

You are the way you structurally talk: structural-temporal neighbourhoods of posts to characterize users in online forums

Alberto Lumbreras*, Bertrand Jouve[†], Marie Guégan[‡], Julien Velcin[§]

*Technicolor, France

Email: alberto.lumbreras@gmail.com

[†]Université de Toulouse; UT2; FRAMESPA/IMT,
5 allée Antonio Machado, 31058 Toulouse, cedex 9

Email: jouve@univ-tlse2.fr

[‡]Technicolor

975 Avenue des Champs Blancs
35576 Cesson-Sevigné,
France

Email: marie.guegan@technicolor.com

[§]Laboratoire ERIC, Université de Lyon,
5 avenue Pierre Mendès France, 69676, Bron
France

Email: julien.velcin@univ-lyon2.fr

Abstract—The abstract goes here.

I. INTRODUCTION

The popularization of online forums has brought a growing interest on their underlying dynamics. As any other complex system, the dynamic of online forums can be studied at different levels, from the most macro to the most micro. Macro dynamics are, for instance, the evolution of some global properties of the social graph such as its diameter, or its distribution degree. Micro dynamics are, for instance, the triadic motifs that represent local phenomena such as transitivity (friends of my friends are also my friends).

An interesting question in online communities is that concerning roles. In sociology, roles are generally seen as the set of expected behaviours that are attached to a position in the community. Extrapolating the notion of role, some researchers have looked for roles in online forums. Some others have tried to detect the roles and the users who hold that roles.

Roles can also be studied from the macro or the micro perspective. If studied from the macro, we can analyse the number of users, the percentage of replied post, its centrality in the network, and so forth.

In this paper, we focus on the analyse of roles at a micro level, and more specifically at the discussion level. We would like to answer the following question: are there different types of users in terms of the kind of conversation they participate in?

Our intuition is that some users like participating in some kind of discussion rather than other. Certainly, an analysis of the textual content will tell us much about a discussion.

However, due to the huge diversity of topics, vocabulary, and the difficulty of current algorithms to capture the language subtleties such as humour, irony, or changing contexts, we turn our attention towards the structure of the discussions. More precisely, we analyse the local graph structure in which a user post is embedded, in the hope that this structure will be meaningful since it also reflects the kind of conversation in that part of the thread. Formally, these local graphs are known as neighbourhoods.

The remaining of the paper is as follows. We first discuss about the convenience of the classic neighbourhood definition in dynamic graphs such as discussion trees. Then we introduce two new definitions that takes into account the time to overcome some of the limitations of the structural neighbourhood. We will finally apply these time-based neighbourhoods to detect clusters of users that tend to appear in the same time-based neighbourhoods.

II. DISCUSSION TREES

We represent a conversation thread as a tree graph where vertex represent posts and edges represent replies from some post to another. The tree is rooted at the post who started the thread. Figure 1 shows some real examples of trees in a Reddit¹ forum.

The *root* post of the tree is the post that starts the discussion. It is the only post that has no parent. A *leaf* is a post with no replies. A *branch* of the tree is the shortest path between the

¹www.reddit.com

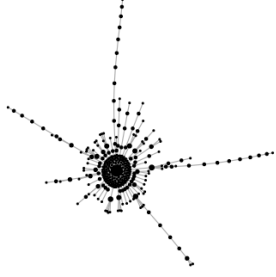


Fig. 1. Representation of discussion threads with post tree graphs. Vertex represent posts and edges represent replies between posts.

root and some of the leafs. Two branches may share their first posts.

III. STRUCTURAL NEIGHBOURHOODS IN DISCUSSION TREES

Extending the classic definition of neighbourhood according to which two vertices are neighbours if the distance between them is one, in this paper we define the *structural neighbourhood* as follows:

Definition 1: Given a tree graph G , the *structural neighbourhood* of radius r of post i , denoted as $\mathcal{N}_i(r)$, is the induced graph composed of all the vertices that are at distance equal or less than d from post i .

This definition has two drawbacks when used in the context of conversation trees. First, the dynamics of the conversation (time or order in which posts are attached to the tree) are not entirely captured in the structure of a tree representation. We know, for instance, that the time in which a node was attached to the tree is always posterior to that of its parent. But it is impossible to say, by just looking at the structure of the tree, the order in which a set of sibling posts replied to their common parent post. Thus, a single *structural neighbourhood* may sometimes correspond to very different dynamics. Second, a structural neighbourhood at a given radius r has an unbounded number of posts, and therefore the space of possible neighbourhoods is infinite. This poses a problem when trying to categorize conversations since many conversations, while structurally different, can be considered semantically equivalent (see Figure 2).

IV. STRUCTURO-TEMPORAL NEIGHBOURHOODS IN DISCUSSION TREES

In this section, we propose two new time-based definitions of neighbourhoods.

A. Order-based

Our first definition is based on the order in which posts are attached to the tree.

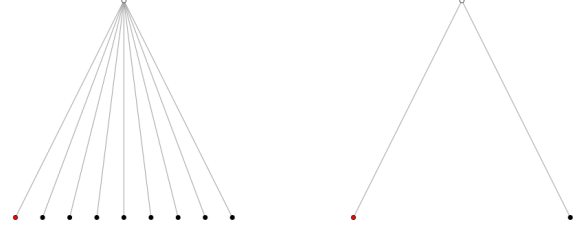


Fig. 2. The size of the neighbourhood with radius r is unbounded. These two graphs represent frequent neighbourhoods of a post (red) that replied to the root (white). However, the tree in the left corresponds to a very successful root while the three in the right has not brought the attention of too many users.

Definition 2: Given a tree graph G , the *structural-temporal neighbourhood* of distance d of post i , denoted as $\mathcal{N}_i^T(r, n)$ is the induced subgraph from its structural neighbourhood composed of the n vertices that are closer to i in time and for which there exists a path to i in $\mathcal{N}_i^T(r, n)$.

This definition has two advantages over the *structural neighbourhood*. First, the temporal aspect of the conversation is better taken into account since the neighbourhood only includes posts that are structurally and temporally close. Second, the size of the neighbourhood has an upper bound of $\max(\mathcal{N}_i(r), o)$.

B. Time-based

Our second definition uses the real timestamps of posts in order to decide where the boundaries of a neighbourhood are. We might naively set fixed time-based boundaries for the neighbourhood by only including in the neighbourhood those posts whose timestamp p_i is at distance less than τ from the ego post $|t_i - t_{ego}| < \tau$. However, the pace at which posts are added to the conversation may be very different between conversations, and even between different parts of the same conversation tree. Instead, we propose a definition based on the concept of *local dynamic*.

Definition 3: The temporal neighbourhood $\mathcal{N}_i^T(r)$ is the maximal subgraph of the structural neighbourhood $\mathcal{N}_i(r)$ where all the posts belong to the same local dynamic than the ego post.

Our definition of *local dynamic* is based on the detection of changepoints in the time stamps of the posts. Given a sequence of consecutive posts in the same branch b of a tree, denoted as p_s, \dots, p_e we say that they belong to the same vertical dynamic if there is no (vertical) changepoint p_i in b such that $p_1 \prec p_i \preceq p_n$. Similarly, given a sequence of chronologically sorted siblings s , denoted as p_s, \dots, p_e , we say that they belong to the same horizontal dynamic if there is no (horizontal) changepoint p_i in s such that $p_1 \prec p_i \preceq p_n$. Now we can outline the complete algorithm for time-based neighbourhood (Algorithm 1). Figure 3 illustrates a tree with a

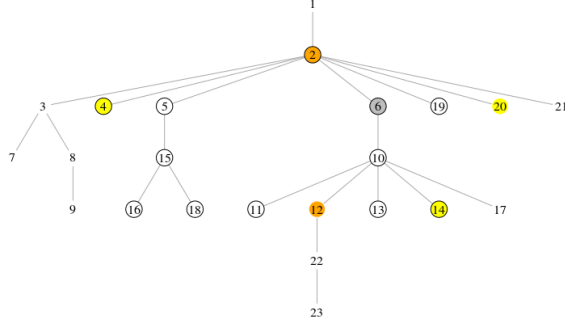


Fig. 3. Time-based neighbourhood. Horizontal changepoints (yellow) and vertical changepoints (orange) represent posts that are temporally far from their predecessors (siblings or parents) and therefore set the limits of the neighbourhood. Post 6 is the ego, and posts with circular borders are those belonging to the time-based neighbourhood.

set of breakpoints and the temporal neighbourhood for a given node.

Algorithm 1 Extraction of time-based neighbourhood

Input: Posts tree g , vertical breakpoints, horizontal breakpoints, ego post ego

Output: Subgraph of g with all vertices in $V(g)$

Compute structural neighbourhood $\mathcal{N}_i(r)$

ancestors \leftarrow ancestors(ego) in $\mathcal{N}_i(r)$

older_siblings \leftarrow older_siblings(ego) in $\mathcal{N}_i(r)$

dump $\leftarrow \emptyset$

for $bp \in$ vertical breakpoints **do**

if $bp \in$ ancestors **then**

 dump \leftarrow dump \cup ancestors(bp)

else

 dump \leftarrow dump \cup descendants(bp) \cup bp

end if

end for

for $bp \in$ horizontal breakpoints **do**

if $bp \in$ (older_siblings \cup ancestors) **then**

 dump \leftarrow dump \cup older_siblings(bp)

else

 dump \leftarrow dump \cup younger_siblings(bp) \cup bp

end if

end for

$\mathcal{N}_i^{(t)}(r) \leftarrow$ delete(dump_posts) from $\mathcal{N}_i(r)$

C. Colored neighbourhoods

In conversation trees, the root vertex plays an important role since it represents the post that opened the conversation and thus set the topic of the thread. Therefore, a reply to the root post is clearly different from a reply to any other post in the tree. We distinguish root posts from non-root posts by

assigning root posts a unique color (white). We also assign a unique color to the ego post (red).

D. Neighbourhood pruning

V. APPLICATION TO REDDIT FORUMS

In this section, we use our definition of structural-temporal neighbourhood to detect different types of users according to the neighbourhoods in which their posts are embedded.

Our dataset consists of all posts from March 2014 to May 2015 of the Podemos forum in Reddit²³. The Podemos forum was conceived in March 2014 as a tool for internal democracy, and forum members used it to debate ideological and organizational principles that were later formalized in their first party congress hold in Madrid the October 18th and 19th 2014. Nowadays, its members use it mainly to share and discuss about political news. The forum contains 83,6119 posts spread over 47,803 threads and written by 26,193 users. (see Figure 4)

A. Neighbourhood census

In this section we analyse the neighbourhood census of our dataset. A counter for every neighbourhood structure is created the first time we detect them. For every post, we extract its neighbourhood (order-based or time-based) and prune it as explained above (Section IV-D). Then check whether it is isomorphic to some of the already seen neighbourhoods. If this is the case, then we increment the counter for the neighbourhood previously seen. Otherwise we create a new entry for the current neighbourhood.

The census obtained with both type of neighbourhood are similar for the most frequent structures? That means that both kind of neighbourhoods are isomorphic for the most popular discussion structures.

B. Conversation-based clustering of users

In this section we categorize users based on the neighbourhoods of their posts. We limit our study to a set of 100 *active users* who wrote more than 100 posts. For those users, we create an initial feature matrix $U \times N$ where U is the number of users and N is the number of neighbourhoods in the census, and where the the position (u, n) is a counter of the number of times that a post written by user u has a neighbourhood isomorphic to n . We drop those feature columns that are zero for every user (these are neighbourhoods seen only around posts of users with low-level activity). To make the feature vector of a user independent on the number of posts, we transform the counts into percentages. And since some features have much higher percentages than others for most users, we scale and normalize the matrix so that every feature has mean 0 and variance 1. To avoid non-significant scores, we remove also the feature column corresponding to neighbourhoods that have a frequency less than 50 among the active users.

²<https://www.reddit.com/r/podemos>

³I have also the data for *gameofthrones*, *france*, *datascience*, *machinelearning*, *complexsystems*, *philosophy*, *twoxchromosomes*, *trees*, *sex*

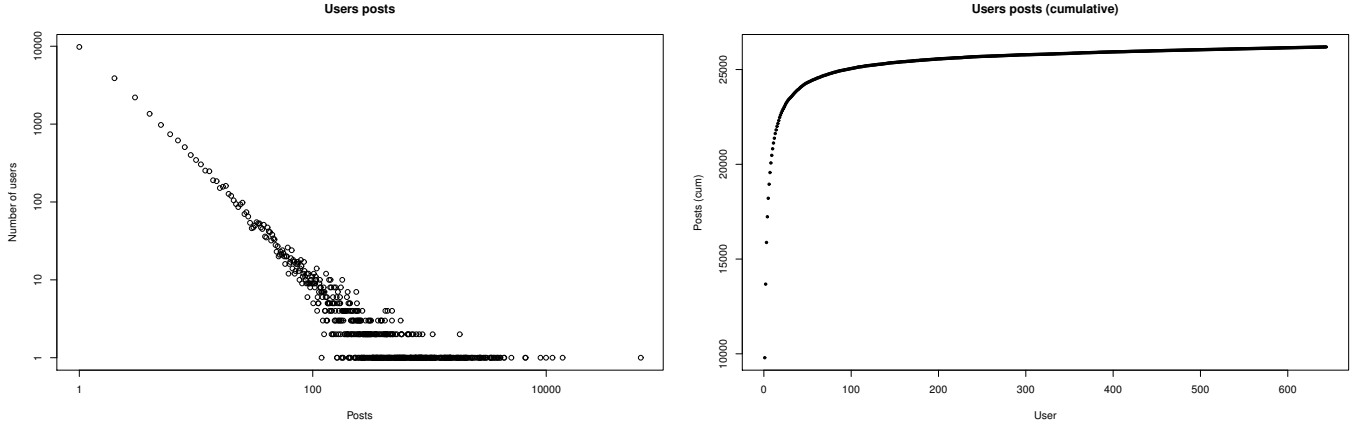


Fig. 4. Log-log and cumulative distributions and of posts per user.

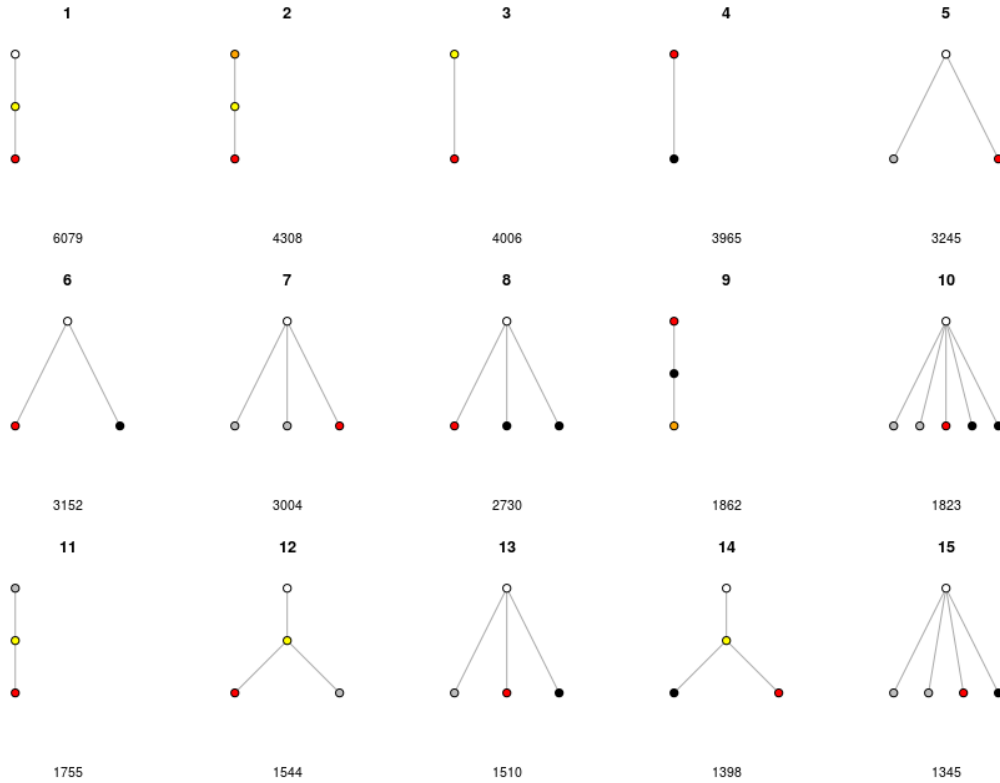
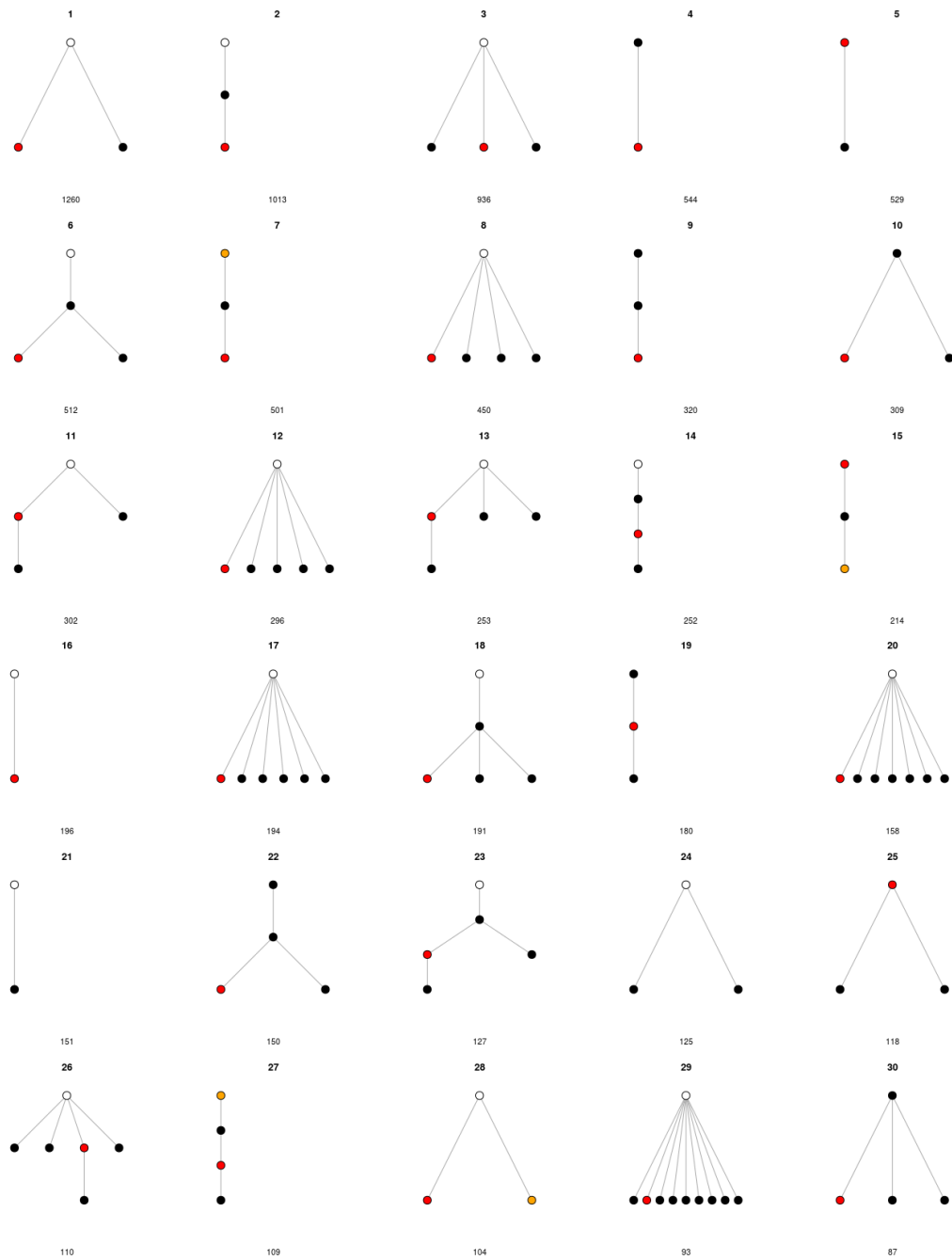


Fig. 5. Most frequent order-based neighbourhoods with $r = 2$ and $n = 4$

We use k-means to find the clusters, though one can use any other clustering method. Since the plot over the Within-Cluster Sum of Squares for $k = 1, \dots, 20$ did not show any clear elbow we chose $k = 3$ clusters so that clusters are easier to interpret. We did the clustering over a feature matrix with order-based neighbourhoods and another feature matrix with time-based neighbourhoods. Figures ?? and shows the PCA projections of the users coloured by their assigned cluster.

VI. CONCLUSIONS

We presented a method to characterise conversations of users in online threads. Due to the tree nature of online threads, traditional patterns such as triads are not able to capture much of relevant dynamics of a conversation. Our defined order-based and temporal-based neighbourhoods are able to capture a very rich variety of structures. We used this neighbourhoods to characterise users in terms of the structure of the conversations they participate in and showed that, indeed, there are different types of structural conversationalists



The concept of structural-temporal neighbourhood opens the door to some interesting paths of research. One might wonder whether other pruning are more pertinent than the proposed here. Also, even after pruning some neighbourhoods suggest the same type of conversation, so a manual merge might be convenient. Maybe sociologists can help to merge the found neighbourhoods in a meaningful way.

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.