COMMUNITY EVOLUTION ANALYSIS WITH ENSEMBLE LOUVAIN

Bojan Evkoski

Master Thesis
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Asst. Prof. Dr. Petra Kralj Novak, Central European University in Vienna, Austria, and Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Ljupčo Todorovski, Chair, Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
Prof. Dr. Zoran Levnajić, Member, Faculty of Information Studies in Novo mesto, and Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Dr. Petra Kralj Novak, Member, Central European University in Vienna, Austria, and Jožef Stefan Institute, Ljubljana, Slovenia

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Bojan Evkoski

COMMUNITY EVOLUTION ANALYSIS WITH ENSEMBLE LOUVAIN

Master Thesis

ANALIZA RAZVOJA SKUPNOSTI Z METODO ENSEMBLE LOUVAIN

Magistrsko delo

Supervisor: Asst. Prof. Dr. Petra Kralj Novak

Ljubljana, Slovenia, November 2022

To Baba Anče

Acknowledgments

Firstly, I would like to express my gratitude to my supervisor, Prof. Dr. Petra Kralj Novak, for these wonderful two years of studying and researching under her guidance. Thank you for giving me the chance that I did not think I deserve, and thank you for giving me the support, knowledge and tools to make the best out of it. You are a true mentor, without whom this work would not have been possible.

I would also like to thank Igor Mozetič, Nikola Ljubešić and Andraž Pelicon for their warm welcome to our research team right from day one. Thank you for always believing in my capabilities and pushing me forwards in my research. Our weekly meetings always kept me inspired and gently led me toward progress, even when I felt like I was failing to do so.

Thank you to everyone from the Knowledge Technologies Department for the stimulating and supportive environment. You all provided a great foundation for my first baby steps into research.

Acknowledgments to the Public Scholarship, Development, Disability and Maintenance Fund of the Republic of Slovenia through its scholarship program, as well as the EU REC Programme (2014–2020) project IMSyPP (grant no. 875263) for funding my master studies.

On a personal note, I would like to thank my friends and family for being by my side through this entire journey and helping me write the thesis. With your companionship, I never felt lonesome, even in the most difficult pandemic times.

To my parents. For your unconditional love. For your daily sacrifice in the effort to ensure a better life for me. For supporting all of my aspirations.

To my partner, for gifting my mind and heart a home to which I can always return.

Abstract

Complex networks, or simply networks, are robust structures for representing relationships between entities (nodes) connected in nontrivial ways. These networks can be used to model many types of relationships and processes in physical, biological, social, and information systems. A valuable aspect of network analysis is the identification of strongly connected groups, as this allows the creation of a large-scale map of a system. These groups are called communities and often provide information about the function of the system represented by the network.

The expansion of data popularized the field of network analysis and opened up the possibility of closely observing the represented processes in a temporal manner. Temporal analysis of communities is known as community evolution and aims to detect and explain changes in the collective behaviour of groups. It can be a precious tool in answering many phenomena, especially for social networks, such as economical or political shifts, measuring the influence of topics in forming and dissolving communities, the evolution of echo chambers, etc.

In this thesis, we present two contributions to the field of community detection and evolution. The first is a community detection method named Ensemble Louvain, which produces stable communities with high quality, suitable for evolution analysis. It uses ensembles of a famous community detection algorithm, significantly outperforming it and other ensemble methods that utilize it.

The second contribution is a novel strategy for using artificial networks for community detection benchmarking. The Lancichinetti-Fortunato-Radicchi (LFR) benchmark is the most widely accepted algorithm for generating artificial networks that resemble real-world networks. In the commonly used setting, the diversity of LFR networks is limited. Because the performance of community detection algorithms can vary depending on other network properties, conclusions based on a single set of LFR parameter values can be misleading. Therefore, we propose a comprehensive benchmarking of community detection algorithms that avoids the shortcomings of the standard LFR benchmarking, called the Unconstrained LFR benchmark.

Finally, we present three of our published works where we apply our Ensemble Louvain on a real-world dynamic social network, gaining insight into the development of Twitter communities. We observe the communities by analyzing the evolution of their influential users, discussion topics, and hate speech use. With this, we show a clear example of how community evolution can be used in applied quantitative research in the interdisciplinary field of complex networks.

Povzetek

Kompleksna omrežja, ali preprosto omrežja, so diskretne strukture za predstavitev odnosov med netrivialno povezanimi entitetami (vozlišči). Kompleksna omrežja lahko uporabljamo za modeliranje različnih odnosov in procesov v fizičnih, bioloških, družbenih in informacijskih sistemih. Osnovna metoda analize omrežij je identifikacija močno povezanih skupin, saj to omogoča poenostavljen vpogled v strukturo obravnavanega sistema. Te skupine močno povezanih vozlišč imenujemo skupnosti in pogosto razkrivajo informacije o delovanju sistema, ki ga predstavlja omrežje.

Dostop do velike količine podatkov odpira možnost analize opazovanih sistemov skozi čas. Časovna analiza skupnosti, poimenovana tudi razvoj skupnosti, je namenjena odkrivanju in razlagi sprememb v kolektivnem vedenju skupin. To omogoča spremljanje in analizo številnih pojavov zlasti na družbenih omrežjih, saj lahko zaznamo gospodarske in politične premike, merimo vpliv tem pri oblikovanju in razpadu skupnosti, razvoj komor odmevov (ang. echo chambers) in podobno.

V tem magistrskem delu predstavljamo dva prispevka na področju odkrivanja in razvoja skupnosti v kompleksnih omrežjih. Prvi prispevek je nova metoda odkrivanja skupnosti po imenu Ensemble Louvain, ki odkrije stabilne skupnosti visoke kakovosti. Ensemble Louvain je metoda ansamblov, ki temelji na znanem algoritmu za zaznavanje skupnosti Louvain. Ensemble Louvain znatno prekaša tako algoritem Louvain kot tudi druge sorodne metode ansamblov za odkrivanje skupnosti. Predvsem daje stabilne rezultate, primerne za analizo razvoja skupnosti.

Drugi prispevek tega magisterskega dela je nova metoda za evalvacijo algoritmov odkrivanja skupnosti na sintetičnih omrežjih. Lancichinetti-Fortunato-Radicchi (LFR) je splošno uveljavljen algoritem za generiranje umetnih omrežij z znanimi skupnostmi, ki so podobna omrežjem resničnega sveta. V običajno uporabljenem scenariju je raznolikost omrežij LFR omejena zaradi uporabe enega samega seta nastavitev parametrov. Ker se lahko uspešnost algoritmov za odkrivanje skupnosti razlikuje glede na ostale lastnosti omrežji, so lahko sklepi na podlagi enega samega niza vrednosti parametrov LFR zavajajoči. Zato predlagamo metodo za celovito primerjalno analizo algoritmov za odkrivanje skupnosti, ki se izogne pomanjkljivostim standardne primerjalne analize LFR, imenovano Unconstrained LFR.

Nazadnje predstavljamo tri svoja objavljena dela, v katerih uporabljamo naš Ensemble Louvain algoritem na dinamičnem družbenem omrežju z resničnega sveta. Razvoj skupnosti na slovenskem delu socialnega omrežja Twitter opazujemo v obdobju treh let preko njihovih najbolj vplivnih uporabnikov, tem za razpravo in uporaba sovražnega govora. S tem prikazujemo jasen primer, kako je mogoče razvoj skupnosti uporabiti v kvantitativnih raziskavah na interdisciplinarnem področju kompleksnih omrežij.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Networks are one of the most used mathematical structures that present pairwise relations (edges) between objects (nodes). These relations can be found in real systems, oftentimes representing non-trivial topological features. Thus, the networks can be used to model many types of relations and processes in physical, biological, social, and information systems. With the digital expansion of society, these systems offer increasingly more data that can be used for network analysis with the purpose of extracting knowledge on how objects (people, molecules, words, etc.) relate, communicate, and influence each other as well as how they propagate information [1]–[3].

1.1 Motivation and Background

The expansion of data did not just popularize the field of network analysis. It also opened up new possibilities on how we analyze networks, one of which is temporal analysis. The idea of temporal analysis is tracking the changes in network properties, detecting trends, and predicting the future development of topologies. The networks which contain temporal information and allow temporal analysis are called *dynamic networks* [4]–[6].

Another valuable method of analyzing these real-world complex networks is discovering groups of highly connected nodes. One can use these groups to identify and understand collective attributes, behavior, influence, etc. These methods are usually known under the name of *community detection* [7]–[9]. The groupings detected by the methods are called *partitions*.

Community evolution is the intersection of temporal analysis and community detection [10], [11]. Here, one tries to detect changes in the collective behavior of groups. It can be a precious tool in answering many diachronic phenomena, especially for social networks, such as economical or political shifts, measuring the influence of topics in forming and dissolving communities, the evolution of echo chambers, and many others.

Community evolution has many challenges to overcome in order for it to be easily applicable [10]. The main matter is that temporal analysis assumes the atomic granularity of events in (dynamic) networks, while community detection requires aggregated (static) networks. This fundamental difference opens up three major fronts of discussion on tackling community evolution: (dynamic) network representation, temporal adaptation of community detection methods, and correct evaluation of the results the first two steps produce.

In this master thesis, we focus on tackling several technical aspects of community evolution on real-world networks, each represented in the following introductory subsections. We also present three of our published works that apply community evolution analysis in discovering social phenomena.

1.1.1 Dynamic networks representation

The way temporality is represented in networks has major implications on how community detection is designed and applied for community evolution. The atomic events we want to represent in the networks are: new nodes emerging in the network, old nodes disappearing from the network, new edge connections between nodes, and edge connections fading away. Generally, three dynamic network representations exist: static, snapshot, and temporal networks [12].

The simplest representation of dynamic networks are static networks. The static networks are also called "frozen in time" networks since the temporal dimension is not really explicitly represented. One operates with a single network that aggregates the whole period of interest, thus freezing the time data into a single object. Several aggregation approaches exist, depending if we are building unweighted or weighted networks, but we drop going into details since these networks are standard complex networks. Yet, it is interesting to follow the genesis of why these networks are the most common approach in dynamic settings, even today. Historically, this absence of the time dimension has two reasons: the graph theoretic origin of the field, and the scarcity of data at the time when the field of complex networks emerged [13]. Graphs were always considered static objects and their mathematical origin simply excluded changing data. The second reason gives the answer to why dynamic networks were not a working subject at the time: there was simply not enough data available (hourly, daily, weekly, monthly, etc.), and splitting the data into more chunks instead of aggregation usually led to networks losing their capacity to even capture basic structures, making aggregation always the go-to option. Now, with modern data expansion and high daily volumes, aggregation strategies are not preferred since they cannot capture dynamics. These aggregations produce massive static networks, making them impractical for computation tasks (too large for computer memory and too complex for CPUs to apply methods such as community detection in a reasonable time).

The second dynamic network representation is the snapshot: a temporally ordered series of network snapshots. Here, we place a fixed time window (temporal granularity) and then produce networks for the time windows. It can be applied with overlapping or non-overlapping windows. The approach allows for efficient tracking of changes in the network structure and increases the expressiveness of the models but at a cost of higher analytical complexity [12]. Yet, it opens up two exclusive issues which have to be addressed: first, how to keep track of the multiple stages of a network or community's life; second, how to harmonize the analytical results obtained in a snapshot with the outcome of subsequent ones.

The third dynamic network representation is temporal networks, that model dynamic phenomena without any aggregation, keeping all temporal details. The main limitation of snapshots and aggregations is that temporal granularity needs to be fixed. The identification of the granularity is not trivial and is mostly context-dependent, yet it often profoundly impacts analytical results. By avoiding aggregation, temporal networks [14] utilize a complete and fine-grained description of network dynamics. However, this solution immensely increases the complexity of the network model and requires the definition of new analytical methodologies.

Different problems and data impose different modeling choices. Static networks are mostly used to identify stable patterns and to describe the general status of a network. Snapshots are proxies that are being used when there are more volatile interactions and there is a need for studying an increasingly dynamic scenario. And, temporal networks are the most detailed, where starting from fine-grained temporal data/networks, it is possible to generate all the other models by subsequent aggregations. Since we are interested in community detection approaches that deal with temporally annotated graphs, we proceed

in more detail with the major strengths and weaknesses of the most used models in this context: Network Snapshots and Temporal Networks.

1.1.1.1 Network snapshots

Network snapshots are partitioning the network history into a series of time-windowed networks, each one of them constructed as an aggregation of the observed interaction during the time window period. A snapshot graph G_t is defined by an ordered set $< G_1, G_2 ... G_t >$ where each snapshot $Gi = (V_i, E_i)$ is created by the time windowed corresponding sets of nodes V_i and edges E_i. The network snapshots can be effectively used, for instance, to model a phenomenon that generates network changes (almost) at regular intervals, applying time-bound observations describing a precise, static discretization of the network life. The snapshot-based analysis is frequently adopted for networks that have a natural regular pattern since they can provide a balance between model complexity and expressivity. Moreover, this analysis allows one to apply static network tools, such as community detection, to evolving networks.

The snapshot approach crucially depends on the representation of time in the network relations. Two main scenarios were considered so far: perfect memory networks (also known as accumulative growth scenario), and limited memory networks [12]. Perfect memory permits only aggregation of nodes and edges, where the old nodes/edges cannot disappear. The limited memory scenario allows for nodes/edges to disappear over time. This is suitable in social network analysis, where the edge disappearance could indicate the decay of social ties. The limited memory networks are implemented with various methods, including static, sliding, or dynamic-sized time windows, each method with its own strengths and weaknesses.

1.1.1.2 Temporal networks

A temporal network is a dynamic object in which both nodes and edges may appear and disappear as time goes by. More formally, A it is a graph $G = (V, E, T)$ where V is a set of triplets of the form (v, t_s, t_e) with v as a node and the birth and death timestamps of the corresponding node as t_s and t_e ($t_s <= t_e$); E is a set of quadruplets (u, v, t_s, t_e) where u and v are nodes of the graph, while t_s and t_e represent the birth and death of their interaction with an edge. These quadruplets can be undirected or directed.

A distinction between two types of temporal networks exists: interaction networks and relation networks. The former defines interactions with duration (phone calls, face-to-face communication, etc.) or without (emails, short messages, etc.) that can repeat as time goes by. These networks have a very low density of edges and usually describe relatively short events. On the other hand, relation networks model more stable relations (friendships, co-worker networks, etc.) which usually last longer than connections in the interaction networks, making them denser and more stable at any given point. In relation to networks, the state of the graph is well-defined at any given time and can be studied through classical static analysis tools. On the other hand, for the interaction networks, one first needs to retrieve a stable dynamic graph by aggregating small time frames. This distinction is important because many methods, in particular the ones applying analysis on edge streams, update the state of communities after each atomic network change. Most methods discussed in this work consider the relation networks approach, as interaction networks lack literature for community detection.

1.1.1.3 Temporal vs. Snapshot networks

The decision of whether one uses snapshots or temporal networks for community evolution depends on the type of data and network complexity produced by the selected representation. If the data describes the punctual states of the network evolution (daily, weekly, monthly, etc.), the snapshot representation is the preferred method to use. If, on the other hand, more precise information is available (e.g. exact timestamps of e-mails), both solutions can be considered. Then, complexity comes to play. Snapshot complexity depends mostly on the number of snapshot networks since if there are N aggregated snapshots, it means that we need to apply analysis to all N networks.

In the thesis, we discuss methods and applications of community evolution using snapshot networks, as they are very suitable for social networks and enable the usage of classical community detection algorithms, which we heavily utilize in our research. In one of our recent works, presented in Chapter 3, we developed a combined method that circumvents some of the drawbacks of the existing snapshot strategies by building a weight-decaying sliding time window network. We then applied a snapshot selection method where fixed static time windows that contain the most information about the dynamics of the communities is selected.

1.1.2 Community detection

Classically defined, a community is a group of nodes that is more densely connected inside compared to nodes not in the group [13]. In a dynamic network scenario, a dynamic community C can be defined as a set of distinct (node v_i, period P_j) pairs: $C = (v_1, P_1), (v_2, P_2)...(v_n, P_n)$ where $P_n = ((t_{s_0}, t_{e_0}), (t_{s_0}, t_{e_0})...(t_{s_N}, t_{e_N}))$. Basically, for each node v_i in a specific C, we have the periods P_j in which the node belongs in a particular community j for the exact intervals defined in P_j. Dynamic community detection (or community evolution) aims to identify the set of all dynamic communities in a dynamic network. These communities can be both overlapping and non-overlapping. In this subsection, we introduce community detection on static networks, a technique that can be also utilized for community evolution.

The idea of community detection methods is to decompose the networks into meaningful substructures, which could later give us more information about a particular network. This generally accepted definition, although intuitive, does not specify what *meaningful substructures* actually are. A meaningful substructure for one does not mean that it is meaningful for others. For example, some want to find friend substructures in social networks, others are interested in topic-specific substructures (politics, philosophical views, music, product preferences, etc.). Networks often contain multiple meaningful substructures of different granularity. Even if we go into a more technical view of what a subgroup is (density, modularity, etc.), we see that there are many metrics that concentrate on different "meaningfulness". With that, there is no single solution to this problem, and we say that community detection is an ill-posed problem [8].

Since there is no universally accepted definition of communities, there is no universal metric of the goodness of communities. This resulted in a plethora of community detection algorithms [15]–[19], with a wide variety of methods to detect the communities. A 2022 study classifies the community detection algorithms in two categories: descriptive and inferential [20]. A descriptive algorithm attempts to find communities according to some context-dependent notion of a good division of the network into groups. It typically implements a greedy search to optimize a pre-specified metric of goodness. An inferential algorithm, on the other hand, applies a generative model of the network to determine which node partitions are more likely responsible for the observed network. Although the latter

are considered state-of-the-art in the network science community, the descriptive community detection methods are still more popular because of their practicality, accessibility, and most importantly, scalability to large real-world networks.

For community evolution, one major challenge is that most of the community detection algorithms suffer from a common issue—the instability of results (partitions) [21]. With the algorithms producing unstable communities, one cannot be sure if a change really happened in the communities, or if it is just a random false event caused simply by the instability of the method. This instability is due to the fact that most community detection algorithms are based on a greedy optimization of partitions on some metric, making them prone to produce different results each run, as they get stuck in local maxima through the optimization process. This issue becomes even bigger when we use the time-dependent approach to detecting communities as a "bad" run of the community detection algorithm influences the results of detection at the subsequent snapshots.

In application-oriented papers, where Louvain is applied, but community detection is not the core focus, the authors typically either use results of a single run of Louvain [22]–[24] or apply ad-hoc stabilization solutions. One example citation from [25]: "Due to the stochasticity of the clustering algorithm (Louvain), we ran 50 trials with different random initialization and assigned each node to the community it was most frequently associated with." Another example from [26]: "In order to prevent the modularity function from being stuck in a local maximum, the Louvain algorithm is repeated 1.000 times which is a number of runs ensuring the algorithm stability. In order to better explore the solution space of the modularity function Q, a node-reshuffling procedure has been performed: . . . ".

To address this issue, in this thesis, we present a method that we call Ensemble Louvain: a wrapper algorithm of the Louvain, which performs hundreds of Louvain runs with different seeds and then aggregates results to get significantly more stable results with higher community quality.

1.1.3 Community evaluation

Finding a reliable way to evaluate communities is a significant issue to address while approaching community detection. As previously discussed, one of the main issues of community detection lies in the absence of a unique definition of a community. Thus, many different strategies exist for comparing results obtained by different community detection methods, each one focusing on one specific characteristic that a proper network partition should express. Since we identified stability as a challenge in community evolution, one needs to also find a way to robustly evaluate community stability, an aspect that is consistently overshadowed by community quality in the research community.

We can evaluate using synthetic data or real-world data. For synthetic data, several network models were introduced during the 20th century: random graphs [27], small-world networks [28], preferential attachment models [29], the Forest Fire models [30] and Community Affiliation Graphs [31]. Once the network is created, the next step is to generate a community ground truth for which one defines probabilities of intra and inter-cluster edges. These evaluations are also known as Girvan and Newman benchmarks for static non-overlapping communities. Currently, Lancichinetti—Fortunato—Radicchi (LFR) benchmark [32] is the most widely accepted algorithm that generates synthetic benchmark networks that resemble real-world networks. Its advantage over other methods is that it accounts for the heterogeneity in the distributions of node degrees and of community sizes.

The LFR method creates synthetic networks using a few important parameters which control the structure of the network. Previous works on LFR [18], [33]–[36], vary μ, the mixing parameter, as a partition difficulty criteria (higher μ means a higher percentage of

edges going out of the community, hence finding the ground truth communities becomes more difficult) and run the algorithms on one or few different network sizes (parameter n). Meanwhile, all other parameters remain fixed, even though their values can significantly alter the community structure of the network [32]. The disadvantage of this approach is that by fixing most of the parameters, the evaluation is done only on a very small subspace of the possible synthetic networks. In the thesis, we refer to this methodology as *Standard LFR benchmark* and introduce a more diverse benchmark alternative, which we refer to as *Unconstrained LFR benchmark*. The idea of the Unconstrained LFR benchmark is to create artificial networks which vary in multiple aspects (parameters), and not just the mixing parameter μ, which then would allow a broader community detection evaluation.

1.1.4 Community detection metrics

Throughout the thesis, we use three stability and performance metrics which compare the algorithm output partitions with the ground-truth: Normalized Mutual Information (*NMI*) [37], Adjusted Rand Index (*ARI*) [38] and our own improvement[1] of the *BCubed* F_1 metric [39], [40]. Although the three metrics (*NMI*, *ARI* and F_1) measure the same property (community quality) and show comparable results (visible in Figure 2.2 and Figure 2.3), there are some practical differences. Specifically, the *NMI* score is more forgiving when nodes are left out as one-node or few-node islands out of their original community since the calculation is community-wise. On the other hand, the F_1 (node-wise) adds a value of zero to the total F_1 for the specific node or close to zero for the few-node islands, before normalizing by the number of nodes. This maximally penalizes the F_1. The *ARI* penalizes in a similar way, but in a pair-wise manner. Thus, if one demands precision (positive predictive value), *NMI* is a more suitable metric, while if sensitivity is focus, *ARI* and F_1 are more adequate. Finally, the most important benefit of our F_1 metric compared to the other two is that it can also compare disjoint sets of nodes, making it a suitable tool for comparing the similarity of adjacent partitions in community evolution. We define it and extensively use it in our application works on retweet networks, covered in Chapter 3. For detailed formulations of all three metrics, see Appendix B.

In the absence of ground truth, which is the case in most real-world problems, a standard way to compare different algorithms is by internal quality evaluations of the communities in the partition. The most widely used metric of this type is *Modularity*[41]. Values of modularity approaching 1 indicate partitions with a strong community structure, while lower values (with -0.5 being the theoretical minimum) indicate that the partition does not correspond to a community structure in the network. Modularity has also been extensively used in community detection algorithms where the aim is to optimize for a higher modularity score. Yet, the modularity score has been disputed in recent years, where researchers show that partitions with the optimal (global maximum) modularity do not necessarily correspond to what one expects to be good communities [42]. Other scores exist, such as: conductance [43], expansion [43], internal density [43] etc.

Although there are many scores that can help one get insight into the quality of the communities, using a metric of this type introduces a major drawback: it favors methods that are designed to maximize it. And, though helpful, they can be quite misleading if we stick to only one particular metric. Thus, when building a new community detection algorithm, whether for dynamic or static communities, it is recommended that one uses multiple evaluation metrics.

[1] https://github.com/boevkoski/bcubed_f1

1.2 Related Work

In this subsection, we cover the related work of the main challenge we tackle in the thesis — stabilizing community detection methods for their applicability in community evolution analysis.

1.2.1 Community detection instability

The instability of the greedy community detection algorithms, such as Louvain [15], is a major problem if consistent solutions are crucial, such as in community evolution analysis. When the communities detected at times t and $t+1$ differ, one has to be sure that these are genuine differences, and not the result of the stochastic instability of the community detection algorithm. A number of solutions were proposed to solve (or at least mitigate) this ambiguity. Their goal is to smooth out the evolution of communities. Rossetti and Cazabet [10] identify four techniques for mitigating the instability: explicit smoothing, implicit smoothing, global smoothing, and smoothing by bootstrap.

Explicit smoothing is when the algorithm explicitly introduces smoothing in its definition [44], [45], requiring the partition at step t to have similarity with the partition found at $t-1$. These algorithms typically introduce a parameter that determines the trade-off between the quality of the partition at t and its similarity with the partition at $t-1$.

Implicit smoothing is the idea to maximize the similarity between consecutive partitions by favoring. Some methods [46] use the communities found at the previous step as seeds for the current one. Others try to locally update communities that have been directly affected by modifications between the previous and the current step [47].

Global smoothing uses the idea of searching for communities that are coherent in time by examining all steps of evolution simultaneously. Primarily, this is done by creating a single network from different snapshots [21].

1.2.2 Ensemble approaches

What if the aim is not to smooth out the evolution of the communities, but actually identify major shifts in the community structure? Then, one needs to fix the instability of the algorithm on a snapshot level instead of using the timeline of snapshots. That can be done with smoothing by bootstrap, which is the technique we use in Ensemble Louvain, the method covered in this thesis to apply stable community detection. The idea here is to run an algorithm on the same snapshot multiple times, and find stable parts in the community structure which are more easily trackable. It relies on the use of community detection *ensembles*. Analogous to the ensemble approaches from machine learning [48], the idea is to combine partitions produced by multiple runs of the community detection algorithm to derive one superior partitioning, both in terms of stability and quality of the detected communities [49], [50].

The most well-known ensemble approach is the Consensus Clustering method by Lancichinetti and Fortunato [51]. The method applies a greedy detection algorithm r times, computing a consensus matrix D, where an entry D_{ij} is the number of partitions in which vertices i and j are assigned to the same cluster, divided by r. Then, all the entries of D below a chosen threshold τ are set to zero. This is followed by another r times of running the greedy algorithm on the modified consensus matrix D. If the partitions are all equal (or, in other words, the new consensus matrix is block-diagonal), the procedure stops and returns the final communities. Otherwise, the last step is repeated on the new consensus matrix, until the partitions stabilize. The main downside of this approach is the uncertainty on how many partitionings are to be applied before the final communities stabilize,

and if they eventually stabilize at all, since the algorithm is not guaranteed to converge. Moreover, the optimal combinations of the parameters r and τ are underexplored.

Another approach is the Node-based Fusion of Communities (NFC) [52]. Here, the main idea is to apply hierarchical clustering on the consensus matrix D. Then, the authors select the partition where the communities give the highest modularity. The challenges of hierarchical clustering are multiple. First is the usage of distance metrics between the clusters, as it is difficult to find a robust way of measuring the distance between a node and a community, or between two communities. Second, it introduces another type of instability while trying to solve the original, since merging of clusters through the distance metric will very often produce ambiguous situations with multiple "equally good" merges, where arbitrary choices lead to divergent end results. Finally, hierarchical clustering is a slow algorithm that quadratically depends on the number of samples.

Ensemble Clustering for Graphs (ECG) [36] is one more similar approach for tackling instability by using ensembles. It begins with running a community detection algorithm multiple times (the authors recommend it be not more than 16). Then, it creates a weighted meta-network using the consensus matrix D. To get the output partition, it runs a final partitioning on the weighted meta-network. The main issue with ECG is that it does not solve the instability problem completely, as it still depends on greedy modularity optimization to find the final partition on the weighted meta-network. Thus, ECG inherits the instability property of the main algorithm, still making it unsuitable for environments where consistency is crucial.

Finally, a very interesting, yet insufficiently researched idea is YASCA – Yet Another Seed-centric Community detection Algorithm [53]. Here, the method first detects seed nodes that could potentially be central in a community. This is done by utilizing centrality metrics for networks. Next, it applies local community detection for each of the seed nodes. It uses the local communities to create a consensus matrix D and then it applies a final greedy optimization community detection algorithm on the matrix, outputting the final partition.

Ensemble Louvain, the proposed method in this thesis, solves the instability problem without introducing additional ambiguous steps. It uses similar ideas as in the first steps of the Consensus Clustering algorithm, yet shows better results by a simplification of the follow-up procedure (after the consensus or co-membership matrix has been created). It includes a finer evaluation, a broader optimal parameter analysis, and a faster implementation.

1.3 Thesis Structure

The rest of the thesis is organized as follows. In Chapter 2, we describe the methods proposed in this thesis. First, Ensemble Louvain, a novel method for stable community detection suitable for community evolution analysis. Second, Unconstrained LFR, an improvement of a well-known community detection benchmark that enables intuitive comparison between algorithms both in terms of community quality and community stability. In Chapter 3, we present three of our applied research publications on community evolution analysis with Ensemble Louvain: "Community evolution in retweet network", "Retweet communities reveal the main source of hate speech" and "Evolution of topics and hate speech in retweet network communities". In Chapter 4, we discuss the content of the thesis and conclude.

Chapter 2

Methods

This chapter covers our main contributions to the field of community detection. The first, is Ensemble Louvain, a novel method for stable community detection based on ensembles, suitable for community evolution analysis. And the second is the Unconstrained LFR, a community detection benchmark that aims to generate diverse artificial networks and allow intuitive comparison of algorithms.

2.1 Ensemble Louvain

First presented in 2008 by Blondel et al. [15], Louvain is the base for our proposed Ensemble Louvain algorithm. It uses a greedy optimization of a community quality metric, called modularity, to find the most suitable partitioning. Modularity (Q) measures the relative density of edges inside communities with respect to edges outside communities.

Formally, it is defined as follows:

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j),$$

where A_{ij} represents the edge weight between nodes i and j; k_i and k_j are the sums of weights of the edges connecting nodes i and j; m is the sum of all of the edge weights in the graph; c_i and c_j are the communities of the nodes, and δ is the Kronecker delta function.

Optimizing modularity theoretically produces the best possible grouping of nodes of a given network. Yet, its optimization (or any other node-based community quality metric) is NP-hard [54] and this group of community detection algorithms are searching for local maxima of the score using greedy optimization approaches.

The greedy optimization of the Louvain method has two phases that are repeated iteratively. First, each node in the network is assigned to its own community. Then, the algorithm tries to put each node in the community of all other nodes and calculate the overall modularity of all possible combinations of moving a node from one community to another. In the second phase, the combination with maximum modularity is selected and the procedure continues with the new set of communities. This is repeated until no improvement in modularity is observed. Very often, multiple choices in the combinatorial explosion of possible communities give an equivalent overall score of modularity, but only one combination is chosen (the common implementation is selecting the first maximum by random node ordering). Although equivalent, these choices oftentimes diverge in dissimilar outcomes a few steps down the line, leading to bountiful differences in the community structure between partitions and introducing instability.

To overcome the instability of Louvain, we propose the aforementioned Ensemble Louvain. A simple, but powerful approach that uses the instability flaw of Louvain to its

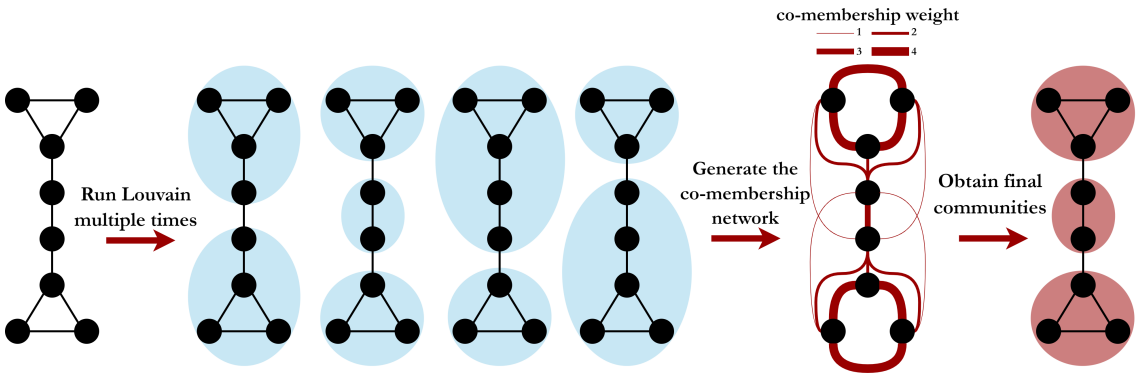

Figure 2.1: **Ensemble Louvain steps.** The outline of Ensemble Louvain consists of the following steps: running standard Louvain multiple times; generating a weighted co-membership network (an edge between two nodes exists if Louvain assigns them to the same community at least once; the edge weight is the number of co-memberships); removing edges of the co-membership network under a given threshold (typically 85%); obtaining final communities by extracting the connected components.

advantage, as it combines multiple unstable Louvain partitions to produce one superior final partition. As shown in Fig 2.1, Ensemble Louvain has the following steps:

- Runs the standard Louvain algorithm *multiple r times* in parallel, with different starting positions in the node order resulting in varying partitions.
- Then, it builds a co-membership (or consensus) network where the edges between the original nodes are weighted co-membership scores (i.e., how many times two nodes appeared in the same community in the multiple Louvain runs).
- This is followed by an edge-removal phase: if a co-membership score is below the *co-membership threshold ct* (see Appendix A.2 for sensitivity analysis of both *ct* and r hyperparameters), the edge is removed from the meta-network.
- The output of the edge-removal phase is a set $\{C_1, ..., C_n\}$ of connected components which represent our final partition P. The final partition is obtained from a very simple non-stochastic process, mitigating the instability of the separate Louvain partitions.

Compared to other ensemble algorithms for community detection, namely Consensus clustering [51] and ECG [36], Ensemble Louvain also builds a co-membership (consensus) network in an identical manner. The difference lies in how the co-membership network is used to produce the final communities. Consensus clustering continues by applying the same procedure of multiple Louvain runs on the co-membership network repeatedly, until it achieves unambiguous stable communities where all co-memberships are either zero or maximum. On the other hand, ECG applies a final Louvain partitioning on the co-membership network, introducing a new opening for instability in the methodology. Ensemble Louvain applies neither of these techniques. Rather, it simplifies the methodology by immediately inferring the final communities from the co-membership network using the edge-removal phase. Doing so reduces the stochasticity of the process and enables faster convergence.

There are multiple benefits of using Ensemble Louvain. Here, in order to maintain a simple structure of the thesis with a gradual introduction of concepts and techniques,

we only cover the main strong points of Ensemble Louvain. Section 2.2 and Appendix A present the experiments and reasoning which led us to the following conclusions.

First, Ensemble Louvain shows major community stability improvements and comparable community quality results. With this, we achieve our main goal behind the idea to develop the method, since stable performance makes it suitable for community evolution analysis. To prove this, we apply extensive benchmarking, comparing Ensemble Louvain with the original Louvain and other ensemble methods. The benchmarking results, both on artificial and real-world networks, are covered in Section 2.2. To demonstrate the usefulness of Ensemble Louvain regarding our main motivation, Appendix A.4 explores the practical implications of using the method for community evolution analysis. Here, we explore the stability of results when using Louvain vs. Ensemble Louvain on a timeline of 185 retweet networks of the Slovenian tweetosphere. We present evaluations that show that Ensemble Louvain introduces five times less noise in the results, which helps reduce the influence of stochasticity in the interpretation of community evolution results.

Second, although Ensemble Louvain has two hyperparameters to be set, they show low sensitivity yet provide robustly good results across a well-defined range of values. In other words, high-quality partitions are achievable even with little to no hyperparameter tuning. Using a large variety of artificial networks, we identify a co-membership threshold of $ct = 0.85$ and $r = 100$ number of Louvain runs to be a good starting choice of parameters that works across all networks. We consistently use it across all our experiments in Section 2.2. Appendix A.2 presents the hyperparameter exploration and the results behind the decision on the default parameters.

Third, as a side effect of the co-membership thresholding, Ensemble Louvain is able to discover borderline nodes. These borderline nodes lie between communities or at the periphery of the network. They have a co-membership value below the threshold for all their neighbours, thus turning out to be single-node components. Community analysis can benefit from detecting these nodes, not just to stabilize the discovered communities, but also to potentially analyze the "swingers" or the peripheral nodes in the network. The borderline nodes can be very helpful in social network analysis, where one tries to understand who and why is on the borderline between two communities with conflicting views, agendas, ideologies, etc. Or, regarding the peripheral nodes, which are the communities that act as news feeders to passive readers on social media. In Appendix A.1, we present an experiment on the presence of the borderline nodes, as well as their effects both on the analyzed network and the discovered communities.

Fourth, Ensemble Louvain is, to the best of our knowledge, the only ensemble method that has a parallelized implementation, utilizing multiple CPU cores. The independent Louvain runs are processed on separate cores and then collected in the co-membership matrix. The parallelization significantly helps to mitigate the time complexity of running Louvain multiple times, as we show that the introduction of additional CPU cores speeds up the computation close to linearly. The experiments on the effects of parallelized Ensemble Louvain are presented in Appendix A.3.

Last but not least, Ensemble Louvain has an openly accessible and easy-to-use implementation for Python. It is available on Github[1], as well as through the *pip* Python package installer. In its implementation, it utilizes the famous NetworkX library for handling network structures[2], as well as Python-Louvain[3], an implementation of the Louvain algorithm.

[1] https://github.com/boevkoski/ensemble-louvain
[2] https://networkx.org/
[3] https://github.com/taynaud/python-louvain

2.2 Community Detection Evaluation

The most common approach to evaluating and comparing community detection algorithms is using networks with an *a priori* known community structure. Lancichinetti—Fortunato—Radicchi (LFR) benchmark [32] is the most widely accepted algorithm that generates benchmark networks (artificial networks that resemble real-world networks). Its advantage over other methods is that it accounts for the heterogeneity in the distributions of node degrees and of community sizes.

The LFR method creates synthetic networks using a few important parameters which control the structure of the network. These are the number of nodes in the network; the power law exponent for the degree distribution of the network and communities; the fraction of inter-community edges of each node; the maximum and minimum node degree; the maximum and minimum community size. The common pipeline of using the LFR networks for benchmarking [18], [33]–[36] is to only vary μ, the mixing parameter, as a criterion of difficulty. Higher μ means a higher percentage of edges going out of the community, hence the task of finding the ground-truth communities becomes less trivial. Meanwhile, all other parameters remain fixed, even though their values can significantly alter the community structure in the network [32]. We refer to this methodology as *Standard LFR benchmark* and introduce a more diverse benchmark alternative, which we refer to as *Unconstrained LFR benchmark*.

We apply stability and performance benchmarks on the LFR networks, comparing Ensemble Louvain to the standard Louvain, Consensus Clustering, and ECG. We first focus on and present the stability results, as the aim of our algorithm is to primarily improve consistency while maintaining performance. For both Consensus Clustering and ECG, we use Louvain as the backbone algorithm and the recommended parameter values by the authors in the corresponding papers.

Originating from cluster analysis, we use three stability and performance metrics that allow the comparison of two partitions: Normalized Mutual Information (NMI) [37], Adjusted Rand Index (ARI) [38] and the *BCubed* F_1 metric [39], [40]. Detailed definitions of the metrics and how each is applied for measuring stability and performance can be found in Appendix B.

2.2.1 Standard LFR

The main goal of developing Ensemble Louvain was to achieve stable results, suitable for environments where consistency is key, such as tracking communities through time. And, the task of producing stable results is where the proposed algorithm excels. We measure stability in the following manner: First, we apply the community detection algorithm multiple times on a network, but with different seeds (node ordering). Due to the greedy optimization, these multiple runs produce different partitions. We then use the partition similarity metrics to compare each partition pair separately. Finally, if the average similarity is higher for one algorithm than another, we say that the first one is more stable, as it manages to produce more consistent results. Note that higher stability does not correlate with higher community quality.

For the purpose of evaluating stability in our case, we generate Standard LFR networks for three different network sizes: 100, 1000, and 10000, while varying parameterμ. For each network, we run the selected algorithms 10 times and then compare each pair of runs with the defined metrics. A higher metric score (1.0 being maximum) means that the partitions of the different runs are similar, or the algorithm produces more stable results. A lower value (0.0 being minimum) indicates higher instability.

Figure 2.2 shows the results of the Standard LFR stability benchmark. For all three

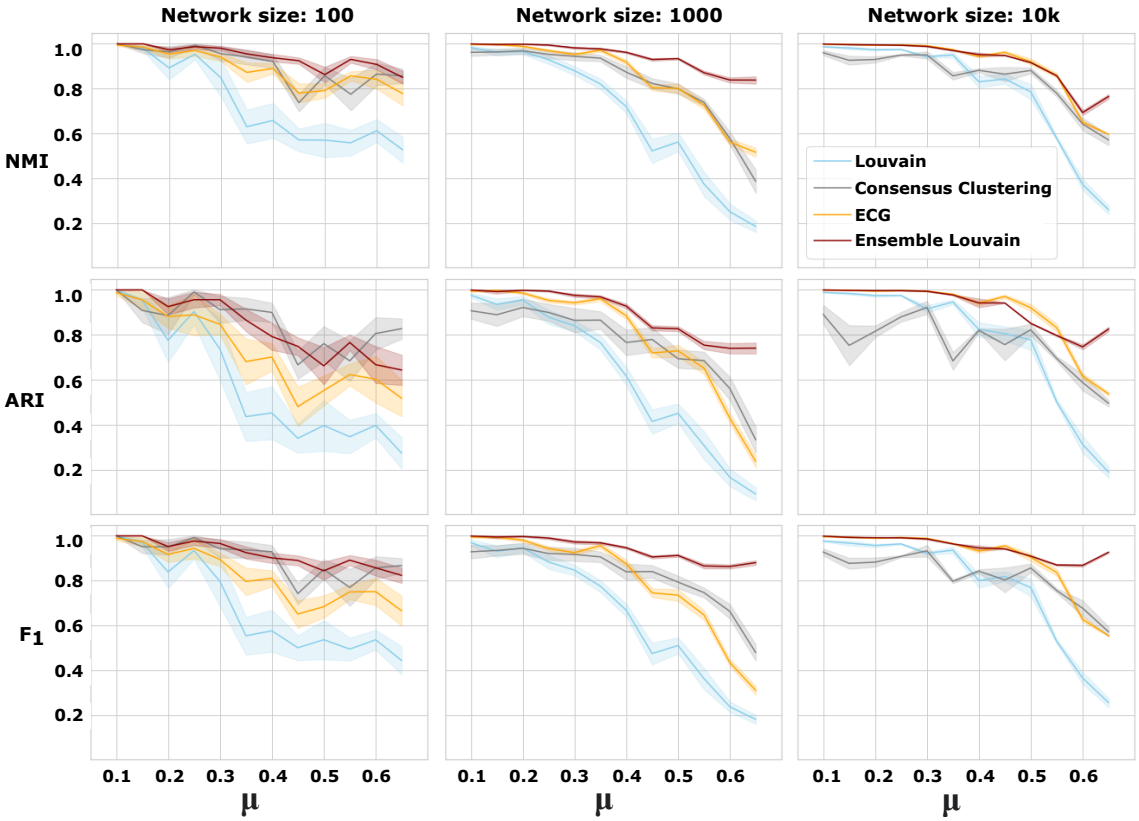

Figure 2.2: **Standard LFR stability benchmark.** Stability results, with a standard LFR setting, consist of a pair-wise comparison of multiple local optimum partitions for a community detection algorithm on the same network. In our case, we show the average pair-wise similarity (stability) and standard error of the mean of ten runs. We apply this on LFR networks with varying μ values (from 0.1 to 0.65, showed on the x-axis), and for three network sizes ($n = 100$, $n = 1000$ and $n = 10000$, with $\tau_1 = 2$ and $\tau_2 = 1.1$). The stability is estimated by three measures (y-axis): *NMI*, *ARI* and F_1. Higher scores imply more similar partitions, or more consistent results through the multiple runs, which mean higher stability.

measures and three network sizes, we observe a similar pattern—Ensemble Louvain shows to be the most stable approach. This is most evident when observing the *NMI* since it does not heavily punish the placing of the borderline nodes into separate communities. As expected, the original Louvain produces the most unstable results, showing the original need for ensemble techniques.

We proceed with a performance benchmark by comparing the algorithm outputs with the ground truth partitions given by the Standard LFR networks. The results are shown in Figure 2.3. All four algorithms produce comparable community quality for all metrics and network sizes, with a slight advantage of Ensemble Louvain and ECG, especially when observing higher values of μ. The only major difference we observe is the lower modularity score of Ensemble Louvain when the value of μ is high. In Subsection A.1, we show that this is due to the effect of the borderline nodes.

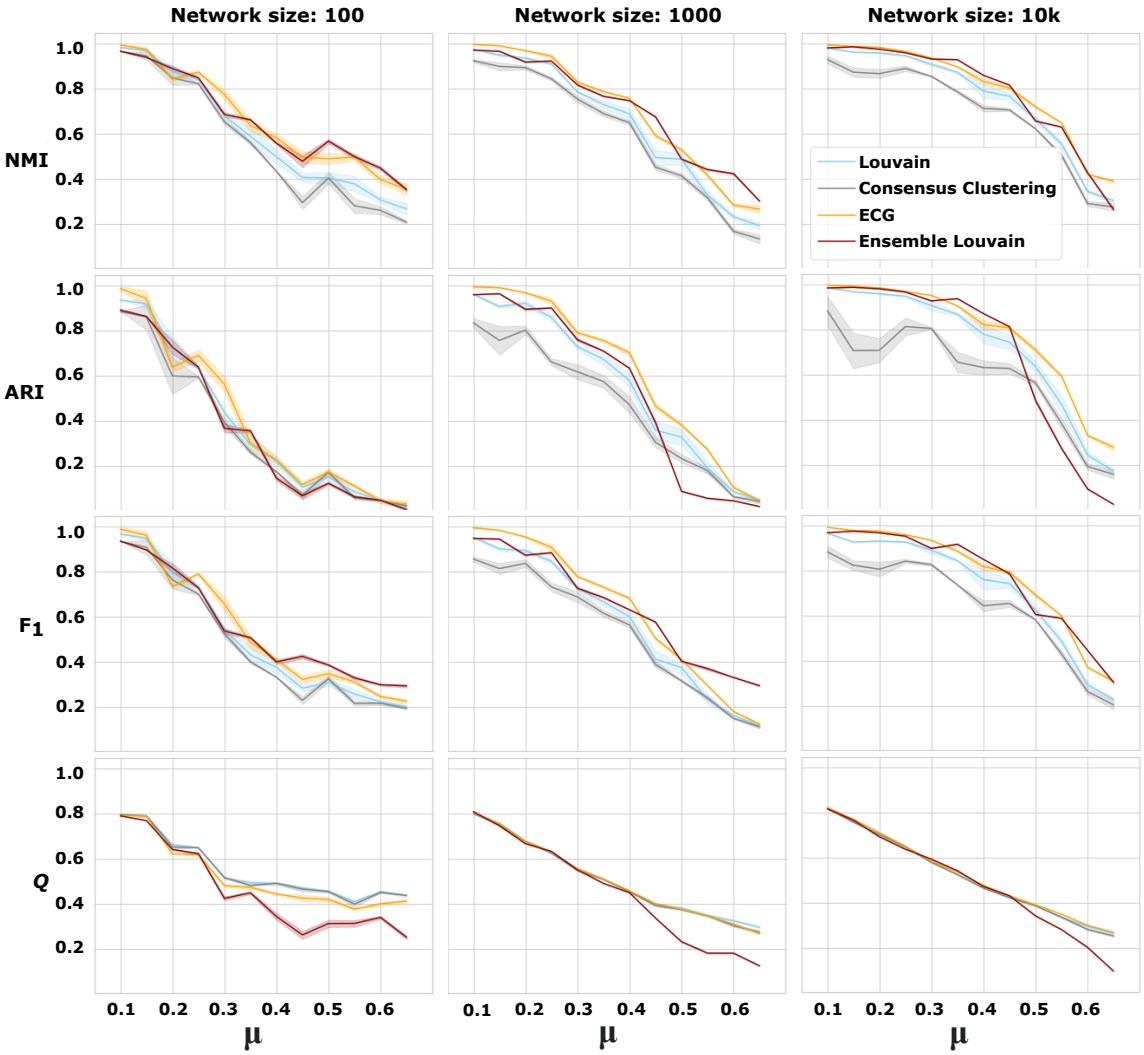

Figure 2.3: **Standard LFR performance benchmark.** Performance results on standard LFR setting, showing average scores of ten runs when compared to the LFR ground truth by varying μ from 0.1 to 0.65 (x-axis) for three network sizes ($n = 100$, $n = 1000$ and $n = 10000$, with $\tau_1 = 2$ and $\tau_2 = 1.1$). The matching of ground truth is estimated by three measures (y-axis): *NMI*, *ARI* and F_1. Higher scores imply a better match of the detected communities to the ground truth of the LFR networks. This figure shows the competence of Ensemble Louvain's community quality compared to the three different methods.

2.2.2 Unconstrained LFR

For the typical Standard LFR comparison methodology, the diversity of the LFR networks is very limited. We argue that the performance of community detection algorithms may vary depending on other network properties, i.e., some algorithms perform better on one set of LFR parameters and other algorithms perform better on others. Consequently, conclusions based on only one set of LFR parameters while only varying the mixing parameter μ can be misleading.

We propose the Unconstrained LFR to perform a more comprehensive benchmarking of community detection algorithms while avoiding the shortcomings of the standard LFR benchmarking. The approach consists of two steps: generating diverse LFR networks and then benchmarking by applying the Friedman test and the post-hoc Nemenyi test. In this

way, the full diversity of the LFR network space can be explored and the potential bias from a single set of LFR parameters is avoided.

2.2.2.1 Network creation

For the network creation part, we randomly generate values for the following parameters: n—number of nodes in the network (from n_{min} to n_{max} nodes); τ_1—power law exponent for the degree distribution of the network (from τ_{1min} to τ_{1max}); τ_2—power law exponent for the community size distribution in the network (from τ_{2min} to τ_{2max}); μ—fraction of inter-community edges of each node (from μ_{min} to μ_{max}); d_{max}—maximum degree allowed for a node (from $\sqrt{n}$ to $n/2$); d_{avg}—average degree of nodes (from $d_{avg,min}$ to $d_{avg,max}$); (c_{min}, c_{max})—minimum and maximum size of a community ($1 < c_{min}$ and $d_{max} < c_{max}$; $c_{min} \in [1, \sqrt{n}]$ and $c_{max} \in [d_{max}, n/2]$). If the combination of parameters fails to generate a valid network, the process is repeated until a valid combination is found.

Once the network creation part is complete, we measure the performance and stability of the community detection algorithms using the previously defined metrics. We then compare the scores by the Friedman-Nemenyi test [55]–[57]. We use the Friedman-Nemenyi combination to simultaneously compare several algorithms on many different networks whose performances by *NMI*, *ARI*, and F_1 are not normally distributed. The result is visualized by critical difference diagrams and can be intuitively presented, especially compared to the Standard LFR where statistical significance is hard to evaluate and interpret. In the ranking diagram, the algorithms are ordered from best performing (on the right) to the worst-performing (on the left). The performance of a pair of algorithms is significantly different if the corresponding average ranks differ by at least the critical difference. Groups of algorithms that are not significantly different are connected by a black CD line. If an algorithm is within one CD of all other algorithms, the correct interpretation is that the experimental data is not sufficient to reach any conclusion regarding this algorithm.

2.2.2.2 The Unconstrained LFR benchmark

As a pilot study, we apply the Unconstrained Friedman-Nemenyi LFR benchmark to our selected four community detection algorithms. We generate 500 ($N = 500$) unconstrained LFR networks using the NetworkX [58] library with the following parameter settings:

- Number of nodes range, $n \in [100, 12500]$,
- Power law exponent for the degree distribution range, $\tau_1 \in [1.1, 3.0]$,
- Power law exponent for the community size distribution range, $\tau_2 \in [1.05, \tau_1]$,
- Fraction of inter-community edges of each node range, $\mu_{min} \in [0.05, 0.70]$,
- Maximum node degree range, $d_{max} \in [\sqrt{n}, n/2]$,
- Average node degree range, $d_{avg} \in [3, 25]$,
- Maximum community size, $c_{max} \in [d_{max} + 1, n/2]$,
- Minimum community size, $c_{min} \in [2, \sqrt{n}]$.

Note that the range for the selection of parameters is only a recommendation based on preliminary experiments. They are chosen so as to most likely yield a viable combination for a network to be generated while preserving varying network and community structures.

We apply the community detection algorithms 10 times on each of the 500 LFR networks and calculate the *NMI*, *ARI* and F_1 measures on their partitions. For stability, we

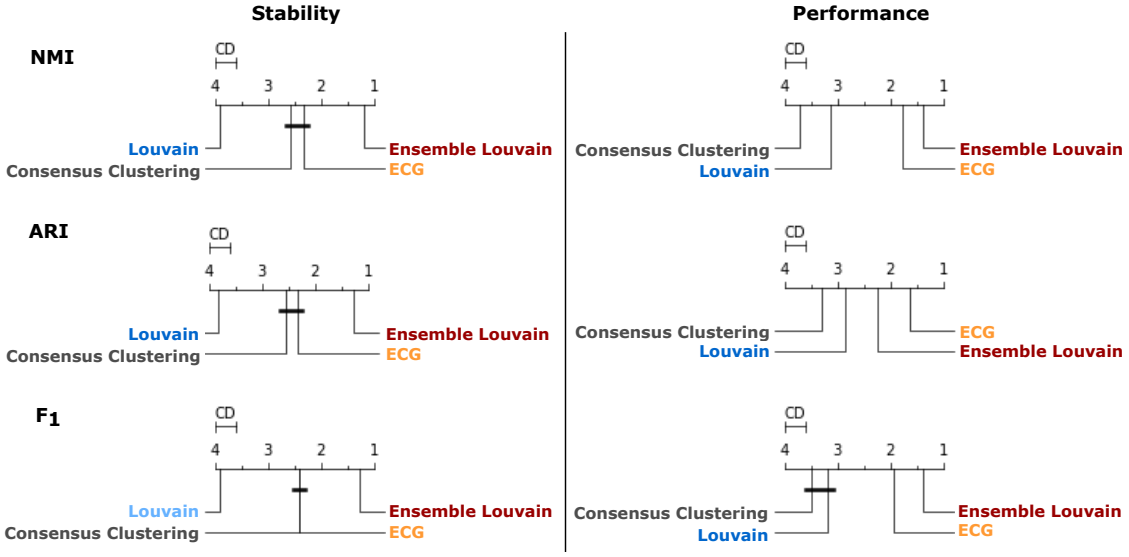

Figure 2.4: **Unconstrained LFR benchmark.** We compare four algorithms on 500 unconstrained LFR benchmark networks, applying the Friedman-Nemenyi significance test. The left-hand side shows the stability results, and the right-hand side the matching to ground truth results. Each individual chart shows the ranks of the four algorithms as estimated by one of the evaluation measures (*NMI*, *ARI*, and F_1). CD denotes the critical difference, and the black bars connect ranks of procedures that are not significantly different at the 5% level.

calculate the partition similarity between pairs of multiple runs of the same algorithm. For performance, we compare the ten runs of each algorithm to the LFR ground truth. The scores are the input to the Friedman-Nemenyi combined test using the Autorank library in Python [59], where we generate rankings of the four algorithms, separately for stability and performance. The results of this experiment are presented in Figure 2.4. A clear distinction in stability and performance can be observed between the algorithms when their ranks differ more than the critical distance (CD). Confirming the stability results of the Standard LFR, yet this time statistically, Ensemble Louvain shows the most stable results by all three metrics. Regarding performance (or community quality), the benchmark shows that Ensemble Louvain shows the best results according to *NMI* and F_1, while it goes to second place according to *ARI*. Again, as *ARI* evaluates the correctness of node placements by comparing pair combinations, it heavily punishes the borderline nodes in Ensemble Louvain, thus evaluating ECG as the most accurate algorithm.

Finally, in Appendix C we present additional experiments which show why we argue that the Unconstrained LFR is a valuable addition to the community detection evaluation tools. Basically, we demonstrate that varying only μ leads to an underexplored space of possible networks and that the performance of community detection algorithms may vary depending on other network properties which stay fixed during a Standard LFR benchmark.

2.2.3 Evaluation on real-world networks

Synthetic networks are a reliable way of comparing community detection methods with respect to ground truth since one knows the network-generating process, or in other words, can justify nodes being in a particular community. On the other hand, for real-world networks, one takes network metadata as ground truth (e.g., country, age, political views, etc. of a user). This metadata cannot guarantee that it is equivalent to the unobserved

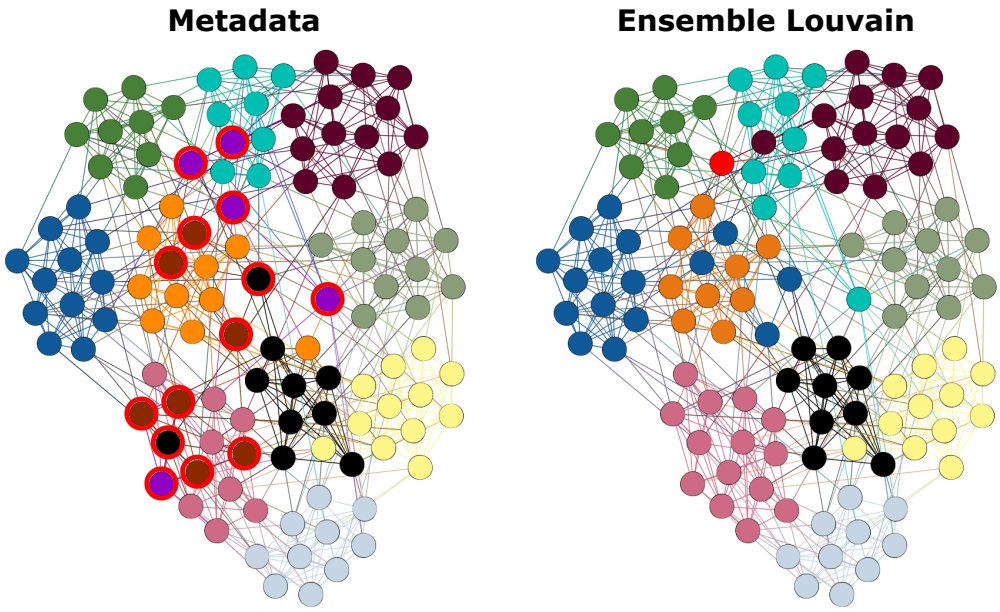

Figure 2.5: **The Football network: Metadata vs. Ensemble Louvain**. Network layout is done using the Force Atlas 2 visualization in Gephi [64]. Node colors represent the metadata on the left and the discovered communities by Ensemble Louvain on the right. Red circles mark nodes that are differently assigned by Ensemble Louvain compared to the metadata.

ground truth. Hence, if one compares the metadata with the detected partitions, there is a simultaneous test of the metadata relevance and the algorithm performance, with no ability to differentiate between the two [60]. This does not imply that efforts of finding the best community detection algorithm on real-world networks are in vain and that metadata should not be used as a benchmark for evaluating or comparing the efficacy of community detection algorithms. Although searching for better community detection results on imperfect metadata may stray from a better understanding of the actual community structure, it could lead to identifying classes of algorithms whose strengths are aligned with the requirements of a specific network category. Here, we analyze the community structure and performance of our algorithm on three real-world networks with metadata: College football [61], Researcher e-mails [62], A European Parliament retweet network [63].

Figure 2.5 shows the metadata and Ensemble Louvain partitions of the College football network [61]. It represents the schedule of United States football games between Division IA colleges during the regular season in the Fall of 2000. Nodes represent teams, while links represent regular season games between the two teams connected. The metadata communities are defined by conferences, each containing around 8 to 12 teams and marked with colors. In principle, teams from one conference are more likely to play games with each other than with teams belonging to different conferences. In general, Ensemble Louvain correctly clusters teams from one conference. Most of the deviation from the conference segmentation comes from the independents, a cluster formed by teams who do not belong to a particular conference, yet frequently play with teams from different conferences.

Full results on all three real-world networks regarding stability and performance are shown in Table A.1. We measure the mean and standard error of the mean of the F_1 score of performance and stability experiments on ten runs. Ensemble Louvain produces the most stable results for all three networks while showing competitive F_1 scores on the

Table 2.1: A comparison of the stability and performance of the four algorithms on three real-world networks. The stability is computed from pairwise comparisons of the detected network partitions over ten runs. The performance is estimated by comparing the detected communities with the metadata of each network over ten runs, showing the mean and its standard error.

Stability (F_1)	Football	Email	EU Parliament
Louvain	0.969 ± 0.027	0.834 ± 0.051	0.951 ± 0.029
Consensus Clustering	0.986 ± 0.021	$\mathbf{0.981 \pm 0.005}$	$\mathbf{0.988 \pm 0.008}$
ECG	0.987 ± 0.012	0.897 ± 0.039	0.953 ± 0.020
Ensemble Louvain	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.980 \pm 0.005}$	$\mathbf{0.995 \pm 0.005}$
Performance (F_1)			
Louvain	0.836 ± 0.018	0.450 ± 0.015	$\mathbf{0.680 \pm 0.014}$
Consensus Clustering	0.808 ± 0.014	0.388 ± 0.002	$\mathbf{0.693 \pm 0.007}$
ECG	$\mathbf{0.894 \pm 0.005}$	$\mathbf{0.536 \pm 0.011}$	0.599 ± 0.012
Ensemble Louvain	0.852 ± 0.000	0.492 ± 0.001	0.674 ± 0.000

metadata. The standard error of the mean for both performance and stability is drastically in favor of Ensemble Louvain, showing the clear consistency of its outputs.

Chapter 3

Applications

Our methodological contributions to community evolution found great use in a trilogy of papers focused on analyzing the Slovenian tweetosphere. In these works, we built a timeline of snapshot retweet networks and analyzed their community evolution from multiple aspects, such as tracking community influencers, hate speech sources, topics, and more.

3.1 Community Evolution in Retweet Networks

“Community evolution in retweet networks" is a joint work by Bojan Evkoski, Igor Mozetič, Nikola Ljubešič, and Petra Kralj Novak. It was published in PLOS One[1], a peer-reviewed open access scientific journal published by the Public Library of Science, in July 2021.

In this paper, we propose an approach that tracks two aspects of community evolution in retweet networks. First is the flow of members in, out, and between the major communities. Second, the influence of members in and out of the communities, as well as the influence of both members and communities in and out of identified super communities. The analysis is applied to the Slovenian tweetosphere, to tweets starting from January 2018 to January 2021 and it is the earliest community evolution analysis that utilizes Ensemble Louvain, the community detection method described in this master thesis.

Methodologically, the paper consists of several contributions relevant to the field of community evolution. It proposes a strategy for representing community evolution that takes static network snapshots in a timeline of 24-week networks using a sliding window. On every following snapshot, it applies an exponential edge weight decay, removing the effect of the trailing end of the window and making the choice of the window size less relevant. The chosen sliding window is one week, providing high temporal resolution. Thus, it proposes a temporal zoom-out to a lower time resolution, by a computationally efficient selection of more distant time points where the network partitions exhibit larger differences. Finally, it extends and applies the BCubed F_1 measure of community similarity, which was originally introduced to evaluate the quality of document clustering but does not appear to be used in the field of complex networks. The F_1 score can measure differences between network communities with only partially overlapping sets of nodes, which is essential for comparing retweet networks in community evolution, where new nodes keep appearing and disappearing from the network snapshots.

The analysis shows that the Slovenian tweetosphere is dominated by politics and that the left-leaning communities are consistently larger, but the right-leaning communities and users exhibit a significantly higher impact. Despite events such as the emergence of the Covid-19 pandemic and the change of government from left-leaning to right-leaning, the

[1] https://journals.plos.org/plosone/

retweet networks change relatively gradually, while the behavior patterns of the left-leaning and right-leaning communities and users remain consistent.

The author of the master thesis contributed to this paper in the conceptualization of the paper, the methodology, and the writing. He was also responsible for the complete software implementation, the experiments, and the visualization of results.

PLOS ONE

RESEARCH ARTICLE

Community evolution in retweet networks

Bojan Evkoski[1,2], **Igor Mozetič**[1], **Nikola Ljubešić**[1], **Petra Kralj Novak**[1]

1 Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia, **2** Jozef Stefan International Postgraduate School, Ljubljana, Slovenia

* igor.mozetic@ijs.si

OPEN ACCESS

Citation: Evkoski B, Mozetič I, Ljubešić N, Kralj Novak P (2021) Community evolution in retweet networks. PLoS ONE 16(9): e0256175. https://doi.org/10.1371/journal.pone.0256175

Editor: Chantal Cherifi, Universite Lumiere Lyon 2, FRANCE

Received: March 31, 2021

Accepted: July 22, 2021

Published: September 1, 2021

Data Availability Statement: All Twitter data were collected through the public Twitter API and are subject to Twitter terms and conditions. The Slovenian Twitter dataset 2018–2020 is available at a public language resource repository clarin.si at https://hdl.handle.net/11356/1423. The code used to implement the Ensemble Louvain algorithm described in the paper is available at the Github repository https://github.com/boevkoski/ensemble-louvain.git.

Funding: The authors acknowledge financial support from the Slovenian Research Agency (research core funding no. P2-103 and P6-0411),

Abstract

Communities in social networks often reflect close social ties between their members and their evolution through time. We propose an approach that tracks two aspects of community evolution in retweet networks: flow of the members in, out and between the communities, and their influence. We start with high resolution time windows, and then select several time-points which exhibit large differences between the communities. For community detection, we propose a two-stage approach. In the first stage, we apply an enhanced Louvain algorithm, called Ensemble Louvain, to find stable communities. In the second stage, we form influence links between these communities, and identify linked super-communities. For the detected communities, we compute internal and external influence, and for individual users, the retweet h-index influence. We apply the proposed approach to three years of Twitter data of all Slovenian tweets. The analysis shows that the Slovenian tweetosphere is dominated by politics, that the left-leaning communities are larger, but that the right-leaning communities and users exhibit significantly higher impact. An interesting observation is that retweet networks change relatively gradually, despite such events as the emergence of the Covid-19 pandemic or the change of government.

Introduction

With the ever-growing base of social media users, platforms such as Twitter are becoming a very valuable source of data for social analysis. Users on social media interact with each other, so it is natural to use graphs (where the users are nodes, and interaction between them are edges) to represent the structure of the user base. Nowadays, a lot of research in the field of complex networks is focused on social networks analysis. Due to the social media volatility, temporal analyses are needed for an in-depth understanding of the underlying phenomena. They can provide insights into the patterns and evolution of the social media landscape, and consequently to the society itself.

Change in the collective behaviour of groups in networks is referred to as community evolution [1], where communities in the networks are defined as groups of densely connected users. However, community detection methods are typically designed for static networks, and consequently have to be adapted for detecting changes in dynamic social media networks.

In our approach, we proceed by creating overlapping snapshots of the network through time, and detect communities in each snapshot. We then track evolution of relevant

the Slovenian Research Agency and the Flemish Research Foundation bilateral research project LiLaH (grant no. ARRS-N6-0099 and FWO-G070619N), and the European Union's Rights, Equality and Citizenship Programme (2014-2020) project IMSyPP (grant no. 875263). The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Competing interests: The authors have declared that no competing interests exist.

communities over time. Several developments are needed to detect community evolution in terms of the flow of members in, out and between the communities, as well as to track the changes in the community influence.

We illustrate our approach to community evolution on a set of Slovenian tweets during the last three years, roughly 13 million Twitter posts. Our initial research, where we performed a static community structure analysis of the data showed strong polarization of the detected communities along the political dimension. In the subsequent research, the basis of the current paper, we compared community structures between different manually selected time windows [2]. In the current paper, we describe a general set of techniques that enable semi-automatic analysis of the evolution of community structures and influence. These techniques make static community detection algorithms applicable to dynamic networks. We show a step-by-step application and insightful results of the proposed techniques on the Slovenian retweet networks.

Related work

The temporal dimension is very valuable in modern analyses of complex networks. This has implications on how dynamic community discovery is designed and applied.

The related approaches mostly depend on the representation of time. One can group them into three types: static/edge-weighted, snapshots, and temporal networks [3]. The first community discovery methods were applied to the so-called "frozen in time" networks, where the temporal dimension is not explicitly represented. One operates with a single network (static or edge-weighted) that aggregates the whole period of interest. This absence of the time dimension has two historical reasons: the graph theoretic origin of the field, and the scarcity of data at the time when the field of complex networks emerged [4]. Aggregation strategies have severe limitations as they cannot capture dynamics, hence are not suitable for dynamic community detection. Consequently, the second representation emerged—temporally ordered series of network snapshots. This approach allows for efficient tracking of changes in the network structure, thus increasing the expressiveness of the models, but at a cost of higher analytical complexity [3]. Finally, temporal networks were proposed that allow for a complete and fine-grained description of the network dynamics [5]. The field of temporal network analysis is still under active development. Explicit temporal network representation is rarely used for dynamic community discovery, as it considerably increases the complexity of the models, and cannot easily make use of the existing community detection algorithms.

The snapshot approach crucially depends on the representation of time in the network relations. Two main scenarios were considered so far: perfect memory networks (also known as accumulative growth scenario), and limited memory networks. Perfect memory permits only aggregation of nodes and edges, where the old nodes/edges cannot disappear. The limited memory scenario allows for nodes/edges to disappear over time. This is suitable in social network analysis, where the edge disappearance could indicate the decay of social ties. The limited memory networks are implemented with various methods, including static, sliding, or dynamic-sized time windows, each method with its own strengths and weaknesses. In subsection Retweet networks we propose a combined method that circumvents the drawbacks of the existing strategies by building a weight-decaying sliding time window network. We then apply a snapshot selection method, described in subsection Selection of timepoints, where fixed static time windows that contain the most information about the dynamics of the communities are selected.

The second significant factor in dynamic community evolution is the way community detection is applied to the network snapshots [1, 3, 6–8]. Most of the existing approaches

consider the following question: How do detected communities from one snapshot affect other snapshots (usually future-adjacent)? There are three groups of approaches: non-evolutionary, evolutionary, and coupling. The first one, also known as instant-optimal or two-stage approach, considers that communities already existing at time *t* depend only on the current state of the network at time *t*. A two-stage approach first detects communities at each snapshot, and then matches the detected communities [9, 10]. The obvious drawback of this approach is that the knowledge gained about the communities at snapshot *t*-1 is not used for communities at snapshot *t*. Yet, our method shows that this is not necessarily a weakness, when one is interested in detecting maximal changes in the community structure. In the evolutionary approach, also known as temporal trade-off, communities at snapshot *t* do not only depend on the network at the same time *t*, but also on the past evolution of the network [11–13]. The coupling approach shifts the focus from detecting communities at snapshot *t*, to community detection considering pairs of adjacent snapshots, or even the whole network evolution [14, 15].

Although there is a plethora of approaches, with all their advantages and drawbacks, most of the methods suffer from a common issue—the instability of community detection algorithms [16]. Community detection algorithms have different weaknesses, but the instability of the results is their common issue in the temporal scenarios. This is specially problematic in the evolutionary approach to dynamic community detection since the local instability also affects the time dependent communities. In other words, a "bad" run of the community detection algorithm influences the results of detection at the subsequent snapshots. This instability is also an issue for the community evolution analysis in our work, as one cannot distinguish if the community differences are due to the real-world events reflected in the dynamic complex network, or are they simply a consequence of the instability of the algorithm. To address this issue, we propose an Ensemble Louvain algorithm which to some extend solves the instability of the well-known Louvain algorithm for community detection.

Structure of the paper

The main body of the paper is in the Results section. We start with a brief overview of the data collected in the Structure of the Twitter data subsection. In the Retweet networks subsection we describe how the network snapshots are created. Network partitions, generated by an extension of the Louvain algorithm, are described in the Community detection subsection. Evolving communities in adjacent partitions are compared in the Structure of the Measuring community similarity subsection. In Selection of timepoints we show how to select just a few relevant timepoints out of the whole timeline sequence. Two types of transitions are depicted in the Visualization of community transitions subsection. We then define internal and external influence in the Structure of the Identification of super-communities subsection. In the last subsection Retweet h-index influence we show the most influential users in our dataset. In Conclusions we wrap up our approach to community evolution and present main plans for future research. The Methods section provides some additional details. The Data collection subsection describes a specialized tool used for Twitter acquisition. Ensemble Louvain gives details of the community detection algorithm applied, and some preliminary evaluation results. The last subsection on BCubed measure of community similarity defines the measures used throughout the paper.

Results

Twitter data

Social media, and Twitter in particular, are widely used to study various social phenomena. For this study, we collected a set of all Slovenian tweets in the three year period, from January

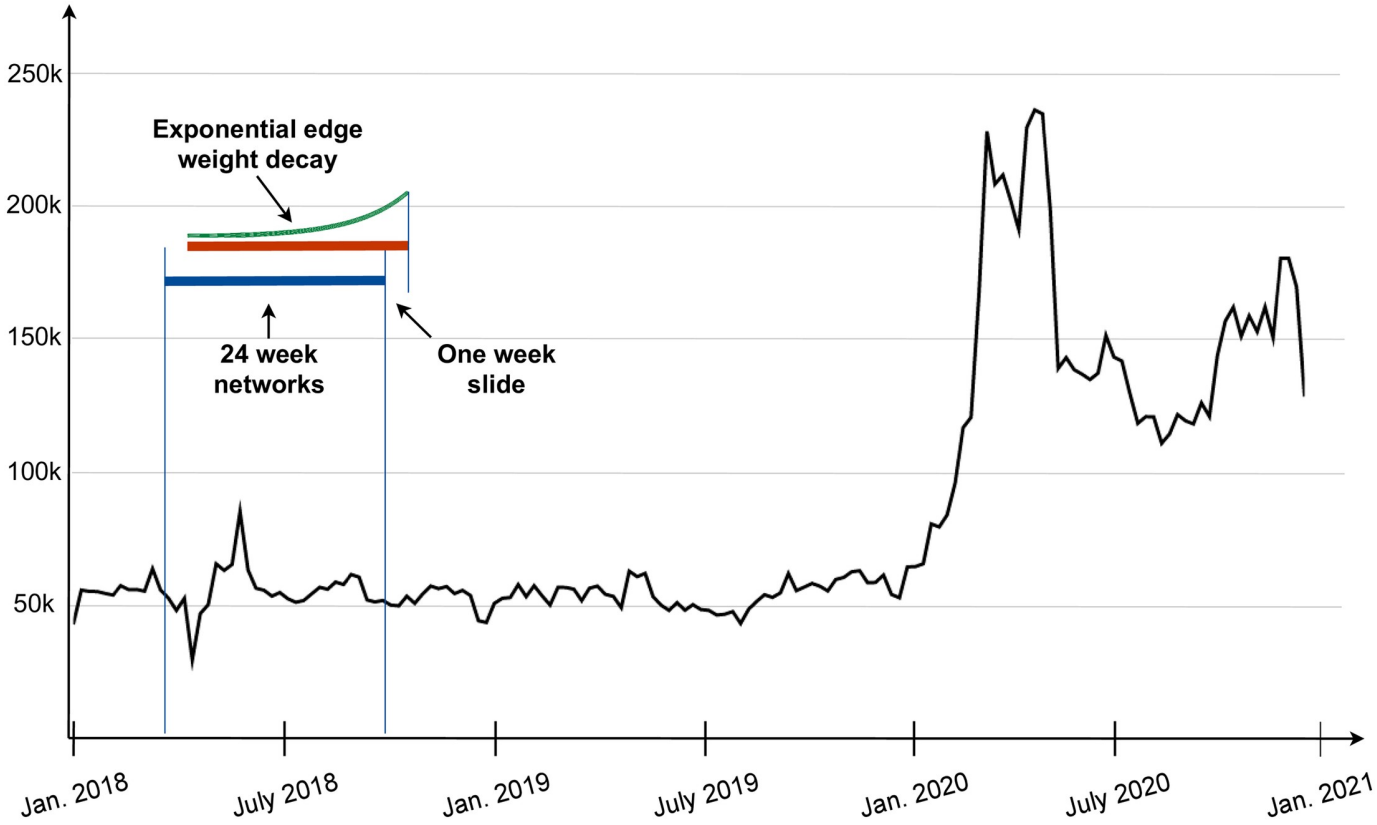

Fig 1. Weekly volume of Slovenian Twitter data, collected over the three year period. The retweet network observation window is 24 weeks (blue and red lines), with exponential weight decay (half-time of 4 weeks, green curve), and one week sliding window (difference between the red and blue line). Note a large increase of Twitter activities at the emergence of the Covid-19 pandemic, which also coincided with the change of the left-wing to the right-wing government in Slovenia.

https://doi.org/10.1371/journal.pone.0256175.g001

1, 2018 until December 28, 2020. The set of almost 13 million tweets represents an exhaustive collection of Twitter activities in Slovenia. See the Methods section for details of the Twitter data acquisition.

Fig 1 shows the weekly volume of tweets collected during the three years. The number of tweets is fairly stable, around 50,000 per week, until the emergence of the Covid-19 pandemic in March 2020. At this point, we observe a four-fold increase of Twitter activities. This also coincides with a change of government in Slovenia, from the left-wing to the right-wing. A minor peak can also be observed around June 2018, at the time of the snap parliamentary elections. It turns out that most of the tweets are related to politics, and, after March 2020, to policies concerning the handling of the pandemic. The following is a list of the most important political events in Slovenia during the last three years:

- March 14, 2018—left-wing government resignation ($PM-sep14-sep18),
- June 8, 2018—snap parliamentary elections,
- September 13, 2018—new left-wing government formation ($PM-sep18-mar20),
- January 27, 2020—left-wing government resignation ($PM-sep18-mar20),
- March 13, 2020—right-wing government formation ($PM-mar20-now),

- March, 2020—emergence of the Covid-19 pandemic in Slovenia.

In parenthesis we give anonymized Twitter handles of the Slovenian prime ministers (PM) at the time since they have important roles in their respective communities. PLoS ONE policy requires to remove information which identifies and names individual Twitter users.

Retweet networks

Twitter provides different forms of interactions between the users: follows, mentions, replies, and retweets. The most useful indicator of social ties between the Twitter users are retweets. When a user retweets a post, it is distributed to all of its followers, just as if it were an originally authored post. Users retweet content that they find interesting or agreeable. Despite the fact that it does not always signify an endorsement (e.g., tweets by the former U.S. president Trump), in large number of cases retweets indicate links between the like-minded users. In particular, in politics retweets very well reflect the actual political alignments and influence. For example, it was demonstrated that political parties and nationalities of the members of the European Parliament can be reconstructed solely from their retweet activities [17]. There is also a correspondence between the co-voting and retweeting in the European Parliament, while higher Twitter activity was observed for the right-wing parties [18]. In the case of Brexit, the Leave proponents showed much higher activity and influence on Twitter than the Remain proponents [19].

A retweet network is a directed graph G. The nodes are Twitter users and the edges are retweet links between the users. An edge is directed from the user A who posts a tweet to the user B who retweets it. The edge weight is the number of retweets posted by A and retweeted by B. For the whole three year period of Slovenian tweets, there are in total 18,821 users (nodes) and 4,597,865 retweets (sum of all weighted edges).

To study dynamics of the retweet networks, we form several network snapshots from our Twitter data. In particular, we select a network observation window of 24 weeks (about six months), with a sliding window of one week. This provides a relatively high temporal resolution between subsequent networks, but later we show how to select the most relevant intermediate timepoints (see subsection Selection of timepoints). Additionally, we employ an exponential edge weight decay, with half-time of 4 weeks (see Fig 1). The reason for this temporal weight decay is to eliminate the effects of the trailing end of the moving network snapshots.

The set of network snapshots thus consists of 133 overlapping observation windows, with temporal delay of one week. The snapshots start with network G_0 (January 1, 2018–June 18, 2018) and end with network G_{132} (July 13, 2020–December 28, 2020).

Community detection

Informally, a network community is a subset of nodes more densely linked between themselves than with the nodes outside the community. There are several formal definitions of communities and different methods to detect them. A practical review that provides strengths and weaknesses of the most popular methods is provided in [20].

A standard community detection method is the Louvain algorithm [21]. Louvain finds a partitioning of the network into communities, such that the modularity of the partition is maximized. For a partition, the modularity measures the density and structure of its communities: the fraction of edges within the communities, as compared to the expected fraction of randomly distributed edges in the network [22]. The Louvain algorithm is computationally

efficient, well suited for large networks, and does not require ex-ante assumptions about the number or size of the communities [23].

However, there are several problems with the modularity maximization [20]. One, from a theoretical point of view, is that there are typically exponentially many distinct partitions whose modularity scores are very close to the global maximum [24]. As a consequence, from a practical point of view, the Louvain algorithm yields different partitions for different trials on the same network (see Fig 7 in Methods for an example).

We address this instability problem of Louvain by applying the Ensemble Louvain algorithm. We run 100 trials of Louvain and compose communities with nodes that co-occur in the same community above a given threshold, 90% of the trials in our case. This results in relatively stable communities of approximately the same size as produced by individual Louvain trials. Details of the Ensemble Louvain algorithm are in the Methods section.

Our 133 retweet network snapshots are directed graphs, $G_0, \ldots, G_{132}$, with weighted edges. For community detection, we transform them into undirected graphs. When a pair of nodes is linked with two weighted edges of the opposite direction, we create an undirected edge with the sum of the original edge weights. When a pair of nodes is linked with a single directed edge, we simply drop the direction. We then run the Ensemble Louvain on all the 133 undirected network snapshots, resulting in 133 network partitions, $P_0, \ldots, P_{132}$.

Measuring community similarity

The sequence of network partitions, $P_0, \ldots, P_{132}$, produced by Ensemble Louvain, varies. Community structure changes, new nodes join some communities, and some nodes disappear from a network snapshot. To study community evolution, one has to compare subsequent network partitions.

There are several measures to evaluate network communities, in particular in relation to the "ground truth". Two widely used measures are Adjusted Rand Index (ARI) [25] and Normalized Mutual Information (NMI) [26]. In this study we use the BCubed measure, extensively evaluated in the context of clustering [27]. BCubed yields evaluation results similar to ARI and NMI (see Fig 7 in Methods). However, there are several advantages of BCubed, useful in the context of community evolution. In particular, we extend the BCubed measure to account for the new and lost nodes between two network partitions.

BCubed decomposes evaluation into calculation of precision and recall of each node in the network. The precision (*Pre*) and recall (*Rec*) are then combined into the F_1 score, the harmonic mean:

$$F_1 = 2\,\frac{Pre \cdot Rec}{Pre + Rec}.$$

Details of computing *Pre* and *Rec* for individual nodes, communities and network partitions are in the Methods section. Here we just emphasize that our extended BCubed F_1 is different and more general than the F_1 score proposed by Rossetti [28].

In the following, we refer to our extended BCubed F_1 score as simply F_1. When we compare two network partitions, P_t and P_{t-1}, we consider a partition earlier in time P_{t-1} as "ground truth", and evaluate the subsequent partition P_t with respect to the previous one. We write $F_1(P_t|P_{t-1})$ to denote the similarity of P_t to P_{t-1}. F_1 ranges from 0 to 1, where increasing F_1 indicates higher similarity between the two partitions.

There are two special cases of F_1. When the two partitions consist of the same nodes, just distributed differently between the communities, F_1 degenerates into *core*-F_1. *core*-F_1 is directly compatible to ARI and NMI. When the two partitions differ in the constituent nodes, i.e.,

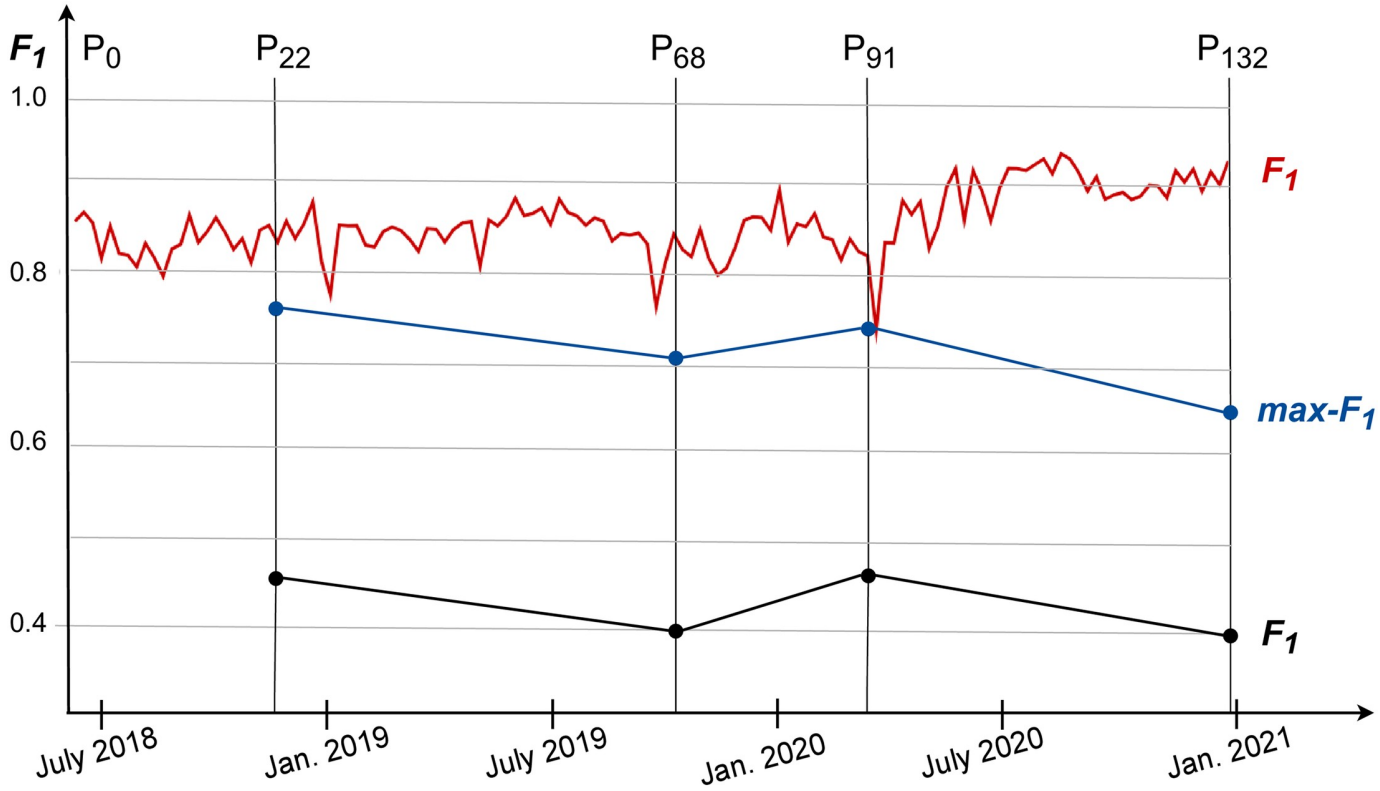

Fig 2. Differences between the adjacent network partitions measured by the F_1 score. The red line at the top shows weekly differences $F_1(P_t|P_{t-1})$ at timepoints $t = 1, 2, \ldots, 132$. The five selected partitions are denoted by $P_0, P_{22}, \ldots, P_{132}$. The middle blue line shows the theoretical maximum $max\text{-}F_1$ differences between distant partitions at the selected timepoints $t = 0, 22, 68, 91, 132$. The bottom black line shows the standard F_1 differences.

https://doi.org/10.1371/journal.pone.0256175.g002

there are new and lost nodes, one can compute the theoretical maximum similarity, $max\text{-}F_1$, where all the nodes common to both partitions (the intersection) are assumed to be in one community. $max\text{-}F_1$ thus measures similarity of two sets and is directly related to the Jaccard index. See Methods for details.

Fig 2 (red line) shows pairwise $F_1(P_t|P_{t-1})$ differences between the retweet network partitions at weekly timepoints $t = 1, 2, \ldots, 132$. The F_1 scores are relatively high, typically in the range [0.8, 0.9]. The largest negative peak, indicating the highest dissimilarity, $F_1(P_{92}|P_{91}) = 0.74$, occurs between the partitions which end on March 16 and 23, 2020, respectively. These dates closely follow the change of government in Slovenia and first policy reactions to the emergence of the Covid-19 pandemic.

Selection of timepoints

The weekly differences between the network partitions are relatively low. The retweet network communities apparently do not change drastically at this relatively high time resolution. Moving to lower time resolution means choosing timepoints which are further apart, and where the network communities exhibit more pronounced differences.

We formulate the timepoint selection task as follows. Let assume that the initial and final timepoints are fixed, corresponding to the partitions P_0 and P_n, respectively. For a given k,

select k intermediate timepoints such that the differences between the corresponding partitions are maximized, i.e., the F_1 scores are minimized:

$$min\left(\sum_{i=1}^{k} F_1(P_i|P_{i-1}) + F_1(P_n|P_k)\right).$$

There are $\binom{n-1}{k} \cdot k!$ possible selections of timepoints, i.e., the number of selections grows exponentially with k. We therefore propose a simple heuristic algorithm which finds k approximate timepoints. The algorithm works top-down and starts with the full, high resolution timeline with $n + 1$ timepoints, $t = 0, 1, \ldots, n$ and corresponding partitions P_t. At each step, it finds a triplet of adjacent partitions P_{t-1}, P_t, P_{t+1} with minimal differences:

$$max\left(F_1(P_t|P_{t-1}) + F_1(P_{t+1}|P_t)\right)$$

and eliminates P_t from the timeline. At the next step, the difference $F_1(P_{t+1}|P_{t-1})$ fills the gap of the eliminated timepoint P_t. The algorithm thus finds the k (non-optimal) timepoints in $n - 1 - k$ steps. While efficient, this approach to the relevant timepoint selection is not suitable for incremental, stream-based network processing since it assumes that the final timepoint is fixed.

For our retweet networks, we experimented with several values of k and eventually settled with $k = 3$ which provides much lower, but still meaningful time resolution. This resulted in the selection of the following network partitions: P_0, P_{22}, P_{68}, P_{91}, P_{132}. Fig 2 shows the F_1 differences (black line) between the adjacent partitions.

The selected timepoints are on average 26 weeks apart, varying between five and ten months. The differences between the network partitions are increasing with temporal distance, but are still relatively uniform, F_1 is in the range [0.4, 0.5]. Due to these small differences, the timepoint selection procedure is not very robust. The selected timepoints should be considered approximate and can vary for several weeks in both directions. As a consequence, the selected timepoints should not be interpreted as indicators of specific events at specific dates, but should rather help in understanding longer terms qualitative transitions in community evolution.

Fig 2 also shows the theoretical maximum differences $max\text{-}F_1$ (blue line), where it is assumed that all the common nodes in two adjacent partitions are in one community, and only the intersection size and the number of new and lost nodes affect the score. The $max\text{-}F_1$ scores, dropping from 0.77 to 0.63, show increasing fluctuation of nodes in and out of the partitions. In the next subsection we show two visualizations of transitions between these five network partitions.

Visualization of community transitions

We present two visualizations of transitions between selected network partitions as Sankey diagrams. A Sankey diagram is a type of flow diagram in which the width of the bands is proportional to the flow rate.

In Fig 3 we show inflows of new nodes, outflows of lost nodes, and transition flows of core (intersection) nodes between the selected network partitions. Note that only about half of the nodes remain in the core transitions. Therefore it is crucial that the community similarity measure takes new and lost nodes into account. This diagram ignores the internal community structure, and corresponds to the theoretical maximum $max\text{-}F_1$ (shown in Fig 2) where all the core nodes are assumed to be in the same community.

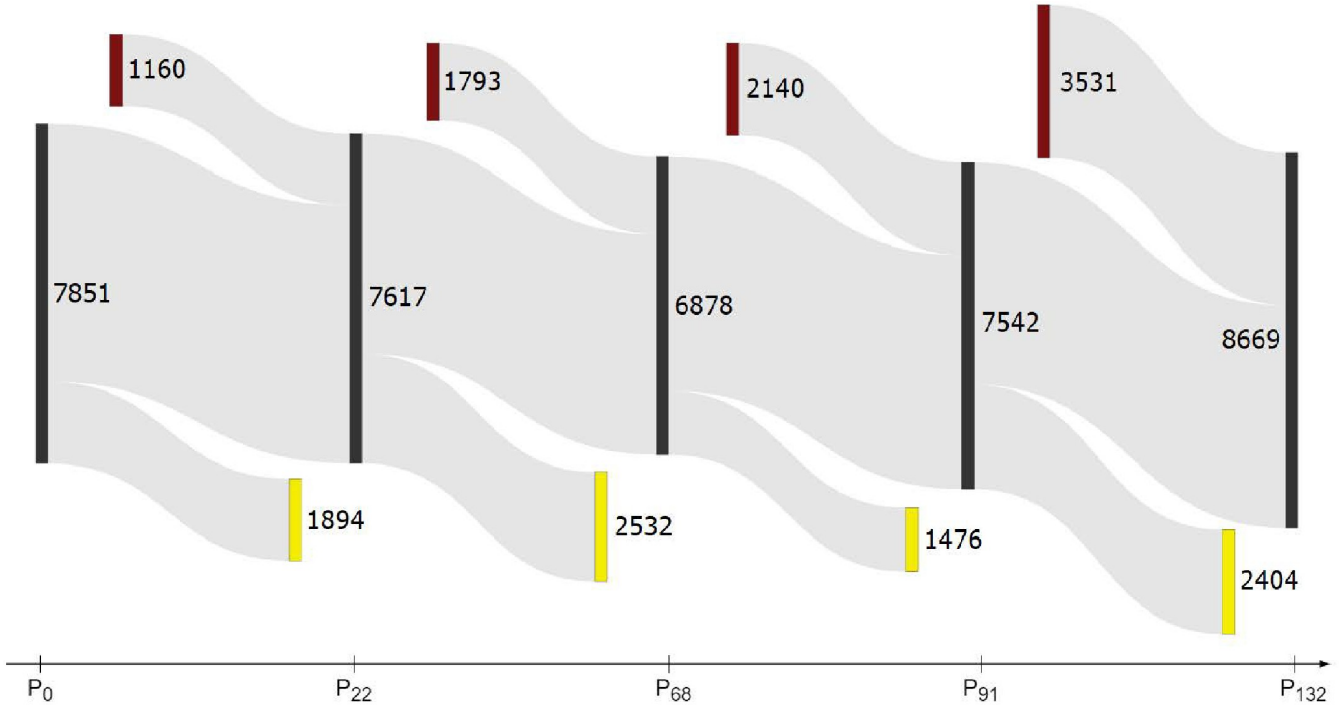

Fig 3. A Sankey diagram showing major transitions between the five selected timepoints P_0, P_{22}, . . ., P_{132}. The numbers indicate core nodes (black), new nodes (brown, at top), and lost nodes (yellow, at bottom) between two adjacent network partitions. The differences between the adjacent partitions are quantified by $max\text{-}F_1(P_i|P_{i-1})$, shown with blue line in Fig 2. Note a relatively large in- and out-flow of new and lost nodes between the partitions.

https://doi.org/10.1371/journal.pone.0256175.g003

Fig 4 is more detailed and shows the internal community structure of the cores, with the top five communities C1, . . ., C5 at each selected timepoint. All the remaining smaller communities are appended together into a single Small community.

The top communities were manually scanned for the most influential users (see subsection Retweet h-index influence) and discussion topics. It turns out that most of the communities are structured around political figures (either politicians, public figures, or journalists with clear political orientation) [29], and that political and ideological topics are prevailing [30]. Thus, the top communities can be classified into three categories: left-leaning (most influential users are part of the left-wing structures), right-leaning (most influential users are part of the right-wing), and Sports (users and topics are clearly related to sports). In Fig 4, the left-leaning communities are in shades of red and the right-leaning communities are in shades of blue. The only non-political community is Sports, in green, represented by the following sequence of communities:

$$C3_0 \mapsto C4_{22} \mapsto (C \subset Small)_{68} \mapsto C5_{91} \mapsto C4_{132}.$$

A community Ci at timepoints t is denoted by Ci_t. Note that at timepoint $t = 68$ the Sports community is absorbed into the Small community.

The political communities are considerably larger than Sports. Let us first consider some right-leaning communities, which feature the current Slovenian prime minister of the right-wing government, with an anonymized Twitter handle $PM-mar20-now. He was initially a member of relatively small communities that at timepoint $t = 22$ did not even make it into the

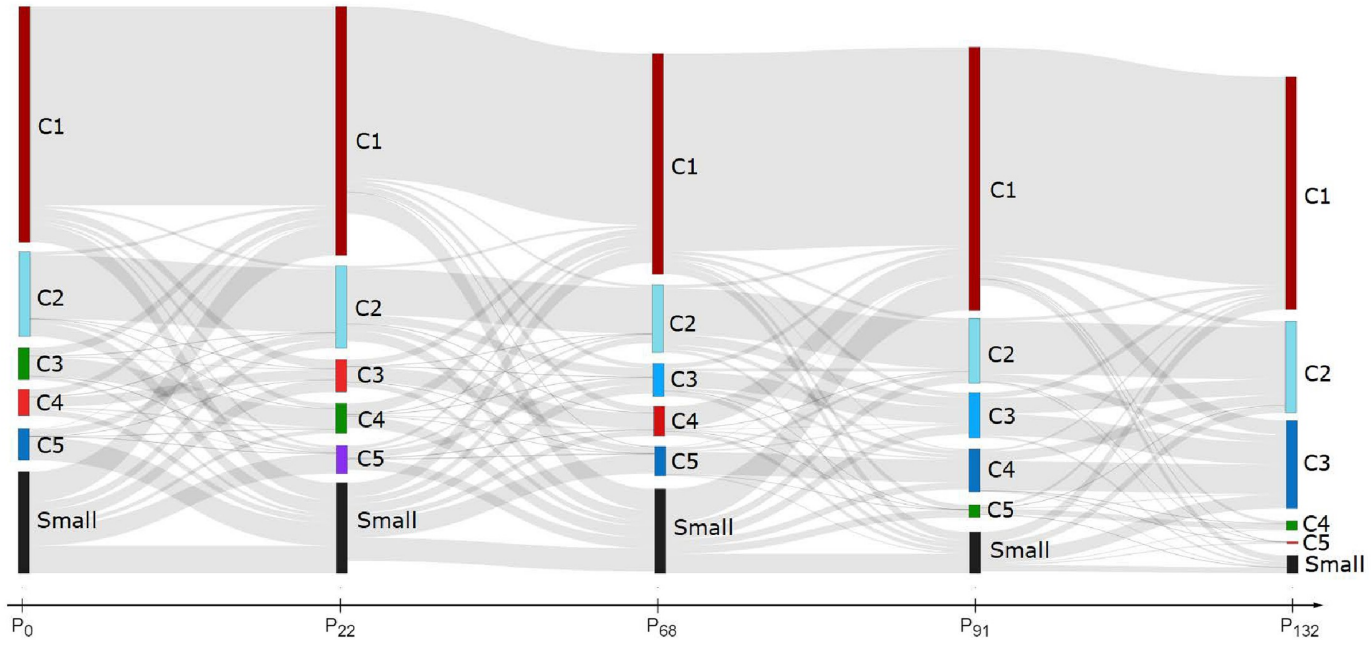

Fig 4. A Sankey diagram showing transitions between the five largest communities C1, . . ., C5 at the selected timepoints P_0, P_{22}, . . ., P_{132}. The remaining smaller communities are labeled as Small, and new and lost nodes are not shown here. The differences between the adjacent partitions are quantified by $F_1(P_i|P_{i-1})$, shown with black line in Fig 2. The left-leaning communities are in shades of red, the right-leaning communities are in shades of blue, and the Sports community is green.

https://doi.org/10.1371/journal.pone.0256175.g004

top five:

$$C5_0 \mapsto (C \subset Small)_{22} \mapsto C5_{68} \mapsto C4_{91} \mapsto C3_{132}.$$

Only after the right-wing government took over in March 2020 (timepoints $t = 91, 132$) did his community grow considerably.

On the political left-wing, there is the $C1$ community that grows and shrinks with time, but remains by far the largest community throughout the three year period. The former Slovenian prime minister (between September 2018 and March 2020), with an anonymized Twitter handle \$PM-sep18-mar20, was a member of $C1$ for most of the time:

$$C1_0 \mapsto C1_{22} \mapsto C4_{68} \mapsto C1_{91} \mapsto C1_{132}.$$

Only in the second half of his government ($t = 68$) did he feature prominently in his own community $C4$. The left-wing Slovenian prime minister before him (until March 2018), with an anonymized Twitter handle \$PM-sep14-sep18, was initially a member of smaller communities on the left-wing, and recently joined $C1$:

$$C4_0 \mapsto C3_{22} \mapsto C4_{68} \mapsto C1_{91} \mapsto C1_{132}.$$

It is interesting to observe the official Slovenian government Twitter account @vladaRS. It moves from the left-leaning to the right-leaning communities as the left-wing government is replaced by the right-wing one, but with some delay:

$$C4_0 \mapsto C3_{22} \mapsto C4_{68} \mapsto C1_{91} \mapsto C3_{132}.$$

@vladaRS matches the $PM-sep14-sep18 community at $t = 0, 22$, the $PM-sep18-mar20 community at $t = 68, 91$, and the $PM-mar20-now community at $t = 132$. This is another piece of evidence that retweet communities evolve gradually and that it takes a while before events with a high impact are reflected in a new community structure.

To further characterize political polarization and community evolution, we now turn attention from the community membership to the retweet links between the communities.

Identification of super-communities

Twitter users differ in how prolific they are in posting tweets, and in the impact these tweets make on the other users. One way to estimate the influence of a Twitter user is to consider how often are its posts retweeted. Similarly, the influence of a community can be estimated by the total number of retweets of their posts. Retweets within the community indicate internal influence, and retweets outside of the community indicate external influence. This approach to characterize influential users and communities was already applied to a wide range of environmental issues discussed on Twitter [31].

In this subsection we focus on community influence and subsequent identification of super-communities. Another measure of individual influence is described in the next subsection Retweet h-index influence. In our retweet networks, the number of retweets is represented by the weighted out-degree of a node. Let W_{ij} denote the sum of all weighted edges between communities C_i and C_j. The average community influence I is defined as:

$$I(C_i) = \frac{\sum_j W_{ij}}{|C_i|},$$

i.e., the weighted out-degree of C_i, normalized by its size. The influence I consists of the internal I_{int} and external I_{ext} component, $I = I_{int} + I_{ext}$, where

$$I_{int}(C_i) = \frac{W_{ii}}{|C_i|},$$

and

$$I_{ext}(C_i, C_j) = \frac{\sum_{i \neq j} W_{ij}}{|C_i|}.$$

We compute internal and external influence of the retweet communities detected at the selected timepoints $t = 0, 22, 68, 91, 132$. The communities which are politically left- or right-leaning are shown in Fig 5, the Sports community is omitted. A community, proportional to its size, is depicted as a pie chart, indicating its internal and external influence. A pair of communities C_i, C_j is linked by a weighted directed edge from C_i to C_j, with the weight equal to the external influence $I_{ext}(C_i, C_j)$.

The meta-networks in Fig 5 support clear identification of two super-communities: Left-wing and Right-wing. A super-community exhibits relatively strong external influence links between its constituent communities. However, there are considerable differences between the Left-wing and Right-wing super-communities. The Left-wing is larger, and its communities have higher internal influences. The Right-wing, on the other hand, has stronger inter-community links, its communities have higher external influences, and appears more cohesive. Note that there are barely any links between the Left-wing and Right-wing communities, a characteristics of echo chambers and political polarization [32].

In Fig 6 we show the total influence of both super-communities. Total influence of a super-community is the sum of weighted out-degrees of all its members, without normalization. The

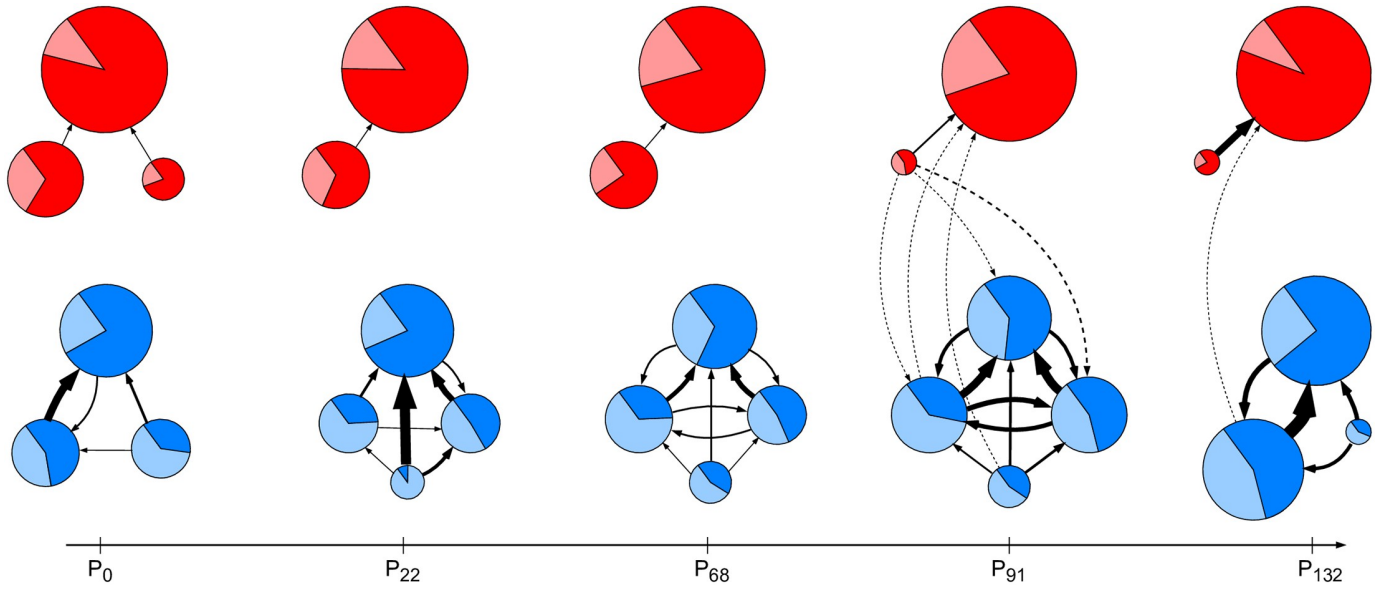

Fig 5. Identification of super-communities from the meta-networks. Nodes are detected communities C1, . . ., C7 at different timepoints, and edges denote average external influences. A node diameter is proportional (cube-root) to the number of community members, darker area corresponds to the internal influence, and lighter area to the external influence. Red communities are part of the Left-wing super-community, blue communities are part of the Right-wing super-community, and the remaining Sports community is not shown. Dashed edges show rare and relatively weak links between the Left- and Right-wing communities.

https://doi.org/10.1371/journal.pone.0256175.g005

Right-wing super-community is typically half the size of the Left-wing, approaching in size only at the last timepoint $t = 132$. However, the influence of the Right-wing is always considerably higher, with the gap even increasing after the right-wing government took over in March 2020 (timepoints $t = 91, 132$).

Retweet h-index influence

Weighted out-degree is a useful measure of influence for communities and super-communities. However, we propose a different measure of influence for individual Twitter users. The user influence is estimated by their retweet h-index, an adaptation of the well known Hirsch index [33] to Twitter. The retweet h-index takes into account the number of tweets posted, as well as the impact of individual posts in terms of retweets.

A user with an index of h has posted h tweets and each of them was retweeted at least h times. Let RT be the function that returns the number of retweets for each original post. The values of RT are ordered in decreasing order, from the largest to the lowest, and i indicates the ranking position in the ordered list. The h-index is then computed as follows:

$$\text{h-index}(RT) = \max_i \min(RT(i), i).$$

To the best of our knowledge, the retweet h-index was first used on Twitter data in the context of Brexit, to measure the influence of the Leave and Remain proponents [19]. Later, this measure of influence was termed a retweet h-index [34], a term we also adopt here.

We compute the h-index and the h-index rank for all the users on Slovenian Twitter during the three year period. For each super-community, Left-wing and Right-wing, we show the top ten most influential users by h-index, ordered by the h-index rank (Table 1). The users are ranked for the overall three year period, but the h-index and relative ranks are also provided

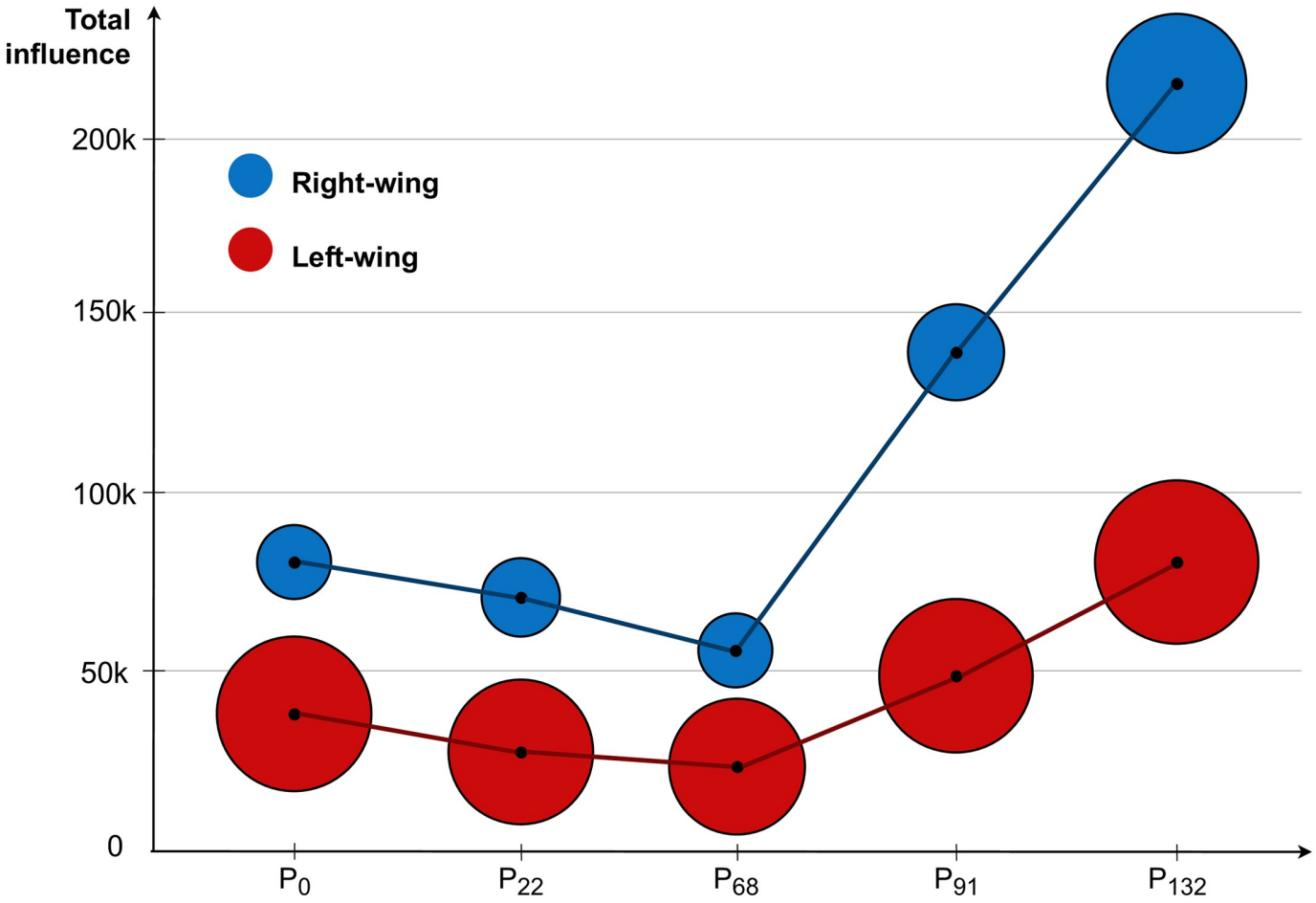

Fig 6. Total weighted out-degree influence for both super-communities. A super-community size is proportional to the number of its members. Total influence is the sum of weighted out-degree influences of all super-community members. Note that the influence of the Right-wing super-community is at least twice as large as the influence of the Left-wing super-community and increasing with time, despite the fact that it is considerably smaller.

https://doi.org/10.1371/journal.pone.0256175.g006

for the selected timepoints $t = 0, 22, 68, 91, 132$. Two of the top Twitter users, @vladaRS and @ukclj, do not remain in the same super-community, but move from the Left-wing to the Right-wing as the government changed.

There is a large difference between the members of the Right-wing and Left-wing super-communities. The Right-wing members consistently take the top h-index ranks, while the Left-wing members barely make it into the top 100 h-index ranks. This reaffirms the super-community influence results from the subsection Identification of super-communities, and is consistent with our previous results. In the case of the European Parliament, higher Twitter activity was observed for the right-wing parties [18]. In the case of Brexit, the Leave proponents showed much higher activity and influence on Twitter than the Remain proponents [19].

As per PLoS ONE policy, individual Twitter users have to be anonymized. Therefore we replace each individual Twitter handle @User with an anonymous handle $Type. The user types for the top 890 users were determined manually [29]. There are three types of individual users: Politician, Public_figure, and Journalist, and an additional Anonymous type for users

Table 1. Top ten influential users from each super-community, ranked by the overall h-index. Individual Twitter users are anonymized and their handles start with $. Left-to-Right denotes users which moved from the Left-wing to the Right-wing super-community (@vladaRS: the official Slovenian government account, and @ukclj: University Medical Centre Ljubljana). The top users in each super-community are $PM-mar20-now (current prime minister), and $PM-sep18-mar20 (former prime minister), respectively. Each user is assigned the h-index rank (h-rank), the h-index (h-ind) for the overall three year period and the five selected timepoints (P_0, . . ., P_{132}), and the overall unweighted out-degree (out-deg). Note that the top Left-wing influential users barely reach the h-index rank of top 100.

User	Overall			P_0		P_{22}		P_{68}		P_{91}		P_{132}	
	h-rank	h-ind	out-deg	h-rank	h-ind	h-rank	h-ind	h-rank	h-ind	h-rank	h-ind	h-rank	h-ind
Right-wing:													
$PM-mar20-now	1	168	2621	1	93	1	92	1	79	1	93	1	140
$Journalist1	2	111	2465	5	56	3	55	2	60	2	69	2	96
$Journalist2	3	99	1724	7	47	4	53	5	47	6	50	6	60
$Public_figure1	4	95	2169	6	54	5	49	10	40	5	51	4	77
$Politician1	5	92	1609	32	29	39	28	34	28	20	37	3	80
$Anonymous1	6	83	2228	4	58	2	56	7	44	12	46	5	62
$Public_figure2	7	79	1715	34	28	7	46	23	31	7	48	10	56
$Politician2	8	76	1403	31	29	10	39	4	47	4	52	27	43
$Public_figure3	9	75	2273	30	30	25	33	25	31	21	36	13	55
$Public_figure4	10	75	1998	28	30	12	39	8	42	3	56	14	53
Left-to-Right:													
@vladaRS	14	72	2287	446	9	586	8	489	8	52	26	9	57
@ukclj	32	59	2170	/	/	545	8	254	11	61	24	34	41
Left-wing:													
$PM-sep18-mar20	97	41	889	203	13	443	9	490	8	200	14	96	28
@necenzurirano_	103	40	876	/	/	/	/	/	/	225	13	48	37
$Public_figure5	131	37	1349	117	17	52	25	258	11	227	13	84	29
$Public_figure6	133	37	1014	1550	4	1480	4	793	6	365	10	68	32
@strankalevica	140	36	886	170	14	160	16	219	12	114	19	105	27
$Journalist3	141	35	1060	274	11	432	9	348	9	563	8	95	28
$Politician3	149	35	613	3621	1	4595	1	3869	1	267	12	83	29
$Journalist4	182	32	1236	239	12	144	17	218	12	157	16	124	25
@STA_novice	186	32	2204	367	10	255	13	325	10	143	17	162	23
@SpletnaMladina	246	28	1424	135	16	200	14	166	14	177	15	197	21

https://doi.org/10.1371/journal.pone.0256175.t001

that cannot be easily identified. An artificial handle for the current and former prime ministers, $PM-*, was already introduced. Institutional Twitter accounts remain unchanged.

The Right-wing and Left-wing super-communities are led by the current ($PM-mar20-now) and former ($PM-sep18-mar20) prime minister of Slovenia, respectively. The other top members are either politicians, journalists, or public figures active on Twitter. In the Left-wing, there are some media account (@necenzurirano_, @STA_novice, @SpletnaMladina), and a political party account (@strankalevica—'The Left').

The only two users in Table 1 which are clearly unrelated to politics are @ukclj (University Medical Centre Ljubljana) and $Public_figure6 (a biochemist from the National Institute of Chemistry). They reached the rank of top 100 influencers only after the emergence of the Covid-19 pandemic (@ukclj at $t = 91$, and $Public_figure6 at $t = 132$). They post tweets about the medical issues, drugs and vaccines related to the pandemic. There are other influential users, tracking and commenting on the pandemic development, which emerged recently, but they did not yet make it into the overall top ten h-index list.

Conclusions

Social media, and Twitter in particular, are a rich source of data that reflects social relations between the users. In the paper we exploit a specific type of networks where retweets are used as links between the users. We demonstrate that in the retweet networks meaningful communities are formed. We show that retweet influence reveals important differences between different communities as well as between individual Twitter users. The main focus of the paper is on the evolution of communities and influence through time, and we address several issues relevant for the field of dynamic networks.

One problem is the instability of detected static communities by a standard community detection Louvain algorithm. We propose to run an ensemble of Louvain trials, and detect stable communities through frequent co-occurrence of nodes across the trials. Preliminary evaluations of the Ensemble Louvain algorithm on some benchmark networks with known "ground truth" communities show promising results, and this is certainly one of the directions that needs to be further explored in the future.

We study network evolution by taking several static network snapshots with a sliding window. One has to decide on the window size and the temporal resolution between the snapshots. We decided on the 24 weeks window size and an exponential edge weight decay, with half-time of 4 weeks. The edge decay removes the effect of the trailing end of the window, and thus makes the choice of the window size less relevant. The choice of the half-time decay, on the other hand, is subject to experimentation, and depends on the volume of Twitter data. The chosen sliding window of one week provides high temporal resolution, but again the choice of this parameter is not crucial. We propose a temporal zoom-out to a lower time resolution, by computationally efficient selection of more distant timepoints where the network partitions exhibit larger differences. An analysis of how robust is this selection and what are meaningful ranges of distant timepoints is required in the future.

We apply and extend a measure of community similarity BCubed, which was originally introduced to evaluate quality of document clustering, but does not appear to be used in the field of complex networks. The F_1 score can measure differences between network communities with only partially overlapping set of nodes. This is essential for comparing retweet networks, where new nodes keep appearing and disappearing from the network snapshots. An additional nice property of F_1 is that it degenerates into a well-known set comparison coefficient, directly related to the Jaccard index.

A specially interesting result of this research is clear identification of super-communities from external influence links between the detected communities. The exiting problem, worth addressing in the future, is how to design a multi-stage super-community detection algorithm. This seems relevant for retweet networks in particular, where a standard community detection algorithm produces a large set of fractured communities.

There are two follow-up directions of the current research, already undertaken: classification of tweets by the level of hate speech and detection of discussion topics [30], and attribution of the hate speech to the detected communities and types of users [29]. The results show that most of the hate speech has the form of offensive tweets, and that over 60% of them can be attributed to a single right-leaning community of moderate size.

We illustrate our approach on a well-defined set of Slovenian tweets, of reasonable size, but not extremely large. Our next step is to apply the same approach on two different, but somehow related sets of Croatian and Serbian tweets. This will reveal which parameters need to be tuned to specific datasets, and what seem to be domain-invariant properties and methods, applicable to a wide range of domains.

Methods

Data collection

The three years of comprehensive Slovenian Twitter data cover the period from January 1, 2018 until December 28, 2020. In total, 12,961,136 tweets were collected. We used the TweetCat tool [35] for Twitter data acquisition.

The TweetCat tool is specialized on harvesting Twitter data of less frequent languages. It searches continuously for new users that post tweets in the language of interest by querying the Twitter Search API for the most frequent and unique words in that language. Once a set of new potential users posting in the language of interest are identified, their full timeline is retrieved and the language identification is run over their timeline. If it is evident that specific users post predominantly in the language of interest, they are added to the user list and their posts are being collected for the remainder of the collection period. In the case of Slovenian Twitter, the collection procedure started at the end of 2017 and is still running. As a consequence, we are confident that the full Slovenian tweetosphere is well covered.

Ensemble Louvain

The Ensemble Louvain algorithm addresses the problem of instability of the Louvain community detection algorithm. The instability is manifested by different results of community detection in the same network, run with different initial seeds. This is due to theoretical issues with modularity maximization, and to heuristic nature of an efficient implementation of the algorithm.

We address this instability problem with a new approach called Ensemble Louvain. The steps of the algorithm are as follows:

1. run several trials of Louvain on the same network,
2. built a new network where a pair of the original nodes is linked if their total co-membership across all the Louvain trials is above a given threshold (e.g., 90%),
3. identify the disjoints sets which represent the resulting communities.

More trials eventually lead to more stable partitioning (see Fig 7), but increase the computation time. We found a reasonable trade-off between 50 and 500 trials, depending on the network size.

We are not the first to use ensembles for community detection. A combination of several different algorithms to create a refined partitioning was proposed in [36]. Re-sampling methods with variations of the same network were used by [37]. [38] create weighted consensus graphs and then detect communities in the consensus graph.

We measure the stability of Ensemble Louvain by Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). An initial comparison between the standard Louvain versus Ensemble Louvain is performed on three well-known datasets: the Football network (115 nodes), the Email EU core (1005 nodes), and a Slovenian retweet network (3992 nodes). 100 separate experiment runs show that Ensemble Louvain yields significantly more stable results, especially on the larger networks, where the variation between possible solutions grows.

We measure the performance with respect to the "ground truth" for the Football and Email EU Core networks. The initial results (presented by the mean ± standard deviation of the scores) show a significant improvement of Ensemble Louvain over the standard Louvain:

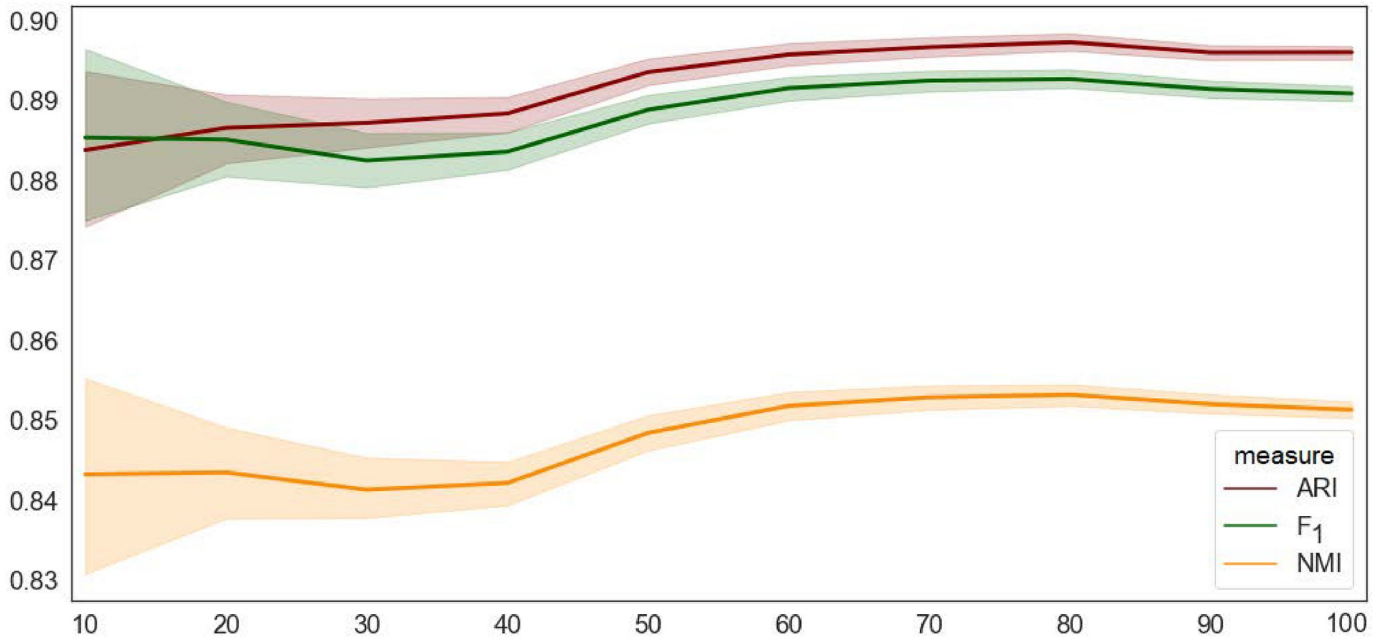

Fig 7. A comparison of the BCubed F_1 measure with Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). The comparison is run on the initial G_0 Slovenian retweet network. On the x-axis is the number of standard Louvain trials, $N = 10, 20, \ldots, 100$. For each N, all the resulting partitions are pairwise compared by the three measures, ARI, F_1, and NMI (y-axis). Solid lines show the mean values and shaded areas the 95% confidence intervals.

https://doi.org/10.1371/journal.pone.0256175.g007

- The Football network,
  standard Louvain: NMI = 0.88±0.015 and ARI = 0.78±0.041,
  Ensemble Louvain: NMI = 0.92±0.008 and ARI = 0.89±0.019.
- The Email EU Core network,
  standard Louvain: NMI = 0.58±0.016 and ARI = 0.32±0.032,
  Ensemble Louvain: NMI = 0.72±0.005 and ARI = 0.52±0.012.

BCubed measure of community similarity

The BCubed measure was originally proposed to evaluate effectiveness of document clustering [39]. Its properties were compared to a wide range of other extrinsic clustering evaluation metrics, with the conclusion that BCubed satisfies all the required qualitative properties [27]. Since data clustering and community detection in networks produce analogous results, one can also apply the BCubed measure to evaluate the detected communities. Communities can be evaluated against the "ground truth" when available, or compared to each other, as is the case with evolving communities.

The BCubed measure is applicable to individual nodes, communities, and network partitions in general. It decomposes the evaluation into calculating the precision and recall associated with each node in the network. The precision (*Pre*) and recall (*Rec*) are then combined into the F_1 score:

$$F_1 = 2\,\frac{Pre \cdot Rec}{Pre + Rec}.$$

The F_1 score is a special case of Van Rijsbergen's effectiveness measure [40], where precision and recall can be combined with different weights. In the following we focus on definitions of precision and recall for different cases, and assume a balanced definition of the F_1 score as the harmonic mean. We first define the BCubed measure for a node, and then proceed with definitions of *core*-F_1, *standard* F_1, and theoretical *max*-F_1.

Let $L(n)$ denote the "ground truth" community and $C(n)$ the detected community of the node n, $n \in L(n)$, $C(n)$. *Pre* and *Rec* for a node are defined as follows:

$$Pre(n) = \frac{|L(n) \cap C(n)|}{|C(n)|},$$

$$Rec(n) = \frac{|L(n) \cap C(n)|}{|L(n)|}.$$

***Core*-F_1.** Let first assume a special case when a pair of network partitions consist of the same set of nodes. In this case, we name the BCubed measure *core*-F_1. Let $Ls = \{L_i\}$ denote a set of "ground truth" communities L_i, and $Cs = \{C_i\}$ a set of detected communities C_i. Constituent *Pre* and *Rec* for the partition *Cs* with respect to *Ls* are defined as:

$$Pre(Cs|Ls) = \frac{1}{|Cs|} \sum_{n \in C_i, C_i \in Cs} Pre(n),$$

$$Rec(Cs|Ls) = \frac{1}{|Ls|} \sum_{n \in L_i, L_i \in Ls} Rec(n).$$

The F_1 measure proposed by Rossetti [28] is a special case of the *core*-F_1. In our case, the *Pre* and *Rec* are computed with respect to all the communities *Ci* and *Li*, while Rossetti computes the *Pre* and *Rec* just between a pair of communities with the largest overlap.

***Standard* F_1.** In general, a pair of partitions P_0, P_1 has some overlapping nodes, and some nodes that are present in only one of the partitions. Let *Ls*, *Cs* denote communities with overlapping nodes, and R_0, R_1 the nodes specific to the respective partitions P_0, P_1. We have:

$$P_0 = Ls \cup R_0, \quad P_1 = Cs \cup R_1.$$

Pre and *Rec* of partition P_1 with respect to the "ground truth" partition P_0 are then computed as follows:

$$Pre(Cs|P_0) = Pre(Cs|Ls),$$

$$Pre(P_1|P_0) = \frac{|Cs|}{|Cs| + |R_1|} Pre(Cs|Ls),$$

$$Rec(Cs|P_0) = \frac{|Ls|}{|Ls| + |R_0|} Rec(Cs|Ls),$$

$$Rec(P_1|P_0) = Rec(Cs|P_0).$$

***Max*-F_1.** A theoretical maximum value of F_1 can be computed under the assumption that all the overlapping nodes of the two partitions P_0, P_1 form one community. Let $C = L$ denote

the community with the intersecting nodes, R_0 extra nodes in P_0 (w.r.t. P_1), and R_1 extra nodes in P_1 (w.r.t. P_0):

$$C = L = P_1 \cap P_0, \quad P_0 = L \cup R_0, \quad P_1 = C \cup R_1.$$

Pre and *Rec* of P_1 with respect to the "ground truth" P_0 are computed as:

$$Pre(P_1|P_0) = \frac{|C|}{|C| + |R_1|} = \frac{|C|}{|P_1|},$$

$$Rec(P_1|P_0) = \frac{|C|}{|L| + |R_0|} = \frac{|C|}{|P_0|}.$$

The *max*-F_1 score is then:

$$F_1(P_1|P_0) = 2\,\frac{Pre(P_1|P_0) \cdot Rec(P_1|P_0)}{Pre(P_1|P_0) + Rec(P_1|P_0)} = 2\,\frac{|C|}{|P_1| + |P_0|} = 2\,\frac{|P_1 \cap P_0|}{|P_1| + |P_0|}.$$

This measure of similarity of two sets, P_0 and P_1, is also known as Sørensen-Dice coefficient [41, 42]. It is directly related to the Jaccard index:

$$Jacc(P_1|P_0) = \frac{|P_1 \cap P_0|}{|P_1 \cup P_0|}.$$

The transformation between the Jaccard index and F_1 is as follows:

$$Jacc = \frac{F_1}{2 - F_1}, \quad F_1 = \frac{2 \cdot Jacc}{1 + Jacc}.$$

The BCubed-based F_1 measure therefore has two special cases, *core*-F_1 for comparing completely overlapping network partitions, and *max*-F_1 for comparing two partitions with emerging (new) and disappearing (lost) nodes. The later case is specially relevant in evolving retweet networks, when new users appear and some users leave the network at different time windows.

The F_1 can be compared to standard community evaluation measures, such as Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). In Fig 7 we compare the three measures on the same network, running several trials of the standard Louvain with different initial seeds. The F_1 in this case is actually the *core*-F_1, compatible to ARI and NMI. ARI and NMI cannot be applied in the case when network partitions differ in the sets of respective nodes.

Author Contributions

Conceptualization: Bojan Evkoski, Igor Mozetič, Petra Kralj Novak.

Data curation: Nikola Ljubešić.

Methodology: Bojan Evkoski, Igor Mozetič, Petra Kralj Novak.

Software: Bojan Evkoski.

Supervision: Petra Kralj Novak.

Visualization: Bojan Evkoski.

Writing – original draft: Bojan Evkoski, Igor Mozetič, Nikola Ljubešić, Petra Kralj Novak.

References

1. Dakiche N, Tayeb FBS, Slimani Y, Benatchba K. Tracking community evolution in social networks: A survey. Information Processing & Management. 2019; 56(3):1084–1102. https://doi.org/10.1016/j.ipm.2018.03.005
2. Evkoski B, Mozetič I, Ljubešić N, Novak PK. Evolution of political polarization on Slovenian Twitter. In: Complex Networks 2020, Book of abstracts; 2020. p. 325–327.
3. Rossetti G, Cazabet R. Community discovery in dynamic networks. ACM Computing Surveys. 2018; 51 (2):1–37. https://doi.org/10.1145/3172867
4. Coscia M, Giannotti F, Pedreschi D. A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining: The ASA Data Science Journal. 2011; 4(5):512–546. https://doi.org/10.1002/sam.10133
5. Holme P, Saramäki J. Temporal networks. Physics reports. 2012; 519(3):97–125. https://doi.org/10.1016/j.physrep.2012.03.001
6. Aynaud T, Fleury E, Guillaume JL, Wang Q. Communities in evolving networks: definitions, detection, and analysis techniques. In: Dynamics On and Of Complex Networks, Volume 2. Springer; 2013. p. 159–200.
7. Hartmann T, Kappes A, Wagner D. Clustering evolving networks. In: Algorithm engineering. Springer; 2016. p. 280–329.
8. Lambiotte R, Masuda N. A guide to temporal networks. vol. 4. World Scientific; 2016.
9. Chen Z, Wilson KA, Jin Y, Hendrix W, Samatova NF. Detecting and tracking community dynamics in evolutionary networks. In: 2010 IEEE International Conference on Data Mining Workshops. IEEE; 2010. p. 318–327.
10. Bóta A, Csizmadia L, Pluhár A. Community detection and its use in real graphs. Matcos. 2010.
11. Alvari H, Hajibagheri A, Sukthankar G. Community detection in dynamic social networks: A game-theoretic approach. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE; 2014. p. 101–107.
12. Agarwal MK, Ramamritham K, Bhide M. Real time discovery of dense clusters in highly dynamic graphs: Identifying real world events in highly dynamic environments. In: Proc. VLDB. vol. 5; 2012. p. 980–991.
13. Crane H, Dempsey W. Community detection for interaction networks; 2015. Available from: https://arxiv.org/abs/1509.09254.
14. Aynaud T, Guillaume JL. Multi-step community detection and hierarchical time segmentation in evolving networks. In: Proc. 5th SNA-KDD workshop; 2011. p. 69–103.
15. Gauvin L, Panisson A, Cattuto C. Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. PLoS ONE. 2014; 9(1):e86028. https://doi.org/10.1371/journal.pone.0086028 PMID: 24497935
16. Aynaud T, Guillaume JL. Static community detection algorithms for evolving networks. In: 8th International symposium on modeling and optimization in mobile, ad hoc, and wireless networks. IEEE; 2010. p. 513–519.
17. Cherepnalkoski D, Mozetič I. Retweet networks of the European Parliament: Evaluation of the community structure. Applied Network Science. 2016; 1(1):2. https://doi.org/10.1007/s41109-016-0001-4 PMID: 30533494
18. Cherepnalkoski D, Karpf A, Mozetič I, Grčar M. Cohesion and coalition formation in the European Parliament: Roll-call votes and Twitter activities. PLoS ONE. 2016; 11(11):e0166586. https://doi.org/10.1371/journal.pone.0166586 PMID: 27835683
19. Grčar M, Cherepnalkoski D, Mozetič I, Kralj Novak P. Stance and influence of Twitter users regarding the Brexit referendum. Computational Social Networks. 2017; 4(1):6. https://doi.org/10.1186/s40649-017-0042-6 PMID: 29266132
20. Fortunato S, Hric D. Community detection in networks: A user guide. Physics Reports. 2016; 659:1–44. https://doi.org/10.1016/j.physrep.2016.09.002
21. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008; 2008(10):P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008
22. Newman MEJ. Modularity and community structure in networks. Proceedings of the National Academy of Sciences. 2006; 103(23):8577–8582. https://doi.org/10.1073/pnas.0601602103
23. Lancichinetti A, Fortunato S. Community detection algorithms: a comparative analysis. Physical review E. 2009; 80(5):056117. https://doi.org/10.1103/PhysRevE.80.056117 PMID: 20365053

24. Good BH, de Montjoye YA, Clauset A. Performance of modularity maximization in practical contexts. Phys Rev E. 2010; 81:046106. https://doi.org/10.1103/PhysRevE.81.046106 PMID: 20481785

25. Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985; 2(1):193–218. https://doi.org/10.1007/BF01908075

26. Danon L, Díaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment. 2005; 2005(09):P09008–P09008. https://doi.org/10.1088/1742-5468/2005/09/P09008

27. Amigó E, Gonzalo J, Artiles J, Verdejo F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval. 2009; 12(4):461–486. https://doi.org/10.1007/s10791-008-9066-8

28. Rossetti G, Pappalardo L, Rinzivillo S. A novel approach to evaluate community detection algorithms on ground truth. In: 7th Workshop on Complex Networks; 2016.

29. Evkoski B, Pelicon A, Mozetič I, Ljubešić N, Novak PK. Retweet communities reveal the main sources of hate speech; 2021. Available from: https://arxiv.org/abs/2105.14898.

30. Novak PK, Ljubešić N, Pelicon A, Mozetič I. Hate speech detection as a knowledge discovery process; 2021.

31. Sluban B, Smailović J, Battiston S, Mozetič I. Sentiment leaning of influential communities in social networks. Computational Social Networks. 2015; 2(1):9. https://doi.org/10.1186/s40649-015-0016-5

32. Del Vicario M, Vivaldo G, Bessi A, Zollo F, Scala A, Caldarelli G, et al. Echo chambers: Emotional contagion and group polarization on Facebook. Scientific Reports. 2016; 6(1):37825. https://doi.org/10.1038/srep37825 PMID: 27905402

33. Hirsch JE. An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences. 2005; 102(46):16569–16572. https://doi.org/10.1073/pnas.0507655102 PMID: 16275915

34. Gallagher RJ, Doroshenko L, Shugars S, Lazer D, Welles BF. Sustained online amplification of COVID-19 elites in the United States. Social Media + Society. 2021; 7(2):20563051211024957. https://doi.org/10.1177/20563051211024957

35. Ljubešić N, Fišer D, Erjavec T. TweetCaT: A tool for building Twitter corpora of smaller languages. In: Proc. 9th Intl. Conf. on Language Resources and Evaluation. European Language Resources Association (ELRA); 2014. p. 2279–2283.

36. Chakraborty T, Park N, Agarwal A, Subrahmanian VS. Ensemble detection and analysis of communities in complex networks. ACM/IMS Transactions on Data Science. 2020; 1. https://doi.org/10.1145/3313374

37. Dahlin J, Svenson P. Ensemble approaches for improving community detection methods; 2013. Available from: https://arxiv.org/abs/1309.0242.

38. Lancichinetti A, Fortunato S. Consensus clustering in complex networks. Scientific Reports. 2012; 2(1):336. https://doi.org/10.1038/srep00336 PMID: 22468223

39. Bagga A, Baldwin B. Entity-based cross-document coreferencing Using the Vector Space Model. In: Proc. 17th Intl. Conf. on Computational Linguistics (COLING). Stroudsburg, PA, USA; 1998. p. 79–85.

40. Van Rijsbergen CJ. Information Retrieval. 2nd ed. Newton, MA, USA: Butterworth; 1979.

41. Sørensen T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Kongelige Danske Videnskabernes Selskab. 1948; 5(4):1–34.

42. Dice LR. Measures of the amount of ecologic association between species. Ecology. 1945; 26(3):297–302. https://doi.org/10.2307/1932409

3.2 Retweet Communities Reveal the Main Source of Hate Speech

“Retweet communities reveal the main source of hate speech" is the second work in the trilogy of community evolution analysis with Ensemble Louvain, this time with the focus on identifying the groups within the Slovenian tweetosphere responsible for the hate speech. It is a joint work by Bojan Evkoski, Andraž Pelicon, Igor Mozetič, Nikola Ljubešič and Petra Kralj Novak. It was published in PLOS One in March 2022.

The hate speech community evolution analysis is done by first training a Slovenian hate speech model which uses a language model and achieves a comparable score to the inter-annotator agreement of the manual labeling [65]. The analysis reveals that the share of unacceptable tweets moderately increases with time, from the initial 20% in January 2020, to 30% by the end of 2020. Moreover, about 60% of all unacceptable tweets are produced by a single right-leaning community of only moderate size. We also investigate which types of Twitter accounts spread most of the hate speech. It turns out that institutional and media accounts post significantly fewer unacceptable tweets than individual accounts. In fact, the main source of unacceptable tweets are anonymous accounts and accounts that were suspended or closed during the years 2018-2020.

The author of the master thesis contributed to this paper by conducting the community evolution analysis. He was also responsible for most of the experiments as well as the visualization of the results. He did not take part in training and evaluating the hate speech models.

PLOS ONE

RESEARCH ARTICLE

Retweet communities reveal the main sources of hate speech

Bojan Evkoski[1,2], Andraž Pelicon[1,2], Igor Mozetič[1]*, Nikola Ljubešić[1,3], Petra Kralj Novak[1]

1 Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia, **2** Jozef Stefan International Postgraduate School, Ljubljana, Slovenia, **3** Faculty of Information and Communication Sciences, University of Ljubljana, Ljubljana, Slovenia

* igor.mozetic@ijs.si

Citation: Evkoski B, Pelicon A, Mozetič I, Ljubešić N, Kralj Novak P (2022) Retweet communities reveal the main sources of hate speech. PLoS ONE 17(3): e0265602. https://doi.org/10.1371/journal.pone.0265602

Editor: Antonio Scala, Italian National Research Council, ITALY

Received: May 17, 2021

Accepted: March 2, 2022

Published: March 17, 2022

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0265602

Data Availability Statement: Data availability: The Slovenian Twitter dataset 2018-2020, with retweet links and assigned hate speech class, is available at a public language resource repository CLARIN.SI at https://hdl.handle.net/11356/1423. Model

Abstract

We address a challenging problem of identifying main sources of hate speech on Twitter. On one hand, we carefully annotate a large set of tweets for hate speech, and deploy advanced deep learning to produce high quality hate speech classification models. On the other hand, we create retweet networks, detect communities and monitor their evolution through time. This combined approach is applied to three years of Slovenian Twitter data. We report a number of interesting results. Hate speech is dominated by offensive tweets, related to political and ideological issues. The share of unacceptable tweets is moderately increasing with time, from the initial 20% to 30% by the end of 2020. Unacceptable tweets are retweeted significantly more often than acceptable tweets. About 60% of unacceptable tweets are produced by a single right-wing community of only moderate size. Institutional Twitter accounts and media accounts post significantly less unacceptable tweets than individual accounts. In fact, the main sources of unacceptable tweets are anonymous accounts, and accounts that were suspended or closed during the years 2018–2020.

Introduction

Hate speech is threatening individual rights, human dignity and equality, reinforces tensions between social groups, disturbs public peace and public order, and jeopardises peaceful coexistence. Hate speech is among the "online harms" that are pressing concerns of policymakers, regulators and big tech companies [1]. Reliable real-world hate speech detection models are essential to detect and remove harmful content, and to detect trends and assess the sociological impact of hate speech.

There is an increasing research interest in the automated hate speech detection, as well as competitions and workshops [2]. Hate speech detection is usually modelled as a supervised classification problem, where models are trained to distinguish between examples of hate and normal speech. Most of the current approaches to detect and characterize hate speech focus solely on the content of posts in online social media [3–5]. They do not consider the network structure, nor the roles and types of users generating and retweeting hate speech. A systematic literature review of academic articles on racism and hate speech on social media, from 2014 to

availability: The model for hate speech classification of Slovenian tweets is available at a public language models repository Huggingface at https://huggingface.co/IMSyPP/hate_speech_slo.

Funding: The authors acknowledge financial support from the Slovenian Research Agency (research core funding no. P2-103 and P6-0411), the Slovenian Research Agency and the Flemish Research Foundation bilateral research project LiLaH (grant no. ARRS-N6-0099 and FWO-G070619N), and the European Union's Rights, Equality and Citizenship Programme (2014-2020) project IMSyPP (grant no. 875263). The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Competing interests: The authors have declared that no competing interests exist.

2018 [6], finds that there is a dire need for a broader range of research, going beyond the text-based analyses of overt and blatant racist speech, Twitter, and the content generated mostly in the United States.

In this paper, we go a step further from detecting hate speech from Twitter posts only. We develop and combine a state-of-the-art hate speech classification model with estimates of tweet popularity, retweet communities, influential users, and different types of user accounts (individual, organization, or "problematic"). More specifically, we address the following research questions:

- Are hateful tweets more likely to be retweeted than acceptable tweets?
- Are there meaningful differences between the communities w.r.t. hateful tweets?
- How does the hateful content of a community change over time?
- Which types of Twitter users post more hateful tweets?

As a use case, we demonstrate the results on an exhaustive set of three years of Slovenian Twitter data. We report a number of interesting results which are potentially relevant also for other domains and languages. Hate speech is dominated by offensive tweets, while tweets inciting violence towards target groups are rare. Hateful tweets are retweeted significantly more often than acceptable tweets. There are several politically right-leaning communities which form a super-community. However, about 60% of unacceptable tweets are produced by a single right-leaning community of only moderate size. Institutional and media Twitter accounts post significantly less unacceptable tweets than individual account. Moreover, the main sources of unacceptable tweets are anonymous accounts, and accounts that were closed or suspended during the years 2018–2020.

Related works

Identifying hate speech and related phenomena in social media has become a very active area of research in natural language processing in recent years. Early work targeted primarily English, and focused on racism and sexism on Twitter [7], harassment in online gaming communities [8], toxicity in Wikipedia talk pages [9], and hate speech and offensive language on Twitter [10]. Results on non-English languages emerged soon after, with early work focusing on, inter alia, hate towards refugees in Germany [11], newspaper comment moderation in Greek [12], Croatian and Slovenian [13], and obscenity and offensiveness of Arabic tweets [14].

There is very little research addressing hate speech in terms of temporal aspects and community structure on Twitter. The most similar research was done on the social media platform Gab (https://Gab.com) [15]. The authors study the diffusion dynamics of the posts by 341,000 hateful and non-hateful users on Gab. The study reveals that the content generated by the hateful users tends to spread faster, farther, and reach a much wider audience as compared to the normal users. The authors also find that hateful users are far more densely connected between themselves, as compared to the non-hateful users. An additional, temporal analysis of hate speech on Gab was performed by taking temporal snapshots [16]. The authors find that the amount of hate speech in Gab is steadily increasing, and that the new users are becoming hateful at an increasingly high rate. Further, the analysis reveals that the hate users are occupying the prominent positions in the Gab network. Also, the language used by the community as a whole correlates better with the language of hateful users than with the non-hateful users.

Our research addresses very similar questions on the Twitter platform. Most of our results on Twitter are aligned with the findings on Gab, however, there are some important differences. Twitter is a mainstream social medium, used by public figures and organizations, while

Gab is an alt-tech social network, with a far-right user base, described as a haven for extremists. We analyse an exhaustive dataset covering all Twitter communication within Slovenia and in the Slovenian language, while Gab covers primarily the U.S. and the English language.

A dynamic network framework to characterize hate communities, focusing on Twitter conversations related to Covid-19, is proposed in [17]. Higher levels of community hate are consistently associated with smaller, more isolated, and highly hierarchical network communities across both the U.S. and the Philippines. In both countries, the average hate scores remain fairly consistent over time. The spread of hate speech around Covid-19 features similar reproduction rates as other Covid-related information on Twitter, with spikes of hate speech at the same times as the highest community-level organization. The identity analysis further reveals that hate in the U.S. initially targets political figures, and then becomes predominantly racially charged. In the Philippines, on the other hand, the targets of hate over time consistently remain political.

In [18], the authors propose a user-centric view of hate speech. They annotate 4,972 Twitter users as hateful or normal, and find that the hateful users differ significantly from the normal users in terms of their activity patterns, word usage, and network structure. In our case, we manually annotate the 890 most influential users for their type, but the level of hate speech of their tweets is automatically assigned by the hate speech classification model.

The relation between political affiliations and profanity use in online communities is reported in [19]. The authors address community differences regarding creation/tolerance of profanity and suggest a contextually nuanced profanity detection system. They report that a political comment is more likely profane and contains an insult or directed insult than a non-political comment.

The work presented here is an extension of our previous research in the area of evolution of retweet communities [20]. The results, obtained on the same Twitter dataset as used here, show that the Slovenian tweetosphere is dominated by politics and ideology [21], that the left-leaning communities are larger, but that the right-leaning communities and users exhibit significantly higher impact. Furthermore, we empirically show that retweet networks change relatively gradually, despite significant external events, such as the emergence of the Covid-19 pandemic and the change of government. In this paper, the detection and evolution of retweet communities is combined with the state-of-the-art models for hate speech classification.

Structure of the paper

The main results of the paper are in the Structure of the paper Results and discussion section. We first give an overview of the data collected, and how various subsets are used in the Twitter data subsection. In the Hate speech classification subsection we provide a detailed account on training and evaluation of deep learning models. The differences between the detected communities and their roles in posting and spreading hate speech are in the subsection on Communities and hate speech. In subsection Twitter users and hate speech we classify the most influential Twitter users and show the roles of different user types in producing hate speech. In Conclusions we wrap up our combined approach to community evolution and hate speech classification, and present some ideas for future research. The Methods section provides more details regarding the Twitter data acquisition, community detection and evolution, selection of informative timepoints, and retweet influence.

Results and discussion

This section discusses the main results of the paper. We take two independent approaches of analyzing the same set of Twitter data and then combine them to reveal interesting

conclusions. On one hand, we develop and apply a state-of-the-art machine learning approach to classify hate speech in Twitter posts. On the other hand, we analyze network properties of Twitter users by creating retweet networks, detecting communities, and estimating their influence. This combination allows to distinguish between the communities in terms of how much hate speech they originate and how much they contribute to spreading the hate speech by retweeting. A classification of Twitter users by user types provides additional insights into the structure of the communities and the role of the most influential users in posting and spreading the hate speech.

Twitter data

Social media, and Twitter in particular, have been widely used to study various social phenomena [22–25]. For this study, we collected a set of almost 13 million Slovenian tweets in the three year period, from January 1, 2018 until December 28, 2020. The set represents an exhaustive collection of Twitter activities in Slovenia. Fig 1 shows the timeline of Twitter volumes and types of speech posted during that period. The hate speech class was determined automatically by our machine learning model. Note a large increase of Twitter activities at the beginning of 2020 when the Covid-19 pandemic emerged, and the left-wing government was replaced by the right-wing government (in March 2020). At the same time, the fraction of hate speech tweets increased.

Our machine learning model classifies Twitter posts into four classes, ordered by the level of hate speech they contain: acceptable, inappropriate, offensive, and violent. It turns out that inappropriate and violent tweets are relatively rare and cannot be reliably classified. Therefore, for this study, all the tweets that are not considered **acceptable** are jointly classified as **unacceptable**. See the next subsection on Hate speech classification for details on the machine learning modelling and extensive evaluations.

Twitter posts are either original tweets or retweets. Table 1 gives a breakdown of the 13-million dataset collected in terms of how different subsets are used in this study. A large subset of

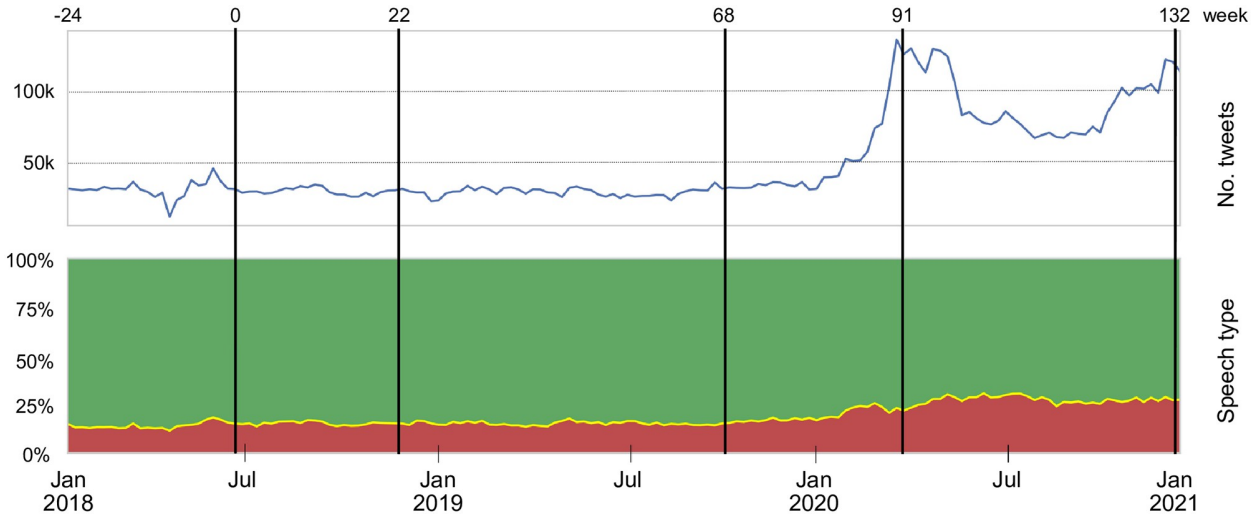

Fig 1. Slovenian Twitter posts, collected over the three year period: Weekly volume of collected original tweets (top) and distribution of hate speech classes (bottom). Green area denotes the fraction of acceptable tweets, yellow (barely visible) inappropriate tweets, and red offensive tweets. Tweets inciting violence are not visible due to their low volume (around 0.1%). During the three years, there are 133 time windows from which we create retweet networks. Each window comprises 24 weeks of Twitter data, and subsequent windows are shifted for one week. Vertical lines show five endpoints of automatically selected time windows, with weeks labeled as $t = 0, 22, 68, 91, 132$.

https://doi.org/10.1371/journal.pone.0265602.g001

Table 1. Slovenian Twitter datasets used in this paper. Out of almost 13 million tweets collected, a selection of original tweets is used for hate speech annotation, training of classification models, and their evaluation. The retweets are used to create retweet networks, detect communities and influential users.

Dataset	Period	No. tweets	Role
All tweets	Jan. 2018–Dec. 2020	12,961,136	collection, hate speech classification
Original tweets	Jan. 2018–Dec. 2020	8,363,271	hate speech modeling
Retweets	Jan. 2018–Dec. 2020	4,597,865	network construction
Training set	Dec. 2017–Jan. 2020	50,000	hate speech model learning and cross valid.
Evaluation set	Feb. 2020–Aug. 2020	10,000	hate speech model eval.

https://doi.org/10.1371/journal.pone.0265602.t001

the original tweets is used to train and evaluate hate speech classification models. On the other hand, retweets are used to create retweet networks and detect retweet communities. See the subsection on Retweet networks and community detection in Methods for details.

Out of the 13 million tweets, 8.3 million are original tweets, the rest are retweets. Out of 8.3 million original tweets, less than one million are retweeted, and most of them, 7.3 million, are not. Given a hate speech classification model, one can compare two properties of a tweet: is it retweeted or not vs. is it acceptable or unacceptable in terms of hate speech. A proper measure to quantify association between two events is an odds ratio (see Comparing proportions in Methods for definition).

Table 2 provides a contingency table of all the original tweets posted. Less than one million of them were retweeted (12%), possibly several times (therefore the total number of retweets in Table 1 is almost five times larger). On the other hand, the fraction of unacceptable tweets is more than 21%. The odds ratio with 99% confidence interval is 0.678±0.004 and the log odds ratio is -0.388±0.006. This confirms a significant negative correlation between the acceptable tweets and their retweets.

A tweet can be classified solely from the short text it contains. A retweet, on the other hand, exhibits an implicit time dimension, from the time of the original tweet to the time of its retweet. Consequently, retweet networks depend on the span of the time window used to capture the retweet activities. In this study, we use a time span of 24 weeks for our retweet networks, with exponential half-time weight decay of four weeks, and sliding for one week. This results in 133 snapshot windows, labeled with weeks $t = 0, 1, \ldots, 132$ (see Fig 1). It turns out that the differences between the adjacent snapshot communities are small [20]. We therefore implement a heuristic procedure to find a fixed number of intermediate snapshots which maximize the differences between them. For the period of three years, the initial, final, and three intermediate network snapshots are selected, labeled by weeks $t = 0, 22, 68, 91, 132$ (see Fig 1). Details are in subsection Selection of timepoints in Methods.

Hate speech classification

Hate speech classification is approached as a supervised machine learning problem. Supervised machine learning requires a large set of examples labeled with types of speech (hateful or

Table 2. Proportions of the (un)acceptable and (not) retweeted tweets over the three year period. The odds ratio (OR) statistic confirms that acceptable tweets are retweeted significantly less often than the unacceptable tweets.

Original tweets	Acceptable	Unacceptable	Total (99%)
Retweeted	708,094	270,282	978,376 (12%)
Not retweeted	5,866,259	1,518,636	7,384,895 (88%)
Total	6,574,353	1,788,918	8,363,271 (99%)
	(79%)	(21%)	ln(OR) = -0.388±0.006

https://doi.org/10.1371/journal.pone.0265602.t002

normal) to cover different textual expressions of speech [26]. Classification models are then trained to distinguish between the examples of hate and normal speech [5]. We pay special attention to properly evaluate the trained models.

The hate speech annotation schema is adopted from the OLID [27] and FRENK [28] projects. The schema distinguishes between four classes of speech on Twitter:

- Acceptable—normal tweets that are not hateful.
- Inappropriate—tweets containing terms that are obscene or vulgar, but they are not directed at any specific person or group.
- Offensive—tweets including offensive generalization, contempt, dehumanization, or indirectly offensive remarks.
- Violent—tweets that threaten, indulge, desire or call for physical violence against a specific person or group. This also includes tweets calling for, denying or glorifying war crimes and crimes against humanity.

The speech classes are ordered by the level of hate they contain, from acceptable (normal) to violent (the most hateful). During the labeling process, and for training the models, all four classes were used. However, in this paper we take a more abstract view and distinguish just between the normal, **acceptable** speech (abbreviated A), and the **unacceptable** speech (U), comprising inappropriate (I), offensive (O) and violent (V) tweets.

We engaged ten well qualified and trained annotators for labeling. They were given the annotation guidelines [29] and there was an initial trial annotation exercise. The annotators already had past experience in a series of hate speech annotation campaigns, including Facebook posts in Slovenian, Croatian, and English. In this campaign, they labeled two sets of the original Slovenian tweets collected: a training and an evaluation dataset.

Training dataset. The training set was sampled from Twitter data collected between December 2017 and January 2020. 50,000 tweets were selected for training different models.

Out-of-sample evaluation dataset. The independent evaluation set was sampled from data collected between February and August 2020. The evaluation set strictly follows the training set in order to prevent data leakage between the two sets and allow for proper model evaluation. 10,000 tweets were randomly selected for the evaluation dataset.

Each tweet was labeled twice: in 90% of the cases by two different annotators and in 10% of the cases by the same annotator. The tweets were uniformly distributed between the annotators. The role of multiple annotations is twofold: to control for the quality and to establish the level of difficulty of the task. Hate speech classification is a non-trivial, subjective task, and even high-quality annotators sometimes disagree on the labelling. We accept the disagreements and do not try to force a unique, consistent ground truth. Instead, we quantify the level of agreement between the annotators (the self- and the inter-annotator agreements), and between the annotators and the models.

There are different measures of agreement, and to get robust estimates, we apply three well-known measures from the fields of inter-rater agreement and machine learning: Krippendorff's Alpha-reliability, accuracy, and F-score.

Krippendorff's Alpha-reliability (*Alpha*) [30] was developed to measure the agreement between human annotators, but can also be used to measure the agreement between classification models and a (potentially inconsistent) ground truth. It generalizes several specialized agreement measures, such as Scott's π, Fleiss' K, Spearman's rank correlation coefficient, and Pearson's intraclass correlation coefficient. *Alpha* has the agreement by chance as the baseline, and an instance of it, used here, *ordinal Alpha* takes ordering of classes into account.

Table 3. The annotator agreement and the overall model performance. Two measures are used: ordinal Krippendorff's *Alpha* and accuracy (*Acc*). The first line is the self-agreement of individual annotators, and the second line is the inter-annotator agreement between different annotators. The last two lines are the model evaluation results, on the training and the out-of-sample evaluation sets, respectively. Note that the overall model performance is comparable to the inter-annotator agreement.

		No. of tweets	Overall	
			Alpha	*Acc*
Self-agreement		5,981	0.79	0.88
Inter-annotator agreement		53,831	0.60	0.79
Classification model	Train.set	50,000	0.61	0.80
	Eval.set	10,000	0.57	0.80

https://doi.org/10.1371/journal.pone.0265602.t003

Accuracy (*Acc*) is a common, and the simplest, measure of performance of the model which measures the agreement between the model and the ground truth. Accuracy does not account for the (dis)agreement by chance, nor for the ordering of hate speech classes. Furthermore, it can be deceiving in the case of unbalanced class distribution.

F-score (F_1) is an instance of the well-known class-specific effectiveness measure in information retrieval [31] and is used in binary classification. In the case of multi-class problems, it can be used to measure the performance of the model to identify individual classes. In terms of the annotator agreement, $F_1(c)$ is the fraction of equally labeled tweets out of all the tweets with label c.

Tables 3 and 4 present the annotator self-agreement and the inter-annotator agreement jointly on the training and the evaluation sets, in terms of the three agreement measures. Note that the self-agreement is consistently higher than the inter-annotator agreement, as expected, but is far from perfect.

Several machine learning algorithms are used to train hate speech classification models. First, three traditional algorithms are applied: Naïve Bayes, Logistic regression, and Support Vector Machines with a linear kernel. Second, deep neural networks, based on the Transformer language models, are applied. We use two multi-lingual language models, based on the BERT architecture [32]: the multi-lingual BERT (mBERT), and the Croatian/Slovenian/English BERT (cseBERT [33]). Both language models are pre-trained jointly on several languages but they differ in the number and selection of training languages and corpora.

The training, tuning, and selection of classification models is done by cross validation on the training set. We use blocked 10-fold cross validation for two reasons. First, this method provides realistic estimates of performance on the training set with time-ordered data [34]. Second, by ensuring that both annotations for the same tweet fall into the same fold, we prevent data leakage between the training and testing splits in cross validation. An even more

Table 4. The annotator agreement and the model performance for individual hate speech classes. The identification of individual classes is measured by the F_1 score. The lines correspond to Table 3. The last three columns give the F_1 scores for the three detailed hate speech classes which are merged into a more abstract, Unacceptable class (F_1(U)), used throughout the paper. Note relatively low model performance for the Violent class (F_1(V)).

		Acceptable	Unacceptable	Inappropriate	Offensive	Violent
		F_1(A)	F_1(U)	F_1(I)	F_1(O)	F_1(V)
Self-agreement		0.92	0.87	0.62	0.85	0.69
Inter-annotator agreement		0.85	0.75	0.48	0.71	0.62
Classification model	Train.set	0.85	0.77	0.52	0.73	0.25
	Eval.set	0.86	0.71	0.46	0.69	0.26

https://doi.org/10.1371/journal.pone.0265602.t004

realistic estimate of performance on yet unseen data is obtained on the out-of-sample evaluation set.

An extensive comparison of different classification models is done following the Bayesian approach to significance testing [35]. Bayesian approach is an alternative to the null hypothesis significance test which has the problem that the claimed statistical significance does not necessarily imply practical significance. One is really interested to answer the following question: What is the probability of the null and the alternative hypothesis, given the observed data? Bayesian hypothesis tests compute the posterior probability of the null and the alternative hypothesis. This allows to detect equivalent classifiers and to claim statistical significance with a practical impact.

In our case, we define that two classifiers are practically equivalent if the absolute difference of their *Alpha* scores is less than 0.01. We consider the results significant if the fraction of the posterior distribution in the region of practical equivalence is less than 5%. The comparison results confirm that deep neural networks significantly outperform the three traditional machine learning models (Naïve Bayes, Logistic regression, and Support Vector Machine). Additionally, language-specific cseBERT significantly outperforms the generic, multi-language mBERT model. Therefore, the cseBERT classification model is used to label all the Slovenian tweets collected in the three year period.

The evaluation results for the best performing classification model, cseBERT, are in Tables 3 and 4. The F_1 scores in Table 4 indicate that the acceptable tweets can be classified more reliably than the unacceptable tweets. If we consider classification of the unacceptable tweets in more detail, we can see low F_1 scores for the inappropriate tweets, and very low scores for the violent tweets. This low model performance is due to relatively low numbers of the inappropriate (around 1%) and violent tweets (around 0.1%, see Table 5) in the Slovenian Twitter dataset. For this reason, the detailed inappropriate, offensive and violent hate speech classes are merged into the more abstract unacceptable class.

The overall *Alpha* scores in Table 3 show a drop in performance estimate between the training and evaluation set, as expected. However, note that the level of agreement between the best model and the annotators is very close to the inter-annotator agreement. This result is comparable to other related datasets, where the annotation task is subjective and it is unrealistic to expect perfect agreement between the annotators [36, 37]. If one accepts an inherent ambiguity of the hate speech classification task, there is very little room for improvement of the binary classification model.

Table 5 shows the distribution of hate speech classes over the complete Slovenian Twitter dataset. We also provide a breakdown of the unacceptable speech class into its constituent subclasses: inappropriate, offensive, and violent. Offensive tweets are prevailing, inappropriate tweets are rare, and tweets inciting violence are very rare. There is also a considerable difference between the unacceptable original tweets and retweets. Offensive retweets are more frequent (an increase from 20% to 31%), while inappropriate and violent retweets are more rare in comparison to the original tweets.

Table 5. Distribution of hate speech classes across the original and the retweeted tweets.

Tweets	No. of tweets	Acceptable	Unacceptable		
			Inappropriate	Offensive	Violent
Original tweets	8,363,271	6,574,353 (79%)	88,813 (1.1%)	1,687,730 (20%)	12,375 (0.15%)
Retweets	4,597,865	3,146,906 (68%)	20,535 (0.4%)	1,427,477 (31%)	2,947 (0.06%)

https://doi.org/10.1371/journal.pone.0265602.t005

Communities and hate speech

The methods to analyze community evolution through time are described in detail in our related work [20]. They cover formation of retweet networks, community detection, measuring community similarity, selection of coarse-grained timepoints, various measures of influence, and identification of super-communities. In the current paper we use these methods to observe the development of hate speech on Slovenian Twitter during the years 2018–2020.

Fig 2 shows the top seven communities detected at the five selected timepoints. Each node represents a community, where its diameter is a cube-root of the community size (to stifle the large differences in sizes). An edge from the community C_i to C_j indicates average external influence of C_i to C_j in terms of tweets posted by C_i and retweeted by C_j. See subsection Retweet influence in Methods for definitions of various types of retweet influence.

The nodes (communities) and edges (external influence links) in Fig 2 form meta-networks. We call communities in meta-networks super-communities. In analogy to a network community, a super-community is a subset of detected communities more densely linked by external influence links than with the communities outside of the super-community. We use this informal definition to identify super-communities in our retweet networks. It is an open research problem, worth addressing in the future, to formalize the definition of super-communities and to design a multi-stage super-community detection algorithm.

In our case, in Fig 2, one can identify three super-communities: the political left-leaning (top), the Sports (middle), and the political right-leaning (bottom) super-community. The prevailing characterization and political orientation of the super-communities is determined by their constituent communities. A community is defined by its members, i.e., a set of Twitter users. A label assigned to a community is just a shorthand to characterize it by its most influential users [20], their types (see subsection Twitter users and hate speech), and tweets they post. Left and Right are generic communities with clear political orientation. SDS is a community

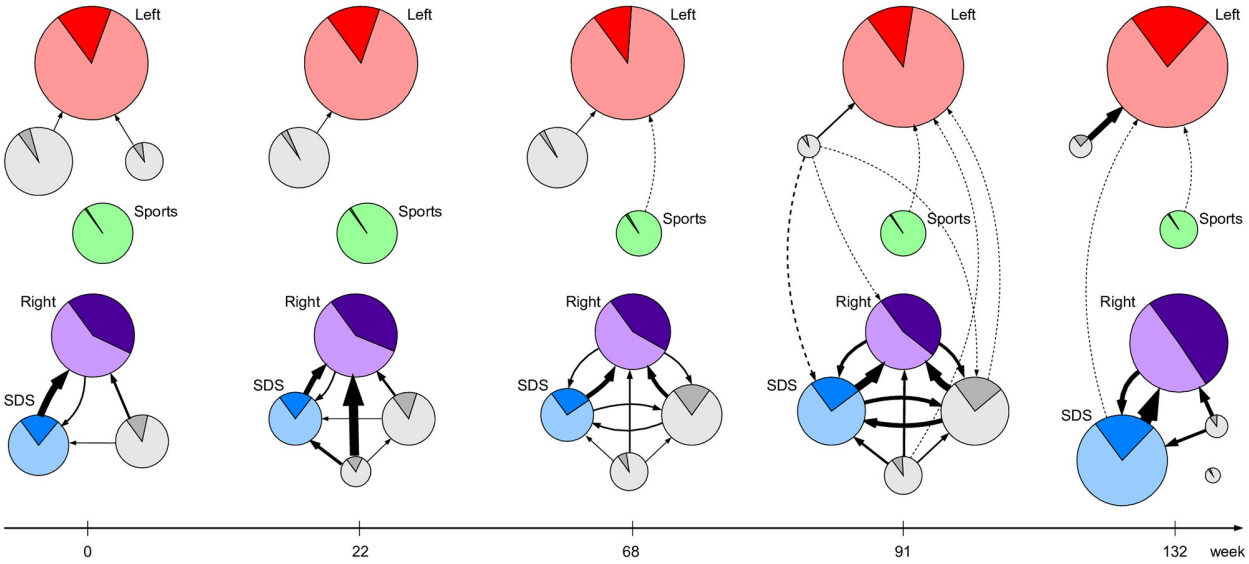

Fig 2. Fractions of unacceptable tweets posted by different communities. Nodes are the largest detected communities at timepoints $t = 0, 22, \ldots, 132$. The node size indicates the community size, darker areas correspond to unacceptable tweets, and lighter areas to acceptable tweets. An edge denotes the external influence of community C_i to C_j. Linked communities form super-communities: left-leaning (Left, top), Sports (middle), and right-leaning (Right and SDS, bottom).

https://doi.org/10.1371/journal.pone.0265602.g002

with a large share of its influential members being politicians and also members of the right-leaning SDS party (Slovenian Democratic Party).

While super-communities exhibit similar political orientation, their constituent communities are considerably different with respect to the hate speech they post. In the following, we compare in detail the largest left-leaning community Left (red), two right-leaning communities, namely Right (violet) and SDS (blue), and a non-political Sports community (green). Left and Right are consistently the largest communities on the opposite sides of the political spectrum. The SDS community was relatively small in the times of the left-leaning governments in Slovenia (until January 2020, $t = 0, 22, 68$), but become prominent after the right-wing government took over (in March 2020, $t = 91, 132$), at the same time as the emergence of the Covid-19 pandemic.

Communities in Fig 2 are assigned proportions of unacceptable tweets they post. Darker areas correspond to fractions of unacceptable tweets, and lighter areas correspond to fractions of acceptable original tweets. Several observations can be made. First, the prevailing Twitter activities are mostly biased towards political and ideological discussions, even during the emergence of the Covid-19 pandemic [21]. There is only one, relatively small, non-political community, Sports. Second, political polarization is increasing with time. Larger communities on the opposite poles grow even larger, and smaller communities are absorbed by them. There are barely any links between the left and right-leaning communities, a characteristics of the echo chambers and political polarization [38]. Third, the fraction of unacceptable tweets posted by the two largest communities, Left and Right, is increasing towards the end of the period. This is clearly visible in Fig 3.

Fig 3 shows the overall increase of unacceptable Twitter posts in the years 2018–2020. Regardless of the community, the fraction of unacceptable tweets in all posts (dotted black line) and in posts that were retweeted (solid black line), are increasing. The same holds for the

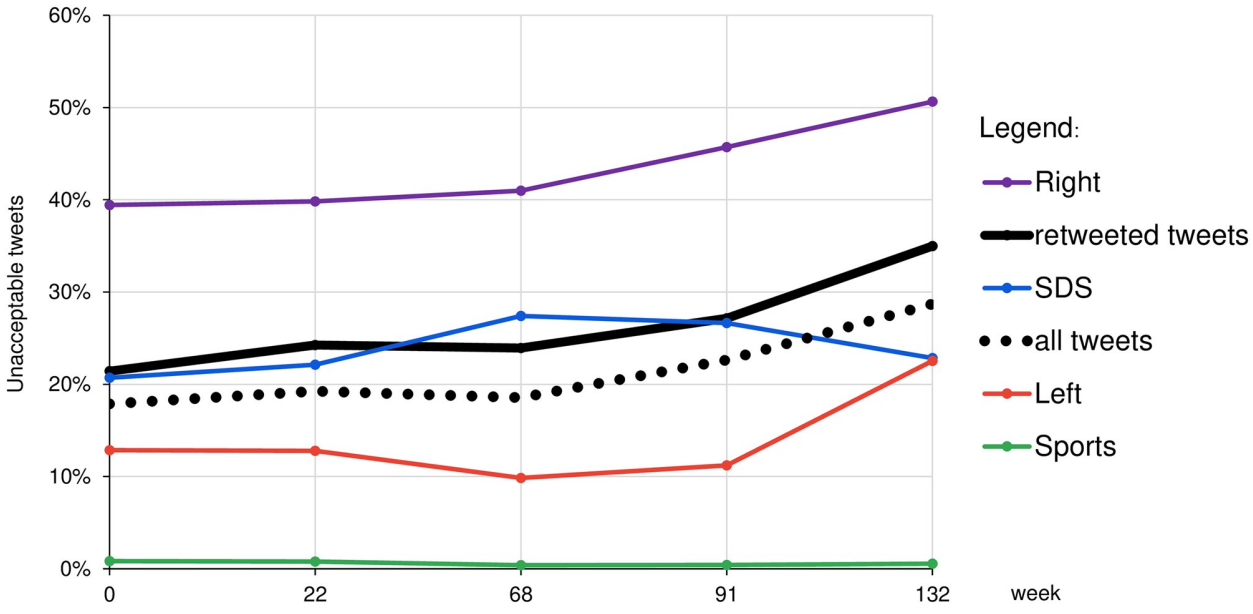

Fig 3. Fractions of unacceptable tweets posted by the major communities and overall, at weekly timepoints $t = 0, 22, \ldots, 132$. The solid black line represents all the tweets that were retweeted and are used to form retweet networks and communities. The dotted black line represents all the tweets posted. The largest communities are Left, Right, and SDS. For a comparison, we also show Sports, a small community with almost no unacceptable tweets.

https://doi.org/10.1371/journal.pone.0265602.g003

largest Left (red) and Right (violet) communities. However, the right-wing SDS community (blue), shows an interesting change of behaviour. During the left-wing governments in Slovenia, when the SDS party was in opposition (until March 2020, $t = 0, 22, 68$), the fraction of unacceptable tweets they posted was increasing. After SDS became the main right-wing government party (in March 2020, $t = 91, 132$), the fraction of unacceptable tweets they post is decreasing. By the end of 2020 ($t = 132$), SDS and the largest left-leaning community Left converge, both with about 23% of their posted tweets classified as unacceptable. Note that at the same time ($t = 132$), over 50% of the tweets by the Right community is unacceptable. For a comparison, there is also a non-political and non-ideological community Sports (green) with almost no unacceptable tweets.

Fig 4 shows the distribution of unacceptable tweets posted through time. We focus just on the three major communities, Left, Right and SDS. All the remaining communities are shown together as a single Small community (yellow). At any timepoint during the three years, the three major communities post over 80% of all the unacceptable tweets. By far the largest share is due to the Right community, about 60%. The Left and SDS communities are comparable, about 10–20%. However, the three communities are very different in size and in their posting activities.

Fig 5 clearly shows differences between the major communities. We compare the share of unacceptable tweets they post (the leftmost bar), the share of unacceptable tweets they retweet (the second bar from the left), the share of retweet influence (the total number of posted tweets that were retweeted, the third bar from the left), and the size of each community (the rightmost bar). The community shares are computed as the average shares over the five timepoints during the three year period.

The Right community (violet) exhibits disproportional share of unacceptable tweets and retweets w.r.t. its size. Its retweet influence share (the total number of posted tweets that were retweeted) is also larger than its size, which means that its members are more active. However, even w.r.t. to its influence, the share of unacceptable tweets and retweets is disproportional.

The Left community (red) is the most moderate of the three, in terms of unacceptable tweets and retweets. The shares of its posted tweets and retweet influence (weighted out-

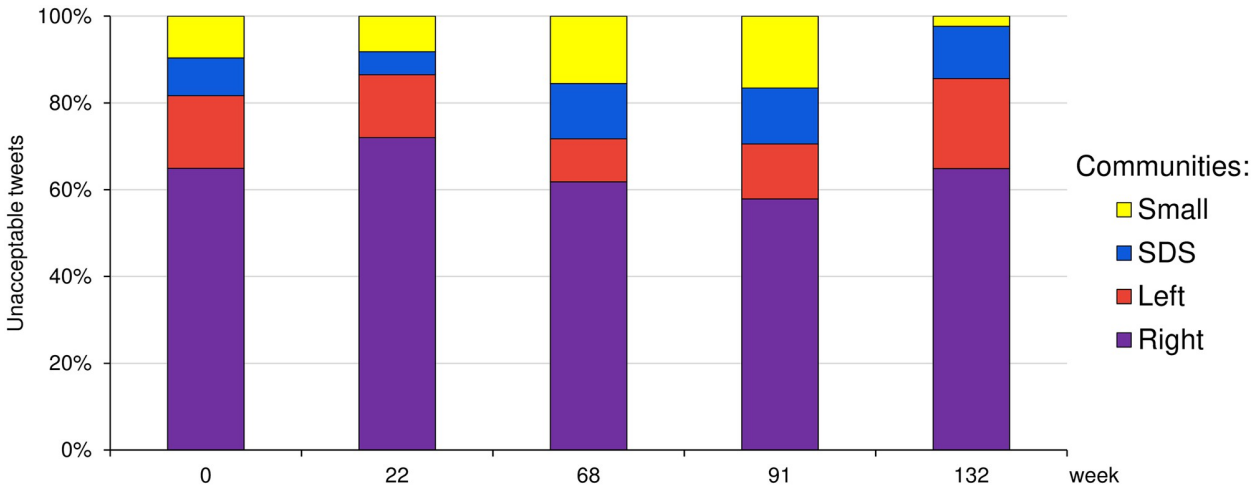

Fig 4. Distribution of posted unacceptable tweets between the three major communities through time. Left is the largest, left-leaning community, two right-leaning communities are Right and SDS, and Small denotes all the remaining smaller communities. Weekly timepoints are marked by $t = 0, 22, \ldots, 132$. The Right community posts the largest share of the unacceptable tweets, over 60% at four out of five timepoints. The Left and SDS communities are comparable, each with the share of about 10–20% of all unacceptable tweets posted.

https://doi.org/10.1371/journal.pone.0265602.g004

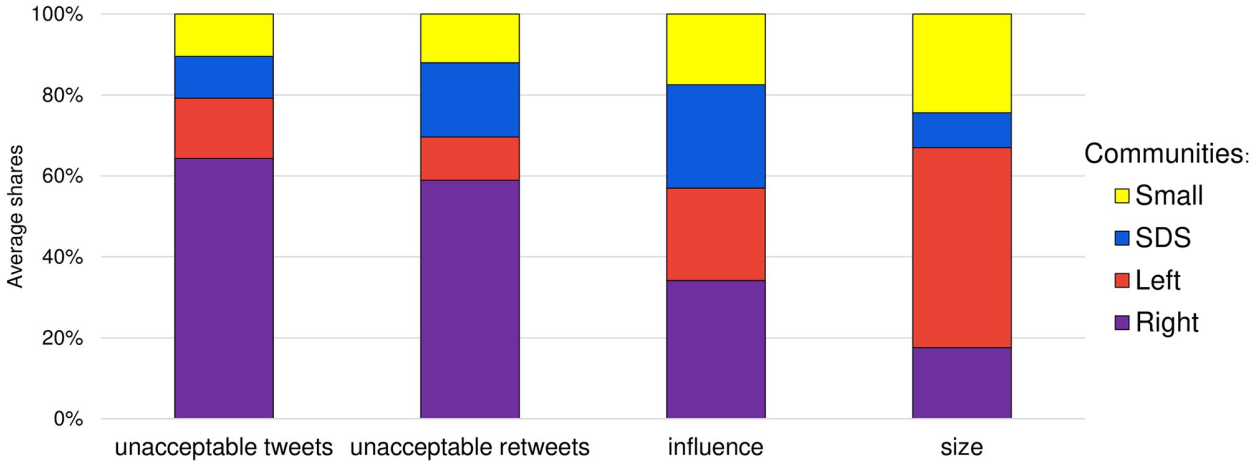

Fig 5. Comparison of the three major communities in terms of four different properties. Each bar is composed of the Right, Left, SDS, and the remaining Small communities, from bottom to top. Bars correspond to the average shares (over the five weekly timepoints) of posted unacceptable tweets, retweeted unacceptable tweets, community influence (weighted out-degree), and size of the community, from left-to-right, respectively.

https://doi.org/10.1371/journal.pone.0265602.g005

degree) w.r.t. its size, are lower in comparison to the Right and SDS communities. This indicates that its members are, on average, less active and less influential.

The SDS community (blue) posts about the same share of unacceptable tweets as is expected for its size. However, its share of unacceptable retweets is larger. It is also very active and the most influential of the three, and in this respect its share of unacceptable tweets posted is lower w.r.t. its influence share.

The differences between proportions of various community aspects can be quantified by Cohen's h [39]. Cohen's h quantifies the size of the difference, allowing one to decide if the difference is meaningful. Namely, the difference can be statistically significant, but too small to be meaningful. See subsection Comparing proportions in Methods for details. Table 6 gives the computed h values for the three major communities. The results are consistent with our interpretations of Fig 5 above.

Twitter users and hate speech

The analysis in the previous subsection points to the main sources of unacceptable tweets posted and retweeted at the community level. In this subsection, we shed some light on the composition of the major communities in terms of the user types and their individual influence.

We estimate a Twitter user influence by the retweet h-index [40], an adaptation of the well known Hirsch index [41] to Twitter. A user with a retweet index h posted h tweets and each of

Table 6. Comparison of the three major communities by Cohen's *h*. The headings denote the first property (the proportion p_1) vs. the second property (the proportion p_2). The values of h in the body have the following interpretation: positive sign of h shows that $p_1 > p_2$, and the value of $|h|$ indicates the effect size. In bold are the values of $h > 0.50$, indicating at least medium effect size.

Community	Unacc. tweets vs. Size	Unacc. retweets vs. Size	Influence vs. Size	Unacc. tweets vs. Influence
Right	**1.00**	**0.88**	0.38	**0.61**
Left	-0.77	-0.89	-0.56	-0.20
SDS	0.06	0.29	0.46	-0.41

https://doi.org/10.1371/journal.pone.0265602.t006

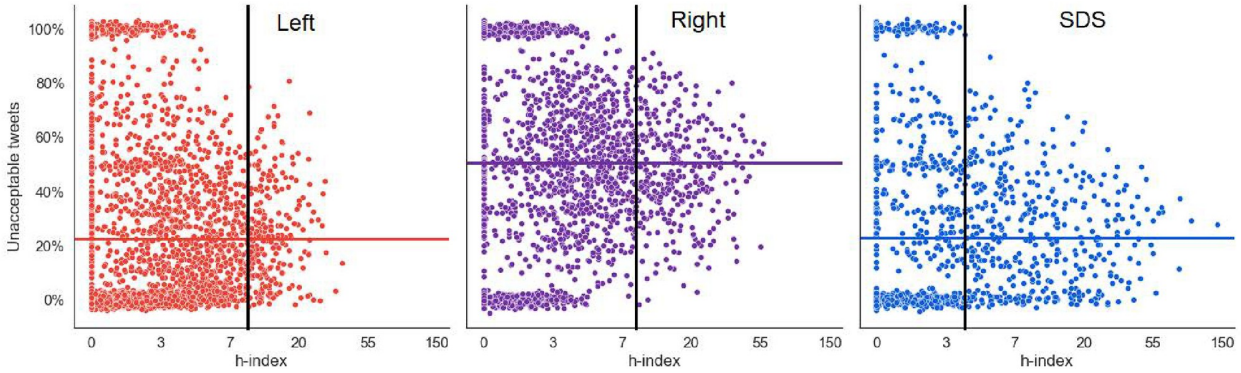

Fig 6. Scatter plot of the three major communities at the last timepoint, $t = 132$. Each point represents a Twitter user, with its retweet h-index and a fraction of unacceptable tweets posted. Horizontal lines show the average fraction of unacceptable tweets per community. Vertical bars delimit the low influence from the high influence Twitter users. For the most influential users, with an h-index right of the vertical bar, the user types are determined (see Table 8).

https://doi.org/10.1371/journal.pone.0265602.g006

them was retweeted at least h times. See subsection Retweet influence in Methods for details. It was already shown that members of the right-leaning super-community exhibit much higher h-index influence than the left-leaning users [20]. Also, influential users rarely switch communities, and when they do, they stay within their left- or right-leaning super-community.

In Fig 6 we show the distribution of Twitter users from the three major communities detected at the end of the three year period ($t = 132$). The scatter plots display individual users in terms of their retweet h-index (x-axis, logarithmic scale) and fraction of unacceptable tweets they post (y-axis). The average proportions of unacceptable tweets posted by the community members are displayed by horizontal lines. The results are consistent with Fig 3 at the last timepoint $t = 132$, where the Left, Right and SDS communities post 23%, 51% and 23% of unacceptable tweets, respectively. More influential users are at the right hand-side of the plots. Consistent with Fig 5, the members of the SDS and Right communities are considerably more influential than the members of the Left. In all the communities, there are clusters of users which post only unacceptable tweets (at the top), or only acceptable tweets (at the bottom). However, they are not very prolific nor do they have much impact, since their retweet h-index is very low. Vertical bars delimit the low influence from the high influence Twitter users.

Fig 6 shows that the distribution of influence in terms of retweet h-index is different between the three communities. We compute the concentration of influence by the Gini coefficient, a well-known measure of income inequality in economics [42]. Gini coefficient of 0 indicates perfectly equal distribution of influence, and Gini of 1 indicates the extreme, i.e., all the influence in concentrated in a single user. The results in Table 7 show that the highest concentration of influence is in the SDS community, followed by the Right, and that the Left community has more evenly distributed influence.

For the most influential Twitter users, right of the vertical bars in Fig 6, we inspect their type and their prevailing community during the whole time period. They are classified into

Table 7. Gini coefficients of influence distribution for the three major communities at the last timepoint, $t = 132$.

Community	Gini coefficient
Left	0.50
Right	0.57
SDS	0.64

https://doi.org/10.1371/journal.pone.0265602.t007

Table 8. Twitter user types and their prevailing communities. The top 890 users from the major communities, ranked by the retweet h-index, are classified into different types. When possible (in over 72% of the cases) the prevailing community across the five timepoints is determined. The rest of the users shift between different communities through time. There is an interesting transition community Left→SDS that corresponds to the government transition from the left-wing to the right-wing, and consists mostly of the governmental institutions.

User type subtype	Share	Prevailing community			
		Left	Right	SDS	Left→SDS
Individual	486 (55%)	137 (59%)	85 (38%)	104 (60%)	3 (20%)
Politician	143	20	16	72	3
Public figure	101	40	22	8	0
Journalist	100	41	12	11	0
Other	142	36	35	13	0
Organization	129 (14%)	41 (18%)	9 0(4%)	36 (21%)	12 (80%)
Institution	59	20	3	9	10
Media	46	16	4	15	1
Political party	24	5	2	12	1
Unverified	275 (31%)	55 (23%)	130 (58%)	32 (19%)	0 0(0%)
Anonymous	148	37	47	16	0
Closed	95	14	58	13	0
Suspended	32	4	25	3	0
Total	890 (99%)	233 (99%)	224 (99%)	172 (99%)	15 (99%)

https://doi.org/10.1371/journal.pone.0265602.t008

three major categories: Individual, Organization, and Unverified. The Unverified label is not meant as the opposite of the Twitter verification label, but just lumps together the users for which the identity was unclear (Anonymous), their accounts were closed (Closed) or suspended by Twitter (Suspended). The Individual and Organization accounts are further categorized into subtypes.

Table 8 provides the categorization of 890 users into types and subtypes. We selected the top users from the major communities, ranked by their retweet h-index. When the user did not switch between the communities (in 644 out of 890 cases, 72%) we assign its prevailing community across the whole time period from the community membership at individual timepoints. We introduce an additional transition community, Left→SDS, that encompasses Twitter accounts which switched from the Left community to the current government SDS community at the time of the government transition from the left-wing to the right-wing. This transition community consists mostly of governmental accounts (ministries, army, police, etc.) and demonstrates surprisingly well how detected communities in time reflect the actual changes in the political landscape.

The 890 users, classified into different types, represent less than 5% of all the users active on Slovenian Twitter during the three year period. However, they exhibit the bulk of the retweet influence. They posted almost 10% of all the original tweets, and, even more indicative, over 50% of all the retweeted tweets were authored by them. The details are given in Table 9.

Fig 7 shows how many unacceptable tweets are posted by different user types and subtypes. Over 40% of tweets posted by unverified accounts are unacceptable. In this category, the suspended accounts lead with over 50% of the tweets classified as unacceptable. This demonstrates

Table 9. Influential users. The share of influential users in terms of their number, the original tweets they post, and their tweets that were retweeted.

	Users		Original tweets		Retweeted tweets	
All	18,821		8,363,271		978,376	
Influential	890	(4.7%)	812,862	(9.7%)	529,110	(54.1%)

https://doi.org/10.1371/journal.pone.0265602.t009

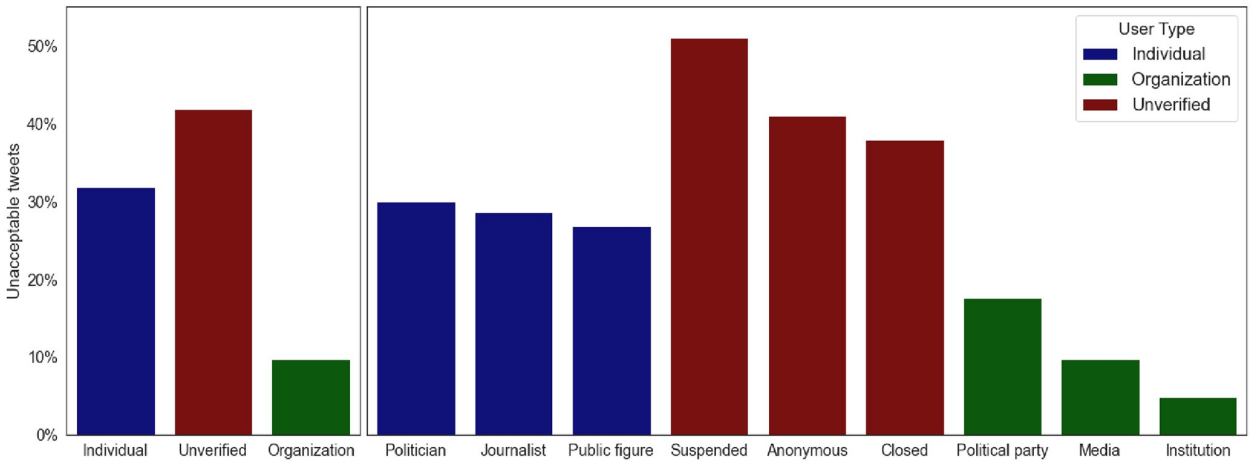

Fig 7. Fractions of unacceptable tweets posted by different types of users. The left bar chart shows major user types. Each major user type consists of three subtypes, shown at the right bar chart with the same color. Individual bars indicate the fraction of all tweets posted by the user (sub)type that are unacceptable.

https://doi.org/10.1371/journal.pone.0265602.g007

that Twitter is doing a reasonable job at suspending problematic accounts, and that our hate speech classification model is consistent with the Twitter criteria. Note that on the global level, Twitter is suspending an increasing number of accounts (776,000 accounts suspended in the second half of 2018, and one million accounts suspended in the second half of 2020).

Individual accounts, where politicians dominate between the influential users, post over 30% of tweets as unacceptable. Organizational accounts post mostly acceptable tweets (90%). In this category, accounts from the political parties dominate, with a fraction of 17% of their tweets being unacceptable. There is an interesting difference between the individual journalists and media organizations. Official media accounts post about 10% of tweets as unacceptable, while for the influential journalists, this fraction is 28%.

At the end, we can provide a link between the major retweet communities and the user types. We use Cohen's h again to quantify the differences between the representation of the main user types in the communities and their overall share. The community-specific proportions (first property) and the overall share (second property) of the main user types are taken from Table 8. The h values in Table 10 then quantify which user types post disproportional fractions of unacceptable tweets (first column in Table 10), and in which communities are they disproportionately represented (columns 2-4 in Table 10).

Results in Table 10 confirm that the Unverified accounts produce a disproportionate fraction of unacceptable tweets, and that they are considerably over-represented in the Right community. On the other hand, Individual and Organization accounts are under-represented in the Right community.

Table 10. Comparison of the three major user types by Cohen's *h*. The headings denote the first property (the proportion p_1) vs. the second property (the proportion p_2). The values of h in the body have the following interpretation: positive sign of h shows that $p_1 > p_2$, and the value of $|h|$ indicates the effect size. In bold is the value of $h > 0.50$, indicating at least medium effect size.

User type	Unacc. tweets vs. Share	Left vs. Share	Right vs. Share	SDS vs. Share
Individual	-0.04	0.08	-0.34	0.12
Organization	-0.26	0.08	-0.38	0.17
Unverified	0.21	-0.16	**0.55**	-0.29

https://doi.org/10.1371/journal.pone.0265602.t010

Conclusions

Retweets play an important role in revealing social ties between the Twitter users. They allow for the detection of communities of like-minded users and super-communities linked by the retweet influence links. In our Slovenian Twitter dataset, the two main super-communities show clear political polarization between the left and right-leaning communities [20]. The right-leaning communities are closely linked, and exhibit significantly higher retweet influence than the left-leaning communities. This is consistent with the findings about the European Parliament [43] and polarization during the Brexit referendum [40]. However, in terms of hate speech, the super-communities are not homogeneous, and there are large differences between the communities themselves.

Regarding the hate speech classification, we demonstrate that the best model reaches the inter-annotator agreement. This means that a model with such level of performance can replace a human annotator, and that without additional information, the model cannot be improved much. The additional information, if properly taken into account, might be in the form of a context. Textual context, such as previous tweets or a thread, is difficult to incorporate in the machine learning models. The user context, on the other hand, can provide additional features about the user history and community membership, and seems very relevant and promising for better hate speech classification.

Our hate speech classification model distinguishes between three classes of hate speech on Twitter: inappropriate, offensive, and violent. Specially tweets inciting violence, and directed towards specific target groups, are essential to detect since they may be subject to legal actions. However, in our training data sample of 50,000 tweets, the annotators found only a few 100 cases of violent tweets. The evaluation results show that the model cannot reliably detect violent hate speech, therefore we classified all three classes of hate speech together, as unacceptable tweets. Our previous experience in learning Twitter sentiment models for several languages [36] shows that one needs several 1,000 labelled tweets to construct models which approach the quality of human annotators. This calls for additional sampling of a considerably larger set of potentially violent tweets, which should be properly annotated and then used for model training.

Another dimension of hate speech analysis are the topics which are discussed. The results of topic detection on Slovenian Twitter show that political and ideological discussions are prevailing, accounting for almost 45% of all the tweets [21]. The sports-related topic, for example, is subject of only about 12% of all the tweets, and 90% of them are acceptable. This is also consistent with the very low fraction of unacceptable tweets posted by the Sports community in Fig 3. The distribution of topics within the detected communities, the levels of topic-related hate speech, and the evolution through time are some of the interesting results reported in [21].

We identify one, right-leaning, community of moderate size which is responsible for over 60% of unacceptable tweets. In addition, we show that this community consists of a disproportional share of anonymous, suspended, or already closed Twitter accounts which are the main source of hate speech. The other right-leaning community, corresponding to the main party of the current right-wing Slovenian government, shows more moderation, in particular after it took over from the left-wing government in March 2020. While these results are specific for the Slovenian tweetosphere, there are two lessons important for other domains and languages. One is the concept of super-communities which can be identified after the standard community detection process [20, 44], and share several common properties of the constituent communities. Another is the insight that hate speech is not always evenly spread within a super-community, and that it is important to analyze individual communities and different types of users.

Methods

Data collection

The three years of comprehensive Slovenian Twitter data cover the period from January 1, 2018 until December 28, 2020. In total, 12,961,136 tweets were collected, indirectly through the public Twitter API. The data collection and data sharing complies with the terms and conditions of Twitter. We used the TweetCaT tool [45] for Twitter data acquisition.

The TweetCaT tool is specialized on harvesting Twitter data of less frequent languages. It searches continuously for new users that post tweets in the language of interest by querying the Twitter Search API for the most frequent and unique words in that language. Once a set of new potential users posting in the language of interest are identified, their full timeline is retrieved and the language identification is run over their timeline. If it is evident that specific users post predominantly in the language of interest, they are added to the user list and their tweets are being collected for the remainder of the collection period. In the case of Slovenian Twitter, the collection procedure started in August 2017 and is still running. As a consequence, we are confident that the full Slovenian tweetosphere is covered in the period of this analysis.

Comparing proportions

Odds ratio and Cohen's h are two measures of association and effect size of two events. Odds ratio can be used when both events are characterized by jointly exhaustive and mutually exclusive partitioning of the sample. Cohen's h is used to compare two independent proportions.

Odds ratio. An odds ratio is a statistic that quantifies the strength of the association between two events. The (natural logarithm of the) odds ratio L of a sample, and its approximate standard error SE are defined as:

$$L = ln\left(\frac{n_{11} \cdot n_{00}}{n_{10} \cdot n_{01}}\right), \quad SE = \sqrt{\frac{1}{n_{11}} + \frac{1}{n_{00}} + \frac{1}{n_{10}} + \frac{1}{n_{01}}},$$

where n_{ij} are the elements of a 2×2 contingency table. A non-zero log odds ratio indicates correlation between the two events, and the standard error is used to determine its significance.

Cohen's *h*. The difference between two independent proportions (probabilities) can be quantified by Cohen's h [39]. For two proportions, p_1 and p_2, Cohen's h is defined as the difference between their "arcsine transformations":

$$h = 2\arcsin\sqrt{p_1} - 2\arcsin\sqrt{p_2}.$$

The sign of h shows which proportion is greater, and the magnitude indicates the effect size. Cohen [39, p. 184–185] provides the following rule of thumb interpretation of h: 0.20–small effect size, 0.50–medium effect size, and 0.80–large effect size.

Retweet networks and community detection

Twitter provides different forms of interactions between the users: follows, mentions, replies, and retweets. The most useful indicator of social ties between the Twitter users are retweets [44, 46]. When a user retweets a tweet, it is distributed to all of its followers, just as if it were an originally authored tweet. Users retweet content that they find interesting or agreeable.

A retweet network is a directed graph. The nodes are Twitter users and edges are retweet links between the users. An edge is directed from the user A who posts a tweet to the user B who retweets it. The edge weight is the number of retweets posted by A and retweeted by B.

For the whole three year period of Slovenian tweets, there are in total 18,821 users (nodes) and 4,597,865 retweets (sum of all weighted edges).

To study dynamics of the retweet networks, we form several network snapshots from our Twitter data. In particular, we select a network observation window of 24 weeks (about six months), with a sliding window of one week. This provides a relatively high temporal resolution between subsequent networks, but in the next subsection Selection of timepoints we show how to select the most relevant intermediate timepoints. Additionally, in order to eliminate the effects of the trailing end of a moving network snapshot, we employ an exponential edge weight decay, with half-time of 4 weeks.

The set of network snapshots thus consists of 133 overlapping observation windows, with temporal delay of one week. The snapshots start with a network at $t = 0$ (January 1, 2018–June 18, 2018) and end with a network at $t = 132$ (July 13, 2020–December 28, 2020) (see Fig 1).

Informally, a network community is a subset of nodes more densely linked between themselves than with the nodes outside the community. A standard community detection method is the Louvain algorithm [47]. Louvain finds a partitioning of the network into communities, such that the modularity of the partition is maximized. However, there are several problems with the modularity maximization and stability of the Louvain results [48]. We address the instability of Louvain by applying the **Ensemble Louvain** algorithm [20, 49]. We run 100 trials of Louvain and compose communities with nodes that co-occur in the same community above a given threshold, 90% of the trials in our case. This results in relatively stable communities of approximately the same size as produced by individual Louvain trials. We run the Ensemble Louvain on all the 133 undirected network snapshots, resulting in 133 network partitions, each with slightly different communities.

Selection of timepoints

There are several measures to evaluate and compare network communities. We use the BCubed measure, extensively evaluated in the context of clustering [50]. BCubed decomposes evaluation into calculation of precision and recall of each node in the network. The precision (*Pre*) and recall (*Rec*) are then combined into the F_1 score, the harmonic mean:

$$F_1 = 2\,\frac{Pre \cdot Rec}{Pre + Rec}.$$

Details of computing *Pre* and *Rec* for individual nodes, communities and network partitions are in [20]. We write $F_1(P_i|P_j)$ to denote the F_1 difference between the partitions P_i and P_j. The paper also provides a sample comparison of BCubed with the Adjusted Rand Index (ARI) [51] and Normalized Mutual Information (NMI) [52]. Our F_1 score extends the original BCubed measure to also account for new and disappearing nodes, and is different and more general than the F_1 score proposed by Rossetti [53].

The weekly differences between the network partitions are relatively small. The retweet network communities do not change drastically at this relatively high time resolution. Moving to lower time resolution means choosing timepoints which are further apart, and where the network communities exhibit more pronounced differences.

We formulate the timepoint selection task as follows. Let us assume that the initial and final timepoints are fixed (at $t = 0$ and $t = n$), with the corresponding partitions P_0 and P_n, respectively. For a given k, select k intermediate timepoints such that the differences between the corresponding partitions are maximized. The number of possible selections grows exponentially with k. Therefore, we implement a simple heuristic algorithm which finds the k (non-optimal) timepoints. The algorithm works top-down and starts with the full, high resolution timeline

with $n + 1$ timepoints, $t = 0, 1, \ldots, n$ and corresponding partitions P_t. At each step, it finds a triplet of adjacent partitions P_{t-1}, P_t, P_{t+1} with minimal differences (i.e., maximum F_1 scores):

$$max(F_1(P_t|P_{t-1}) + F_1(P_{t+1}|P_t)).$$

The partition P_t is then eliminated from the timeline:

$$P_0, \ldots, P_{t-1}, P_t, P_{t+1}, \ldots, P_n \mapsto P_0, \ldots, P_{t-1}, P_{t+1}, \ldots, P_n.$$

At the next step, the difference $F_1(P_{t+1}|P_{t-1})$ fills the gap of the eliminated timepoint P_t. The step is repeated until there are k (non-optimal) intermediate timepoints. The heuristic algorithm thus requires $n - 1 - k$ steps.

For our retweet networks, we fix $k = 3$, which provides much lower, but still meaningful time resolution. This choice results in a selection of five network partitions P_t at timepoints $t = 0, 22, 68, 91, 132$.

Retweet influence

Twitter users differ in how prolific they are in posting tweets, and in the impact these tweets make on the other users. One way to estimate the influence of Twitter users is to consider how often their tweets are retweeted. Similarly, the influence of a community can be estimated by the total number of retweets of tweets posted by its members. Retweets within the community indicate **internal influence**, and retweets outside of the community indicate **external influence** [44, 54].

Let W_{ij} denote the sum of all weighted edges between communities C_i and C_j. The average community influence I is defined as:

$$I(C_i) = \frac{\sum_j W_{ij}}{|C_i|},$$

i.e., the weighted out-degree of C_i, normalized by its size. The influence I consists of the internal I_{int} and external I_{ext} component, $I = I_{int} + I_{ext}$, where

$$I_{int}(C_i) = \frac{W_{ii}}{|C_i|},$$

and

$$I_{ext}(C_i, C_j) = \frac{\sum_{i \neq j} W_{ij}}{|C_i|}.$$

We compute internal and external influence of the retweet communities detected at the selected timepoints $t = 0, 22, 68, 91, 132$. Fig 2 shows the communities and the external influence links between the detected communities. One can observe a formation of super-communities, with closely linked communities. There are two super-communities, the political left-leaning and right-leaning, and an apolitical Sports.

Weighted out-degree is a useful measure of influence for communities. For individual Twitter users, a more sophisticated measure of influence is used. The user influence is estimated by their **retweet h-index** [40, 55], an adaptation of the well known Hirsch index [41] to Twitter. The retweet h-index takes into account the number of tweets posted, as well as the impact of individual tweets in terms of retweets. A user with an index of h has posted h tweets and each of them was retweeted at least h times.

Author Contributions

Conceptualization: Igor Mozetič.

Data curation: Nikola Ljubešić.

Software: Bojan Evkoski, Andraž Pelicon.

Supervision: Petra Kralj Novak.

Validation: Andraž Pelicon.

Visualization: Bojan Evkoski.

Writing – original draft: Igor Mozetič.

References

1. Bayer J, Bárd P. Hate speech and hate crime in the EU and the evaluation of online content regulation approaches. Directorate-General for Internal Policies, European Union; 2020. Available from: https://www.europarl.europa.eu/RegData/etudes/STUD/2020/655135/IPOL_STU(2020)655135_EN.pdf.
2. MacAvaney S, Yao HR, Yang E, Russell K, Goharian N, Frieder O. Hate speech detection: Challenges and solutions. PloS ONE. 2019; 14(8):e0221152. https://doi.org/10.1371/journal.pone.0221152 PMID: 31430308
3. Basile V, Bosco C, Fersini E, Debora N, Patti V, Pardo FMR, et al. Semeval-2019 Task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In: Proc. 13th International Workshop on Semantic Evaluation. ACL; 2019. p. 54–63.
4. Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In: Proc. 13th International Workshop on Semantic Evaluation. ACL; 2019. p. 75–86. Available from: https://www.aclweb.org/anthology/S19-2010.
5. Zampieri M, Nakov P, Rosenthal S, Atanasova P, Karadzhov G, Mubarak H, et al. SemEval-2020 Task 12: Multilingual offensive language identification in social media (OffensEval); 2020. Available from: https://arxiv.org/abs/2006.07235.
6. Matamoros-Fernández A, Farkas J. Racism, Hate Speech, and Social Media: A Systematic Review and Critique. Television & New Media. 2021; 22(2):205–224. https://doi.org/10.1177/1527476420982230
7. Waseem Z, Hovy D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: Proc. NAACL Student Research Workshop. ACL; 2016. p. 88–93. Available from: https://www.aclweb.org/anthology/N16-2013.
8. Bretschneider U, Peters R. Detecting cyberbullying in online communities. In: Proc. 24th European Conference on Information Systems (ECIS). Istanbul, Turkey; 2016.
9. Wulczyn E, Thain N, Dixon L. Ex Machina: Personal Attacks Seen at Scale. In: Proc. 26th International Conference on World Wide Web; 2017. p. 1391–1399. Available from: https://doi.org/10.1145/3038912.3052591.
10. Davidson T, Warmsley D, Macy M, Weber I. Automated hate speech detection and the problem of offensive language. In: Proc. International AAAI Conference on Web and Social Media. vol. 11; 2017.
11. Ross B, Rist M, Carbonell G, Cabrera B, Kurowsky N, Wojatzki M. Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. In: Proc. 3rd Workshop on Natural Language Processing for Computer-Mediated Communication; 2016.
12. Pavlopoulos J, Malakasiotis P, Androutsopoulos I. Deeper Attention to Abusive User Content Moderation. In: Proc. 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2017. p. 1125–1135. Available from: https://aclanthology.info/papers/D17-1117/d17-1117.
13. Ljubešić N, Erjavec T, Fišer D. Datasets of Slovene and Croatian Moderated News Comments. In: Proc. 2nd Workshop on Abusive Language Online (ALW2); 2018. p. 124–131.
14. Mubarak H, Darwish K, Magdy W. Abusive Language Detection on Arabic Social Media. In: Proc. 1st Workshop on Abusive Language Online. ACL; 2017. p. 52–56. Available from: https://www.aclweb.org/anthology/W17-3008.
15. Mathew B, Dutt R, Goyal P, Mukherjee A. Spread of hate speech in online social media. In: Proc. 10th ACM conference on web science; 2019. p. 173–182.

16. Mathew B, Illendula A, Saha P, Sarkar S, Goyal P, Mukherjee A. Hate begets hate: A temporal study of hate speech. In: Proc. ACM on Human-Computer Interaction. vol. 4; 2020. p. 1–24.

17. Uyheng J, Carley KM. Characterizing network dynamics of online hate communities around the COVID-19 pandemic. Applied Network Science. 2021; 6(20). https://doi.org/10.1007/s41109-021-00362-x PMID: 33718589

18. Ribeiro M, Calais P, Santos Y, Almeida V, Meira Jr W. Characterizing and detecting hateful users on Twitter. In: Proc. International AAAI Conference on Web and Social Media. vol. 12; 2018.

19. Sood S, Antin J, Churchill E. Profanity use in online communities. In: Proc. SIGCHI Conference on Human Factors in Computing Systems; 2012. p. 1481–1490.

20. Evkoski B, Mozetič I, Ljubešić N, Novak PK. Community evolution in retweet networks. PLoS ONE. 2021; 16(9):e0256175. https://doi.org/10.1371/journal.pone.0256175 PMID: 34469456

21. Evkoski B, Ljubešić N, Pelicon A, Mozetič I, Novak PK. Evolution of topics and hate speech in retweet network communities. Applied Network Science. 2021; 6(96). https://doi.org/10.1007/s41109-021-00439-7 PMID: 34957317

22. Cinelli M, Cresci S, Galeazzi A, Quattrociocchi W, Tesconi M. The limited reach of fake news on Twitter during 2019 European elections. PLoS ONE. 2020; 15(6):e0234689. https://doi.org/10.1371/journal.pone.0234689 PMID: 32555659

23. Bollen J, Mao H, Zeng X. Twitter mood predicts the stock market. Journal of Computational Science. 2011; 2(1):1–8. https://doi.org/10.1016/j.jocs.2010.12.007

24. Gil de Zúñiga H, Koc Michalska K, Römmele A. Populism in the era of Twitter: How social media contextualized new insights into an old phenomenon. New Media & Society. 2020; 22(4):585–594. https://doi.org/10.1177/1461444819893978

25. Wu S, Hofman JM, Mason WA, Watts DJ. Who says what to whom on Twitter. In: Proc. 20th International Conference on World Wide Web; 2011. p. 705–714.

26. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y. Abusive language detection in online user content. In: Proc. 25th International Conference on World Wide Web; 2016. p. 145–153.

27. Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R. Predicting the type and target of offensive posts in social media. In: Proc. 2019 Conference NAACL. ACL; 2019. p. 1415–1420.

28. Ljubešiš N, Fišer D, Erjavec T. The FRENK Datasets of Socially Unacceptable Discourse in Slovene and English; 2019. Available from: https://arxiv.org/abs/1906.02045.

29. Novak PK, Mozetič I, Pauw GD, Cinelli M. IMSyPP deliverable D2.1: Multilingual hate speech database; 2021. Available from: http://imsypp.ijs.si/wp-content/uploads/IMSyPP-D2.1-Hate-speech-DB.pdf.

30. Krippendorff K. Content Analysis, An Introduction to its methodology. 4th ed. Thousand Oaks, CA, USA: Sage Publications; 2018.

31. Van Rijsbergen CJ. Information Retrieval. 2nd ed. Newton, MA, USA: Butterworth; 1979.

32. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. 2019 Conference NAACL: Human Language Technologies, vol. 1. ACL; 2019. p. 4171–4186.

33. Ulčar M, Robnik-Šikonja M. FinEst BERT and CroSloEngual BERT. In: International Conference on Text, Speech, and Dialogue (TSD); 2020. p. 104–111.

34. Mozetič I, Torgo L, Cerqueira V, Smailović J. How to evaluate sentiment classifiers for Twitter time-ordered data? PLoS ONE. 2018; 13(3):e0194317. https://doi.org/10.1371/journal.pone.0194317 PMID: 29534112

35. Benavoli A, Corani G, Demšar J, Zaffalon M. Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis. The Journal of Machine Learning Research. 2017; 18(1):2653–2688.

36. Mozetič I, Grčar M, Smailović J. Multilingual Twitter sentiment classification: The role of human annotators. PLoS ONE. 2016; 11(5):e0155036. https://doi.org/10.1371/journal.pone.0155036 PMID: 27149621

37. Cinelli M, Pelicon A, Mozetič I, Quattrociocchi W, Novak PK, Zollo F. Dynamics of online hate and misinformation. Scientific Reports. 2021; 11(22083). https://doi.org/10.1038/s41598-021-01487-w PMID: 34764344

38. Del Vicario M, Vivaldo G, Bessi A, Zollo F, Scala A, Caldarelli G, et al. Echo chambers: Emotional contagion and group polarization on Facebook. Scientific Reports. 2016; 6(37825). https://doi.org/10.1038/srep37825 PMID: 27905402

39. Cohen J. Statistical power analysis for the behavioral sciences. 2nd ed. Routledge; 1988.

40. Grčar M, Cherepnalkoski D, Mozetič I, Kralj Novak P. Stance and influence of Twitter users regarding the Brexit referendum. Computational Social Networks. 2017; 4(1):6. https://doi.org/10.1186/s40649-017-0042-6 PMID: 29266132

41. Hirsch JE. An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences. 2005; 102(46):16569–16572. https://doi.org/10.1073/pnas.0507655102 PMID: 16275915

42. Gini C. On the measure of concentration with special reference to income and statistics. Colorado College Publication, General Series. 1936; 208(1):73–79.

43. Cherepnalkoski D, Karpf A, Mozetič I, Grčar M. Cohesion and coalition formation in the European Parliament: Roll-call votes and Twitter activities. PLoS ONE. 2016; 11(11):e0166586. https://doi.org/10.1371/journal.pone.0166586 PMID: 27835683

44. Durazzi F, Müller M, Salathé M, Remondini D. Clusters of science and health related Twitter users become more isolated during the COVID-19 pandemic. Scientific Reports. 2021; 11(19655). https://doi.org/10.1038/s41598-021-99301-0 PMID: 34608258

45. Ljubešić N, Fišer D, Erjavec T. TweetCaT: A tool for building Twitter corpora of smaller languages. In: Proc. 9th Intl. Conf. on Language Resources and Evaluation. ELRA; 2014. p. 2279–2283. Available from: http://www.lrec-conf.org/proceedings/lrec2014/pdf/834_Paper.pdf.

46. Cherepnalkoski D, Mozetič I. Retweet networks of the European Parliament: Evaluation of the community structure. Applied Network Science. 2016; 1(1):2. https://doi.org/10.1007/s41109-016-0001-4 PMID: 30533494

47. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008;(10). https://doi.org/10.1088/1742-5468/2008/10/P10008

48. Fortunato S, Hric D. Community detection in networks: A user guide. Physics Reports. 2016; 659:1–44. https://doi.org/10.1016/j.physrep.2016.09.002

49. Evkoski B, Mozetič I, Novak PK. Community evolution with Ensemble Louvain. In: Complex Networks 2021, Book of abstracts; 2021. p. 58–60.

50. Amigó E, Gonzalo J, Artiles J, Verdejo F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval. 2009; 12(4):461–486. https://doi.org/10.1007/s10791-008-9066-8

51. Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985; 2(1):193–218. https://doi.org/10.1007/BF01908075

52. Danon L, Díaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment. 2005;(9). https://doi.org/10.1088/1742-5468/2005/09/P09008

53. Rossetti G, Pappalardo L, Rinzivillo S. A novel approach to evaluate community detection algorithms on ground truth. In: 7th Workshop on Complex Networks; 2016.

54. Sluban B, Smailović J, Battiston S, Mozetič I. Sentiment leaning of influential communities in social networks. Computational Social Networks. 2015; 2(1):9. https://doi.org/10.1186/s40649-015-0016-5

55. Gallagher RJ, Doroshenko L, Shugars S, Lazer D, Welles BF. Sustained online amplification of COVID-19 elites in the United States. Social Media + Society. 2021; 7(2):20563051211024957. https://doi.org/10.1177/20563051211024957

3.3 Evolution of Topics and Hate Speech in Retweet Network Communities

"Evolution of topics and hate speech in retweet network communities" is the last of the trilogy of papers that focus on community evolution analysis on the Slovenian tweetosphere in the years from 2018 to 2021. In this final work, Bojan Evkoski, Nikola Ljubešić, Andraž Pelicon, Igor Mozetič, and Petra Kralj Novak, investigate the relationship between topics discussed and hate speech exhibited, the respect to communities discovered and tracked via community evolution.

Using topic modeling, we detect six broad topics: health, family, politics, ideology, local, and sports. Utilizing the information on community memberships, as well as the hatefulness and topics of tweets, we draw the following conclusions: politics and ideology are the prevailing topics despite the emergence of the Covid-19 pandemic; the same two topics attract the highest proportion of hateful tweets; while the membership of retweet communities changes, the topic distribution remains stable; finally, the detected super-communities are very different in terms of the discussion topics, with the right-leaning ones showing more interest in politics and ideology.

The author of the master thesis contributed to this work by performing most of the experiments and producing the figures. He did not take part in the topic and hate speech modeling.

Evkoski *et al. Applied Network Science* 2021, **6**(1):96
https://doi.org/10.1007/s41109-021-00439-7

Applied Network Science

RESEARCH Open Access

Evolution of topics and hate speech in retweet network communities

Bojan Evkoski[1,2], Nikola Ljubešić[1,3], Andraž Pelicon[1,2], Igor Mozetič[1*] and Petra Kralj Novak[1,4]

*Correspondence:
igor.mozetic@ijs.si
[1] Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia
Full list of author information is available at the end of the article

Abstract

Twitter data exhibits several dimensions worth exploring: a network dimension in the form of links between the users, textual content of the tweets posted, and a temporal dimension as the time-stamped sequence of tweets and their retweets. In the paper, we combine analyses along all three dimensions: temporal evolution of retweet networks and communities, contents in terms of hate speech, and discussion topics. We apply the methods to a comprehensive set of all Slovenian tweets collected in the years 2018–2020. We find that politics and ideology are the prevailing topics despite the emergence of the Covid-19 pandemic. These two topics also attract the highest proportion of unacceptable tweets. Through time, the membership of retweet communities changes, but their topic distribution remains remarkably stable. Some retweet communities are strongly linked by external retweet influence and form super-communities. The super-community membership closely corresponds to the topic distribution: communities from the same super-community are very similar by the topic distribution, and communities from different super-communities are quite different in terms of discussion topics. However, we also find that even communities from the same super-community differ considerably in the proportion of unacceptable tweets they post.

Keywords: Twitter, Retweet networks, Network communities, Community evolution, Hate speech classification, Topic detection

Introduction

Social media, and Twitter in particular, are widely used to study various social phenomena, see for example (Wu et al. 2011; Bollen et al. 2011; Gil de Zúñiga et al. 2020; Cinelli et al. 2020). Network analyses play an important role in these studies since social media exhibit typical network properties. Collective behaviour is captured by the network communities, defined as groups of densely connected users. Changes in the behaviour of groups are referred to as community evolution (Dakiche et al. 2019). Temporal analyses provide insights into the patterns and developments of the social media landscape, and are increasingly relevant in modern analyses of complex networks (Rossetti and Cazabet 2018).

Temporal network analysis

There are several approaches to temporal network analyses, one of them is taking temporally ordered series of network snapshots. This approach allows for efficient tracking of changes in the network structure, thus increasing the expressiveness of the models, but at a cost of higher analytical complexity (Rossetti and Cazabet 2018). The snapshot approach depends on the representation of time in the networks, e.g., the limited memory scenario allows for nodes/edges to disappear over time. This is suitable in social network analysis, where the edge disappearance indicates possible decay of social ties. In our approach, we create overlapping snapshots of the network through time, detect communities in each snapshot, and then track evolution of relevant communities over time.

An issue in dynamic community evolution is how community detection is applied to the network snapshots (Aynaud et al. 2013; Hartmann et al. 2016; Masuda and Lambiotte 2016; Dakiche et al. 2019; Rossetti and Cazabet 2018). The problem is the instability of community detection algorithms (Aynaud and Guillaume 2010). To address this issue, we developed the Ensemble Louvain algorithm which considerably improves the stability of the well-known Louvain algorithm for community detection (Evkoski et al. 2021a).

Hate speech detection

Hate speech in online media is among the "online harms" that are pressing concerns of policymakers, regulators and big tech companies. There is an increasing research interest in the automated hate speech detection, with organized competitions and workshops (MacAvaney et al. 2019). Hate speech detection is usually addressed as a supervised classification problem, where models are trained to distinguish between examples of hate and normal speech. A systematic literature review of academic articles on hate speech on social media, between 2014 and 2018 (Matamoros-Fernández and Farkas 2021), found that research was limited to text-based analyses of racist hate speech, to the Twitter platform, and to the content mostly from the U.S.

There is not much research addressing hate speech in terms of temporal aspects and community structure on Twitter. The most similar work was done on the social media platform Gab (https://Gab.com) (Mathew et al. 2019, 2020). The authors find that the content posted by the hateful users spreads faster and further, and that they are more densely connected between themselves. The amount of hate speech on Gab is steadily increasing and hateful users are occupying more prominent positions in the Gab network. Our research addresses very similar questions on the Twitter platform and most of our results are aligned with the findings on Gab. However, there are some important differences. Twitter is a mainstream social medium, used by public figures and organizations, while Gab is an alt-tech social network, with a far-right user base, described as a haven for extremists.

In Uyheng and Carley (2021) the authors propose a dynamic network framework to characterize hate communities, focusing on Twitter conversations related to Covid-19. Higher levels of community hate are consistently associated with smaller, more isolated, and highly hierarchical network communities. The identity analysis reveals that hate speech in the U.S. initially targets political figures and then becomes predominantly

racially charged, while in the Philippines, the targets of hate speech over time remain political. Another study of political affiliations and profanity use (Sood et al. 2012) finds that a political comment is more likely profane and contains an insult than a non-political comment. These results are similar to our findings that politics and ideology attract the highest proportions of unacceptable tweets.

Topic detection

In a typical simplistic analysis of the content on Twitter, hashtags posted in tweets are used as semantic indicators. A more advanced approach represents tweets as bag-of-words and then applies k-means clustering to group together tweets about similar topics. We take a more sophisticated approach to topic modeling by applying a variant of Latent Dirichlet Allocation (Blei et al. 2003), named probabilistic topic models (Steyvers and Griffiths 2007). The approach is based on the assumptions that semantic information can be derived from word–tweet co-occurrences, that dimensionality reduction is essential, and that the semantic properties of words and tweets are expressed in terms of probabilistic topics.

Structure of the paper

In the paper we address the following research questions:

- Which topics are prevailing and which draw the most hate speech in Twitter discussions?
- How do retweet communities differ in topics they discuss?
- How do topics evolve through time with respect to the communities and hate speech?

This work is an extension of our previous research on the evolution of retweet communities (Evkoski et al. 2021a), and identification of the main sources of hate speech (Evkoski et al. 2021c). We illustrate our approach to the evolution of topics, hate speech and communities on an exhaustive set of Slovenian tweets, collected during the 3 year period 2018–2020. In the Methods section we provide a brief overview of the methods used in the previous research, and the topic detection approach used here. The Results and discussion section gives answers to the research questions addressed. In Conclusions we summarize each components of the analysis, and wrap up the analyses of the Slovenian tweets.

Methods

In the paper we apply methods from three research areas that deal with different aspects of data analysis. They are applied to 3 years of Slovenian Twitter data to study the evolution of communities, hate speech and discussion topics through time. We first give an overview of the Twitter data collected, and the roles that different parts of the data have in the analyses (subsection Overview). We then outline individual research methods applied. Network analysis is used to construct retweet networks, detect communities, and study their evolution through time (subsection Evolving retweet communities). Machine learning is applied to train and evaluate a hate speech classification model

(subsection Hate speech classification). Methods of content analysis are used to detect topics discussed in the tweets (subsection Topic detection). In the next section, Results and discussion, we combine the results of individual methods to reveal some interesting insights gained from the collected Twitter data.

Overview

For this study, we collected a set of almost 13 million Slovenian tweets in the 3 year period, from January 1, 2018 until December 28, 2020. The set represents an exhaustive collection of Twitter activities in Slovenia. The tweets were collected via the public Twitter API, using the TweetCaT tool (Ljubešić et al. 2014). TweetCaT is designed to acquire exhaustive Twitter datasets for less frequent languages, in this case Slovenian.

Figure 1 shows the timeline of Twitter volumes, the types of hate speech posted, and topics discussed during that period. Table 1 gives a breakdown of the 13-million dataset collected in terms of how different subsets are used in this study.

All Twitter posts are either original tweets or retweets. In this study we use the retweets to create retweet networks and detect retweet communities. A retweet network comprises a time window of 24 weeks, and adjacent retweet networks are shifted for 1 week. A selection of five retweet networks, with the largest differences in the detected communities, is indicated by vertical bars in Fig. 1 (top chart). See the subsection on Evolving retweet communities for details.

A large subset of the original tweets is used to manually annotate, train and evaluate hate speech classification models. A machine learning model classifies tweets into four classes: acceptable, inappropriate, offensive, and violent. Inappropriate and violent tweets are relatively rare and cannot be reliably classified. Therefore, for this study, all

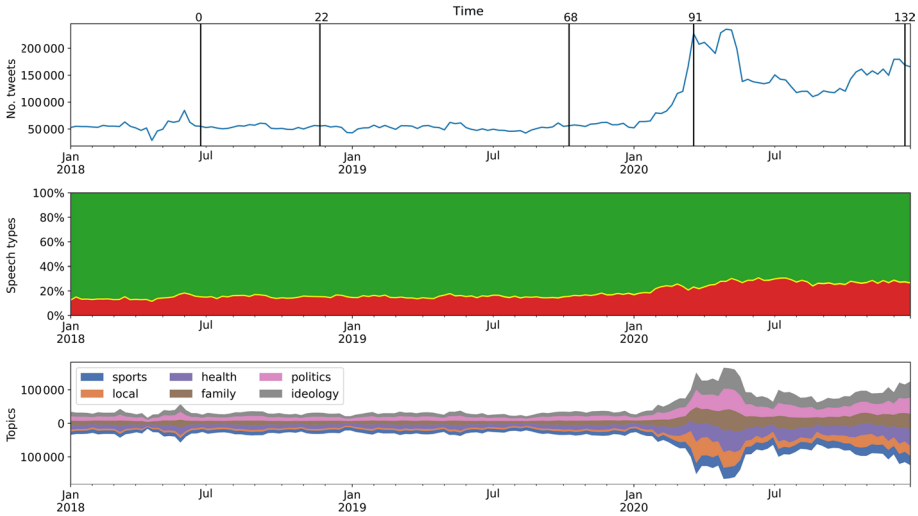

Fig. 1 Three aspects of the Twitter data analysis: Creation of retweet networks at selected timepoints (top), hate speech classification (middle), and topics detected (bottom). (Top) Weekly volume of the Slovenian Twitter data comprises original tweets and retweets collected over the period of 3 years. Vertical bars show five endpoints of automatically selected time windows for retweet networks construction, with weeks labeled as $t = 0, 22, 68, 91, 132$. (Middle) Distribution of hate speech classes: fraction of acceptable tweets (green), inappropriate tweets (yellow, barely visible), and offensive tweets (red); violent tweets are not visible due to low volume. (Bottom) Distribution of the six detected topics

Table 1 The roles of different subsets of the 2018–2020 Slovenian Twitter dataset

Dataset	Period	No. of tweets	Role
All tweets	Jan. 2018–Dec. 2020	12,961,136	Collection, hate speech classification and topic detection
Original tweets	Jan. 2018–Dec. 2020	8,363,271	Hate speech modeling
Retweets	Jan. 2018–Dec. 2020	4,597,865	Network construction and community detection
Training set	Dec. 2017–Jan. 2020	50,000	Hate speech model training and cross validation
Evaluation set	Feb. 2020–Aug. 2020	10,000	Hate speech model evaluation

Out of almost 13 million tweets collected, a sample of the original tweets is used for hate speech annotation, training of classification models, and their evaluation. The retweets are used to create retweet networks, and detect communities. All the tweets are automatically classified by the hate speech classification model, and are used to detect topics

the tweets that are classified as not **acceptable** are jointly classified as **unacceptable**. See the subsection on Hate speech classification for details on the machine learning modelling and extensive evaluations.

All the original tweets and their retweets are used to detect discussion topics. In general, the number of different topics is not fixed, and a typical tweet discusses several topics. For this study we settled for six most distinguishing topics and assigned one prevailing topic to each tweet. Details are in the Topic detection subsection.

Evolving retweet communities

This subsection briefly summarizes our approach to community evolution in retweet networks, extensively described in Evkoski et al. (2021a). Twitter provides different forms of interactions between the users: follows, mentions, replies, and retweets. A very useful indicator of social ties between the Twitter users are retweets (Cherepnalkoski and Mozetič 2016; Durazzi et al. 2021) since a user typically retweets content that he/she finds interesting or agreeable. When a user retweets a tweet, it is distributed to all of its followers, and the link between the original tweet and the final retweet is retained even when several retweeters are in between.

Retweet networks

A retweet network is a directed graph. The nodes are Twitter users and the edges are retweet links between the users. An edge is directed from the user A who posts a tweet to the user B who retweets it. The edge weight is the number of tweets posted by A and retweeted by B. For the whole 3-year period of Slovenian tweets, there are in total 18,821 users (nodes) and 4,597,865 retweets (sum of all the weighted edges).

We form a sequence of network snapshots, with a sliding window of 1 week, to study the evolution of a retweet network. The snapshots are overlapping, where each snapshot comprises an observation window of 24 weeks (about 6 months). We employ an exponential edge weight decay, with half-time of 4 weeks, to eliminate the effects of the trailing end of a moving network snapshot. This provides a relatively high temporal resolution between subsequent networks, but we later select just the most relevant intermediate timepoints.

The set of network snapshots thus consists of 133 overlapping observation windows, with temporal delay of 1 week. The snapshots start with a network at $t = 0$ (January 1,

2018–June 18, 2018) and end with a network at $t = 132$ (July 13, 2020–December 28, 2020) (see Fig. 1).

Retweet communities

Informally, a network community is a subset of nodes more densely linked between themselves than with the nodes outside the community. A standard community detection method is the Louvain algorithm (Blondel et al. 2008). Louvain finds a partitioning of the network into communities, such that the modularity of the partition is maximized. However, there are several problems with statistical fluctuations and stability of the Louvain results (Fortunato and Hric 2016). The instability is manifested by different results of community detection in the same network, run with different initial seeds. This is due to theoretical issues with modularity maximization, and to heuristic nature of an efficient implementation of the algorithm.

We address the instability of Louvain by applying the Ensemble Louvain algorithm (Evkoski et al. 2021a). The steps of Ensemble Louvain are the following:

1. Run several trials of Louvain on the same network (100 trials by default),
2. Build a new network where a pair of the original nodes is linked if their total Co-membership across all the Louvain trials is above a given threshold (90% by default),
3. Identify the disjoints sets which then represent the detected communities.

As a result of using Ensemble Louvain, nodes without a clear community membership (i.e., nodes that do not have consistent co-membership across repeated Louvain trials) are isolated and excluded from further analyses. The resulting communities are of approximately the same size as produced by individual Louvain trials, but with drastically improved stability and reproducibility (Evkoski et al. 2021b).

We run the Ensemble Louvain on all the 133 undirected network snapshots, resulting in 133 network partitions, where the detected communities change through time.

Community evolution

The differences between the network partitions are relatively small at weekly resolution. The retweet network communities do not change much at this relatively high time resolution. Selecting a lower time resolution means choosing timepoints which are further apart, and where the network communities exhibit larger differences.

We formulate the timepoint selection task as follows. Let us assume that the initial and final timepoints are fixed (at $t = 0$ and $t = n$), with the corresponding partitions P_0 and P_n, respectively. For a given k, select k intermediate timepoints such that the differences between the corresponding partitions are maximized. We implement a simple heuristic algorithm which finds the k timepoints. The algorithm works top-down and starts with the full, high resolution timeline with $n + 1$ timepoints, $t = 0, 1, \ldots, n$ and corresponding partitions P_t. At each step, it finds a triplet of adjacent partitions P_{t-1}, P_t, P_{t+1} with minimal differences, and then eliminates P_t from the timeline, until only k intermediate partitions are left.

For our retweet networks, we fix $k = 3$, which provides much lower, but still meaningful time resolution. This choice results in a selection of five distinguishing network partitions at timepoints t:

- $t = 0$: January 1, 2018–June 18, 2018,
- $t = 22$: June 4, 2018–November 11, 2018,
- $t = 68$: April 22, 2019–October 7, 2019,
- $t = 91$: September 30, 2019–March 16, 2020,
- $t = 132$: July 13, 2020–December 28, 2020.

Community transitions

Communities evolve by new nodes joining, some nodes dropping out, and/or by merging and splitting of communities. In Fig. 2 we visualize the evolution of the retweet communities by a Sankey diagram. At each selected timepoint, we show the top four communities and the membership transitions between them. Note that a relatively large number of Twitter users joined or left the retweet communities between the timepoints during the 2018–2020 period.

The top four communities are named Left, Right, SDS, and Sports. The names are derived from their most influential users and the contents of tweets they post. The largest three communities are politically oriented, the left leaning Left, the right leaning Right, and the main right-wing government party SDS (Slovenian Democratic Party). The only non-political community is Sports. All the remaining, smaller communities, are represented as Rest.

Hate speech classification

Hate speech classification is approached as a supervised machine learning problem. Supervised machine learning requires a large set of examples labeled for hate speech, and typically involves a considerable initial effort to produce such labeled examples. The labeled examples are then used to train classification models to distinguish between the

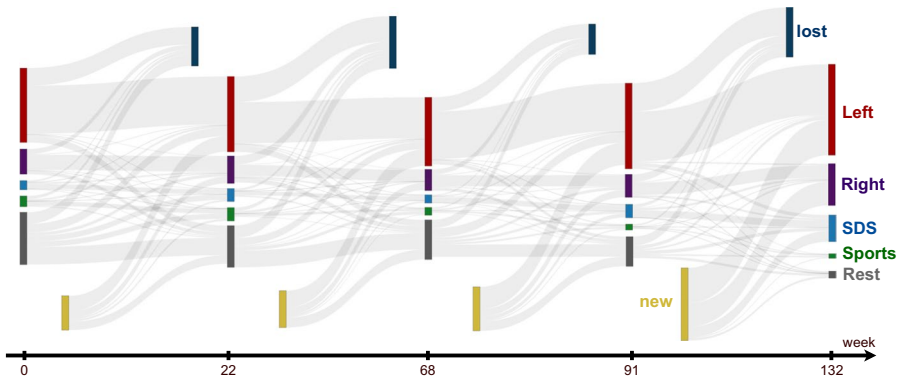

Fig. 2 A Sankey diagram showing community membership transitions at the five selected timepoints $t = 0, 22, 68, 91, 132$. We focus on four major communities: Left (red), Right (violet), SDS (blue), and Sports (green). The remaining, typically smaller, communities are denoted as Rest. At each timepoint, there are new nodes joining the retweet networks, and lost nodes leaving the networks

examples of hate and normal speech (Zampieri et al. 2020). It is important to properly evaluate the trained models to asses their applicability and predictive performance on yet unseen examples of (normal or hate) speech. We pay special attention to the evaluation of the trained models, not only by cross validation (on the training set), but also on a separate, out-of-sample evaluation set. More details are provided in Evkoski et al. (2021c).

Data annotation

The hate speech annotation schema is adapted from OLID (Zampieri et al. 2019) and FRENK (Ljubešić et al. 2019). The schema distinguishes between four classes of speech on Twitter:

- Acceptable—normal tweets, not hateful,
- Inappropriate—tweets contain terms that are obscene or vulgar, but the tweets are not directed at any specific target (a person or a group),
- Offensive—tweets include offensive generalization, contempt, dehumanization, or indirect offensive remarks,
- Violent—the author threatens, indulges, desires or calls for physical violence against a target; this also includes tweets calling for, denying or glorifying war crimes and crimes against humanity.

During the annotation process, and for training the models, all four classes were considered. However, in this paper we take a more abstract view and distinguish just between the normal, **acceptable** speech, and the **unacceptable** speech, i.e., inappropriate, offensive or violent.

We engaged ten well qualified annotators to label a random sample of the Slovenian tweets. The annotators first underwent a training, and were then asked to label each tweet assigned to them by selecting one of the four classes of speech. Two datasets were labeled: a training and an evaluation set.

Training dataset The training set was sampled from Twitter data collected before February 2020. 50,000 tweets were selected for manual annotation and training different models.

Out-of-sample evaluation dataset The independent evaluation set was sampled from data collected between February and August 2020. The evaluation set strictly follows the training set in order to prevent data leakage between the two sets and allow for proper model evaluation. 10,000 tweets were randomly selected for the evaluation dataset.

Each tweet was labeled twice: in 90% of the cases by two different annotators and in 10% of the cases by the same annotator. The role of multiple annotations is twofold: to control for the quality and to establish the level of difficulty of the task. Hate speech classification is a non-trivial, subjective task, and even highly qualified annotators sometimes disagree. We accept the disagreements and do not try to force a unique, consistent ground truth. Instead, we quantify the level of agreement between the annotators (the self- and the inter-annotator agreements), between the annotators and the models, and then compare if a model comes close to the inter-annotator agreement.

Training classification models

Several machine learning algorithms were used to train hate speech classification models. First, three traditional algorithms were applied: Naïve Bayes, Logistic regression, and Support Vector Machine with a linear kernel. Second, deep neural networks, based on the Transformer language models, were applied. We used two multi-lingual language models, based on the BERT architecture (Devlin et al. 2018), the general multi-lingual BERT (mBERT), and the specialized Croatian/Slovenian/English BERT (cseBERT Ulčar and Robnik-Šikonja 2020). The two language models differ in the number and selection of training languages and corpora on which they were pre-trained.

An extensive comparison of different classification models was done following the Bayesian approach to significance testing (Benavoli et al. 2017). Two classifiers are considered practically equivalent if the absolute difference of their scores is less than 1%. We consider two classifiers to be significantly different if the fraction of the posterior distribution in the region of practical equivalence is less than 5%. The comparison results show that deep neural networks significantly outperform the three traditional machine learning models. Additionally, language-specific cseBERT significantly outperforms the general multi-lingual mBERT model. Consequently, the cseBERT classification model was used to label all the Slovenian tweets collected in the 3-year period.

Evaluation measures and procedures

The training, tuning, and selection of classification models was done by cross validation on the training set. We used blocked 10-fold cross validation for two reasons. First, this method provides realistic estimates of performance on the training set with time-ordered data (Mozetič et al. 2018). Second, by ensuring that both annotations for the same tweet fall into the same fold, we prevent data leakage between the training and test splits in cross validation. An even more realistic estimate of performance on yet unseen data is obtained on the out-of-sample evaluation set.

There are different evaluation measures, and to get robust estimates, we apply three well-known measures from the fields of inter-rater agreement and machine learning: Krippendorff's Alpha-reliability, accuracy, and F-score.

Krippendorff's Alpha-reliability (*Alpha*) (Krippendorff 2018) was developed to measure the agreement between human annotators, but can also be used to measure the agreement between classification models and a (potentially inconsistent) ground truth. It generalizes several specialized agreement measures, takes ordering of classes into account, and has the agreement by chance as the baseline.

Accuracy (*Acc*) is the simplest, common measure of performance of models which measures the agreement between the model and the ground truth. Accuracy does not account for the (dis)agreement by chance, nor for the ordering of the values of hate speech classes. Furthermore, it can be deceiving in cases of unbalanced class distribution.

F-score (F_1) is an instance of the well-known effectiveness measure in information retrieval (Van Rijsbergen 1979) and is used in binary classification. In the case

of multi-class problems, it can be used to measure the performance of the model to identify individual classes. In terms of the annotator agreement, $F_1(c)$ is the fraction of equally labeled tweets out of all the tweets with class label c.

Evaluation results

Table 2 presents the annotator self-agreement and the inter-annotator agreement on both the training and the evaluation sets. Note that the self-agreement is consistently higher than the inter-annotator agreement, as expected, but is far from perfect. The results for the best performing classification model (cseBERT) are also in Table 2. The F_1 scores indicate that acceptable tweets can be classified more reliably than unacceptable tweets. The overall *Alpha* scores show a drop in performance estimate between the training and evaluation set, as expected. However, note that the level of agreement between the best model and the annotators is very close to the inter-annotator agreement. If one accepts inherent ambiguity of the hate speech classification task, there is very little room for model improvement, without taking additional information into account.

Topic detection

Topic models provide a simple way to analyze large volumes of unlabeled documents, in our case tweets. A "topic" consists of a cluster of words that frequently occur together and represents a content abstraction of a collection of tweets. The goal of topic modelling in this paper is to identify prevailing topics discussed, to see which topics provoke more hate speech, which topics are of interest to different communities, and how specific topics and unacceptable speech evolve through time.

Topic models (Steyvers and Griffiths 2007) assume that tweets contain a mixture of topics, where a topic is a probability distribution over words. A topic model is a generative model: it specifies a probabilistic procedure by which tweets can be generated. To construct a new tweet, one chooses a distribution over topics. Then, for each word in that tweet, one chooses a topic at random according to that distribution, and picks a word from that topic. Standard statistical techniques are then used to invert this process, inferring the set of topics that were responsible for generating a collection of tweets.

Table 2 The annotator agreement and the model performance

	No. of tweets	Overall		Acceptable	Unacceptable
		Alpha	Acc	F_1(A)	F_1(U)
Self-agreement	5981	0.79	0.88	0.92	0.87
Inter-annotator agreement	53,831	0.60	0.79	0.85	0.75
Classification model					
Training set	50,000	0.61	0.80	0.85	0.77
Evaluation set	10,000	0.57	0.80	0.86	0.71

Three measures are used: ordinal Krippendorff's *Alpha*, accuracy (*Acc*), and F_1 for the classes of acceptable (A) and unacceptable (U) tweets. The first line is the self-agreement of individual annotators, and the second line is the inter-annotator agreement between different annotators. The last two lines are the evaluation results of the model, on the training set (by cross validation) and on the out-of-sample evaluation set, respectively. Note that the model performance is comparable to the inter-annotator agreement

Previous research (Martin and Johnson 2015), as well as our own experience, show that topics are more coherent if topic modelling is run over sequences of lemmas of nouns. We adopt this approach and represent each tweet as a sequence of lemmas of nouns occurring in that tweet. To obtain lemmas and part-of-speech tags, we process the Slovenian Twitter corpus with the CLASSLA pipeline (Ljubešić and Dobrovoljc 2019). The pipeline consists of a Bi-LSTM (Bidirectional Long Short-Term Memory) tagger and a LSTM sequence-to-sequence lemmatizer. We use models that were trained on a combination of standard and non-standard texts, and were additionally augmented for missing diacritics. These models are well suited to deal with language variability and non-standard language used in social media, and are therefore appropriate for our Twitter corpus. The topic detection was implemented by applying the MALLET toolkit (McCallum 2002). MALLET was ran for the default 1000 iterations with the suggested hyperparameter optimization every 10 iterations.

Results and discussion

In this section we combine the results of individual methods applied to the Slovenian Twitter dataset 2018–2020. In subsection Topics and unacceptable tweets we show the major topics detected and the shares of unacceptable tweets in each of them. We then quantify the differences between the top retweet communities in terms of the topics they discuss, and how stable they are through time (subsection Communities and topics). In subsection Evolution of offensive topics we focus on the three prevailing topics, and show the evolution of acceptable and unacceptable tweets posted by the top communities.

Topics and unacceptable tweets

The topic detection method we apply requires to set the number of topics in advance. We experimented with different preset values to find an appropriate level of detail where no obvious topics are neither merged nor split across multiple topics. This experiment resulted in six topics, each defined by a probability distribution over constituent words. In general, a tweet discusses several topics with different probabilities. For easier interpretation of the results, we selected just the most probable topic assigned to each tweet.

A topic is defined by the probability distribution over words, and we provide the top most probable words for each topic. Each topic is assigned a shorthand label to adequately characterize it and to facilitate further analyses. We assigned the topic labels manually, on the basis of the most probable words, and by inspecting several tweets for each topic. The six detected topics are listed below:

- **local** Ljubljana, year, price, municipality, road, city, Slovenia, car, water, vehicle, center, Maribor, Euro, apartment, shop, house, registration, firefighter, mayor;
- **sports** match, year, Slovenia, show, win, season, movie, team, book, city, Ljubljana, league, Maribor, award, interview, concert, weekend, game;
- **health** measure, human, mask, virus, government, epidemic, Slovenia, infection, country, coronavirus, doctor, week, health, number, case, work, life, help, school;
- **family** child, year, human, school, life, woman, head, hand, parent, world, thank you, man, word, language, end, thing, mother, book, family;

Table 3 Distribution of hate speech classes and subclasses, and detected topics across the complete 2018–2020 Slovenian Twitter dataset

Hate speech			Topics	
Acceptable	75%		Local	12.5%
			Sports	12.3%
Unacceptable:	25%		Health	14.0%
Inappropriate		0.84%	Family	17.1%
Offensive		24.14%	Politics	22.9%
Violent		0.12%	Ideology	21.2%

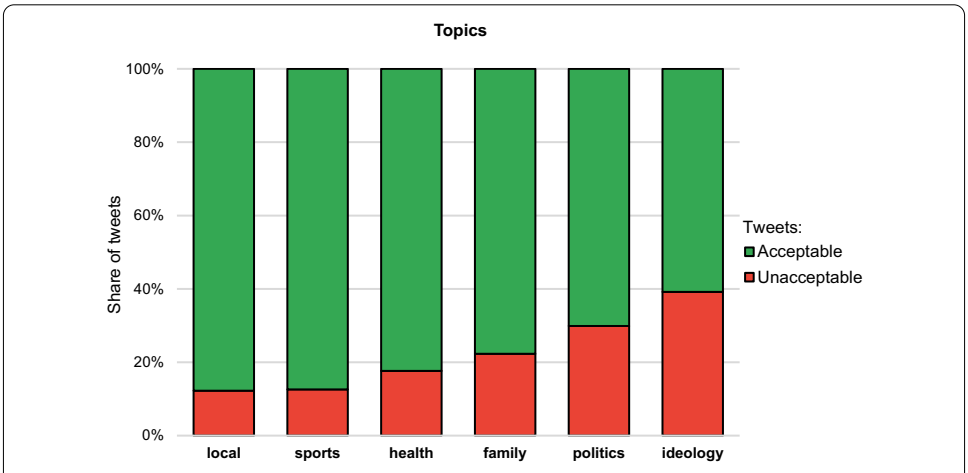

Fig. 3 Shares of unacceptable tweets for different topics

- **politics** government, party, state, year, money, Slovenia, minister, media, president, election, work, salary, law, parliament member, human, Janša, Šarec, court, politics;
- **ideology** Slovenia, country, human, year, Slovenian, nation, border, migrant, war, communist, government, Europe, Janša, power, army, world, media, justice, leftist.

In Table 3 we summarize the distribution of hate speech and detected topics across the complete set of almost 13 million Slovenian tweets. The distribution of hate speech classes shows that inappropriate and violent tweets are rare. This justifies our decision to merge all the tweets labeled by the model as not acceptable into a single class of unacceptable tweets. The unacceptable tweets, predominantly offensive, account for a quarter of all the original and retweeted tweets. The topics detected are much more evenly distributed, but we can observe that politics and ideology are prevailing, accounting for almost 45% of all the tweets.

Figure 3 shows the shares of unacceptable tweets for different topics. The two dominant topics, politics and ideology, also exhibit the highest share of unacceptable tweets, between 30 and 40%. Interestingly, the topic of sports, which often triggers passionate cheering and heated debates between the fans, shows a very low level of unacceptable tweets, about 10% only.

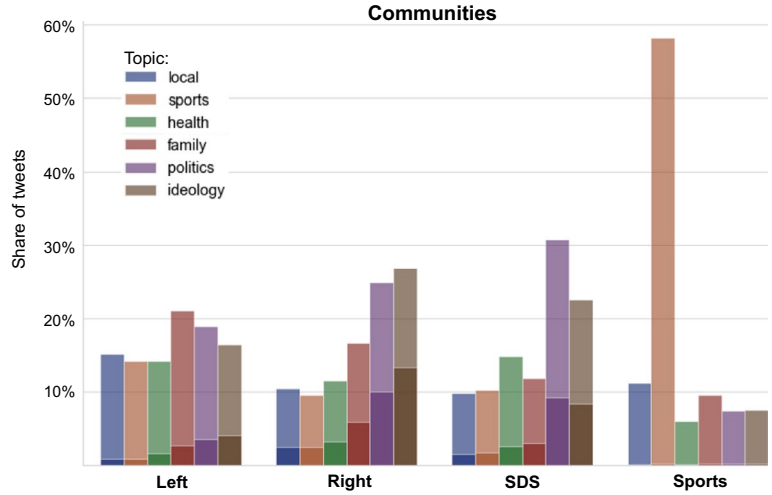

Fig. 4 Cumulative topic distribution for the four major communities: Left, Right, SDS, and Sports. Darker areas represent fractions of unacceptable tweets in individual topics

Communities and topics

In this subsection we turn attention to the topic distribution per community. We focus just on the top four communities, already identified in Fig. 2: Left, Right, SDS, and Sports. Figure 4 shows the cumulative topic distribution for the four major communities. The Right and SDS communities are similar as they both favor topics of politics and ideology. These two topics represent more that 50% of their original tweets or retweets. On the other hand, the Left community is more balanced in terms of its topic distribution, with slight preference for the family topic. The Sports community represents another extreme, with almost 60% of its tweets and retweets about sports, and a low level of interest in the other topics.

Figure 4 also shows fractions of unacceptable tweets per community and topic. The Sports community posts almost exclusively acceptable tweets. On the other hand, the political Right community posts about one half of its tweets, on the topics of politics and ideology, as unacceptable. The governmental SDS posts about one third of its tweets, on the topics of politics and ideology, as unacceptable. The political Left, in opposition to the right-wing government, is more modest, but it also posts the largest fraction of unacceptable tweets on the topics of politics and ideology. A detailed analysis of the distribution of hate speech between the communities and different types of Twitter users, regardless of topics, is discussed in Evkoski et al. (2021c).

If one wants to compare communities in terms of their topic distributions, between themselves and through time, one needs to quantify the similarities between distributions. A suitable measure of the similarity between two probability distributions, P and Q, is defined by the Jensen–Shannon divergence (JSD) (Lin 1991):

$$JSD(P \parallel Q) = \frac{1}{2}KLD(P \parallel M) + \frac{1}{2}KLD(Q \parallel M),$$

where M is the average of the two distributions:

$$M = \frac{1}{2}(P + Q).$$

JSD is defined in terms of the Kullback–Leibler divergence (*KLD*) (Kullback and Leibler 1951):

$$KLD(P \parallel Q) = \sum_x P(x) \cdot log_2\left(\frac{P(x)}{Q(x)}\right)$$

The square root of *JSD*, which makes the measure a metric, is known as Jensen–Shannon distance (*JS*) (Endres and Schindelin 2003):

$$JS(P \parallel Q) = \sqrt{JSD(P \parallel Q)}, \quad 0 \leq JS(P \parallel Q) \leq 1.$$

$JS(P \parallel Q)$ of 0 indicates that P and Q are identical distributions, while values close to 1 indicate very different distributions.

Let C_t denote a probability distribution of topics in tweets posted by the community C, at timepoint t. We denote by $C_\cup$ a cumulative distribution of topics in all the tweets by C across the five timepoints $t = 0, 22, 68, 91, 132$. We can compare how the topic distribution in a community C changes over time by computing the distances between subsequent timepoints $JS(C_t \parallel C_{t+1})$, or the distances of individual timepoints to the cumulative distribution $JS(C_t \parallel C_\cup)$. We can also compare the differences between pairs of communities *Ci* and *Cj* by computing the distance between their cumulative distributions $JS(Ci_\cup \parallel Cj_\cup)$.

Results with the differences in topic distributions are in Table 4. The left-hand side of the table shows that for individual communities, topic distribution does not change much over time. The table gives the distances to the cumulative distribution, but the distances between subsequent timepoints are similarly low. We only observe some change in topic distribution for SDS (bold numbers on the left-hand side of Table 4), from the initial timepoints, when the party was in opposition, to the final timepoints, when SDS became the main government party.

The right-hand side of Table 4 gives pairwise distances between different communities. The results show that the Right and SDS communities are the most similar to each other, which corroborates the visual impression from Fig. 4. Both, Right and SDS, are some distance from the Left community (bold numbers on the right-hand side of Table 4). As

Table 4 Differences in topic distributions in terms of Jensen–Shannon distance (*JS*)

	Timepoint *t*					**Community**			
Community	**0**	**22**	**68**	**91**	**132**	**Left**	**Right**	**SDS**	**Sports**
Left	0.052	0.051	0.060	0.008	0.057	0.0	**0.146**	**0.172**	*0.406*
Right	0.051	0.047	0.049	0.019	0.034	–	0.0	0.092	*0.481*
SDS	**0.101**	**0.114**	0.091	0.028	0.044	–	–	0.0	*0.482*
Sports	0.074	0.020	0.036	0.087	0.082	–	–	–	0.0

The left-hand side of the table shows the *JS* distances for each community *C*, between its cumulative distribution $C_\cup$ and individual timepoints C_t, $JS(C_t \parallel C_\cup)$. The right-hand side is a symmetrical matrix, with the *JS* distances between the cumulative distributions for all pairs *i*, *j* of communities, $JS(Ci_\cup \parallel Cj_\cup)$. In bold are the *JS* distances $0.1 < JS \leq 0.4$, and in italics $0.4 < JS$

expected, the Sports community is considerably different from the other three in terms of the topic distribution (numbers in italics on the right-hand side of Table 4).

Similarities between the communities in terms of topic distributions are consistent with the formation of super-communities. A super-community is a set of communities that are densely linked together by the external influence links, i.e., retweets (Evkoski et al. 2021a). In our case, Right and SDS (with other smaller communities) form the right-wing super-community, Left (with other smaller communities) is part of the left-wing super community, and Sports is isolated in its own super-community. This formation of super-communities closely matches the similarities in terms of *JS* distances. We find it interesting that two different methods, super-community formation and topic detection, yield very similar results. In fact, it is surprising that some detected communities (such as Right and SDS) exhibit higher similarities in terms of their topic distribution than in terms of their membership.

Evolution of offensive topics

In this subsection we focus just on the top three largest, political communities: Left, Right, and SDS. The goal is to show the evolution of the most interesting topics through time. We pinpoint the differences between the acceptable and unacceptable (predominantly offensive) tweets posted by the three communities.

The three communities are very different in size and in their Twitter activities. Figure 5 (left panel) shows how the membership (the number of Twitter users) changed through the 3-year period, 2018–2020. We see that the Left is considerably larger than the right-wing communities, Right and SDS, and that its membership is gradually increasing. On the other hand, the sizes of the Right and SDS communities considerably increased after the right-wing government was formed (in March 2020, timepoints $t = 91, 132$). Even more drastic is the increase in the number of tweets posted and retweeted (Fig. 5, right panel), corresponding to the change of government and the emergence of the Covid-19 pandemic. In the last period ($t = 132$) the Right even surpassed the Left community, despite the fact that it is considerably smaller. The governmental SDS, which was barely active when in opposition (timepoints $t = 0, 22, 68$) shows a five-fold increase in the Twitter activities during the last period. This is consistent with the observed smaller size and higher activities of the

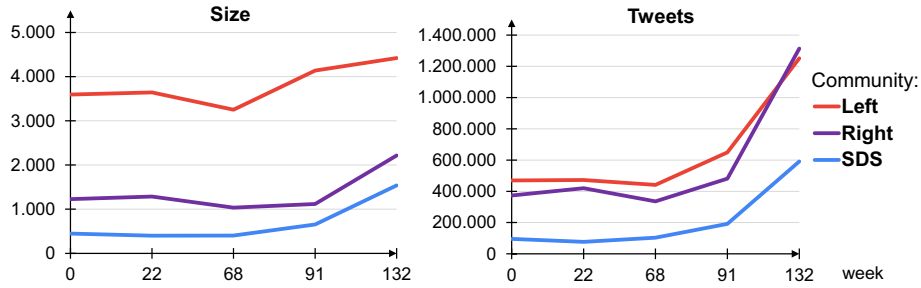

Fig. 5 Evolution of the community size and the number of tweets posted and retweeted for the three major communities: Left (red), Right (violet), and SDS (blue). The x-axis are the five timepoints at weeks $t = 0, 22, 68, 91, 132$

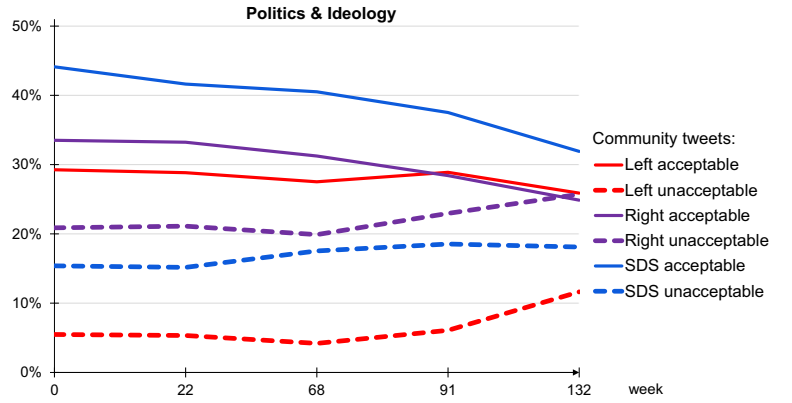

Fig. 6 Evolution of the two topics merged, politics and ideology, for the three major communities: Left (red), Right (violet), and SDS (blue). Solid lines represent acceptable tweets, and dashed lines correspond to unacceptable tweets. The y-axis represents percentages of tweets with a topic of politics or ideology out of all the tweets posted or retweeted by a community. The x-axis are the five timepoints at weeks $t = 0, 22, 68, 91, 132$

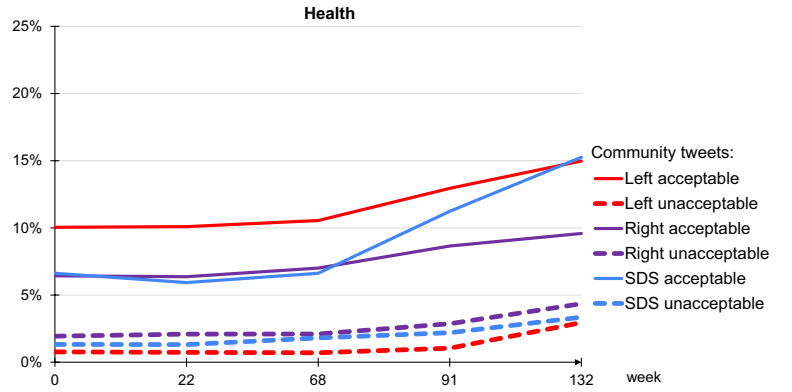

Fig. 7 Evolution of the health topic, including the Covid-19 pandemic, for the three major communities: Left (red), Right (violet), and SDS (blue). Solid lines represent acceptable tweets, and dashed lines correspond to unacceptable tweets. The y-axis represents percentages of tweets with a topic of health out of all the tweets posted or retweeted by a community. Note that the range of the y-axis here is half the range of the y-axis in Fig. 6. The x-axis are the five timepoints at weeks $t = 0, 22, 68, 91, 132$

right-wing parties in the European Parliament (Cherepnalkoski et al. 2016), and the Leave proponents during the Brexit referendum (Grčar et al. 2017).

Out of the six topics detected, we first consider the two prevailing topics, politics and ideology, taken together. Figure 6 shows the evolution of the two topics through the 3-year period. For the selected communities, Left, Right and SDS, the percentages of acceptable (solid lines) and unacceptable (dashed lines) tweets are given. For all three communities, the fractions of acceptable tweets are decreasing, while the unacceptable tweets are increasing. We speculate that this is due to the change of the government from the left-wing to the right-wing, and increased political polarization in the last period (after March 2020, timepoints $t = 91, 132$). Taken all tweets together,

throughout the 3-year period, Right and SDS post more than 50% of their tweets on politics and ideology, and Left is approaching 40%.

The change of the government in Slovenia in 2020 coincides with the emergence of the Covid-19 pandemic. In Fig. 7 we show the evolution of the health topic which also covers the pandemic-related issues (keywords: mask, virus, epidemic, infection, coronavirus, ...). The figure shows a considerable increase in the Twitter activities at the last two timepoints (after March 2020, $t = 91, 132$). The most pronounced is the increase for the SDS community which corresponds to the main party in the right-wing government, and which undertook major activities during the pandemic. However, the overall volume is still much lower in comparison to the topics of politics and ideology (less than 20%). Note that the range of the y-axis in Fig. 7 is only half the range of the y-axis in Fig. 6.

In contrast to the politics and ideology, the health topic draws relatively low number of unacceptable tweets. However, as the pandemic progressed, and increasingly more unpopular public measures were taken, so has the volume of unacceptable tweets increased.

Conclusions

This paper concludes a trilogy on the analysis of a comprehensive Slovenian Twitter data corpus, from the 2018–2020 period. In the first part (Evkoski et al. 2021a) we propose methods to study the evolution of retweet communities through time. We developed an extension of the Louvain community detection algorithm, Ensemble Louvain, to improve the stability of the detected communities, which is important in time-changing networks (Evkoski et al. 2021b). We found that in our data retweet communities change relatively slowly, and we speculate that the time window snapshots can be taken further apart, in the order of months, not weeks. We also proposed several measures of influence, and demonstrated that external retweet influence links similar communities into super-communities. The detected super-communities show clear signs of increasing political polarization in Slovenia in the years 2018–2020.

The second part of the trilogy (Evkoski et al. 2021c) introduces an analysis of hate speech in Twitter posts. We developed a state-of-the-art hate speech classification model with the performance close to the human annotators. We found that communities which form the same super-community can be very different in the amount of hate tweets they post. We identified a single right-wing retweet community which posts a disproportional amount of unacceptable tweets with respect to its size. We also found that the main source of unacceptable tweets are personal Twitter accounts, which were either anonymous or suspended during the 3-year period.

In the current paper we add another aspect to the analysis, namely topic detection. We confirm what was already indicated before, that politics and ideology are the prevailing topics during the years 2018–2020. These two topics also draw the highest proportion of unacceptable tweets. Interestingly, distribution of topics discussed by individual communities shows high similarity between the communities which form the same super-community. On one hand, we find high similarity between the communities by means of external retweet influence links and topics they discuss. On the other hand, they are very different in the amount of hate speech produced. This also indicates that community

membership can be a useful additional feature if one wants to improve the hate speech classification models.

In our case, the performance of the binary classification model, acceptable vs. unacceptable tweets, is already close to the inter-annotator agreement. Our results are comparable to the performance of models on similarly subjective and difficult tasks, on different social media platforms (Twitter, Facebook, YouTube comments) and in other languages (Zollo et al. 2015; Mozetič et al. 2016; Cinelli et al. 2021). However, the performance can be improved if user-related context is taken into account (Gao and Huang 2017; Fehn Unsvåg and Gambäck 2018). Previous works (Mishra et al. 2019; Mosca et al. 2021), as well as our results, indicate that combining community information with textual information can considerably improve the hate speech classification models.

Abbreviations
Acc: Accuracy; *Alpha*: Krippendorff's Alpha-reliability; F_1: F-score; *JSD*: Jensen–Shannon divergence; *JS*: Jensen–Shannon distance; *KLD*: Kullback–Leibler divergence; LSTM: Long short-term memory neural network; SDS: A community denoting Slovenian democratic party.

Acknowledgements
Not applicable.

Authors' contributions
BE constructed the retweet networks, implemented the Ensemble Louvain algorithm, and performed most of the experiments. NL collected the Twitter data and implemented the topic detection method. AP implemented and evaluated the hate speech classification models. IM wrote the initial draft of the paper. PKN supervised the work. All the authors analyzed the results, contributed to, read, and approved the final manuscript.

Funding
The authors acknowledge financial support from the Slovenian Research Agency (research core Funding No. P2-103 and P6-0411), the Slovenian Research Agency and the Flemish Research Foundation bilateral research project LiLaH (Grant Nos. ARRS-N6-0099 and FWO-G070619N), and the European Union's Rights, Equality and Citizenship Programme (2014–2020) project IMSyPP (Grant No. 875263). The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Data availability
The Slovenian Twitter dataset 2018–2020, with retweet links and assigned hate speech class, is available at a public language resource repository CLARIN.SI at https://hdl.handle.net/11356/1423.

Code availability
The code used to implement the Ensemble Louvain algorithm is available at the Github repository at https://github.com/boevkoski/ensemble-louvain.git.

Model availability
The model for hate speech classification of Slovenian tweets is available at a public language models repository Huggingface at https://huggingface.co/IMSyPP/hate_speech_slo.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia. [2]Jozef Stefan International Postgraduate School, Ljubljana, Slovenia. [3]Faculty of Information and Communication Sciences, University of Ljubljana, Ljubljana, Slovenia. [4]Central European University, Vienna, Austria.

Received: 9 October 2021 Accepted: 10 December 2021
Published online: 20 December 2021

References
Aynaud T, Guillaume J-L (2010) Static community detection algorithms for evolving networks. In: 8th international symposium on modeling and optimization in mobile, ad hoc, and wireless networks, pp 513–519. IEEE

Aynaud T, Fleury E, Guillaume J-L, Wang Q (2013) Communities in evolving networks: definitions, detection, and analysis techniques. In: Ganguly N, Deutsch A, Mukherjee A (eds) Dynamics on and of complex networks, vol 2. Springer, Berlin, pp 159–200. https://doi.org/10.1007/978-1-4614-6729-8_9
Benavoli A, Corani G, Demšar J, Zaffalon M (2017) Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. J Mach Learn Res 18(1):2653–2688
Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3(4–5):993–1022
Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech: Theory Exp 2008(10):10008
Bollen J, Mao H, Zeng X (2011) Twitter mood predicts the stock market. J Comput Sci 2(1):1–8
Cherepnalkoski D, Mozetič I (2016) Retweet networks of the European parliament: evaluation of the community structure. Appl Netw Sci 1(1):2. https://doi.org/10.1007/s41109-016-0001-4
Cherepnalkoski D, Karpf A, Mozetič I, Grčar M (2016) Cohesion and coalition formation in the European parliament: roll-call votes and Twitter activities. PLoS ONE 11(11):0166586. https://doi.org/10.1371/journal.pone.0166586
Cinelli M, Cresci S, Galeazzi A, Quattrociocchi W, Tesconi M (2020) The limited reach of fake news on Twitter during 2019 European elections. PLoS ONE 15(6):0234689. https://doi.org/10.1371/journal.pone.0234689
Cinelli M, Pelicon A, Mozetič I, Quattrociocchi W, Novak PK, Zollo F (2021) Dynamics of online hate and misinformation. Sci Rep. https://doi.org/10.1038/s41598-021-01487-w
Dakiche N, Tayeb FB-S, Slimani Y, Benatchba K (2019) Tracking community evolution in social networks: a survey. Inform Process Manag 56(3):1084–1102
Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
Durazzi F, Müller M, Salathé M, Remondini D (2021) Clusters of science and health related Twitter users become more isolated during the COVID-19 pandemic. arXiv:2011.06845
Endres DM, Schindelin JE (2003) A new metric for probability distributions. IEEE Trans Inf Theory 49(7):1858–1860. https://doi.org/10.1109/TIT.2003.813506
Evkoski B, Mozetič I, Ljubešić N, Novak PK (2021a) Community evolution in retweet networks. PLoS ONE 16(9):0256175 . https://doi.org/10.1371/journal.pone.0256175. arXiv:2105.06214
Evkoski B, Mozetič I, Novak PK (2021b) Community evolution with Ensemble Louvain. In: Complex networks 2021, Book of Abstracts
Evkoski B, Pelicon A, Mozetič I, Ljubešić N, Novak PK (2021c) Retweet communities reveal the main sources of hate speech. arXiv:2105.14898
Fehn Unsvåg E, Gambäck B (2018) The effects of user features on Twitter hate speech detection. In: Proceedings of 2nd workshop on abusive language online (ALW2), pp 75–85. ACL. https://aclanthology.org/W18-5110
Fortunato S, Hric D (2016) Community detection in networks: a user guide. Phys Rep 659:1–44. https://doi.org/10.1016/j.physrep.2016.09.002
Gao L, Huang R (2017) Detecting online hate speech using context aware models. In: Proceedings of international conference recent advances in natural language processing (RANLP), pp 260–266. https://doi.org/10.26615/978-954-452-049-6_036
Gil de Zúñiga H, Koc Michalska K, Römmele A (2020) Populism in the era of Twitter: How social media contextualized new insights into an old phenomenon. New Media Soc 22(4):585–594
Grčar M, Cherepnalkoski D, Mozetič I, Kralj Novak P (2017) Stance and influence of Twitter users regarding the Brexit referendum. Comput Soc Netw 4(1):6. https://doi.org/10.1186/s40649-017-0042-6
Hartmann T, Kappes A, Wagner D (2016) Clustering evolving networks. In: Sanders P (ed) Algorithm engineering. Springer, Berlin, pp 280–329
Krippendorff K (2018) Content analysis, an introduction to its methodology, 4th edn. Sage Publications, Thousand Oaks
Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86. https://doi.org/10.1214/aoms/1177729694
Lin J (1991) Divergence measures based on the Shannon entropy. IEEE Trans Inf Theory 37(1):145–151. https://doi.org/10.1109/18.61115
Ljubešić N, Dobrovoljc K (2019) What does neural bring? Analysing improvements in morphosyntactic annotation and lemmatisation of Slovenian, Croatian and Serbian. In: Proceedings of 7th workshop on Balto-Slavic natural language processing, pp 29–34. https://doi.org/10.18653/v1/W19-3704
Ljubešić N, Fišer D, Erjavec T (2014) TweetCaT: a tool for building Twitter corpora of smaller languages. In: Proceedings of 9th international conference on language resources and evaluation, pp 2279–2283. European Language Resources Association (ELRA), Reykjavik, Iceland. http://www.lrec-conf.org/proceedings/lrec2014/pdf/834_Paper.pdf
Ljubešić N, Fišer D, Erjavec T (2019) The FRENK datasets of socially unacceptable discourse in Slovene and English. arXiv:1906.02045
MacAvaney S, Yao H-R, Yang E, Russell K, Goharian N, Frieder O (2019) Hate speech detection: challenges and solutions. PLoS ONE 14(8):0221152. https://doi.org/10.1371/journal.pone.0221152
Martin F, Johnson M (2015) More efficient topic modelling through a noun only approach. In: Proceedings of Australasian language technology association workshop, pp 111–115. https://www.aclweb.org/anthology/U15-1013
Masuda N, Lambiotte R (2016) A guide to temporal networks, vol 4. World Scientific, Singapore
Matamoros-Fernández A, Farkas J (2021) Racism, hate speech, and social media: a systematic review and critique. Telev New Media 22(2):205–224
Mathew B, Dutt R, Goyal P, Mukherjee A (2019) Spread of hate speech in online social media. In: Proceedings of 10th ACM conference on web science, pp 173–182
Mathew B, Illendula A, Saha P, Sarkar S, Goyal P, Mukherjee A (2020) Hate begets hate: A temporal study of hate speech. Proc ACM Hum–Comput Interact 4(CSCW2):1–24
McCallum AK (2002) Mallet: a machine learning for language toolkit. http://mallet.cs.umass.edu

Mishra P, Del Tredici M, Yannakoudakis H, Shutova E (2019) Abusive language detection with graph convolutional networks. In: Proceedings of 2019 conference of the North American chapter of the ACL: human language technologies, pp 2145–2150. https://doi.org/10.18653/v1/N19-1221
Mosca E, Wich M, Groh G (2021) Understanding and interpreting the impact of user context in hate speech detection. In: Proceedings of 9th international workshop on natural language processing for social media, pp 91–102
Mozetič I, Grčar M, Smailović J (2016) Multilingual Twitter sentiment classification: the role of human annotators. PLoS ONE 11(5):0155036. https://doi.org/10.1371/journal.pone.0155036
Mozetič I, Torgo L, Cerqueira V, Smailović J (2018) How to evaluate sentiment classifiers for Twitter time-ordered data? PLoS ONE 13(3):0194317. https://doi.org/10.1371/journal.pone.0194317
Rossetti G, Cazabet R (2018) Community discovery in dynamic networks. ACM Comput Surv 51(2):1–37. https://doi.org/10.1145/3172867
Sood S, Antin J, Churchill E (2012) Profanity use in online communities. In: Proceedings of SIGCHI conference on human factors in computing systems, pp 1481–1490
Steyvers M, Griffiths T (2007) Probabilistic topic models. In: Landauer T, McNamara D, Dennis S, Kintsch W (eds) Latent semantic analysis: a road to meaning. Laurence Erlbaum, Mahwah, pp 427–448
Ulčar M, Robnik-Šikonja M (2020) FinEst BERT and CroSloEngual BERT. In: International conference on text, speech, and dialogue. Springer, Berlin, pp 104–111
Uyheng J, Carley KM (2021) Characterizing network dynamics of online hate communities around the covid-19 pandemic. Appl Netw Sci 6(1):1–21
Van Rijsbergen CJ (1979) Information retrieval, 2nd edn. Butterworth, Newton
Wu S, Hofman JM, Mason WA, Watts DJ (2011) Who says what to whom on Twitter. In: Proceedings of 20th international conference on world wide web, pp 705–714
Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R (2019) Predicting the type and target of offensive posts in social media. In: Proceedings of North American Chapter of the ACL
Zampieri M, Nakov P, Rosenthal S, Atanasova P, Karadzhov G, Mubarak H, Derczynski L, Pitenis Z, Çöltekin Ç (2020) SemEval-2020 task 12: multilingual offensive language identification in social media (OffensEval 2020). arXiv:2006.07235
Zollo F, Kralj Novak P, Del Vicario M, Bessi A, Mozetič I, Scala A, Caldarelli G, Quattrociocchi W (2015) Emotional dynamics in the age of misinformation. PLoS ONE 10(9):0138740. https://doi.org/10.1371/journal.pone.0138740

Publisher's Note

Chapter 4

Conclusions

In this master thesis, we have explored the idea of community evolution analysis, a technique that enables tracking groups of densely connected nodes in a complex network. In practice, community evolution aims at detecting and explaining changes in the collective behavior of groups. In order for it to be well executed, it requires many technical choices that would fit the dynamic environment, such as the right strategy for representing the dynamics in a network; applying suitable (stable) and robustly evaluated community detection algorithms; using appropriate community and partition similarity metrics.

Mainly, we presented two contributions to the field of community evolution analysis. First, we proposed Ensemble Louvain, a community detection method based on ensembles of the famous Louvain method. Ensemble Louvain produces stable communities with high quality, making it suitable for evolution analysis. It significantly outperforms Louvain and other ensemble methods. As a side effect, it is also able to detect borderline nodes, which either act as periphery to a community (and the network) or act as influential bridges between several communities.

With the original goal to carefully evaluate Ensemble Louvain, our second proposal is a novel community detection benchmark on artificial networks. It uses the Lancichinetti-Fortunato-Radicchi (LFR) networks, which are the most used networks for evaluation, yet it defines a strategy to create more diverse networks than the common approaches, guaranteeing the evaluation over a bigger space of possible network and community structures. Finally, it uses statistical tests to produce a comprehensive and easily interpretable benchmarking which avoids the shortcomings of the standard LFR benchmarking. We refer to our proposal as the Unconstrained LFR benchmark.

On the applicative side, we presented three of our published works where we combined our Ensemble Louvain with many other techniques for data analysis, resulting in insightful results on the evolution of Twitter communities in the Slovenian tweetosphere. In the first work, we set up a procedure for community evolution analysis that uses overlapping network snapshots with time-decaying edge weights, tracking the major (mostly political) Slovenian Twitter communities from 2018 to 2021. In the second, we combined the knowledge of community evolution and hate speech detection in a study where we discovered the main sources of unacceptable public speech and tracked its share and change throughout the Covid pandemic and major political events. Lastly, the third work adds our final perspective on the evolution of the Slovenian retweet network, this time through topic modeling.

We hope that this thesis can be used as an example of the many challenges community evolution analysis encounters, as well as a read where one can borrow interesting ideas or get inspired for solutions to some of these challenges. In the meantime, we aim to continue to contribute to the field of community detection and community evolution.

Appendix A

Ensemble Louvain Experiments

A.1 Borderline Nodes

Borderline nodes are nodes that do not have a high co-membership with any neighbouring node. They do not clearly belong to any community and are left as islands in the partitions produced by Ensemble Louvain. Community analysis can benefit if these nodes are detected, not just to stabilize and improve the discovered communities, but also to potentially reveal hidden processes connected to these nodes in the network. This could be helpful in social network analysis where the borderline nodes could point out to users who are between two communities with conflicting views, agendas, ideologies etc.

We wanted to explore the frequency of borderline nodes and to what extent they influence the total quality of the partition. Therefore, we compared the Ensemble Louvain results of the full partition with the results on the subpartition that does not include the borderline nodes. For detecting borderline nodes, we took the default value of the co-membership threshold parameter $ct = 0.85$. In other words, if a node does not have a co-membership value of at least 0.85 with any of its neighbours, we label it as borderline.

In Figure A.1, we show the results of the influence of borderline nodes on community quality and modularity score (Q), using the standard LFR benchmark. We show that the increase of the LFR μ parameter heavily influences the percentage of borderline nodes detected by Ensemble Louvain. When comparing the partitions including and excluding the borderline nodes, we notice a drastic difference in both scores in favour of the partitions which exclude these borderline nodes. It shows that Ensemble Louvain outputs even better results when the core communities are in focus and that future community detection evaluations could benefit from the separate comparison of the core and peripheral communities.

The role of these borderline nodes in terms of function and influence in real-world networks is yet to be researched. One pioneer study [66] uses the co-membership (or consensus) matrix to calculate a score named CoI (community inconsistency), which gives insight into how unstable a borderline node is. The authors also name the two types of borderline nodes: outsiders (no firm community membership) and promiscuous (nodes with multiple connections to several communities), yet with no clear method on how to differentiate the two. Nonetheless, Ensemble Louvain's natural detection of borderline nodes in combination with the CoI score can lead to interesting future work on the role of nodes in communities and networks.

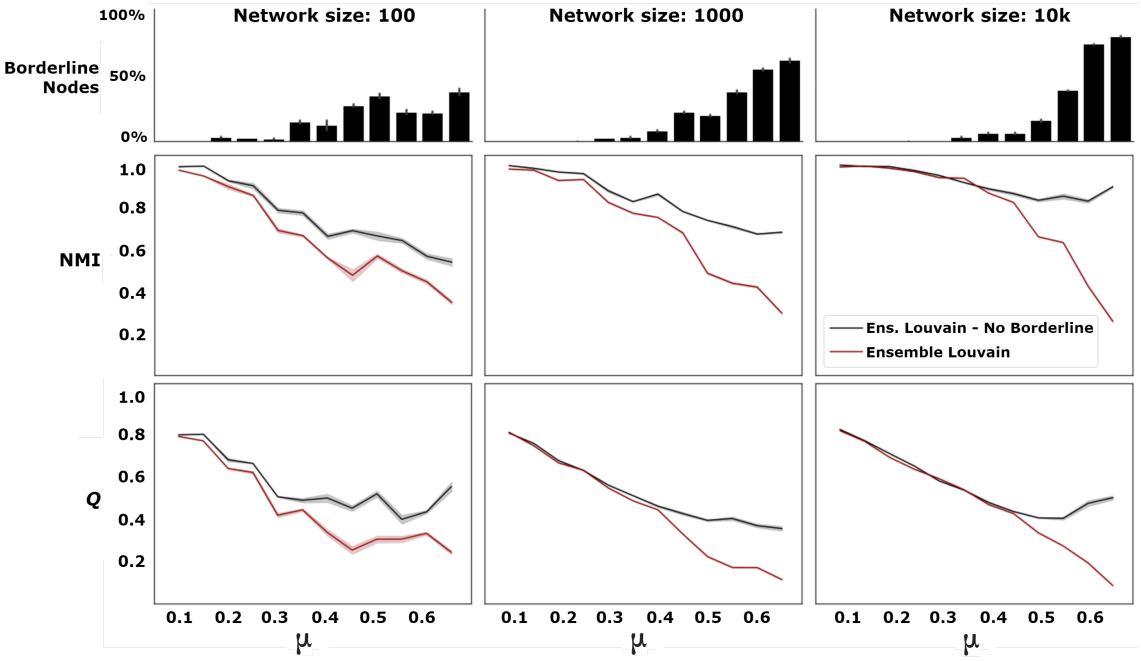

Figure A.1: **Influence of borderline nodes on community quality.** The top chart shows percentages of borderline nodes which cannot be reliably assigned to a community by Ensemble Louvain. The bottom two charts show the influence of borderline nodes on the *NMI* performance and modularity (Q) scores, respectively. Red lines represent the original Ensemble Louvain results, while black lines show results excluding the borderline nodes. The experiments are run on the standard LFR networks of three different network sizes. Results show that the percentage of borderline nodes correlates with the mixing parameter μ, while the quality of the non-borderline communities remains high.

A.2 Ensemble Louvain Parameters Analysis

The proposed Ensemble Louvain algorithm requires two main hyperparameters to be set: the number of Louvain runs r and the co-membership threshold ct for creating the co-membership network. Here, we use the 500 Unconstrained LFR networks and explore the impact of both hyperparameters.

First, we fix r to hundred runs and vary ct from 0.5 to 1.0. We then calculate both the average F_1 score on ground truth (performance) and the pair-wise F_1 (stability) across all networks. Results are presented in Figure A.2. We observe that the algorithm is not sensitive to small changes in the parameters. Moreover, values of ct between 0.8 and 0.9 provide the best stability, while values between 0.7 and 0.9 provide the best performance on ground truth. Thus, we suggest $ct = 0.85$ as a good starting point. Yet, the "best" value depends on the structure of the network one is analyzing. Trying out different values could be beneficial for the task at hand.

We have experimented with the r parameter in a similar manner. Results on the right side of Figure A.2 show consistently good results for performance, even with a small number of Louvain runs (such as $r = 10$). Yet, the number of runs holds a bigger impact on the stability of the partition. When $r = 10$, executions of Ensemble Louvain produce partitions with $F_1 = 0.9$, while changing to $r = 1000$, the stability approaches $F_1 = 1$, showing almost perfect stability.

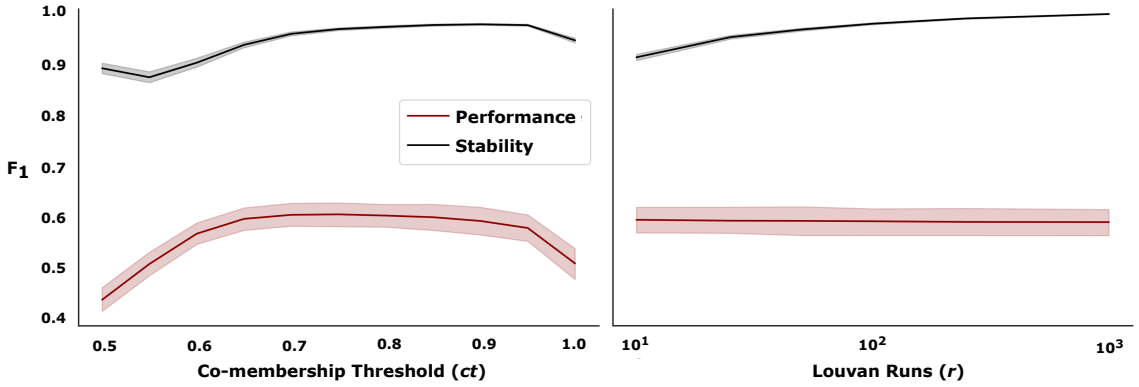

Figure A.2: **Ensemble Louvain parameter sensitivity.** Stability and performance results of Ensemble Louvain on 500 Unconstrained LFR networks. In the left-hand chart, the co-membership threshold (ct) varies from 0.5 to 1, with constant number of Louvain runs $r = 100$. In the right-hand chart, the number of Louvain runs (r) varies from 10 to 1000, with $ct = 0.85$. In both cases, the stability and performance are estimated by the F_1 score. Results suggest that parameter values of $0.7 < ct < 0.9$ and $10^2 < r$ provide consistent results both in terms of stability and performance.

A.3 Ensemble Louvain Parallelism

Finally, we benchmarked Ensemble Louvain's execution times, testing the benefits of parallelism of the Louvain runs. We created LFR networks with varying size (from $n = 100$ to $n = 20k$) and ran Ensemble Louvain (with $r = 100$) using multiple CPU core settings on an AMD Ryzen 7 5800X 8-core processor. The results in Figure A.3 show that the execution time difference between standard Louvain and Ensemble Louvain is constant for all network sizes. The parallelism proves useful for four cores or higher, improving speed close to linearly to the number of cores. When one looks at the execution time when increasing r (the number of Louvain runs), again, the execution time almost linearly increases. We use the term "almost" because the updates of the co-membership matrix are the additional non-parallel step that must be executed except the Louvain runs, adding a small overhead.

A.4 Evaluating Community Evolution with Ensemble Louvain

Techniques that tackle the instability issue in dynamic community detection are referred to as temporal smoothing. Adopting a specific smoothing strategy can lead to computational constraints, but it is a crucial step if one aims for finer results [10]. This is where Ensemble Louvain comes into play, applying temporal smoothing using the ensembles, significantly stabilizing the results, and with that, removing a large portion of the signal noise in the tracking of community evolution. Using ensembles as a smoothing strategy was mentioned as a prospect in the past as well [34], [36], yet without further quantitative experimental analysis of it being used. Here, we compare Ensemble Louvain with the standard Louvain and explore the impact of using the first.

As a continuation of our work [40], [67], we apply the community evolution analysis on the Slovenian Twitter data from January 2018 to December 2021. We create retweet networks out of 24-week data, with a sliding window of one week, resulting in 185 networks, with 01.01.2018-18.06.2018 (*week 0*) being the first, and 12.07.2021-27.12.2021 (*week 185*)

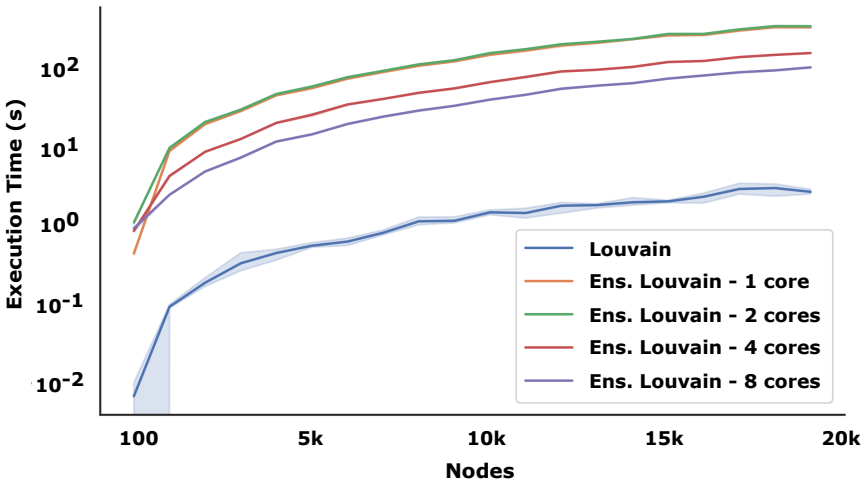

Figure A.3: **Parallelized Ensemble Louvain execution times.** $(r = 100, ct = 0.85)$**.** The chart shows the execution time of Ensemble Louvain in comparison to Louvain, with respect to the number of nodes in standard LFR networks. Ensemble Louvain execution times are reported for runs on various numbers of CPU cores. Execution times show that increasing the number of cores speeds up Ensemble Louvain almost linearly. Although Ensemble Louvain contains a hundred iterations of standard Louvain, its speed is slower by only twelve to fifteen times, when Ensemble Louvain execution times are reported for runs on various numbers of CPU cores.

the last network in the sequence. Additionally, each network is created with exponential decay on the edge weights (from latest to oldest retweets), so that we prevent detecting "changes" due to lost structure patterns of the trailing data. With that, we instead emphasize the actual community behavior shifts due to new events.

To detect community structure changes, we compare the similarity of two adjacent partitions (e.g., the *week 0* with the *week 1 network community partition*). To compare Ensemble Louvain to the standard Louvain, we apply the described procedure three times for the whole timeline and compare all adjacent partition combinations, for both algorithms. A graphical representation of the results is shown in Figure A.3, while the statistics regarding the comparison between Louvain and Ensemble Louvain are presented in Table A.1.

Figure A.4 and Table A.1 show that the Ensemble Louvain produces less noise (more stable results) when comparing the outputs of the experiments. In terms of numbers, the average and total standard deviation (noise) of F_1 is five times lower for Ensemble Louvain. In practice, this means that the randomness of the process is not such a strong factor that could influence the community evolution analysis, ensuring that the observed changes in the partitions are actually data-driven events.

We also observe a generally higher F_1 score throughout the whole timeline for the Ensemble Louvain. This means that, according to the Ensemble Louvain outputs, the adjacent partition differences are significantly lower compared to when analyzed using Louvain. The borderline nodes that Ensemble Louvain detects do not change their behavior rapidly, thus they usually maintain their non-affiliation, increasing the general F_1 score of the adjacent networks. Additionally, having these unstable nodes out of the large communities, the similarity between the communities becomes even higher, ending with a consistently higher score compared to the standard Louvain. Finally, tracking community evolution benefits from these behaviors, as the partition noise is removed from the process

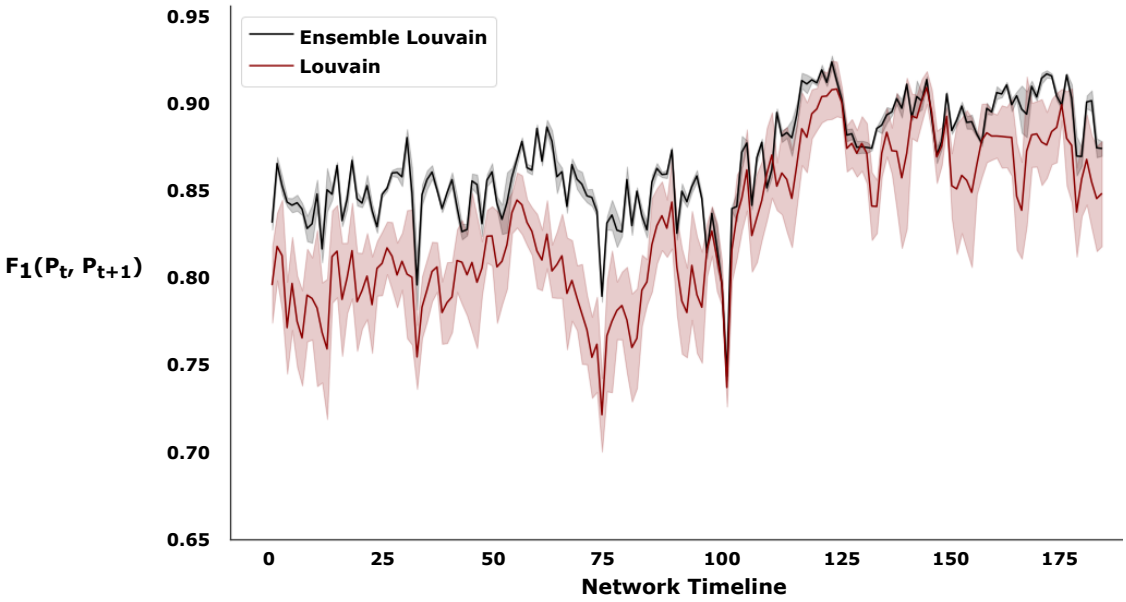

Figure A.4: **Tracking community evolution: Louvain vs. Ensemble Louvain** ($r = 250$, $ct = 0.95$). The x-axis shows the timeline of the detected network partitions in 185 weekly increments over a four-year period. The y-axis shows the F_1 similarity score between the pairs of adjacent partitions in time (partition P_t compared to partition P_{t+1}. A lower F_1 value indicates a larger change in the community structure, while a higher F_1 indicates higher similarity. Shaded areas show the (in)stability of the results over five runs. The shaded area represents the standard error of the mean. The figure shows Ensemble Louvain's general effect on stabilizing results, helping dynamic community tracking.

Table A.1: A comparison of Louvain and Ensemble Louvain on the community evolution case study (see Figure A.3). We compare adjacent network partition P_t and P_{t+1} with the F_1 score. There are 185 pairs of partitions to compare, and the results show an average mean value of F_1 and its standard deviation for both algorithms.

		Louvain	Ensemble Louvain
	Avg. std.	0.023 ± 0.008	0.004 ± 0.003
$F_1(P_t, P_{t+1})$	Avg. mean	0.834 ± 0.043	0.869 ± 0.031

of detecting change, making Ensemble Louvain preferred over the standard Louvain.

Appendix B

Metrics for Community Detection

Here, we describe how we evaluate the community detection algorithms for stability and performance, while also covering the metrics used for evaluation.

Stability evaluation comes down to comparing two output partitions of the same algorithm. If an algorithm is perfectly stable, it means that it always produces the same partition. Once we have the partition outputs of multiple runs, we measure the similarity of each partition pair, acquiring a distribution of similarities, which we then either plot as in Fig.2.2 for the Standard LFR networks or calculate the Friedman-Nemenyi ranking as in Fig.2.4 for the Unconstrained LFR networks.

The performance evaluation is more straightforward, as we only measure the similarity of the multiple runs' partitions with the ground truth, measuring how "close" the results are to the expected output, again, resulting in a distribution of similarities. We present these distributions for Standard and Unconstrained LFR networks in the same manner as the stability results (see Fig.2.3 and Fig.2.4).

Essentially, both performance and stability evaluation is simply measuring the similarity between two partitions. We use three partition similarity metrics: Normalized Mutual Information (NMI), Adjusted Rand (ARI) Index, and the BCubed F1 (or simply F1).

B.1 Normalized Mutual Information – NMI

NMI [37] is an evaluation metric from the field of clustering, which is very similar to the community detection problem. Here, we compare pairs of communities of the two partitions by calculating the probability of a node belonging to the respective communities and their intersection. Let $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ and $\Psi = \{\psi_1, \psi_2, ..., \psi_j\}$ be the two partition community sets we would want to compare (where Ψ would be the ground-truth in performance evaluation, and a second partition output for stability evaluation). And let N be the number of nodes in the network. Normalized mutual information (NMI) is defined as:

$$NMI(\Omega, \Psi) = \frac{I(\Omega; \Psi)}{[H(\Omega) + H(\Psi)]/2}$$

where I is mutual information, calculated as:

$$I(\Omega; \Psi) = \sum_k \sum_j P(\omega_k \cap \psi_j) log \frac{P(\omega_k \cap \psi_j)}{P(\omega_k)P(\psi_j)} = \sum_k \sum_j \frac{|\omega_k \cap \psi_j|}{N} log \frac{|\omega_k \cap \psi_j|}{|\omega_k||\psi_j|}$$

where $P(\omega_k)$, $P(\psi_j)$ and $P(\omega_k \cap \psi_j)$ are the probabilities of a node being in communities ω_k, ψ_j and their intersection. The second equivalence is for a maximum likelihood estimate

of the probabilities (i.e., the estimate of each probability is the corresponding relative frequency). H is entropy, defined as:

$$H(\omega) = -\sum_k P(\omega_k) log P(\omega_k) = -\sum_k \frac{|\omega_k|}{N} log \frac{|\omega_k|}{N}$$

where, again, the second equation is based on maximum likelihood estimates of the probabilities.

$I(\Omega;\Psi)$ measures the amount of information by which our knowledge about the first partition increases when we know about the second partition or vice versa. The minimum of $I(\Omega;\Psi)$ is 0 if at least one of the partitions is random. In that case, knowing about a node in one of the partitions does not give any new information about where the node could be in the second partition. Maximum mutual information is reached for a partition Ω_{exact} that identically recreates the partition Ψ. Yet, it is also reached when Ω_{exact} is further subdivided into smaller communities. In particular, a partition with $K = N$ one-node communities has maximum mutual information. This is because mutual information does not penalize large cardinalities and thus does not formalize our bias that, other things being equal, fewer clusters are better.

The normalization by the denominator $[H(\Omega)] + H(\Psi)]/2$ fixes the problem since entropy tends to increase with the number of clusters. For example, $H(\Omega)$ reaches its maximum $logN$ for $K = N$, which ensures that NMI is low for $K = N$. The particular form of the denominator is chosen because $[H(\Omega)] + H(\Psi)]/2$ is a tight upper bound on $I(\Omega;\Psi)$. Thus, NMI is always a number between 0 and 1. Finally, we refer to NMI as being a community-wise evaluation metric, since it compares node sets instead of individual nodes.

B.2 Adjusted Rand Index – ARI

Adjusted Rand Index [68] is a commonly used metric to compare clustering results against external criteria, making it suitable for community detection evaluation as well. The main idea is to count the pairs of nodes which appear in the same cluster in both partitions.

Once more, let $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ and $\Psi = \{\psi_1, \psi_2, ..., \psi_j\}$ be the two partition community sets we would want to compare. And let N be the number of nodes in the network. Now, ARI is defined as follows:

$$ARI(\Omega;\Psi) = \frac{\sum_{i,j}\binom{|\omega_i \cap \psi_j|}{2} - [\sum_i \binom{|\omega_i|}{2} \sum_j \binom{|\psi_j|}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_i \binom{|\omega_i|}{2} + \sum_j \binom{|\psi_j|}{2}] - [\sum_i \binom{|\omega_i|}{2} \sum_j \binom{|\psi_j|}{2}]/\binom{N}{2}}$$

B.3 BCubed F_1

The BCubed measure was originally proposed to evaluate the effectiveness of document clustering [39]. Its properties were compared to a wide range of other extrinsic clustering evaluation metrics, with the conclusion that BCubed satisfies all the required qualitative properties [69]. Since data clustering and community detection in networks produce analogous results, one can also apply the BCubed measure to evaluate the detected communities.

The BCubed measure is applicable to individual nodes, communities, and network partitions in general. It decomposes the evaluation into calculating the precision and recall associated with each node in the network. The precision (Pre) and recall (Rec) are then combined into the F_1 score:

$$F_1 = 2\,\frac{Pre \cdot Rec}{Pre + Rec}.$$

The F_1 score is a special case of Van Rijsbergen's effectiveness measure [70], where precision and recall can be combined with different weights. Hereinafter, we focus on definitions of precision and recall for different cases, and assume a balanced definition of the F_1 score as the harmonic mean. We first define the BCubed measure for a node, and then proceed with definitions of the network F_1.

Let $\Omega(n)$ denote the community (set of nodes) in which node n belongs in partition Ω and $\Psi(n)$ the community in which it belongs in partition Ψ. *Pre* and *Rec* for a node are defined as follows:

$$Pre(n) = \frac{|\Omega(n) \cap \Psi(n)|}{|\Psi(n)|},$$

$$Rec(n) = \frac{|\Omega(n) \cap \Psi(n)|}{|\Omega(n)|}.$$

Now, the *Pre* and *Rec* between the two partitions are defined as:

$$Pre(\Omega|\Psi) = \frac{1}{N} \sum_n Pre(n),$$

$$Rec(\Omega|\Psi) = \frac{1}{N} \sum_n Rec(n),$$

Since the F_1 is calculated as a sum of the *Pre* and *Rec* of individual nodes, we refer to this metric as node-wise. As the *Pre* and *Rec* are normalized by the number of nodes in the network, the F_1 ranges from 0 (no similarity between the partitions) to 1 (complete match between the partitions). A detailed definition of the F_1 metric, including the general form for non-identical node sets, can be found in our previous work [40].

Appendix C

Unconstrained LFR Experiments

In this final appendix, we present the experiments which led us to state that the Standard LFR benchmark fails to explore the space of possible networks created by LFR and can potentially lead to wrong conclusions when comparing two or more community detection algorithms.

The first experiment was to investigate how much networks vary in structure and density when one flips the method of generating the LFR networks: instead of varying only μ while keeping network size and all other parameters fixed, we fixed $\mu = 0.3$ and network size $n = 1000$ while varying all other parameters. Figure C.1 shows four cherry-picked examples of networks that visibly vary in network structure and density, although they are generated with the same method and have the same size. Intuitively, our assumption was that some methods might perform better on one network than others.

In order to statistically evaluate this, the next experiment was to create 500 networks in this manner, and then evaluate a few properties: number of edges, average degree, number of communities, size of the largest community, the diameter of the network, average clustering coefficient, the modularity optimized by Louvain and the *NMI* score resulted by Louvain. Figure C.2 presents a boxplot for each of these properties and shows astounding differences between the networks, showing a wide range of possible combinations for all parameters. The results which serve the most to our claim are the modularity and *NMI* scores produced by Louvain. Here, the greedily optimized modularity remarkably varies from 0.32 to 0.68 and the *NMI* from 0.75 to 0.98. This points out that network structure and community detection results heavily depend on the choice of the additional LFR parameters and not only on network size and the mixing parameter μ. It indicates that a methodology similar to or in the direction of our suggested Unconstrained LFR is a must when one is comparing the performance or stability of community detection algorithms.

Finally, in order to evaluate in what amount the *NMI* fluctuation is due to the in-

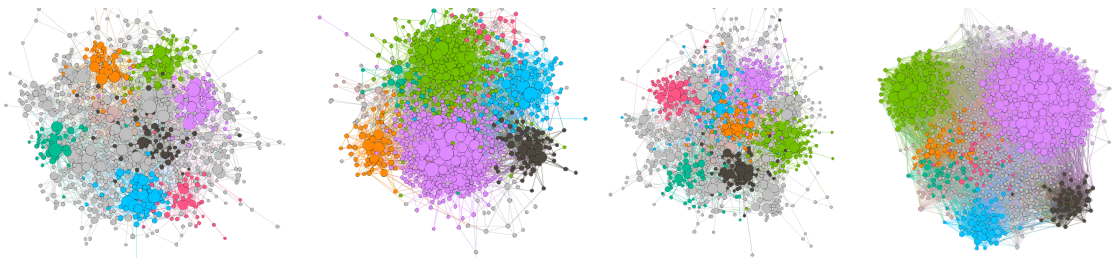

Figure C.1: Examples of Unconstrained LFR networks with changing parameters, but fixed mixing parameter $\mu = 0.3$ and network size $n = 1000$.

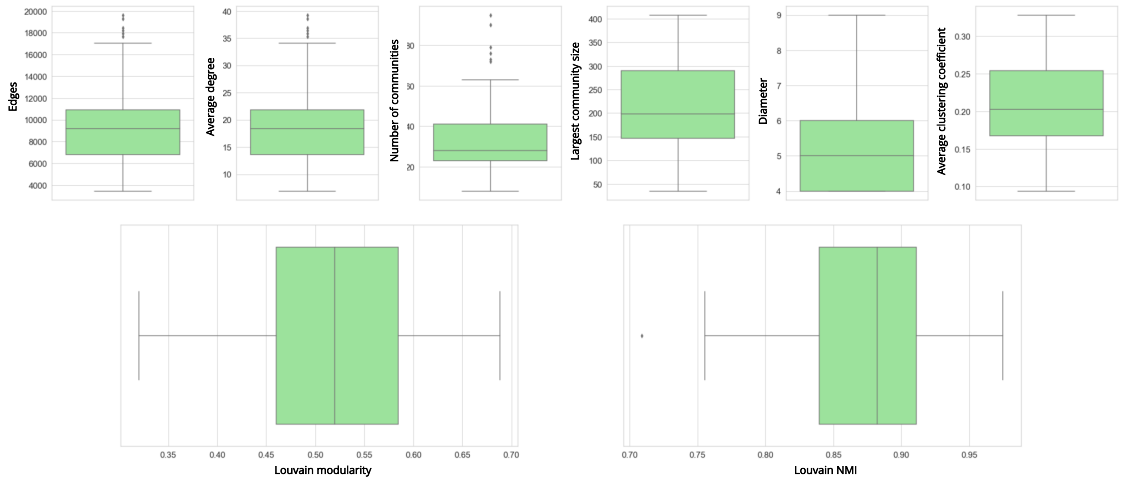

Figure C.2: Network structure and community detection box plots on Unconstrained LFR networks with changing parameters, but fixed $\mu = 0.3$ and $n = 1000$.

stability of the community detection algorithms or due to the actual differences of the networks, we take three popular greedy optimization algorithms (Edmot [16], Leiden [71] and Walktrap [72]) and evaluate the average score and the standard deviation of *NMI* on both Standard LFR and Unconstrained LFR with the same $\mu = 0.3$ and network size $n = 1000$. For the Standard LFR, we used the most common set of parameters, where $\tau_1 = 2$ and $\tau_2 = 1.1$.

Table C.1: NMI scores (means and standard deviations) of several community detection algorithms on Standard LFR networks and Unconstrained LFR networks.

	Standard LFR Benchmark			Unconstrained LFR Benchmark		
Algorithm	Edmot	Leiden	Walktrap	Edmot	Leiden	Walktrap
Mean NMI	0.78	0.81	0.67	0.61	0.62	0.69
Std of NMI	0.03	0.02	0.05	0.23	0.23	0.18

Table C.1 shows that for the Standard LFR all algorithms perform with a standard deviation from 0.02 to 0.05 with Edmot and Leiden expectedly outperforming Walktrap. Yet, for the Unconstrained LFR, the standard deviation of *NMI* drastically jumps to a range from 0.18 to 0.23 depending on the algorithm. Additionally, the average performance of Edmot and Leiden goes considerably lower, suggesting that they failed to recognize the community structure for some of the Unconstrained LFR networks.

References

[1] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] S. H. Strogatz, "Exploring complex networks," *nature*, vol. 410, no. 6825, pp. 268–276, 2001.

[3] L. A. Amaral and J. M. Ottino, "Complex networks," *The European physical journal B*, vol. 38, no. 2, pp. 147–162, 2004.

[4] F. Kuhn and R. Oshman, "Dynamic networks: Models and algorithms," *ACM SIGACT News*, vol. 42, no. 1, pp. 82–96, 2011.

[5] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

[6] D. J. Hill and G. Chen, "Power systems as dynamic networks," in *2006 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2006, 4–pp.

[7] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[8] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016. DOI: 10.1016/j.physrep.2016.09.002.

[9] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical review E*, vol. 80, no. 5, p. 056 117, 2009.

[10] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–37, 2018.

[11] R. Cazabet, G. Rossetti, and F. Amblard, *Dynamic community detection*, 2017.

[12] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks," *ACM Computing Surveys*, vol. 51, no. 2, pp. 1–37, 2018, ISSN: 1557-7341. DOI: 10.1145/3172867. [Online]. Available: http://dx.doi.org/10.1145/3172867.

[13] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 4, no. 5, pp. 512–546, 2011.

[14] P. Holme and J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.

[15] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, P10008, 2008.

[16] P.-Z. Li, L. Huang, C.-D. Wang, and J.-H. Lai, "Edmot: An edge enhancement approach for motif-aware community detection," in *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 479–487.

[17] A. Prat-Pérez, D. Dominguez-Sal, and J.-L. Larriba-Pey, "High quality, scalable and parallel community detection for large real graphs," in *Proc. 23rd International Conference on World Wide Web*, 2014, pp. 225–236.

[18] Z. Lu, J. Wahlström, and A. Nehorai, "Community detection in complex networks via clique conductance," *Scientific reports*, vol. 8, no. 1, pp. 1–16, 2018.

[19] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Physical Review X*, vol. 4, no. 1, p. 011 047, 2014.

[20] T. P. Peixoto, "Descriptive vs. inferential community detection: Pitfalls, myths and half-truths," *arXiv:2112.00183*, 2022.

[21] T. Aynaud and J.-L. Guillaume, "Static community detection algorithms for evolving networks," in *8th International symposium on modeling and optimization in mobile, Ad Hoc, and wireless networks*, IEEE, 2010, pp. 513–519.

[22] G. Lin, S. Liu, A. Zhou, *et al.*, "Community detection in power grids based on louvain heuristic algorithm," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, IEEE, 2017, pp. 1–4.

[23] T.-K. Nguyen, M. Coustaty, and J.-L. Guillaume, "A new image segmentation approach based on the louvain algorithm," in *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, IEEE, 2018, pp. 1–6.

[24] S. Heymann and B. Le Grand, "Visual analysis of complex networks for business intelligence with gephi," in *2013 17th International Conference on Information Visualisation*, IEEE, 2013, pp. 307–312.

[25] F. Durazzi, M. Müller, M. Salathé, and D. Remondini, "Clusters of science and health related twitter users become more isolated during the COVID-19 pandemic," *Scientific Reports*, vol. 11, no. 19655, 2021. DOI: 10.1038/s41598-021-99301-0.

[26] T. Radicioni, T. Squartini, E. Pavan, and F. Saracco, "Networked partisanship and framing: A socio-semantic network analysis of the Italian debate on migration," *PLoS ONE*, vol. 16, no. 8, pp. 1–24, Aug. 2021. DOI: 10.1371/journal.pone.0256705.

[27] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[28] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[29] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[30] P. Bak, K. Chen, and C. Tang, "A forest-fire model and some thoughts on turbulence," *Physics letters A*, vol. 147, no. 5-6, pp. 297–300, 1990.

[31] J. Yang and J. Leskovec, "Community-affiliation graph model for overlapping network community detection," in *2012 IEEE 12th international conference on data mining*, IEEE, 2012, pp. 1170–1175.

[32] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046 110, 2008.

[33] Z. Yang, R. Algesheimer, and C. J. Tessone, "A comparative analysis of community detection algorithms on artificial networks," *Scientific reports*, vol. 6, no. 1, pp. 1–18, 2016.

[34] T. Chakraborty, N. Park, A. Agarwal, and V. Subrahmanian, "Ensemble detection and analysis of communities in complex networks," *ACM/IMS Transactions on Data Science*, vol. 1, 1 2020. DOI: 10.1145/3313374.

[35] G. K. Orman, V. Labatut, and H. Cherifi, "Towards realistic artificial benchmark for community detection algorithms evaluation," *International Journal of Web Based Communities*, vol. 9, no. 3, pp. 349–370, 2013.

[36] V. Poulin and F. Théberge, "Ensemble clustering for graphs: Comparisons and applications," *Applied Network Science*, vol. 4, no. 1, pp. 1–13, 2019.

[37] T. O. Kvålseth, "On normalized mutual information: Measure derivations and properties," *Entropy*, vol. 19, no. 11, p. 631, 2017.

[38] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[39] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *Proc. 17th Intl. Conf. on Computational Linguistics (COLING)*, Stroudsburg, PA, USA, 1998, pp. 79–85.

[40] B. Evkoski, I. Mozetič, N. Ljubešić, and P. Kralj Novak, "Community evolution in retweet networks," *Plos one*, vol. 16, no. 9, e0256175, 2021.

[41] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026 113, 2004.

[42] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 46–65, 2014.

[43] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the national academy of sciences*, vol. 101, no. 9, pp. 2658–2663, 2004.

[44] F. Folino and C. Pizzuti, "An evolutionary multiobjective approach for community discovery in dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1838–1852, 2013.

[45] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, pp. 1–31, 2009.

[46] R. Görke, P. Maillard, A. Schumm, C. Staudt, and D. Wagner, "Dynamic graph clustering combining modularity and smoothness," *Journal of Experimental Algorithmics (JEA)*, vol. 18, pp. 1–1, 2013.

[47] R. Cazabet, F. Amblard, and C. Hanachi, "Detection of overlapping communities in dynamical social networks," in *2010 IEEE second international conference on social computing*, IEEE, 2010, pp. 309–314.

[48] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.

[49] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5249–5253, 2004.

[50] M. Rosvall and C. T. Bergstrom, "Mapping change in large networks," *PLoS ONE*, vol. 5, no. 1, e8694, 2010.

[51] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific Reports*, vol. 2, no. 1, p. 336, 2012. DOI: 10.1038/srep00336.

[52] J. Dahlin and P. Svenson, "Ensemble approaches for improving community detection methods," *arXiv:1309.0242*, 2013.

[53] R. Kanawati, "Yasca: An ensemble-based approach for community detection in complex networks," in *International Computing and Combinatorics Conference*, Springer, 2014, pp. 657–666.

[54] U. Brandes, D. Delling, M. Gaertler, *et al.*, "Maximizing modularity is hard," *arXiv:0608255*, 2006.

[55] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[56] P. B. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton University, USA, 1963.

[57] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. Jan, pp. 1–30, 2006.

[58] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[59] S. Herbold, "Autorank: A python package for automated ranking of classifiers," *Journal of Open Source Software*, vol. 5, no. 48, p. 2173, 2020. DOI: 10.21105/joss.02173. [Online]. Available: https://doi.org/10.21105/joss.02173.

[60] L. Peel, D. B. Larremore, and A. Clauset, "The ground truth about metadata and community detection in networks," *Science advances*, vol. 3, no. 5, e1602548, 2017.

[61] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[62] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 2–es, 2007.

[63] D. Cherepnalkoski and I. Mozetič, "Retweet networks of the european parliament: Evaluation of the community structure," *Applied network science*, vol. 1, no. 1, pp. 1–20, 2016.

[64] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *Third international AAAI conference on weblogs and social media*, 2009.

[65] P. Kralj Novak, T. Scantamburlo, A. Pelicon, M. Cinelli, I. Mozetič, and F. Zollo, "Handling disagreement in hate speech modelling," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, 2022, pp. 681–695.

[66] H. Kim, S. H. Lee, *et al.*, "Relational flexibility of network elements based on inconsistent community detection," *Physical Review E*, vol. 100, no. 2, p. 022 311, 2019.

[67] B. Evkoski, I. Mozetič, and P. K. Novak, "Community evolution with Ensemble Louvain," *Complex networks*, pp. 58–60, 2021.

[68] D. Steinley, "Properties of the hubert-arable adjusted rand index.," *Psychological methods*, vol. 9, no. 3, p. 386, 2004.

[69] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[70] C. Van Rijsbergen, *Information Retrieval*, 2nd. Newton, MA, USA: Butterworth, 1979.

[71] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: Guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[72] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *International symposium on computer and information sciences*, Springer, 2005, pp. 284–293.

Bibliography

Publications Related to the Thesis

Journal Articles

B. Evkoski, I. Mozetič, N. Ljubešić, and P. Kralj Novak, "Community evolution in retweet networks," *Plos one*, vol. 16, no. 9, e0256175, 2021.

B. Evkoski, A. Pelicon, I. Mozetič, N. Ljubešić, and P. Kralj Novak, "Retweet communities reveal the main sources of hate speech," *PloS one*, vol. 17, no. 3, e0265602, 2022.

B. Evkoski, N. Ljubešić, A. Pelicon, I. Mozetič, and P. Kralj Novak, "Evolution of topics and hate speech in retweet network communities," *Applied Network Science*, vol. 6, no. 1, pp. 1–20, 2021.

Conference Papers

B. Evkoski, I. Mozetic, N. Ljubešic, and P. K. Novak, "A Slovenian retweet network 2018-2020," *Information Society*, 2020.

B. Evkoski, I. Mozetič, and P. K. Novak, "Community evolution with Ensemble Louvain," *Complex networks*, pp. 58–60, 2021.

B. Evkoski, I. Mozetič, N. Ljubešić, and P. K. Novak, "Evolution of political polarization on Slovenian Twitter," *Complex Networks*, pp. 325–327, 2020.

Other Publications

Conference Papers

V. Andonovikj, P. Boskoski, B. Evkoski, T. Redek, and B. Mileva Boshkoska, "Community analysis in Slovenian labour network 2010-2020," *Journal of Decision Systems*, pp. 1–11, 2022.

L. Jovanovska, B. Evkoski, M. Mirchev, and I. Mishkovski, "Demographic analysis of music preferences in streaming service networks," in *Complex Networks XI*, Springer, 2020, pp. 233–242.

Biography

The author of this thesis was born on September 13, 1996 in Skopje, Republic of Macedonia. He finished his Bachelor studies in 2019, at the Faculty of Computer Science and Engineering - "SS. Cyril and Methodius" University in Skopje, on the topic of "Predictive Models Management in Spark Structured Streaming".

He enrolled in the master studies in Information and Communication Technologies at the International Jožef Stefan Postgraduate School in October 2020. He conducted his research at the Department of Knowledge Technologies under the supervision of Prof. Dr. Petra Kralj Novak. Bojan holds an "AdFutura" scholarship, awarded by the Slovene Human Resources Development and Scholarship Fund.

His research area is in the field of network science, more specifically, community detection. His primary application fields are political and social studies, where he uses network analysis techniques to gain insight into both online and in-person human interaction phenomena.