opinion on certain topics most spontaneously [11, 21, 58]. As a consequence, they are an extremely powerful tool to know the true feelings and thoughts of a person

and, ultimately, to reconstruct her profile [20, 45, 13, 1, 5, 40]<sup>1</sup>. However, while spontaneity is the main strength of comments, it can also become their main weakness. Indeed, just because comments are written on the spot, their content is often unstructured, sometimes apparently confused, other times apparently contradictory. Nevertheless, there is no doubt that an in-depth analysis of a large set of comments, written for example by a single user, could allow the extraction of a “fil rouge”, a common thread representing a thought, a content profile beyond the apparent inconsistencies of single comments. However, identifying this “fil rouge” requires a very thorough and holistic analysis of the content semantics.

In this paper, we aim at providing a contribution in this setting by proposing a data structure and a related approach to extract content semantics from a set of comments. In our experiments, we focused on Reddit comments and posts. However, as we will see, our approach is general and can also be employed in other social platforms. The activities that our approach performs on comment content are many, but they can be grouped into two phases, which we can call “pre-processing” and “knowledge extraction”.

The pre-processing phase aims at cleaning and annotating available comments and, then, selecting the most significant ones. Cleaning is necessary to remove bot-generated content, errors, inconsistencies, etc., as well as to perform tokenization and lemmatization of comments. Annotation allows important information to be added to each lemmatized comment automatically. Examples of this information are the sentiment value associated with the comment, the post which it refers to, the author who wrote it, etc.

Filtering is based on text pattern mining tasks and is used to identify the most significant lemmatized and annotated comments. In order to carry out this activity, our approach takes into account not only the frequency of patterns, as most of the approaches proposed in the past literature do [3, 25, 27], but also, and above all, their utility [27, 4, 46, 29], measured on the basis of a utility function. Interestingly, our approach is orthogonal to the utility function used and, therefore, choosing different utility functions allows it to give priority to certain properties of comments instead of other ones. A first utility function could be the sentiment of the comments in order to select, for instance, patterns involving only positive comments or only negative ones. A second utility function could be the comment rate, which would allow our approach to select, for example, patterns involving only high rate comments or only low rate ones. A third utility function could concern the Pearson’s correlation [49] between sentiment and rate, which would allow it to select, for instance, patterns involving only comments with discordant sentiment and rate or only comments whose sentiment and rate are in agreement with each other.

Once the comments and patterns of interest have been selected, our approach defines a data structure for their representation, which we call CS-Net (Content Semantics Network). The nodes of a CS-Net represent comments’ lemmas. Its arcs can be of two types, reflecting two different perspectives of viewing content semantics. The first is based on the concept of co-occurrence and considers that two semantically related lemmas tend to appear together very often in sentences. It summarizes the results of many researches carried out in the field of Information Retrieval [19]. The second concerns the concept of relationships and semantically related terms. It summarizes many researches carried

---

<sup>1</sup>In this paper, we deal with comments written by people to whom correspond well-defined accounts. We do not consider anonymous comments both because they are less reliable and because they would be useless for our research.

out in the field of Natural Language Processing [12]. The CS-Net model is extensible so that, if we want to consider further content semantics perspectives in the future, it will be sufficient to add another type of arcs for each new perspective.

The last contribution of this paper concerns the definition of an approach to evaluate the semantic similarity of two CS-Nets. It takes into account the two components that are represented by the CS-Net arcs (i.e., co-occurrences and semantic relationships), weighting them differently, based on their extension (and, thus, on the number of their arcs). In particular, our approach privileges the most extended component because it represents a greater portion of the content semantics than the other. Analogously to the CS-Net model, our approach can be easily extended in case we want to add further content semantics perspectives.

Our approach first evaluates separately, and then combines appropriately, the semantic similarity of each pair of subnetworks obtained starting from the original CS-Nets and considering only one content semantics perspective. When evaluating the semantic similarity of a pair of homogeneous subnetworks (i.e., subnetworks of only co-occurrences or subnetworks of only semantic relationships), it considers two additional aspects, namely the topological similarity of the subnetworks and the similarity of the concepts expressed by their nodes. The former is computed using an approach already proposed in the literature, i.e., NetSimile [10]. The second is determined by computing an enhanced version of the Jaccard coefficient, capable of taking synonymies and homonymies into account. Considering these two additional features (i.e., topological and concept similarities) in the computation of the semantic similarity of the subnetworks, together with the two features adopted for the overall networks (i.e., co-occurrences and semantic relationships), makes our overall approach even more holistic.

We believe that the approach and data structure proposed in this paper allow us to extract the “fil rouge” connecting a set of comments. We mentioned above that if these were the comments published by a single user, we could employ the extracted knowledge to reconstruct her profile. However, this is not the only possible application of our approach. In fact, the comments under consideration could also be those written by more users on a single community, or a set of comments on a certain topic (e.g., COVID-19) or a set of comments written during a certain time period (e.g., during the Tokyo Olympics).

Depending on the set of comments, which it operates on, our approach has several applications. These may concern, for example, the construction of content-based or collaborative filtering recommender systems, the construction of new user communities, the identification of outliers or the construction of new thematic forums (e.g., subreddits in Reddit) from the existing ones. Some of the most interesting applications will be described below.

The outline of this paper is as follows: In Section 2, we discuss the related literature. In Section 3, we illustrate the pre-processing activities of our approach, devoted to comment filtering and text pattern extraction. In Section 4, we present the CS-Net model. In Section 5, we describe our approach for evaluating the semantic similarity of two CS-Nets. In Section 6, we present the experiments we performed to test our approach. In Section 7, we provide an overview of some of its possible applications. Finally, in Section 8, we draw our conclusions and have a look at some possible future developments.

## 2 Related work

Research on social networks has undoubtedly gained a lot of attention over the past few decades. This is motivated by both the enormous growth of the social network phenomenon and the variety of ways in which social networks interface with users. In such a scenario, the analysis of content semantics has become a hot topic, because it allows researchers to investigate phenomena in greater depth than they could do with the structural analysis of networks alone. Our paper is positioned exactly in this context.

One of the research lines most closely related to the one characterizing our paper is semantic network analysis [28, 37, 61]. It examines the way in which two words are associated with each other within a set of texts. To this end, it constructs suitable networks whose nodes represent words and whose arcs denote ties between words. In general, dyadic ties are considered to represent how frequently a pair of words co-occurs within a textual atom of analysis, e.g., a paragraph or a sentence. The networks thus constructed are investigated by means of the concepts and theories of classic network analysis. Our approach shares some similarities with the semantic content analysis ones. In particular, it also builds networks from content and relationships between text units, i.e., comments. An important distinction between our approach and the semantic network analysis ones is that the latter does not consider classic text mining; instead, the former leverages pattern mining and utility functions to identify representative concepts in a text. Furthermore, our approach uses two very different sets of arcs between nodes, and further sets may be added in the future. Instead, semantic network analysis approaches use only one set of arcs. As pointed out in [15], semantic network analysis approaches could be improved by including statistical testing in word selection. Our approach goes exactly in that direction when it considers utility functions and text patterns for the selection of the words to consider for the network construction.

An approach using semantic analysis, in combination with social network analysis, is presented in [28]. Here, the authors analyze online travel forums to predict tourism demand. Specifically, they first extract data from TripAdvisor using specifically crafted crawlers. Then, they focus on data from seven European capitals. For each city, they construct a social network whose nodes represent users; an arc from  $u$  to  $v$  indicates that  $u$  responded to a post submitted by  $v$ . After that, they use group degree and betweenness centrality to study the connectivity of created networks. Finally, they analyze language usage by considering two dimensions, namely sentiment and complexity. The former indicates whether the post is positive or negative, while the latter measures the complexity of the language used in the post. The approach of [28] shares few similarities with ours. Indeed, both of them use a measure of sentiment. However, the goals and the way it is used are very different in the two cases.

In [37], the authors present an approach that uses semantic network analysis to study social media rumors in Twitter discourses during a specific event. They collected 16,000 tweets selected through different keywords. After a pre-processing phase, they get 2,300 unique tweets, which they use to perform two analyses. The former is a content analysis to verify if a tweet contains noise or not. The latter uses semantic network analysis and creates three networks starting from three different sets of labeled tweets. The nodes of these networks are selected through a probabilistic approach; specifically, a node appears in a network if the corresponding word has the highest frequency probability in the related set of tweets. The authors combine content and semantic network analysis to study the three

networks and identify clusters. A component of the approach of [37] uses semantic network analysis; from this point of view, we can consider this approach related to ours. However, the goals of the two approaches and the methodologies to achieve them are very different. Furthermore, in our approach, the CS-Net model has two types of arcs, representing co-occurrences and semantic relationships, and further arc types could be added in the future. The network used in the approach of [37] is more classical and has only one type of arcs.

In [61], the authors adopt semantic network analysis to investigate user experiences on mental disorders shared on Reddit. Specifically, they consider two subreddits, namely `/r/Bipolar` and `/r/Depression`. They initially collect posts from these two subreddits. Then, they perform a pre-processing activity to obtain the set of words present in them, along with the corresponding Term Frequency - Inverse Document Frequency (TF-IDF) [7] values. Starting from that, they build a word matrix and a semantic network for both subreddits. In addition, they analyze the emotional component of words by means of LIWC (Linguistic Inquiry and Word Count) [56], a software that allows the analysis and categorization of texts according to word class, emotions and speech features. Finally, they exploit additional indicators, such as authenticity and emotional tone. Using all these tools, they compare the characteristics of the two networks and draw several conclusions. The approach of [61] shares several similarities with ours. In particular, both of them consider text content, although the former restricts the analysis to words for semantic network construction, while the latter considers text patterns, occurrences and semantic relationships also obtained from utility functions and knowledge bases. Both approaches consider emotional values. However, our approach adopts them to define the utility functions contributing to the extraction of patterns, which are then used to build the network. Instead, the approach of [61] performs only a static analysis of such values.

In the context of community detection approaches, the content and semantics of the underlying network are often analyzed. In [50], the authors propose a community detection approach using topological and content information. In particular, they adopt a non-negative matrix factorization. They also address the overlapping community discovery problem. The approach of [50] uses a network in which each node is associated with one or more attributes. It considers the mismatch between network topology and content to measure how much a community represents a set of similar nodes. The approach of [50] and ours are similar in that both of them consider content to achieve their goals. However, they differ both in their goals and in the methodology for achieving them. In particular, the approach of [50] is more focused on the topological aspects of the network and does not consider semantic relationships between words. Moreover, it also handles the problem of community overlapping, which is not addressed by our approach. However, the latter is scalable and allows the easy addition of new types of arcs to the network, each representing a new perspective that we decide to handle.

In [39], the authors propose a community detection approach using Markov-network based models and frequent pattern mining. Here, the goal is to study the behavioral interactions among users and discover latent links between social objects. To perform the latter task, the authors employ frequent patterns in behavioral observation. Specifically, they first mine frequent patterns and then build a Markov-based model from them. The latter is used in the algorithm for community detection, which aims to find a certain type of maximum clique. Both the approach of [39] and ours use frequent pattern mining in Social Network Analysis. However, our approach extends this technique by considering the

utility of a pattern. Furthermore, it also considers semantic relationships between words.

In [52], the authors propose a community detection framework in ranking-based social networks. They aim to find overlapping communities in which members are interested in the same topic, with their relationships measured based on the rate of their viewpoints. In particular, the authors study the case of a social network where users can rate movies and each movie has a genre. Their approach creates topological subgroups for each genre. It computes the semantic relationships between users by weighing their communications; these also include the rating of the same movie. The approach of [52] and ours share the generic idea of using content for their goals. However, they are quite different. In fact, the approach of [52] considers a limited representation of content in a social network to identify communities.

Topic oriented community detection is also studied in [62]. Here, the authors combine social objects clustering and link analysis to define the semantics within a network. Their methodology is very similar to the one proposed in [52]. However, they consider the text involved in user interactions as social objects. In fact, they define a text social object as a set of pairs  $\langle w, m \rangle$ , where  $w$  is a word and  $m$  is a measure of  $w$ , e.g., the corresponding TF-IDF value. They cluster users involved in these social objects to identify topical communities. The main similarity between the approach of [62] and ours is in the use of content to achieve their goals. However, the latter and the way to reach them are very different from our approach.

An interesting recent work is the one presented in [51]. Here, the authors perform a socio-semantic analysis of a particular context concerning the Italian “twittersphere” along a period of eight months. In particular, they collect a set of approximately 5 million targeted tweets from a collection of hashtags. Then, they identify a set of communities and analyze them from both a structural and a temporal viewpoint. After that, they consider the semantics of collected data and identify conductive hashtags, i.e., the most relevant hashtags representing entities like topics, actors, etc. Finally, they study communities at a mesoscale level by applying both a k-core and a core-periphery decomposition. The latter analysis allows them to conclude that hashtags are hierarchically arranged within discussions and that the most relevant ones are located at the innermost k-shell of the studied semantic network. The approach of [51] and ours share the attempt to analyze the extracted networks by tracing the semantics they express. However, at the methodological level, the two approaches are very different. In fact, the approach of [51] does not consider any form of content within the collected tweets; consequently, the network extraction is carried out starting only from structural relationships. On the other hand, content is one of the two main “ingredients” for determining the semantics of the CS-Nets in our approach.

The community detection context has been influenced in the past by methods using pattern mining. For example, in [46], the authors propose an approach adopting frequent pattern mining on operations performed by users, such as posting and suggesting content. This approach employs the database of user actions as input for pattern mining algorithms. In this case, a pattern represents a sequence of users performing similar operations. Extracted patterns are then employed to identify homogeneous groups of users performing similar operations on the social network. Both the approach of [46] and ours build networks based on interactions and patterns. However, our approach also considers content within the network. It also extracts patterns based not only on their frequency but also on their utility.

In [59], the authors introduce the concept of cosine pattern mining and use it to detect communities from large scale social networks. This approach mines sets of nodes based on an extended cosine similarity. Both our approach and the one of [59] first select patterns whose frequency is higher than a certain threshold. Then, the approach of [59] discards some of these patterns based on the value of cosine pattern similarity, while our approach performs the same task based on the content exchanged by users and their interactions.

Frequent pattern mining is also applied in [2] to perform community detection. In this case, the authors model a dataset of entities as a social network. Then, they apply frequent pattern mining algorithms to generate features representing information between entities. In this context, patterns are adopted to model the set of user tasks. The main similarity between the approach of [2] and ours concerns the adoption of pattern mining. However, the two approaches have important differences. In particular, our approach is strongly based on content and the semantics it expresses.

### 3 Comment filtering and text pattern extraction

In this section, we present our approach to filter the starting set of comments and construct a set of text patterns from them. These represent the core for the construction of the CS-Nets to be used in the various applications of interest and which we illustrate in Section 7.

Our approach receives a set of comments. These should hopefully be homogeneous (e.g., comments related to the same post, comments written by the same user, comments present in a certain subreddit, comments related to a very specific topic or written at a very particular time of the year). Actually, in principle, comments should also be randomly selected, although this would make little sense in real applications.

Our approach first proceeds with a phase of Data Cleaning and Annotation. During this phase, it performs:

- The removal of bot-generated content.
- The cleaning of the textual content present in the comments and the next tokenization and lemmatization of these last ones.
- The annotation of data performed by associating a sentiment value with each comment; for this purpose, we use the compound score [35]. This last technique returns a sentiment value between -1 (most extreme negative) and +1 (most extreme positive).
- The enrichment of comments with features regarding them, their users and the posts they refer to.

Once the Data Cleaning and Annotation activities have been completed, our approach proceeds with the extraction of text patterns from the comments thus obtained. In this activity, an important role is played by pattern mining. This is a well known task in the literature, which aims at extracting text patterns with certain characteristics from a set of lemmatized texts (which, in our case, are the lemmatized comments obtained at the end of the previous phase).



Generally, the extraction of patterns is carried out based on their frequency assuming that a pattern is more important the more frequent it is [27, 4, 46, 29]. This assumption is true in most cases, but there are situations where it does not hold. In fact, there could exist patterns characterized by a low frequency but an extremely high utility (given a certain notion of it). For example, in a sales database, a pattern may have a low co-occurrence frequency but may provide a higher profit than more frequent patterns (think, for instance, of the pattern  $\langle car, car\ alarm \rangle$  against the pattern  $\langle windshield\ washer\ fluid, new\ windshield\ wipers \rangle$ ).

Several utility functions have been introduced to handle this situation. In this way, the focus shifts from frequent pattern mining to High Utility Pattern Mining (hereafter, HUPM) [26, 30, 60]. In this case, a utility function denotes an ordering of user preferences over a set of choices [32]. Consequently, it is a subjective measure and depends on the user’s preferences. Clearly, the utility of an item or a pattern can be defined from different points of view according to the preferences of the user who wants to adopt it. This is especially true in our reference scenario where users, posts and comments can be considered from multiple perspectives. To better address this issue, in this paper, we extend the standard notion of HUPM, which considers only one utility function. In this way, we pass from a one-dimensional to a multi-dimensional view of the utility concept. According to this view, several utility functions can coexist simultaneously and interact with each other. It follows that the values they assume for an item or a pattern can be properly combined to obtain an overall value to be associated with it.

Having introduced the notions of frequency and utility, we are now able to illustrate our approach for the extraction of high utility patterns. It starts from a set  $\mathcal{L}$  of possible lemmas and a set  $\mathcal{C}$  of lemmatized comments. Both each comment of  $\mathcal{C}$  and a pattern  $p_h$  can be represented as a set of lemmas, and thus as a subset of  $\mathcal{L}$ . We call  $\mathcal{C}_h$  the set of comments in which  $p_h$  is present. The frequency of  $p_h$  is given by the cardinality of  $\mathcal{C}_h$ , while the set of features of  $p_h$  consist of the set of features of the comments of  $\mathcal{C}_h$ . An utility function of  $p_h$  is a function applied to the features of  $p_h$  or on a subset of them. The choice of the features and the utility function determines the point of view that is being adopted in the pattern analysis. For example, if we focus on the compound score and the *avg* function, the utility function of  $p_h$  calculates the average value of the compound scores of the comments of  $\mathcal{C}_h$ . It can be used, for instance, to select those patterns whose presence in the comments leads to a positive sentiment (or, conversely, to a negative one). Once the features of interest and the suitable utility functions have been defined, our approach can proceed with the selection of the patterns having frequency and utility values greater than a certain threshold. In particular, if we desire to give a higher weight to utility than to frequency, we can set a low frequency threshold in order to filter out only very rare patterns and keep all the others. These are then selected based on utility.

Our approach works as follows. First, it extracts patterns having a frequency higher than a minimum threshold. For this purpose, it can use one of the classical techniques for frequent pattern mining, such as FPGrowth [34]. Then, it associates each pattern with the features appearing in the comments it is present in. These features will be used for the next analyses. Afterwards, it applies the chosen utility function to each pattern for computing the pattern’s utility value. Finally, it selects and returns those patterns whose utility value is greater than a minimum threshold.

If we choose to filter only extremely rare patterns, and therefore to give a little weight to frequency,

the utility function plays a key role in filtering patterns and allows us to direct the pattern selection towards a strategy rather than another. Two utility functions very interesting in our case are the following:

- The average sentiment value of the comments which the pattern of interest, say  $p_j$ , refers to. It can be formalized as:

$$f_s(p_j) = avg_{c_{jk} \in \mathcal{C}_j} \{\gamma(c_{jk})\}$$

Here: (i)  $f_s(\cdot)$  is the utility function we are defining; (ii)  $p_j$  is the generic pattern, of which we want to compute the utility function; (iii)  $\mathcal{C}_j$  is the set of comments in which  $p_j$  is present; (iv)  $\gamma(\cdot)$  is a function that receives a comment and returns its compound score (and, therefore, its sentiment value); (v)  $avg(\cdot)$  is a function computing the average of the values received as input.

- The Pearson’s correlation [49] between the sentiment and the score of the comments where a certain pattern  $p_j$  is present. Here, we feel it appropriate to point out that the score of a comment is a very different concept from the compound score mentioned above. In fact, by score of a comment we mean the evaluation that, in most social platforms, users can give to each comment posted by another user. For example, considering Reddit, the score of a comment is given by the difference between the number of upvotes and the number of downvotes it received. Having clarified this aspect, we can proceed with the application of the Pearson’s correlation to our case. Remember that it is a measure of the linear correlation between two sets of data. Its value belongs to the real interval  $[-1, 1]$ , where -1 (resp., 1) denotes a negative (resp., positive) linear correlation, while 0 indicates a lack of correlation. It can be formalized as follows:

$$f_p(p_j) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Here: (i)  $p_j$  and  $\mathcal{C}_j$  have been already explained for  $f_s(\cdot)$ ; (ii)  $X$  (resp.,  $Y$ ) is the set of sentiment values (resp., score) related to the comments of  $\mathcal{C}_j$ ; (iii)  $x_i$  (resp.,  $y_i$ ) indicates the  $i^{th}$  element of  $X$  (resp.,  $Y$ );  $\bar{x}$  (resp.,  $\bar{y}$ ) represents the mean of the values of  $X$  (resp.,  $Y$ ). Note that a positive (resp., negative) value of  $f_p(\cdot)$  indicates that there is a direct (resp., inverse) correlation between the sentiment elicited by a comment and the score it gets. During the experimental campaign, which we describe in Section 6, we observed that there exist many patterns and comments with negative values of  $f_p(\cdot)$ . This allows us to say that a positive (resp., negative) sentiment in a comment does not necessarily lead it to receive a high (resp., low) score. This is especially true for certain kinds of comment, e.g., those related to Not Safe For Work (resp., NSFW) posts, which are the ones investigated in the experiments of this paper. Regarding this utility function, some observations are in order. In fact, correlation coefficients are one of the most common topics in statistics and are also widely used in many data analytics applications. Various correlation coefficients have been proposed in the past literature. For example, in addition to the Pearson’s correlation mentioned above, other very common forms of correlation are the Spearman’s rank correlation [54], the Kendall’s rank correlation [54], the association strength

between non-linear random variables [55], the logistic regression [43], etc. In our approach, we decided to use the Pearson’s correlation for several reasons. First, it can be easily interpreted and explained. Second, it does not require a high computational cost, and this property is crucial in a context like ours where the number of elements involved in the computation of the correlation between two features is huge. Third, the variables whose correlation we want to compute (e.g., sentiment values and scores) are both quantitative. In presence of this kind of variable, it is possible to apply the Pearson’s coefficient and it is not necessary to adopt more sophisticated correlation coefficients capable of handling non-quantitative variables. Finally, the relationship between the variables of interest are linear and, again, this is a case handled very well by the Pearson’s correlation. In presence of non-linear relationships we would have had to use other more sophisticated forms of correlation. These forms (for example, the distance correlation) would have been able to handle non-linear relationships, but at the price of a computational cost higher than that required by the Pearson’s correlation.

We end this section by pointing out that many other utility functions could be defined. Here, we have focused on  $f_s(\cdot)$  and  $f_p(\cdot)$  to give an idea of their potential and possible variety. These two utility functions, like others we might define in the future, have pros and cons. In particular,  $f_s(\cdot)$  models a simple and intuitive relationship between posts and the sentiments of the corresponding comments. Simplicity is both the strength and the weakness of this utility function. In fact, it guarantees to  $f_s(\cdot)$  a very low computational cost. However, it makes  $f_s(\cdot)$  not able to catch possible more sophisticated information (for example, the presence of trends or spikes in the sentiment of the comments related to a post). The second utility function, that is  $f_p(\cdot)$ , allows the identification of those patterns whose presence in the comments with high (resp., low) score is accompanied by a positive (resp., negative) sentiment. The identification of this correlation between score and sentiment is a valuable and not obvious task, since there could be comments with high scores and a null or negative sentiment or comments with a low score and a null or positive sentiment. A potential drawback of  $f_p(\cdot)$  is that, in presence of rare patterns, a high correlation between score and sentiment returned by  $f_p(\cdot)$  may not be statistically significant.

## 4 Content Semantics Network definition

Let  $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$  be a set of lemmatized comments and let  $\mathcal{L} = \{l_1, l_2, \dots, l_q\}$  be the set of all lemmas that can be found in a comment of  $\mathcal{C}$ . Each comment  $c_k \in \mathcal{C}$  can be represented as a set of lemmas  $c_k = \{l_1, l_2, \dots, l_m\}$ . As a consequence, we have that  $c_k \subseteq \mathcal{L}$ .

A text pattern  $p_h$  is a set of lemmas; more specifically,  $p_h \subseteq \mathcal{L}$ . In principle,  $p_h$  can occur in zero, one or more comments of  $\mathcal{C}$ . Actually, as pointed out above, we are interested in those patterns whose frequency and utility function are higher than two suitable thresholds. In the following, we call  $\mathcal{P}$  this set of patterns.

Actually, as pointed out above, we are interested in those patterns whose values of frequency and utility function belong to a suitable interval. In particular, as for frequency, it is presumable that we are interested in patterns with a frequency value higher than a certain threshold. Instead, as far as the utility function is concerned, the situation is more articulated, and everything depends on the

utility function used and the context in which we are adopting our approach. Specifically:

- If we are using  $f_s(\cdot)$ , we can select patterns characterized by a compound score (and, therefore, a sentiment value) very high (positive patterns), very low (negative patterns) or belonging to a certain interval (for example, neutral patterns, non negative ones or non positive ones).
- If we are using  $f_p(\cdot)$ , we can select: (i) patterns with a high sentiment value that stimulate positive comments; (ii) patterns with a low sentiment value that stimulate negative comments; (iii) patterns with a high sentiment value that stimulate negative comments; (iv) patterns with a low sentiment value that stimulate positive comments. Clearly, in the great majority of applications, the patterns of interest are those pertaining to option (i) or, at most, (ii). However, there may be niche applications, where patterns belonging to options (iii) and (iv) are also of interest.

A Content Semantics Network (hereafter, CS-Net)  $\mathcal{N}$  is defined as:

$$\mathcal{N} = \langle N, A^c \cup A^r \rangle$$

$N$  is the set of nodes of  $\mathcal{N}$ . There is a node  $n_i \in N$  for each lemma  $l_i \in \mathcal{L}$ . Since there exists a biunivocal correspondence between  $n_i$  and  $l_i$ , in the following we will use these two symbols interchangeably.

$A^c$  is the set of co-occurrence arcs. An arc  $(n_i, n_j, w_{ij}) \in A^c$  indicates that the lemmas  $l_i$  and  $l_j$  appear at least once together in a pattern of  $\mathcal{P}$ .  $w_{ij}$  is a real number in the interval  $[0, 1]$  denoting the strength of the co-occurrence. The higher  $w_{ij}$ , the higher this strength. For instance,  $w_{ij}$  can be computed considering the number of patterns in which  $l_i$  and  $l_j$  co-occur.

$A^r$  is the set of semantic relationship arcs. An arc  $(n_i, n_j, w_{ij}) \in A^r$  denotes that there exists a form of semantic relationship between  $l_i$  and  $l_j$ .  $w_{ij}$  is a real number in the interval  $[0, 1]$  denoting the strength of the relationship. The higher  $w_{ij}$ , the higher this strength. For instance,  $w_{ij}$  can be computed using ConceptNet [38] and taking into account the number of times in which  $l_j$  is present in the set of “related terms” of  $l_i$ , along with the values of the corresponding weights.

A comment about the structure of the CS-Net is in order. As specified in the Introduction, in this paper we want to make an effort to define the semantics of a set of contents, for example those published in comments to Reddit posts. The CS-Net is intended as a tool to support this activity. For this purpose, it considers two perspectives derived from the past literature.

The first is related to the concept of co-occurrence and specifies that two semantically related lemmas which tend to appear together very often in sentences. This perspective is probably the most natural one in the field of text mining, where it is well known that the frequency with which two or more lemmas appears together in a text is an index of the correlation existing between them. Its potential weakness is the need to compute the frequency of each pair of lemmas. In addition, this computation must be continuously updated because the addition of a new text to be examined (e.g., a new comment) may lead to a change in all frequencies.

The second concerns the concept of relationships and semantically related terms. These summarize the results of several researches carried out in the past both in Information Retrieval [19] and Natural Language Processing [12]. This perspective takes the meaning of terms, i.e., their semantics, into

account. In fact, semantic relationships between terms (such as, for example, synonymies) are a very common feature in natural languages. Its main problem concerns the need to have available a thesaurus, where all semantic relationships are stored. If such a thesaurus exists, the computation of the strength of the semantic relationships is immediate.

Clearly, additional perspectives could be considered and we also do not exclude doing so in the future. For example, a very interesting perspective would be one in which a human expert is involved in determining the correlation between two terms. It may seem obsolete because it is very time consuming and does not take advantage of the computation power of modern processors or the presence of useful tools, such as thesauruses. However, it can become very important in specialized contexts (e.g., bioinformatics and proteomics), where the existence of relationships between certain terms can only be determined by a human expert.

From this point of view, we highlight that our model is highly scalable. In fact, if we wanted to consider a further perspective, it will be sufficient to flank  $A^c$  and  $A^r$  with an additional set of arcs that represents this new perspective.

## 5 Evaluation of the semantic similarity of two CS-Nets

In this section, we illustrate our approach for computing the semantic similarity of the contents expressed by two CS-Nets  $\mathcal{N}_1$  and  $\mathcal{N}_2$ . In the previous section, we have said that the CS-Net model currently adopts two perspectives for the semantic similarity evaluation, namely co-occurrences and semantic relationship between lemmas (see Section 4). We have also said that this model is scalable allowing the adoption of new perspectives, if desired. We aim to preserve such scalability also in the approach to evaluate the semantic similarity of two CS-Nets we are presenting here.

Given this premise, we are now ready to describe our approach. It receives two CS-Nets  $\mathcal{N}_1$  and  $\mathcal{N}_2$  and returns a coefficient  $\sigma_{12}$  that measures the semantic similarity of the contents represented by  $\mathcal{N}_1$  and  $\mathcal{N}_2$ . For this purpose:

- It constructs two pairs of subnetworks  $(\mathcal{N}_1^c, \mathcal{N}_2^c)$  and  $(\mathcal{N}_1^r, \mathcal{N}_2^r)$ , obtained by selecting only the co-occurrence and semantic relationship arcs from the networks  $\mathcal{N}_1$  and  $\mathcal{N}_2$ , respectively. Specifically:

$$\mathcal{N}_1^c = \langle \mathcal{N}_1, A_1^c \rangle \quad \mathcal{N}_2^c = \langle \mathcal{N}_2, A_2^c \rangle \quad \mathcal{N}_1^r = \langle \mathcal{N}_1, A_1^r \rangle \quad \mathcal{N}_2^r = \langle \mathcal{N}_2, A_2^r \rangle$$

If, in the future, the number of perspectives, and therefore the number of arc sets, increases, it will be sufficient to build a pair of subnetworks for each perspective.

- It determines the weights to be associated with the two subnetworks. These weights are computed as:

$$\omega_1^c = \frac{|A_1^c|}{|A_1^c| + |A_1^r|} \quad \omega_2^c = \frac{|A_2^c|}{|A_2^c| + |A_2^r|} \quad \omega_1^r = 1 - \omega_1^c \quad \omega_2^r = 1 - \omega_2^c$$

$$\omega_{12}^c = \frac{\omega_1^c + \omega_2^c}{2} \quad \omega_{12}^r = \frac{\omega_1^r + \omega_2^r}{2}$$

The reasoning underlying these formulas is that, in determining the overall semantics of a content, the importance of a perspective with respect to the other ones is directly proportional to the number of pairs of lemmas it is able to involve.

- It computes the semantic similarity degree  $\sigma_{12}^c$  and  $\sigma_{12}^r$  for the pairs of networks  $(\mathcal{N}_1^c, \mathcal{N}_2^c)$  and  $(\mathcal{N}_1^r, \mathcal{N}_2^r)$ , respectively. We describe this computation in detail in Subsection 5.1.
- It computes the overall semantic similarity degree  $\sigma_{12}$  associated with the networks  $\mathcal{N}_1$  and  $\mathcal{N}_2$  as a weighted mean of the two semantic similarity degrees  $\sigma_{12}^c$  and  $\sigma_{12}^r$ :

$$\sigma_{12} = \frac{\omega_{12}^c \cdot \sigma_{12}^c + \omega_{12}^r \cdot \sigma_{12}^r}{\omega_{12}^c + \omega_{12}^r}$$

If we set:

$$\alpha = \frac{\omega_{12}^c}{\omega_{12}^c + \omega_{12}^r} = \frac{\omega_1^c + \omega_2^c}{2} = \frac{1}{2} \cdot \left( \frac{|A_1^c|}{|A_1^c| + |A_1^r|} + \frac{|A_2^c|}{|A_2^c| + |A_2^r|} \right)$$

then, the formula for the computation of  $\sigma_{12}$  can be written as:

$$\sigma_{12} = \alpha \cdot \sigma_{12}^c + (1 - \alpha) \cdot \sigma_{12}^r$$

In this formula,  $\alpha$  is a coefficient that weights the semantic similarity defined through co-occurrences against the one defined through semantic relationships between lemmas. The rationale behind the formula of  $\alpha$  is that the greater the amount of information carried by one perspective, compared to another, the greater its weight in defining the overall semantics. Now, since  $|N_1^c| = |N_1^r|$  and  $|N_2^c| = |N_2^r|$ , the amount of information carried by co-occurrences with respect to semantic relationships between lemmas can be computed by considering the cardinality of the corresponding sets of arcs. Finally, note that  $\sigma_{12}$  ranges in the real interval  $[0, 1]$ . The higher  $\sigma_{12}$ , the greater the similarity of  $\mathcal{N}_1$  and  $\mathcal{N}_2$ .

Our approach for the computation of  $\sigma_{12}$  is extensible, because if in the future we want to enrich the CS-Net model with additional perspectives to model content semantics, it will be sufficient to flank to  $\sigma_{12}^c$  and  $\sigma_{12}^r$  an additional similarity coefficient for each perspective and modify the formula for the computation of  $\sigma_{12}$  accordingly.

## 5.1 Semantic similarity degree computation

In the previous section, we have seen that our approach for computing the similarity between two CS-Nets  $\mathcal{N}_1$  and  $\mathcal{N}_2$  constructs “projections” or “subnetworks” for each network (i.e.,  $\mathcal{N}_1^c$  and  $\mathcal{N}_1^r$  for  $\mathcal{N}_1$ , and  $\mathcal{N}_2^c$  and  $\mathcal{N}_2^r$  for  $\mathcal{N}_2$ ), computes the similarity coefficients  $\sigma_{12}^c$  between  $\mathcal{N}_1^c$  and  $\mathcal{N}_2^c$ , and  $\sigma_{12}^r$  between  $\mathcal{N}_1^r$  and  $\mathcal{N}_2^r$  separately, and then combines them appropriately. In this context, the way in which the coefficient  $\sigma_{12}^x$ ,  $x \in \{c, r\}$ , is computed becomes extremely important.

In order to define an approach for the computation of  $\sigma_{12}^x$  as holistic as possible, we strove to define a formula that takes into account more factors that may influence the semantic similarity degree of

two networks  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$ ,  $x \in \{c, r\}$ . In particular, there are at least two factors that we think can contribute to define this semantic similarity degree.

The first factor concerns the topological similarity of the networks, and thus the similarity of their structural features (e.g., number of nodes and arcs, density, clustering coefficient, etc.). In fact, the structure of a network is determined by the arcs existing between the corresponding nodes. In our case, nodes represent lemmas involved in comments and arcs represent features (i.e., co-occurrences or semantic relationships) playing a key role to define the semantics of the lemmas they link. This reasoning is also reinforced by the fact that the definition of the semantics of a lemma is certainly improved by looking at the lemmas to which it is related in the network (in this claim, the extension, to the CS-Net model, of the homophily principle [42] characterizing social networks, comes into play).

The second factor is much more straightforward and concerns the semantic meaning of the concepts expressed by the network nodes, because each of them represents a lemma of the corresponding comments.

As for the first factor, in the literature there are many approaches designed for computing the similarity degree of the structural features of two networks (see [31, 10, 24], just to cite a few of them). We decided to adopt one of them and our choice fell on NetSimile [10]. In fact, this approach has a much shorter computation time than most of the other ones performing the same task proposed in the past literature. Furthermore, the accuracy level it guarantees is adequate for our application context. NetSimile extracts and evaluates the structural characteristics of each node based on the structural characteristics (such as the average clustering coefficient, the average number of nodes and arcs, etc.) of its ego network. As a consequence, in order to obtain the similarity score of two networks, NetSimile computes the similarity degree of their vectors of features.

As far as the second factor is concerned, we decided to consider the portion of nodes with the same meaning, or rather with similar meanings, present in the two subnetworks. A simple, but very effective, way to evaluate this portion could consist of the computation of the Jaccard coefficient between the sets of lemmas associated with the nodes of the two networks. Actually, to increase the result accuracy, it is necessary to take lexicographic relationships (e.g., synonymies and homonymies) [48, 17] between lemmas into account. As we mentioned above, these can be identified from an advanced dictionary, such as ConceptNet [38], which includes WordNet [44], a thesaurus widely used in the past literature for this purpose. In the following, we will adopt the symbol  $J^*$  to denote the Jaccard coefficient enhanced in such a way as to take lexicographic relationships into account.

We are now able to define the formula for computing  $\sigma_{12}^x$ . Specifically, we have:

$$\sigma_{12}^x = \beta^x \cdot \nu(\mathcal{N}_1^x, \mathcal{N}_2^x) + (1 - \beta^x) \cdot J^*(N_1^x, N_2^x)$$

Here:

- $\nu(\mathcal{N}_1^x, \mathcal{N}_2^x)$  is a function computing the topological similarity of  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$  by applying the NetSimile approach.
- $\beta^x$  is a coefficient defining the weight of the topological similarity of the networks with respect to the semantic similarity of the lemmas associated with the corresponding nodes. In order to define a formula for  $\beta^x$ , we made the following reasoning. Intuitively, one can assume that

the denser the networks, the more the information about their topology (and, thus,  $\nu$ ) becomes relevant. In other words, while the information contained in the nodes (expressed by  $J^*$ ) does not vary against the density of the networks, the information contained in the arcs varies. In fact, a larger number of arcs implies an increase of the amount of information available, as well as of the strength of the relationships between the lemmas in the network. This is due to the fact that: (i) arcs represent semantic relationships existing between lemmas; (ii) for the homophily principle, a higher number of arcs implies, for each node, a higher number of neighbors that can contribute to better define the semantics of the lemma associated with it.

The above reasoning is at the basis of our formula for computing  $\beta^x$ . In order to define it, we need to introduce the concept of mean density of a set of CS-Nets. In fact, as will be clear in the following, the formula of  $\beta^x$  depends on whether the density of  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$  is greater or less than the mean density  $\bar{d}^x$  of the CS-Nets generally present in the reference context. In fact, we do not have a predefined set of CS-Nets on which we can operate, but these are derived from the subset  $\bar{\mathcal{C}} \subseteq \mathcal{C}$  of the comments returned at the end of the comment filtering and text pattern extraction activities. Therefore, in order to compute the mean density  $\bar{d}^x$ , we built a set  $\bar{\mathcal{CN}} = \langle \bar{\mathcal{N}}_1, \bar{\mathcal{N}}_2, \dots, \bar{\mathcal{N}}_t \rangle$  of CS-Nets by deriving it randomly from the comments of  $\bar{\mathcal{C}}$ . The process of constructing  $\bar{\mathcal{CN}}$  was as follows. First, we randomly constructed a set  $\bar{\mathcal{CS}} = \langle \bar{\mathcal{C}}_1, \bar{\mathcal{C}}_2, \dots, \bar{\mathcal{C}}_t \rangle$  of comment sets such that  $\bar{\mathcal{C}}_h \subseteq \bar{\mathcal{C}}, 1 \leq h \leq t$ . The randomness in the construction of  $\bar{\mathcal{C}}_h$  involves both its cardinality and the lemmas comprising it. A CS-Net  $\bar{\mathcal{N}}_h = \langle \bar{N}_h, \bar{A}_h = \bar{A}_h^c \cup \bar{A}_h^r \rangle$  can be constructed for each subset  $\bar{\mathcal{C}}_h, 1 \leq h \leq t$ , by applying the approach described in Section 4. Let  $\bar{\mathcal{N}}_h^x = \langle \bar{N}_h^x, \bar{A}_h^x \rangle, x \in \{c, r\}$ , be the subnetworks of  $\bar{\mathcal{N}}_h$  obtained by selecting only the arcs of type  $x$ . Let  $\bar{\mathcal{CN}}^x = \langle \bar{\mathcal{N}}_1^x, \bar{\mathcal{N}}_2^x, \dots, \bar{\mathcal{N}}_t^x \rangle$  be the set of subnetworks of type  $x$ .

The density  $\bar{d}_h^x$  of  $\bar{\mathcal{N}}_h^x$  is defined as:

$$\bar{d}_h^x = \frac{|\bar{A}_h^x|}{\frac{|\bar{N}_h^x| \cdot (|\bar{N}_h^x| - 1)}{2}}$$

The mean density of  $\bar{\mathcal{CN}}^x$  is defined as:

$$\bar{d}^x = \frac{\sum_{h=1}^t \bar{d}_h^x}{t}$$

Consider now the subnetworks  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$  of our interest. We define their average density  $d_{12}^x$  as:

$$d_{12}^x = \frac{d_1^x + d_2^x}{2}$$

where the formula to compute  $d_1^x$  and  $d_2^x$  is the same as the one presented above for  $\bar{d}_h^x$ .

At this point, we are able to define  $\beta^x$ . In particular, we have that:

$$\beta^x = \begin{cases} \min \left( 0.5 + \frac{d_{12}^x - \bar{d}^x}{\bar{d}^x}, \beta_{max}^x \right) & \text{if } d_{12}^x \geq \bar{d}^x \\ \max \left( \beta_{min}^x, 0.5 - \frac{\bar{d}^x - d_{12}^x}{\bar{d}^x} \right) & \text{if } d_{12}^x < \bar{d}^x \end{cases}$$



This definition of  $\beta^x$  takes into account the reasoning expressed above regarding the correlation between the density of  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$  and the importance of their topological components in the computation of  $\sigma_{12}^x$ . However, at the same time, it imposes that  $\beta^x$  can oscillate in a range between  $\beta_{min}^x$  and  $\beta_{max}^x$  (which we set at 0.25 and 0.75, respectively). This constraint allows the contribution of  $\nu$  (resp.,  $J^*$ ) not to become irrelevant, in case the density is very low (resp., high).

Note that  $\sigma_{12}^x$  ranges in the real interval  $[0, 1]$ . The higher  $\sigma_{12}^x$ , the greater the similarity of  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$ .

We will return to the choice of the values of  $\beta^x$  in Section 6.3, where we illustrate an experiment that we conducted about this issue.

We point out that our approach for computing  $\sigma_{12}^x$  is capable of operating on any projection  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$  of the networks  $\mathcal{N}_1$  and  $\mathcal{N}_2$ . The only constraint it imposes is that all arcs must be of the same type  $x$ . This helps making our overall approach scalable in that, if in the future we want to add an additional perspective of modeling content semantics, then the similarity degree of the corresponding projections of  $\mathcal{N}_1$  and  $\mathcal{N}_2$  can be still computed using it.

Note that both the formula for  $\sigma_{12}$  and the one of  $\sigma_{12}^*$  are based on a weighted mean, where the weights are  $\alpha$  and  $\beta$ , respectively. Actually, we could have considered other aggregation operators in these formulas. In fact, this kind of operator has been highly investigated in the literature. Here, researchers have studied what mathematical properties an aggregation operator should enjoy, and have mentioned, for example, boundary conditions, monotonicity, continuity, associativity, symmetry, bisymmetry, absorbent element, neutral element, idempotence, compensation, counterbalancement, reinforcement, stability for a linear function and invariance. They also have identified some behavioral properties that these operators should have, such as decisional behavior, interpretability of parameters, and weights on arguments. Based on these studies they proposed several aggregation operators, each enjoying all or some of the above properties. Examples of such operators are the arithmetic mean, the weighted mean, the median, the minimum and the maximum, the weighted minimum and the weighted maximum, the geometric mean, the harmonic mean, the symmetric sum, the Ordered Weighted Averaging (OWA) operators, the Choquet & Sugeno discrete fuzzy integrals. We do not want to go in detail on this issue because it is beyond the scope of this paper. The interested reader can find a complete discussion in [18].

For our case, the choice fell on the weighted mean because, with only two values to aggregate, the maximum, minimum and median made little sense. The arithmetic mean would have been a simpler operator but would not have taken into account an important part of the information available. The geometric mean and the harmonic mean, like the arithmetic mean, would not have taken into account the weights and would have produced a result that was much less intuitive to understand. All the other operators mentioned above were designed for very complex situations, in which the values to be aggregated are many and with heterogeneous characteristics. Therefore, their application in our case would have been unnecessarily expensive and would have led to unintuitive results.

## 6 Experiments

### 6.1 Dataset

As we mentioned in the Introduction, the set of comments on which we apply our approach should be homogeneous, e.g., related to a specific topic or a specific time of the year, or both. Following this guideline, we decided to focus on comments related to Not Safe For Work (hereafter, NSFW) posts in our experiments. This choice is also motivated by the fact that this topic has its intrinsic interest, regardless of our approach. Therefore, it has a double benefit, i.e., it allows us to test our approach and shed some light on a relevant phenomenon in Reddit, which is still little studied. Reddit is one of the few social networks to handle NSFW content in a straightforward and well-structured way. Despite this, only a few researchers have analyzed the phenomenon of NSFW content in this social platform [41, 47, 16].

In order to build our dataset of comments on NSFW posts, we used the website `pushshift.io` [8], which represents one of the main data repositories for Reddit. Specifically, we considered 449 NSFW adult subreddits listed at the address `https://www.reddit.com/r/ListOfSubreddits/wiki/nsfw` and downloaded comments to all posts published from January 1<sup>st</sup>, 2020 to March 31<sup>st</sup>, 2020. The number of posts considered is 3,064,758, while the total number of comments is 11,627,372.

We performed an ETL (Extraction, Transformation, and Loading) activity on this data. During it, we observed that some of the posts downloaded from `pushshift.io` were published by authors who had left Reddit. We decided to remove these posts and the associated comments from our dataset. Moreover, we removed all the comments related to posts whose field `over_18` was set to `false`. After this ETL activity, the total number of NSFW posts in our dataset is 2,981,601, corresponding to 97% of the initial ones. The total number of NSFW comments present in our dataset is 8,383,499, corresponding to 72.20% of the initial ones.

In Table 1, we report some information about the authors of posts and comments. We can see that the number of authors who wrote comments is much larger than the number of authors who published posts. In addition, we can observe that half of the authors who published posts also published comments.

<i>Parameter</i>	<i>January 2020</i>	<i>February 2020</i>	<i>March 2020</i>	<i>Total</i>
Authors publishing posts	91,894	92,530	110,873	218,433
Authors publishing comments	369,014	351,967	392,871	738,216
Authors publishing both posts and comments	46,427	44,733	53,063	115,686

Table 1: Some parameters regarding authors in the dataset

Figure 1 illustrates the distribution of comments against posts. As we can see, it follows a power law. The values of the corresponding parameters  $\alpha$  and  $\delta$  are 3.0821 and 0.0159, respectively. Figure 2 reports the distribution of scores against comments. As can be seen from this figure, it follows a power law. The values of the corresponding parameters  $\alpha$  and  $\delta$  are 3.8485 and 0.0255, for the left part of the curve, and 2.1456 and 0.0158, for the right part of it. In this figure, the values of  $\alpha$  and  $\delta$  for the left part of the distribution were computed considering the absolute values of scores.

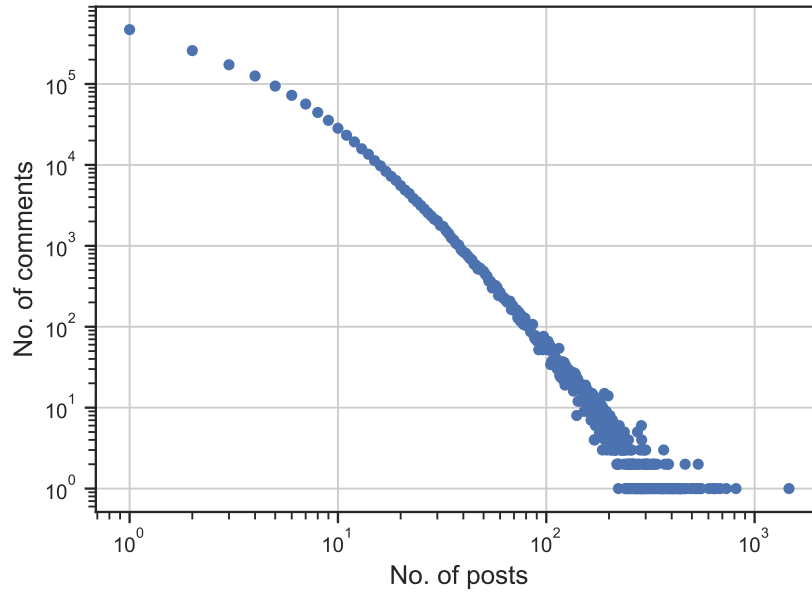


Figure 1: Distributions of comments against posts

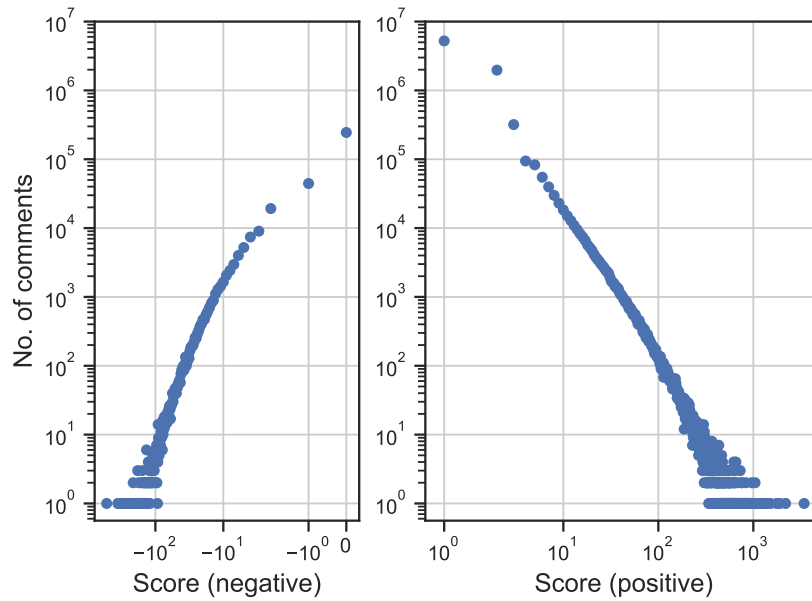


Figure 2: Distributions of scores against comments

## 6.2 Analysis of generated Content Semantic Network

We have seen that our approach extracts text patterns from which it constructs the CS-Nets to analyze. The text pattern detection approaches proposed in the past literature aim at selecting the most frequent patterns. In addition to the frequency of patterns, our approach takes into account their

utility, expressed by a utility function, and selects the patterns with the highest values of this function. However, the main focus of our approach is content semantics. So, it is extremely important to verify whether, besides extracting the most frequent and useful comments, it is able to build CS-Nets having a homogeneous and meaningful semantics.

Remember that, in our approach, semantic links between lemmas are expressed by means of arcs connecting the corresponding nodes. Therefore, we can say that the greater the number of arcs we observe in the generated CS-Nets, the greater the number of semantic links between the corresponding lemmas. Moreover, the greater the number of such links, the greater the semantic significance of the CS-Net and, ultimately, the better the quality of our approach.

To test whether our approach is capable of constructing semantically meaningful CS-Nets from a set of comments, we planned to compare it with an approach that builds the networks randomly and can serve as a null model in a significance test. To this end, we considered four sets of comments  $\mathcal{C}_1, \dots, \mathcal{C}_4$ . They were selected uniformly at random across random posts from our dataset. For each set, we initially applied our approach and constructed the CS-Nets  $\mathcal{N}_1 = \langle N_1, A_1 = A_1^c \cup A_1^r \rangle, \dots, \mathcal{N}_4 = \langle N_4, A_4 = A_4^c \cup A_4^r \rangle$ . Next, we applied the random approach with the goal of constructing the CS-Nets  $\overline{\mathcal{N}}_1 = \langle \overline{N}_1, \overline{A}_1 = \overline{A}_1^c \cup \overline{A}_1^r \rangle, \dots, \overline{\mathcal{N}}_4 = \langle \overline{N}_4, \overline{A}_4 = \overline{A}_4^c \cup \overline{A}_4^r \rangle$ .

In particular, given the set  $\mathcal{C}_k$  of comments, to construct the corresponding CS-Net  $\overline{\mathcal{N}}_k$ , we selected uniformly at random a number of lemmas from  $\mathcal{C}_k$  equal to the cardinality of  $N_k$ , such that  $|\overline{N}_k| = |N_k|$ . In this way,  $\mathcal{N}_k$  and  $\overline{\mathcal{N}}_k$  had the same number of nodes. Then, we constructed  $\overline{A}_k$  as follows: given two nodes  $n_i \in \overline{N}_k$  and  $n_j \in \overline{N}_k$ , we inserted an arc  $a_{ij}^c \in \overline{A}_k^c$  if the lemmas  $l_i$  and  $l_j$ , corresponding to  $n_i$  and  $n_j$ , were simultaneously present in at least one comment of  $\mathcal{C}_k$ . In addition, we inserted an arc  $a_{ij}^r \in \overline{A}_k^r$  if there is a semantic relationship between  $l_i$  and  $l_j$  in ConceptNet.

For each set  $\mathcal{C}_k$  of comments, we performed the random approach described above 30 times. Finally, we computed the number of arcs of  $A_k$  obtained through our approach (applying the two different utility functions  $f_s(\cdot)$  and  $f_p(\cdot)$ ) and the mean of the number of the arcs of  $\overline{A}_k$  obtained by averaging the number of arcs of the 30 CS-Nets  $\overline{\mathcal{N}}_k$  built by applying the random approach. These numbers are shown in Table 2.

Sets of comments	Number of nodes of $\mathcal{N}_k$ and $\overline{\mathcal{N}}_k$	Number of arcs of $\mathcal{N}_k$ Utility function: $f_s(\cdot)$	Number of arcs of $\mathcal{N}_k$ Utility function: $f_p(\cdot)$	Number of arcs of $\overline{\mathcal{N}}_k$
$\mathcal{C}_1$	98	2,351.14	2,116.26	1,587.21
$\mathcal{C}_2$	111	3,191.85	2,872.66	1,834.77
$\mathcal{C}_3$	103	2,400.97	2,160.87	1,798.34
$\mathcal{C}_4$	105	2,527.42	2,274.68	1,311.77

Table 2: Average number of arcs of the CS-Nets generated by applying our approach, with two different utility functions, and the random one

From the analysis of this table, we can observe that, in all cases, our approach returns CS-Nets with a higher number of arcs than the random one.

To assess the significance of this result, we performed the t-test between the outputs of our approach (with the two different utility functions) and those obtained from the null model. More specifically, the objective of the t-test was to check the significance of the difference between the means of the two sets (i.e., the real and the random ones). At the end of this task, we computed the corresponding p-values. They are reported in Table 3.

<i>Sets of comments</i>	$f_s(\cdot)$	$f_p(\cdot)$
$\mathcal{C}_1$	$8.90 \cdot 10^{-25}$	$8.59 \cdot 10^{-20}$
$\mathcal{C}_2$	$4.51 \cdot 10^{-21}$	$7.81 \cdot 10^{-18}$
$\mathcal{C}_3$	$2.40 \cdot 10^{-14}$	$8.59 \cdot 10^{-20}$
$\mathcal{C}_4$	$3.07 \cdot 10^{-15}$	$5.42 \cdot 10^{-19}$

Table 3: p-values obtained by performing the t-test between the outputs of our approach and those returned by the null model

From the analysis of this table, we can observe that, with both utility functions, the p-values are very low, much lower than 0.05. This result leads us to conclude that our approach actually returns CS-Nets with a larger number of arcs, and therefore semantically more homogeneous and meaningful.

Based on what we said at the beginning of this section, this result is very encouraging because it says that our approach not only selects very frequent and useful patterns but also builds high-quality CS-Nets from the content semantics point of view.

We described above our experiment for four sets of comments  $\mathcal{C}_1, \dots, \mathcal{C}_4$ . After obtaining the results described in Table 3, we repeated it with 50 other sets of comments and obtained similar results. Due to space constraints, we cannot report here their details.

### 6.3 Investigating $\beta^x$

In Section 5.1, we have seen that the semantic similarity degree  $\sigma_{12}^x$  between two subnetworks  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$ , obtained from  $\mathcal{N}_1$  and  $\mathcal{N}_2$  considering only arcs of type  $x$ , with  $x \in \{c, r\}$ , depends on a coefficient  $\beta^x$ . This defines the weight of the topological similarity of the networks with respect to the semantic similarity of the lemmas associated with the corresponding nodes. In the same section, we have also defined a formula for  $\beta^x$  and we have seen that it is essentially related to the density of  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$ .

In this experiment, we aim at performing some analyses on the trend of the value of  $\beta^x$  against the number of nodes of  $\mathcal{N}_1^x$  and  $\mathcal{N}_2^x$ . For this purpose, we performed the following tasks:

- We considered 50 sets of comments of different sizes. *Each set was selected uniformly at random across random posts from our dataset.*
- We performed the activities described in Sections 3, 4 and 5 on each set, and obtained 50 CS-Nets of different sizes.
- We considered all possible pairs  $(\mathcal{N}_1, \mathcal{N}_2)$  of CS-Nets that could be constructed from the initial 50 networks.
- For each pair  $(\mathcal{N}_1, \mathcal{N}_2)$  of CS-Nets, we generated two pairs of subnetworks  $(\mathcal{N}_1^c, \mathcal{N}_2^c)$  and  $(\mathcal{N}_1^r, \mathcal{N}_2^r)$ .
- For each pair  $(\mathcal{N}_1^x, \mathcal{N}_2^x)$  of subnetworks,  $x \in \{c, r\}$ , we computed both  $|\mathcal{N}_1^x| + |\mathcal{N}_2^x|$  and  $\beta^x$ . In the following, we call  $\rho^x$  the parameter  $|\mathcal{N}_1^x| + |\mathcal{N}_2^x|$ .
- We constructed 30 bins of values of  $\rho^x$ ; specifically, the first bin groups all values of  $\rho^x$  between 1 and 10, the second bin includes all values of  $\rho^x$  between 11 and 20, and so on. The last bin comprises all values of  $\rho^x$  between 291 and 300.

- We assigned each pair  $(\mathcal{N}_1^x, \mathcal{N}_2^x)$  of subnetworks to the suitable bin, based on the corresponding value of  $\rho^x$ .
- For each bin, we computed the mean value of  $\beta^x$  by averaging the values of  $\beta^x$  of all the pairs of subnetworks assigned to it.

We report the results obtained in the histogram of Figure 3.

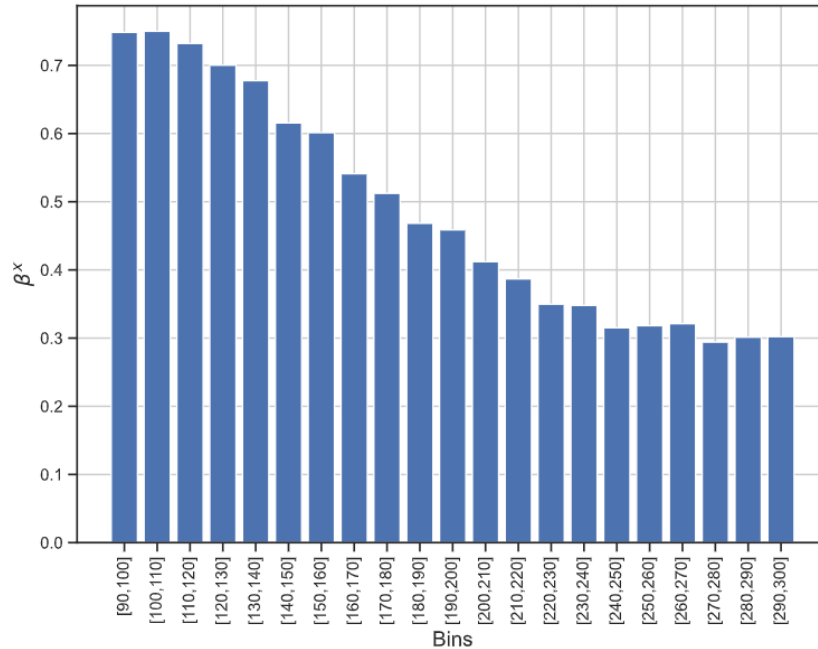


Figure 3: Mean values of  $\beta^x$  against values of  $\rho^x = |N_1^x| + |N_2^x|$

Observe that this histogram starts from the range  $[90, 100]$  of  $\rho^x$  because no pairs of networks fall in lower bins. From the analysis of this figure, we can observe that, as  $\rho^x$  increases, the mean value of  $\beta^x$  decreases, although this trend is gradual. From the graph theory point of view, this can be explained by considering that there is a direct proportionality relationship between  $\beta^x$ , on one side, and  $d_1^x$  and  $d_2^x$ , on the other side. Now, as  $\rho^x$  increases, the denominators of  $d_1^x$  and  $d_2^x$  grow according to a quadratic trend, while their numerators grow at most with a quadratic trend, but generally with a trend between linear and quadratic. This tendency for the numerators to grow less than the denominators is reflected in the trend of  $d_1^x$  and  $d_2^x$  against  $\rho^x$  and, consequently, in the trend of  $\beta^x$  against the same parameter. From our analyses viewpoint, this implies that, as  $\rho^x$  increases, the importance of the semantic similarity against the topological similarity increases too. This is justified taking into account that, as  $\rho^x$  increases, the number of lemmas available to define each network increases as well, and therefore the possibility to better define the semantics expressed by these last ones grows. This semantics is certainly richer than the one that can be defined through the simple topological analysis of the network.

## 6.4 Investigating $\alpha$

In Section 5, we have seen that the semantic similarity degree  $\sigma_{12}$  between two subnetworks  $\mathcal{N}_1$  and  $\mathcal{N}_2$  depends on the coefficient  $\alpha$ . This defines the weight of the semantic similarity expressed by co-occurrences against the one expressed through the semantic relationships between lemmas. In the same section, we have defined a formula for  $\alpha$  and we have seen that it is substantially related to the values of  $|A_1^c|$ ,  $|A_1^r|$ ,  $|A_2^c|$  and  $|A_2^r|$ .

In this experiment, we aim at performing some analyses on the trend of the value of  $\alpha$  against the variation of the four parameters above. To this end, we have carried out the following tasks:

- We considered 50 sets of comments of different sizes. Each set was selected uniformly at random across random posts from our dataset.
- We performed the activities described in Sections 3, 4 and 5 on each set and obtained 50 CS-Nets of different sizes.
- We considered all possible pairs  $(\mathcal{N}_1, \mathcal{N}_2)$  of CS-Nets that could be constructed from the initial 50 CS-Nets.
- For each pair  $(\mathcal{N}_1, \mathcal{N}_2)$  of CS-Nets, we computed the value of the parameter  $\phi = |N_1| + |N_2|$  (i.e., the overall number of nodes of  $\mathcal{N}_1$  and  $\mathcal{N}_2$ ) and the value of  $\alpha$ .
- We constructed 30 bins of values of  $\phi$ ; specifically, the first bin groups all values of  $\phi$  between 1 and 10, the second bin includes all values of  $\phi$  between 11 and 20, and so on. The last bin comprises all values of  $\phi$  between 291 and 300.
- We assigned each pair  $(\mathcal{N}_1, \mathcal{N}_2)$  of subnetworks to the suitable bin, based on the corresponding value of  $\phi$ .
- For each bin, we computed the mean value of  $\alpha$  by averaging the values of  $\alpha$  of all the pairs of CS-Nets assigned to it.

We report the results obtained in the histogram of Figure 4. Analogously to what happens for  $\beta^x$ , the first bins are not present in the histogram because there was no pair of CS-Nets belonging to them.

From the analysis of this figure, we can observe no specific trend in the values of  $\alpha$  against  $\phi$ . This can be explained by considering that, as  $|N_1|$  and  $|N_2|$  grow, it is presumable that  $|A_1^c|$  and  $|A_2^c|$  on the one hand, and  $|A_1^r|$  and  $|A_2^r|$  on the other hand, will also grow. The value of  $\alpha$  depends on how fast these values grow. Specifically, if  $|A_1^c|$  and  $|A_2^c|$  grow faster than  $|A_1^r|$  and  $|A_2^r|$  then  $\alpha$  increases; in the opposite case,  $\alpha$  decreases. However, this fact is totally independent of the growth of  $|N_1|$  and  $|N_2|$ , because it depends exclusively on the number of co-occurrences of the nodes in the text patterns, on the one hand, and the number of semantic relationships between the lemmas corresponding to the nodes, on the other hand. In any case, there is a constant element to observe in Figure 4 and it concerns the fact that  $\alpha$  is always between 0.6 and 0.7. This means that, in the computation of  $\sigma_{12}$ , the component expressing the co-occurrences of lemmas has a higher weight than the one representing

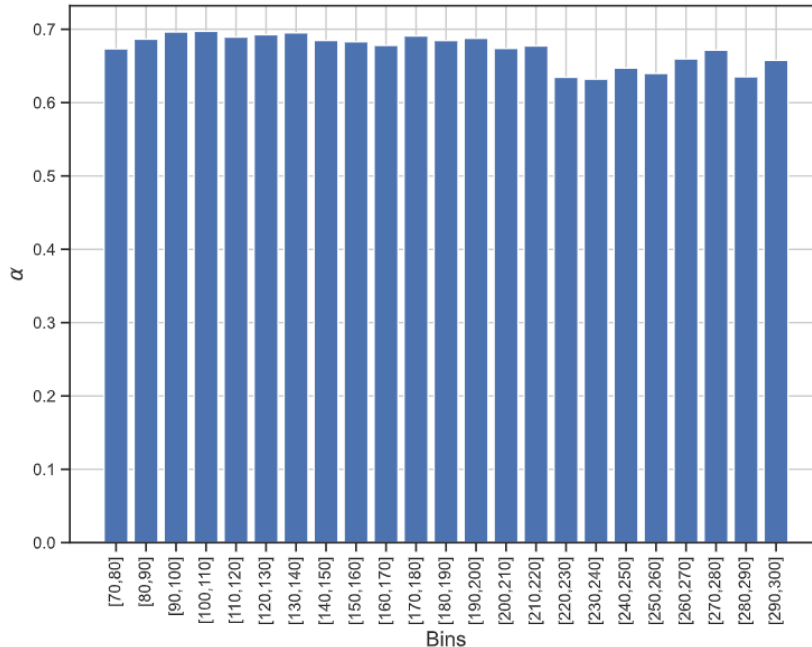


Figure 4: Mean values of  $\alpha$  against values of  $\phi = |N_1| + |N_2|$

the semantic relationships between them. This is reasonable if we consider that the component related to co-occurrences expresses the semantics derived from the dynamic and real use of the lemmas in the comments, while the component related to semantic relationships expresses the semantics as theoretically provided by the language adopted. However, the formula for the computation of  $\alpha$  has been defined in such a way that if, in an application scenario, we have more semantic relationships and much less co-occurrences between lemmas, the weights of the two components are automatically inverted.

Thus, as for the variation of their values against the size of the involved (sub)networks, the parameters  $\alpha$  and  $\beta^x$  show a completely different behavior.

## 6.5 Extracting knowledge from a real world scenario

This latest experiment is intended as a demonstration of the potentialities of our approach in a real world scenario. At the same time, it represents a bridge between the previous subsections, dedicated to experiments, and the next section, concerning applications. In particular, having a Reddit dataset at our disposal, we thought to evaluate, given a user following one or more subreddits, the ability of our approach to recommend new subreddits potentially interesting for her. In this case, our approach would behave as the engine of a content-based recommender system.

The steps of a recommender system employing our approach as an engine and suggesting to a user  $u$  new subreddits to join are the following:

1. Consider the set  $\mathcal{C}_u$  of comments that  $u$  posted in the past.
2. Apply the first two steps of our approach to construct the CS-Net  $\mathcal{N}_u$  associated with  $\mathcal{C}_u$ .



3. Consider a set  $SSet$  of subreddits not yet accessed by  $u$ ; the subreddits of  $SSet$  could be chosen based on parameters like their creation date (favoring the most recent ones), the number of users already accessing them, the number of posts and comments already published in them, etc.
4. For each subreddit  $S_l \in SSet$ , let  $\mathcal{C}_l$  be the set of its comments.
  - 4.1. For each  $\mathcal{C}_l$ , apply the first two steps of our approach to construct the CS-Net  $\mathcal{N}_l$  corresponding to it.
  - 4.2. For each  $\mathcal{N}_l$ , apply the third step of our approach to compute the semantic similarity degree  $\sigma_l$  between  $\mathcal{N}_l$  and  $\mathcal{N}_u$ .
5. Sort the values of  $\sigma_l$  thus obtained in a descending order.
6. Recommend to  $u$  the top  $k$  subreddits of the list. The value of  $k$  can be chosen based on several parameters, such as the seniority of  $u$  on Reddit, the number of subreddits  $u$  is currently accessing, her activity level on Reddit, etc.

We point out that, albeit we presented the previous algorithm with reference to Reddit, it could be applied to several other social networks (such as Facebook and Twitter) with very few changes.

As it is clear from the previous steps, as well as from the way of proceeding of our approach, which is the engine of the recommender system we are describing, the presence of a large set of comments from the user to whom we want to provide recommendations plays a key role on the quality of the results that can be obtained. On the other hand, this is a typical feature of any content-based recommender system. As a consequence, in performing this experiment, we decided to filter out users with few comments. To this end, we computed the distribution of users against comments. It is shown in Figure 5. From the analysis of this figure, we can observe that, even if this distribution does not follow a perfect power law, there are in any case many users posting few comments and few users posting many comments. In our experiment, we judged a number of comments less than 20 as not significant for tracking the interests of a user. Therefore, we selected only users who published more than 20 comments.

To evaluate the performance of the recommender system described above, we borrowed the concepts of *true label* and *Top-k Accuracy* from Machine Learning. Specifically, in the classification task, a true label represents the assignment of a correct class to an observation, while a false label corresponds to a misclassification. The Top-k Accuracy considers the  $k$  predictions of a model having the highest probability. If one of them corresponds to a true label, it considers the prediction as correct; otherwise, it considers the prediction as incorrect. Note that the classical concept of accuracy corresponds to a special case of the Top-k Accuracy one, with  $k = 1$ . Given the complexity of our scenario, in which two or more subreddits could be related to the same topic, and given the huge number of text patterns that could be extracted from a set of comments, we judged that Top-1 Accuracy was a too rigid metric to evaluate the performance of our recommender system and, for this reason, we decided to adopt Top-k Accuracy, with  $1 \leq k \leq 5$ . We point out that we chose the maximum value of  $k$  empirically. In particular, we observed that the values of  $k$  we selected allowed us to obtain the maximum set of subreddits reflecting the scenario of interest. Indeed, as shown in Figure 6, larger values of  $k$  do not lead to an improvement in the hit ratio.

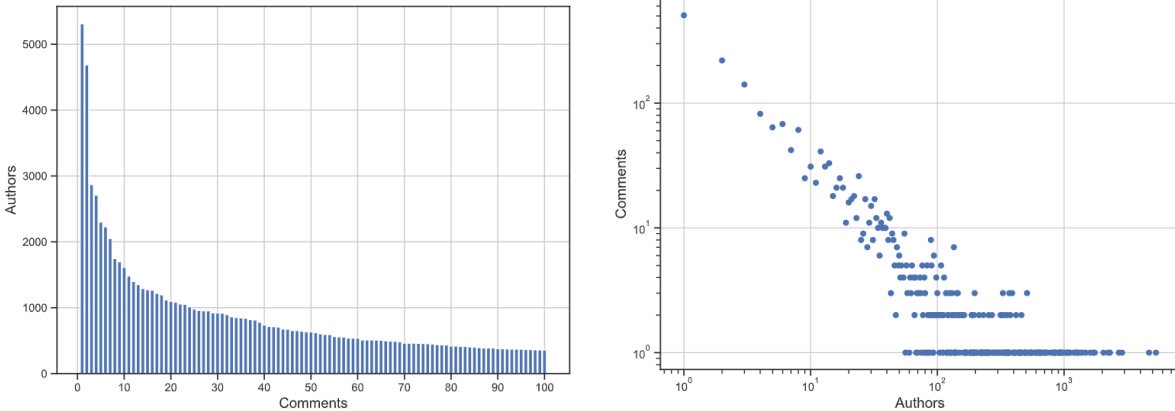


Figure 5: Distribution of authors against comments on linear scale (left) and log-log scale (right)

In order to define the true label of a user, we relied on the homophily principle of Social Network Analysis [42] and made the following assumption: “*the subreddits closest to a specific user are those where she writes the most comments*”. In fact, if a user often visits a subreddit and writes many comments in it, then it means that the topics discussed therein are of her interest. This also means that the patterns characterizing her profile are similar to those used in that subreddit.

Similarly to what we have seen for Top-k Accuracy, we considered that assuming the presence of only one true label for a user is a too rigid hypothesis for the reference context. Therefore, we decided to assume that, for each user,  $h$  true labels are possible,  $1 \leq h \leq 3$  and these are the  $h$  subreddits in which she posted the highest number of comments. Clearly, the comments in these  $h$  subreddits were not used to build  $\mathcal{N}_u$ . Similarly to the range of  $k$ , we set the range of  $h$  empirically. In particular, for larger values of  $h$ , we did not observe significant variations in the hit ratio value against  $k$ , as shown in Figure 6.

Having defined how to proceed and the metrics used in our experiment, we are now able to illustrate how we conducted it. Specifically, we considered all users who posted more than 20 comments. Let  $u$  be one of these users and let  $\mathcal{N}_u$  be the corresponding CS-Net. We ran our recommendation algorithm for her and computed the  $k$  subreddits whose corresponding CS-Nets have the top- $k$  similarity degree with  $\mathcal{N}_u$ . If at least one of the  $k$  subreddits is present in the  $h$  true labels of  $u$ , we considered the whole prediction as a “hit”; otherwise, we categorized it as a “miss”.

In Figure 6, we report the hit ratio, averaged over all users publishing more than 20 eligible comments (i.e., different from those used to build the corresponding profiles), with the values of  $k$  ranging from 1 to 5 and the values of  $h$  ranging from 1 to 3.

From the analysis of this figure we can see that our recommendation algorithm works very well in many cases. The results are already promising for  $h = 1$  (although this is a very stringent condition for the reasons outlined above) as long as the value of  $k$  is greater than or equal to 3. However, we argue that the scenarios best representing the reference context are those with  $h \geq 2$  and  $k \geq 3$ . In this case, the results we obtain are really satisfactory in that the average hit ratio ranges from 81.31% (for  $h = 2$  and  $k = 3$ ) to 93.46% (for  $h = 3$  and  $k = 5$ ).

In this experiment, we used the past data at our disposal, in particular the subreddits already

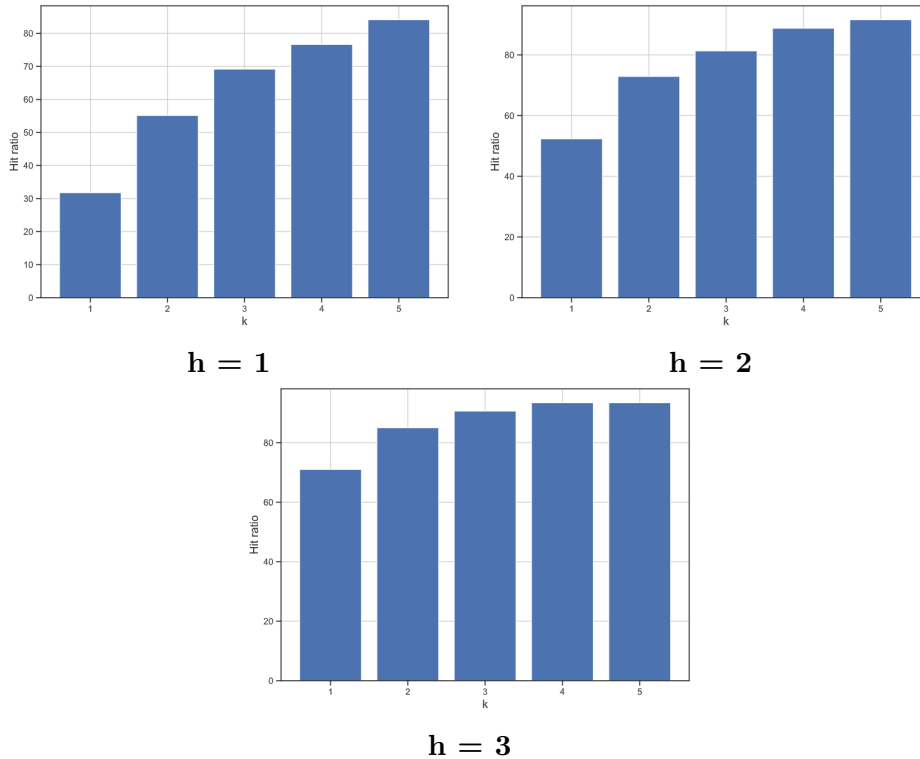


Figure 6: Hit ratio with different values of  $h$  and  $k$

frequented by users, as a test set to evaluate the performance of our recommendation algorithm. It is clear that, in a real world scenario, our recommendation algorithm would not be employed to suggest a user the subreddits that she is already following. Rather, it will be adopted to suggest her subreddits she is not aware of and appearing close to her interests, based on her past behavior.

## 6.6 Discussion

In this subsection, we want to draw some conclusions from examining the results of our experimental campaign and to highlight some lessons learned that may be useful to conduct future research in this area.

First, we note that the dataset available for our tests was derived from `pushshift.io` [8]. This repository has proven to be a valuable and reliable source of Reddit data. An Extraction, Transformation and Loading (ETL) activity was still required, but this was limited. Actually, we believe that it is physiological to perform some ETL in a data science research. In fact, even if there were no errors in the data, it would still be necessary to adapt its format and structure to the investigation that researchers intend to perform. An in-depth study on this part that we could think of carrying out in the future is the realization of a real Exploratory Data Analysis on the starting dataset. Indeed, in this research we have already conducted some descriptive analyses on it, but we had a very specific purpose in mind for the testing campaign we conducted. Therefore, we limited the descriptive analyses to those necessary to verify the achievement of the goals of the current testing campaign. We believe

that an in-depth Exploratory Data Analysis may allow us in the future to identify possible correlations among data and may provide us with insights for further research.

In our opinion, the experiments on the generated Content Semantic Networks represent an important component of our research. Remember that one of the goals of the latter was to demonstrate that high utility patterns can play a key role in the analysis of semantic content. This role could only be played in a very partial way by the most frequent patterns. The results on the density of the CS-Nets obtained represent an important confirmation of the ability of high utility patterns to return semantically cohesive CS-Nets. In the future, we plan to deepen in this analysis by considering various utility functions and determining which of them return the most semantically significant CS-Net. It would also be interesting to understand if different utility functions return CS-Nets with the same level of semantic significance but with different properties and, in the affirmative case, what are the properties that differentiate one from the others.

We do not feel that we need to dwell on discussing in more detail the results of the test for tuning  $\alpha$  and  $\beta^x$ . In fact, although these tests required a considerable design and implementation effort, from the scientific point of view they represent classic tests for tuning the parameters of a research approach.

On the contrary, we consider extremely interesting to focus on the last experiment concerning the extraction of knowledge from a real world scenario. The first result of this experiment concerns the presence of many users posting few comments and the presence of few users posting many comments. This result was somehow expected because most of the posting activities in social networks follow a power law distribution. On the contrary, the second result was not obvious and represents an important strength of our approach. It tells us that the latter can be used as an engine of a content-based recommender system capable of suggesting a subreddit to a user on the basis of her past behavior. This is already an important result in itself, but its relevance goes far beyond we have seen in this experiment. In fact, the recommendation of a community to a user in an Online Social Network is one of the most investigated application issues in the Social Network Analysis literature. Also for this reason, we return to this issue in the next section, where we show how our approach can provide an interesting contribution in this setting.

## 7 Possible Applications

As we mentioned in the previous sections, our approach is general in the sense that it proposes: *(i)* a data model capable of representing and handling a set of comments, regardless of their source; *(ii)* a technique to filter comments based on both their frequency and their utility; *(iii)* a technique to construct a CS-Net for each set of filtered comments; *(iv)* a technique to evaluate the semantic similarity of two CS-Nets.

As a consequence, it may have various applications depending on the origin of comments. In this section, we mention some of them while pointing out that several others can be thought once one or more sets of comments of interest for a given scenario have been identified. Before starting this examination, we would like to point out that the objective of this section is not to fully and thoroughly define the various applications with all their technical details. This study, accompanied by the corresponding tests aimed at highlighting the applications' correctness and performance, will

be the subject of future work. Our goal now is showing that the approach defined in this paper might be exploited in various application scenarios.

We group the application examples that we present in this section into two families, namely recommender systems and community detection. We describe each family in a separate section. In it, we first present the applications of the family, one per subsection. Then, we illustrate the advantages and disadvantages of using our approach for them.

## 7.1 Recommender systems

A recommender system is a content filtering program that creates personalized recommendations for the user to help her in her choices (e.g., it recommends movies on streaming services, products on e-commerce sites, or friends on social networks). Currently, most recommender systems adopt content-based filtering or collaborative filtering [57]. In the next subsections, we discuss the application of our approach to support recommender systems.

### 7.1.1 Content-based recommender systems

Let  $u_1$  be a user and let  $\mathcal{C}_1$  be a set of lemmatized comments that she expressed in a past time interval. The length of the time interval can be arbitrarily defined taking into account that the further back in the past we go, the richer  $\mathcal{C}_1$  could be, but, at the same time, the higher the risk that it includes topics no longer of interest to  $u_1$ . Starting from  $\mathcal{C}_1$ , a set  $\mathcal{P}_1$  of patterns can be derived by applying the techniques explained in Section 3. Once  $\mathcal{P}_1$  has been constructed, it is possible to build a CS-Net  $\mathcal{N}_1$  that indicates the interest of  $u_1$  based on the comments she made in the past. Specifically:

$$\mathcal{N}_1 = \langle N_1, A_1^c \cup A_1^r \rangle$$

$N_1$  is the set of nodes of  $\mathcal{N}_1$ . There is a node  $n_i \in N_1$  for each lemma  $l_i$  present in at least one pattern of  $\mathcal{P}_1$ . An arc  $(n_i, n_j, w_{ij}) \in A_1^c$  indicates that the lemmas  $l_i$  and  $l_j$  occur together in at least one pattern of  $\mathcal{P}_1$ ;  $w_{ij}$  depends on the number of patterns of  $\mathcal{P}_1$  in which  $l_i$  and  $l_j$  occur together. An arc  $(n_i, n_j, w_{ij}) \in A_1^r$  denotes that there is a form of semantic relationship between  $l_i$  and  $l_j$ ; according to what we said about this issue in Section 4,  $w_{ij}$  denotes the strength of that relationship.

Similarly, let  $\mathcal{C}_2$  be a second set of lemmatized comments associated with a set  $PSet_2$  of posts or a subreddit  $S_2$ , which  $u_1$  has not commented yet, e.g., because she does not know of its existence. Starting from  $\mathcal{C}_2$ , it is possible to construct a set  $\mathcal{P}_2$  of patterns, by applying the techniques explained in Section 3, and a CS-Net  $\mathcal{N}_2$  corresponding to  $\mathcal{C}_2$ . The structure and semantics of  $\mathcal{N}_2$  are similar to those of  $\mathcal{N}_1$ .

At this point, by applying the technique expressed in Section 5, it is possible to compute a coefficient  $\sigma_{12}$  that indicates the semantic similarity between  $\mathcal{N}_1$  and  $\mathcal{N}_2$ . If this similarity is high, we can conclude that the set  $PSet_2$  of posts or the subreddit  $S_2$  may be of interest to  $u_1$  and, thus, may be recommended to her. In this way, we can implement a content-based recommender system that can suggest new posts or subreddits to  $u_1$  based on her past history.

### 7.1.2 Collaborative filtering recommender systems

Let  $u_1$  be a user and let  $\mathcal{C}_1$  be the set of lemmatized comments that she expressed in a past time interval. Let  $USet$  be a set of users about whom we make no assumptions. Let  $u_h$  be a user of  $USet$ , let  $\mathcal{C}_h$  be the set of lemmatized comments she expressed in the same time interval considered for  $u_1$ . Applying the same reasoning seen in the previous subsection, we can build a CS-Net  $\mathcal{N}_1$  that represents the profile of  $u_1$  and a CS-Net  $\mathcal{N}_h$  for each user  $u_h \in USet$ .

At this point, it is possible to compute the similarity coefficients between  $u_1$  and each user  $u_h \in USet$  by computing the corresponding similarity between  $\mathcal{N}_1$  and  $\mathcal{N}_2$ . Having these coefficients at disposal, it is possible to apply a k-Nearest-Neighborhood approach to identify the set  $\overline{USet}$  of users with interests most similar to those of  $u_1$ . Thanks to the homophily principle of Social Network Analysis [42], it is possible to assume that the posts and subreddits of interest to users of  $\overline{USet}$  are also of interest to  $u_1$ . As a consequence, if  $u_1$  does not already know them, they can be recommended to her.

In this way, we have realized a collaborative filtering recommender system that can suggest new posts or subreddits to  $u_1$ , based on the behavior of users with interests similar to her.

### 7.1.3 Discussion

As reported in [57], the most used algorithm in the context of content-based recommender systems involves the Vector Space Representation (VSR) of the items of interest to the user and the adoption of methods such as rule induction, nearest neighbor, Rocchio’s algorithm, linear classifier and probabilistic methods. VSR depends strongly on the choice of the item attributes and the approach to measure the closeness between two vectors. The knowledge and choice of the attributes is difficult, and the identification of the closeness function is complex and also depends on the characteristics of the attributes chosen. Our approach is not based on VSR but on network analysis. This makes it independent of attribute knowledge. The closeness between two items is provided by the similarity coefficient  $\sigma_{12}$ , without the need to know the attributes in detail.

As for collaborative filtering recommender systems, as reported in [57], the most widely used algorithm is the k-Nearest Neighbor. There are two subfamilies of collaborative filtering recommender systems, namely user-based and item-based. The first subfamily suffers from scalability problems; to overcome them, the second subfamily was introduced. Our approach belongs to the latter. In it, two important factors to consider are similarity computation and prediction generation. Our approach serves exactly as a core to address these two factors. The main problem of collaborative filtering recommender systems concerns the large level of sparsity in the corresponding dataset. To address this problem, dimensionality reduction techniques, such as Matrix Factorization, Latent Semantic Index (LSI) and Singular Value Decomposition (SVD) have been proposed in the past. These techniques are generally very expensive. Our approach avoids their use as it prevents the sparsity problem during the construction of the network  $\mathcal{N}_1$ . In fact, in  $\mathcal{N}_1$ , a node is connected to at least another node through an arc belonging to  $A^c$  and/or an arc belonging to  $A^r$ .

## 7.2 Community and outlier detection in social platforms

Community detection in a social platform deals with analyzing its members to identify communities. From these investigations, it is possible to deduce the presence of outliers, which are members considered anomalous due to their different behavior with respect to the one of the members of all communities [36].

In the next subsections, we discuss the application of our approach first on a generic social platform and then on Reddit. In the latter case, it allows the creation of new virtual subreddits from real ones.

### 7.2.1 Building new user communities and/or identifying outliers

Let  $USet$  be a set of users on whom we make no initial assumption about their membership in specific communities or about the similarity of their interests. Let  $u_h$  be a user of  $USet$  and let  $\mathcal{C}_h$  be the set of lemmatized comments she expressed in a past time interval. As for the length of this interval, the same considerations seen in Section 7.1.1 can be applied. Performing the procedure seen in that section, we can construct a CS-Net  $\mathcal{N}_h$ , which represents the interests of  $u_h$  as they emerge from  $\mathcal{C}_h$ .

At this point, for each pair of users  $u_1$  and  $u_2$  belonging to  $USet$ , we can compute the semantic similarity coefficient  $\sigma_{12}$  by applying the procedure described in Section 5. The knowledge of this coefficient for each possible pair of users of  $USet$  gives us the possibility to apply on the users of  $USet$  one clustering algorithm among those existing in literature, e.g. DBSCAN [33] that provides very accurate results and allows us to identify outliers. The clusters thus defined allow us to build virtual communities of users (one for each cluster) characterized by similar topics. In Reddit, they could be exploited to build new subreddits.

Furthermore, the outliers thus identified would correspond to users with interests very far from those of the other ones. They could become the “seeds” for new communities dealing with issues different from those already existing (for instance, extremely innovative issues). In other circumstances, the detection of outliers could allow the discovery of users with illegal interests (e.g., fanatics, terrorists, etc.) to be reported to the police.

### 7.2.2 Building new subreddits and/or identifying outliers

Let  $SSet$  be a set of subreddits on which we make no initial assumptions about the similarity of the interests of the users joining them. Let  $S_h$  be a subreddit of  $SSet$ , let  $PSet_h$  be the set of its posts and let  $\mathcal{C}_{h_k}$  be the set of comments corresponding to the post  $p_{h_k} \in PSet_h$ . Applying a procedure similar to the one seen for the users of  $USet$  in Section 7.2.1, we can construct a CS-Net  $\mathcal{N}_{h_k}$  that represents the interests of people involved in  $p_{h_k}$  as they emerge from their past comments.

At this point, we can apply the approach described in the previous section to the resulting CS-Nets. In this way, we can identify clusters of posts (perhaps belonging to different real subreddits) with similar topics. These posts can be grouped into homogeneous virtual subreddits obtained from the real ones. Each virtual subreddit thus obtained can be recommended to each user who had accessed at least one post included in it. In this way, the information, knowledge and opinion exchange between users belonging to different real communities and having similar interests are favored. These users can look very favorably and enthusiastically at this cross-contamination process.

Last, but not the least, the presence of outliers is an indicator of the existence of posts with contents very different from those of the others. These posts could become the seeds of new subreddits, similarly to what we have seen in the previous subsection.

### 7.2.3 Discussion

As reported in [36], various approaches for community detection in social networks have been proposed in the literature. Among them, the most common are: *(i)* the approaches based on the DBSCAN algorithm; *(ii)* those based on the extension of IoT techniques to Social Network Analysis; *(iii)* those using the edge content; *(iv)* those using the arc weight; *(v)* those based on the Newman-Girvan algorithm; *(vi)* those designed for a distributed environment in Web-Scale Networks; *(vii)* those based on Bayesian networks and the Expectation Maximization techniques; *(viii)* those using graph mining; *(ix)* those based on spectral clustering; *(x)* those using overlapping communities. Obviously, there is not a method that is always better than another, but each of them has advantages and disadvantages, as reported in [36].

Our approach can be seen as a hybrid one that brings together the features of edge content-based approaches (which, in our case, would apply to arcs of type  $A^r$ ) with those of weighted network-based approaches (which, in our case, would apply to arcs of type  $A^c$ ). It is intended to bring together the strenghts of the two approaches while avoiding their weaknesses. In particular, as an edge content-based approach, it provides a better supervision to the community detection process. Instead, as a weighted network-based approach, it is able to perform community detection in an accurate way being guided by weights.

Finally, our approach is modular and scalable because, depending on the circumstances, it is possible to make one type of arc prevail over the other. In fact, if necessary, it is possible to completely cancel the contribution of the arcs of  $A^c$  (resp.,  $A^r$ ), relying entirely on the arcs of  $A^r$  (resp.,  $A^c$ ).

## 8 Conclusion

In this paper, we have presented a data structure and a related approach for managing comment semantics in a social platform. Our data structure is network-based and is capable of handling more perspectives about content semantics. It is also easily extensible if additional perspectives are desired in the future. Our approach is based on the mining of text patterns from comments. This activity is carried out based not only on their frequency but also on their utility. The latter is expressed through a utility function that can be chosen according to the reference scenario and the user’s needs. Our approach is also able to compute the semantic similarity degree of two sets of comments.

We have also examined several possible applications of our approach, namely: the realization of content-based and collaborative filtering recommender systems, the construction of new user communities and/or the identification of outliers. Finally, if applied on Reddit, our approach can also be used for building new subreddits.

As for future work, we plan to extend our research efforts in several directions. First, we could investigate the possibility of using our approach to build a system that autonomously identifies offensive content of a certain type (cyberbullism, racism, etc.) in a set of comments (e.g., those of a certain



user or community) on a social platform. To do so, we should first build a meaningful set of comments with characteristics similar to the ones we want to identify and remove. Then, we should construct a CS-Net  $\mathcal{N}_c$  corresponding to these comments. At this point, given a new set  $\mathcal{C}_n$  of comments, if the corresponding CS-Net  $\mathcal{N}_n$  has a very high semantic similarity degree with  $\mathcal{N}_c$ , we can conclude that  $\mathcal{C}_n$  is offensive and should be removed. Extending the previous idea further, we might consider building a virtual moderator. It could not only remove sets of offensive and inappropriate comments, but also favor the most relevant ones to a certain post or comment. Furthermore, it could associate each user with a reputation degree rewarding her when she publishes relevant comments and penalizing her when she submits irrelevant or offensive ones.

A further interesting issue to investigate regards the evolution of CS-Nets over time. In fact, such an analysis would allow us to identify new trends or topics that characterize a social platform.

Last, but not the least, we could use our approach in a sentiment analysis context. In fact, in the literature, there are several studies on how people with anxiety, and/or psychological and emotional disorders, write their posts or comments on social platforms. We could contribute to these studies by considering a set of comments published by users with such characteristics, constructing the corresponding CS-Nets and analyzing them in detail. We could also compare a CS-Net thus obtained with “template CS-Nets”, representative of a certain emotional state, to possibly perform a suitable classification.

## References

- [1] B. Abu-Salih, P. Wongthongtham, K.Y. Chan, K. Yan, and D. Zhu. CredSaT: Credibility ranking of users in big social data incorporating semantic analysis and temporal factor. *Journal of Information Science*, 45(2):259–280, 2019. SAGE Publications Sage UK: London, England.
- [2] M. Adnan, R. Alhajj, and J. G. Rokne. Identifying Social Communities by Frequent Pattern Mining. In *Proc. of the International Conference on Information Visualisation (IV’09)*, pages 413–418, Barcelona, Spain, 2009. IEEE Computer Society.
- [3] R. Agarwal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the International VLDB Conference (VLDB’94)*, pages 487–499, Santiago de Chile, Chile, 1994. Morgan Kaufmann.
- [4] C.C. Aggarwal, M. Bhuiyan, and M. Al Hasan. Frequent pattern mining algorithms: A survey. In J. Han C. Aggarwal, editor, *Frequent Pattern Mining*, pages 19–64. 2014. Springer, Cham.
- [5] S. Ahmadian, M. Afsharchi, and M. Meghdadi. An effective social recommendation method based on user reputation model and rating profile enhancement. *Journal of Information Science*, 45(5):607–642, 2019. SAGE Publications Sage UK: London, England.
- [6] S. Asur and B.A. Huberman. Predicting the future with social media. In *Proc. of the International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT’10)*, volume 1, pages 492–499, Toronto, Ontario, Canada, 2010. IEEE.
- [7] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. 1999. Addison Wesley Longman.
- [8] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn. The pushshift Reddit dataset. In *Proc. of the International AAAI Conference on Web and Social Media (ICWSM’20)*, volume 14, pages 830–839, Atlanta, GA, USA, 2020. AAAI Press.
- [9] J.L. Bender, M.-C. Jimenez-Marroquin, and A.R. Jadad. Seeking support on facebook: A content analysis of breast cancer groups. *Journal of Medical Internet Research*, 13(1):e16, 2011. JMIR Publications.
- [10] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos. Netsimile: A scalable approach to size-independent network similarity. *arXiv preprint arXiv:1209.2684*, 2012.

- [11] P.J. Boczkowski, M. Matassi, and E. Mitchelstein. How young users deal with multiple platforms: The role of meaning-making in social media repertoires. *Journal of Computer-Mediated Communication*, 23(5):245–259, 2018. Oxford University Press.
- [12] Z. Bouraoui, J. Camacho-Collados, and S. Schockaert. Inducing relational knowledge from BERT. In *Proc. of the International Conference on Artificial Intelligence (AAAI 2020)*, volume 34(05), pages 7456–7463, New York, NY, USA, 2020. Association for the Advancement of Artificial Intelligence.
- [13] F. Cauteruccio, E. Corradini, G. Terracina, D. Ursino, and L. Virgili. Investigating Reddit to detect subreddit and author stereotypes and to evaluate author assortativity. *Journal of Information Science*, 2021. SAGE.
- [14] X. Chen, Y. Yuan, and M.A. Orgun. Using Bayesian networks with hidden variables for identifying trustworthy users in social networks. *Journal of Information Science*, 46(5):600–615, 2020. SAGE Publications Sage UK: London, England.
- [15] C. Choi and J.D. Leczy. A semantic network analysis of changes in north korea’s economic policy. *Governance*, 25(4):589–616, 2012. Wiley.
- [16] E. Corradini, A. Nocera, D. Ursino, and L. Virgili. Investigating the phenomenon of NSFW posts in Reddit. *Information Sciences*, 566:140–164, 2021. Elsevier.
- [17] P. De Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of XML Schemas at various “severity” levels. *Information Systems*, 31(6):397–434, 2006.
- [18] M. Detyniecki. Fundamentals on aggregation operators. <http://www.cs.berkeley.edu/~marcin/agop.pdf>, 2001.
- [19] Y. Djenouri, A. Belhadi, P. Fournier-Viger, and J.C. Lin. Fast and effective cluster-based information retrieval using frequent closed itemsets. *Information Sciences*, 453:154–167, 2018. Elsevier.
- [20] R.Y. Dougnon, P. Fournier-Viger, and R. Nkambou. Inferring user profiles in online social networks using a partial social graph. In *Proc. of Canadian Conference on Artificial Intelligence*, pages 84–99, Halifax, Nova Scotia, Canada, 2015. Springer.
- [21] R.E. Dubrofsky and M.M. Wood. Posting racism and sexism: Authenticity, agency and self-reflexivity in social media. *Communication and Critical/Cultural Studies*, 11(3):282–287, 2014. Routledge.
- [22] B. Fazzinga, S. Flesca, F. Furfaro, and E. Masciari. Rfid-data compression for supporting aggregate queries. *ACM Transactions on Database Systems (TODS)*, 38(2):1–45, 2013. ACM New York, NY, USA.
- [23] B. Fazzinga, S. Flesca, F. Furfaro, E. Masciari, and L. Pontieri. Efficiently interpreting traces of low level events in business process logs. *Information Systems*, 73:1–24, 2018. Elsevier.
- [24] M. Fernández and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 22(6-7):753–758, 2001. Elsevier.
- [25] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y.S. Koh, and R. Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.
- [26] P. Fournier-Viger, J.C.W. Lin, R. Nkambou, B. Vo, and V.S. Tseng. *High-Utility Pattern Mining*. 2019. Springer.
- [27] P. Fournier-Viger, J.C.W. Lin, B. Vo, T.T. Chi, J. Zhang, and H.B. Le. A survey of itemset mining. *WIREs Data Mining and Knowledge Discovery*, 7(4):e1207, 2017. Wiley.
- [28] A. Fronzetti Colladon, B. Guardabascio, and R. Innarella. Using social network and semantic analysis to analyze online travel forums and forecast tourism demand. *Decision Support Systems*, 123:113075, 2019. Elsevier.
- [29] L. Gadár and J. Abonyi. Frequent pattern mining in multidimensional organizational networks. *Scientific Reports*, 9(1):1–12, 2019. Nature Publishing Group.
- [30] W. Gan, C. Lin, P. Fournier-Viger, H. Chao, V. Tseng, and P. Yu. A Survey of Utility-Oriented Pattern Mining. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1306–1327, 2021. IEEE.
- [31] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010. Springer.

- [32] H.U. Gerber and G. Pafum. Utility functions: from risk theory to finance. *North American Actuarial Journal*, 2(3):74–91, 1998. Taylor & Francis.
- [33] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques - Third Edition*. 2011. Morgan Kaufmann notes.
- [34] J. Han, J. Pei, Y. Yin, and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004. Springer.
- [35] C.J. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proc. of the International AAAI Conference on Weblogs and Social Media (ICWSM'14)*, pages 216–225, Ann Arbor, MI, USA, 2014.
- [36] M. Khatoon and W.A. Banu. A survey on community detection methods in social networks. *International Journal of Education and Management Engineering*, 5(1):8, 2015. Modern Education and Computer Science Press.
- [37] K.H. Kwon, C. Chris Bang, M. Egnoto, and H. Raghav Rao. Social media rumors as improvised public opinion: semantic network analyses of twitter discourses during korean saber rattling 2013. *Asian Journal of Communication*, 26(3):201–222, 2016. Routledge.
- [38] H. Liu and P. Singh. ConceptNet — a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. Springer.
- [39] W. Liu, K. Yue, H. Wu, X. Fu, Z. Zhang, and W. Huang. Markov-network based latent link analysis for community detection in social behavioral interactions. *Applied Intelligence*, 48(8):2081–2096, 2018. Springer.
- [40] M. Maree, A.B. Kmail, and M. Belkhatir. Analysis and shortcomings of e-recruitment systems: Towards a semantics-based approach addressing knowledge incompleteness and limited domain coverage. *Journal of Information Science*, 45(6):713–735, 2019. SAGE Publications Sage UK: London, England.
- [41] J.N. Matias. Going dark: Social factors in collective action against platform operators in the Reddit blackout. In *Proc. of the International Conference on Human Factors in Computing Systems (ACM CHI 2016)*, pages 1138–1151, San Jose, CA, USA, 2016. ACM.
- [42] M. McPherson, L. Smith-Lovin, and J.M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001. JSTOR.
- [43] H. Midi, S.K. Sarkar, and S. Rana. Collinearity diagnostics of binary logistic regression model. *Journal of Interdisciplinary Mathematics*, 13(3):253–267, 2010. Taylor & Francis.
- [44] A.G. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- [45] A. Mislove, B. Viswanath, K.P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proc. of the third ACM International Conference on Web Search and Data Mining (WSDM'10)*, pages 251–260, New York, NY, USA, 2010. ACM Press.
- [46] S.A. Moosavi, M. Jalali, N. Misaghian, S. Shamshirband, and M.H. Anisi. Community detection in social networks using user frequent pattern mining. *Knowledge and Information Systems*, 51(1):159–186, 2017. Springer.
- [47] B. K. Narayanan and M. Nirmala. Adult content filtering: Restricting minor audience from accessing inappropriate Internet content. *Education and Information Technologies*, 23(6):2719–2735, 2018. Springer.
- [48] L. Palopoli, D. Saccà, G. Terracina, and D. Ursino. Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):271–294, 2003.
- [49] K. Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. The Royal Society.
- [50] M. Qin, D. Jin, K.Lei, B. Gabrys, and K. Musial-Gabrys. Adaptive community detection incorporating topology and content in social networks. *Knowledge-Based Systems*, 161:342–356, 2018. Elsevier.
- [51] T. Radicioni, T. Squartini, E. Pavan, and F. Saracco. Networked partisanship and framing: a socio-semantic network analysis of the italian debate on migration. *CoRR*, abs/2103.04653, 2021. arXiv.

- [52] A. Reihanian, M.R. Feizi-Derakhshi, and H.S. Aghdasi. Overlapping community detection in rating-based social networks through analyzing topics, ratings and links. *Pattern Recognition*, 81:370–387, 2018. Elsevier.
- [53] A. K. Shelton and P. Skalski. Blinded by the light: Illuminating the dark side of social network use through content analysis. *Computers in Human Behavior*, 33:339–348, 2014. Elsevier.
- [54] P. Sprent. *Applied nonparametric statistical methods*. Springer Science & Business Media, 2012.
- [55] G.J. Székely, M.L. Rizzo, and N.K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007. Institute of Mathematical Statistics.
- [56] Y.R. Tausczik and J.W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010. SAGE.
- [57] P.B. Thorat, R.M. Goudar, and S. Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4):31–36, 2015. Foundation of Computer Science.
- [58] G.M. Van Koningsbruggen, T. Hartmann, A. Eden, and H. Veling. Spontaneous hedonic reactions to social media cues. *Cyberpsychology, Behavior, and Social Networking*, 20(5):334–340, 2017. Mary Ann Liebert, Inc. USA.
- [59] Z. Wu, J. Cao, J. Wu, Y. Wang, and C. Liu. Detecting Genuine Communities from Large-Scale Social Networks: A Pattern-Based Method. *The Computer Journal*, 57(9):1343–1357, 2014. Oxford University Press.
- [60] H. Yao, H.J. Hamilton, and L. Geng. A unified framework for utility-based measures for mining itemsets. In *Proc. of the ACM SIGKDD Workshop on Utility-Based Data Mining (UBDM'06)*, pages 28–37, Philadelphia, PA, USA, 2006. ACM.
- [61] M. Yoo, S. Lee, and T. Ha. Semantic network analysis for understanding user experiences of bipolar and depressive disorders on reddit. *Information Processing & Management*, 56(4):1565–1575, 2019. Elsevier.
- [62] J. Zhao, J. Wu, X. Feng, H. Xiong, and K. Xu. Information propagation in online social networks: a tie-strength perspective. *Knowledge and Information Systems*, 32(3):589–608, 2012. Springer.