Structural-temporal neighbourhoods of posts to characterize conversations in online forums

(Draft version: April 30, 2016)

Alberto Lumbreras
Julien Velcin
Laboratoire ERIC
Université de Lyon, France
alberto.lumbreras@univ-lyon2.fr
julien.velcin@univ-lyon2.fr

Bertrand Jouve
FRAMESPA/IMT
Université de Toulouse, France
jouve@univ-tlse2.fr

Marie Guégan
Technicolor, France
marie.guegan@technicolor.com

***Abstract*—Users of social networks are often characterised by extracting some relevant features from the graphs associated to that network. The structural dynamic of the conversation are usually forgotten due to a lack of proper tools. We present a purely graph-based method to cluster users in online forums.**

I. Introduction

The interactions between users in online forums are often modelled as complex networks. As such, they can be studied from many levels, each of which is represented by a graph where vertices and edges may represent different things. The most studied graph is a graph where vertices represent users and edges represent interactions between users (a post from one user replying to a post from another user). From the point of view of the community, some typically analysed properties are the degree distribution, the clustering coefficient, density, or the diameter of the graph. From the point of view of the individual users, analyses are typically focused either on the centrality of users, on the community structure or in blockmodeling (groups of users that tend to interact with the same other groups) [1]. Another common graph is a tree graph where vertices represent posts (or e-mails) and edges from one vertex to another indicate that the first is a reply to the second. Given the different discussion trees of a forum (a forest) the global analyses include depth distribution, branching factors, or even the time between two posts [2]. Studying discussion trees from the point of view of the users means, since trees are discussions, studying the user at a conversational level. Unfortunately, there is a lack of structural tools to perform this analysis. In this paper, we propose to fill this gap through the concept of *structural-temporal neighbourhood*.

Our goal in this paper is two-fold: on the one hand, to illustrate how structural-temporal neighbourhoods can play the role of triads for conversation trees, in the sense that they show us the local structures or dynamics from which the bigger graph emerges. On the other hand, to show how structural-temporal neighbourhoods can be used for the detection of different types of conversationalists in online forums or any other type of online discussion that is representable by a tree structure (e.g.: e-mails).

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 our two definitions of structural-temporal neighbourhoods. To illustrate the kind of neighbourhoods that we obtain in a real data, we analyse the neighbourhood census of a Reddit forum. Finally, we apply both neighbourhood definitions to detect clusters of users that tend to appear in the same neighbourhoods. We close the paper with some suggestions of future research.

II. Discussion trees

We represent a discussion thread by a tree graph $G = (V, E)$ where V is a set of n vertices representing the posts (also called messages or comments) and E is a set of $n-1$ directed edges that say which posts replied to which post. Vertices have two attributes namely time and author. If for two given vertices v_i, v_j there exists an edge $e = (v_i, v_j) \in E$ then v_j is called the *parent* of v_i, denoted as $p(v_i) = v_j$. The *root* of the tree is the only vertex with no parent, and corresponds to the post that starts the discussion. We say that two vertices v_i, v_k are *siblings* if and only if $p(v_i) = p(v_k)$. A vertex v_l is an ancestor of a vertex v_i and v_i is descendant of v_l if and only if v_l is in the path from v_i to the root. A *leaf* is a post with no replies. A *branch* is the path between a leaf and the root. Two vertices v_i and v_m are said to be *neighbours* if either $p(v_i) = v_m$ or $p(v_m) = v_i$. The neighbours of a vertex i are its parent and its children.

III. Structural neighbourhoods

Extending the classic definition of neighbourhood, we recall the definition of *structural neighbourhood*[1]:

Definition 1: Given a tree graph G, the *structure-based neighbourhood* of radius r of post i, denoted as $\mathcal{N}_i(r)$, is the induced graph composed by all the vertices that are at distance equal or less than r from post i.
In the context of discussion threads, this definition has two limitations. First, the decision on whether to include some

[1] We coined the term *structural* to differentiate if from the other neighbourhoods discussed in this paper.

post in the neighbourhood is only based on the structural distance, and therefore two posts that are at distance $d \leq r$ are considered neighbours of i regardless of the time when they were written although, in conversations, time plays an important role. Another consequence of looking only at the structure is that the number of possible neighbourhoods within a radius r is infinite. This poses a problem when trying to categorize conversations since many conversations, while structurally different, can be considered semantically similar (e.g.: we might not want to put in different categories two structures representing, respectively, a post with 50 replies and another with 40).

IV. Structural-temporal neighbourhoods

We propose two refinements of the structure-based neighbourhood to take into account the order and time at which posts are written. In the order-based definition, a new radius is set to include at most the n closest posts in time. In the time-based definition, the new radius is set to include all posts (in the structural-neighbourhood) until the first point where the conversation slows down.

A. Order-based

Definition 2: Given an ordered tree graph G, the *order-based neighbourhood* of radius r and order n of vertex i, denoted as $\mathcal{N}_i^O(r,n)$ is the induced subgraph from its structural neighbourhood $\mathcal{N}_i(r)$ composed by the n vertices that are closest to i in time and for which there exists a path to i in $\mathcal{N}_i^O(r,n)$.

An example of order-based neighbourhood is given in Figure 1. 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 not only near to i in the structure but also in time. Second, the size of the neighbourhood has an upper bound of $min(|\mathcal{N}_i(r))|, n)$ thus making the space of possible neighbourhood structures finite. The main limitation of this definition is that the we have no *a priori* criteria to choose the proper parameter n other than making it small to capture only the local dynamic of the conversation around the post i.

B. Time-based

In order to take time explicitly into account, one might set fixed time-based boundaries for the neighbourhood and include only those posts whose timestamp t_j is at distance less than τ from the ego post i, $|t_j - t_i| < \tau$. However, the pace at which posts are added to the conversation may be very different between conversation threads (and also within a thread) and we have no *a priori* criteria for a proper choice of τ.

Rather than looking for a fixed time radius, we may instead decide it by looking at changes in the pace how new posts are added to the thread. In statistical analysis, a *changepoint* in a sequence $x_1, ...x_n$ is a point that comes from a different probability distribution than its precedent values.

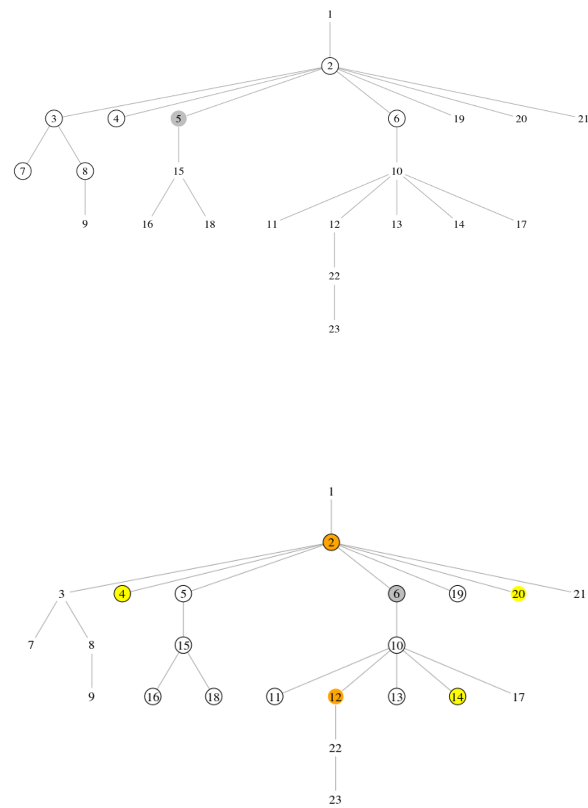

Fig. 1. Illustration of order-based (top) and time-based (bottom) neighbourhoods. Grey nodes represent the ego post. Nodes with circles are those in the neighbourhood. Numbers indicate the order of the post. The order-based neighbourhood has parameters $r = 3$, $n = 6$. The time-based neighbourhood has parameter $r = 3$. In the 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.

If sequences are timestamps, they are monotonic increasing, therefore the changepoints will correspond to sudden pauses in the conversation. In the following, we will differentiate *horizontal changepoints*, that arise between siblings, from *vertical changepoints*, that arise within a branch. We say that a sequence posts with timestamps $t_1, ..., t_n$ belong to the same (vertical or horizontal) *local dynamic* if there is no changepoint t_i in the sequence such that $1 < i \leq n$. For the detection of changepoints we use the PELT algorithm [3][2]. Now we can introduce our second definition of neighbourhood.

Definition 3: Given a tree graph G, the *time-based neighbourhood* of radius r of vertex i, denoted as $\mathcal{N}_i^T(r)$, is the maximal subgraph of the structural neighbourhood $\mathcal{N}_i(r)$ where all the vertices belong to the same vertical and hori-

[2]An implementation of this algorithm written by the authors themselves is available in the R library `changepoint`. We use their `cpt.meanvar` function.

zontal local dynamic as i.

Note that we still have a radius parameter r to guarantee that the resulting structure remains local even if no changepoint is detected near the post i.

Algorithm 1 extracts time-based neighbourhoods of a given post according to the above definition. Since the changepoints only depend on the tree and not on the particular post we analyse, we previously detect the horizontal and vertical changepoints in the tree. In the case of multiple branches with some common posts, we consider that a common post is a vertical changepoint if it is a vertical changepoint in any of the branches. Once we have the changepoints, we can proceed with the algorithm. First, we extract the structural-neighbourhood. The time-based neighbourhood will be a subset of the later. Then, we look for horizontal and vertical changepoints, which mark the frontiers of the time-based neighbourhood. There are four possible cases:

- A *vertical changepoint in the ancestors*: If the changepoint is in the path between the post and the root, then the changepoint started the new local dynamic to which the ego post belongs. Thus, we remove the ancestors of the changepoint, but not the changepoint itself.
- A *vertical changepoint in the descendants*: If the changepoint is a descendant then it started a new different dynamic and therefore we must remove the descendants of the changepoint and the changepoint itself.
- A *horizontal changepoint either in the older siblings or in the ancestors*: If the changepoint is among the older siblings, then it started the horizontal dynamic to which the ego post belongs. Similarly, if the changepoint is among the ancestors, then every older sibling of the changepoint belongs to a previous local dynamic. In both cases, we remove the older siblings of the changepoint but not the changepoint itself.
- A *horizontal changepoint elsewhere*: In any other case, the horizontal changepoint starts a different local dynamic and therefore we remove its younger siblings and the changepoint.

Any time a vertex is removed we also remove its descendants. $\mathcal{N}_i^T(r)$ is a connected induced subgraph of $\mathcal{N}_i(r)$. If there are no changepoints in the structural neighbourhood, then $\mathcal{N}_i^T(r) = \mathcal{N}_i(r)$.

An example of time-based neighbourhood is given in Figure 1 (right).

Algorithm 1 Extraction of time-based neighbourhood

Input: Posts tree G, vertical changepoints, horizontal changepoints, ego post i, radius r
Output: $\mathcal{N}_i^T(r)$: Time-based neighbourhood of i at radius r

```
Compute structural neighbourhood N_i(r)
ancestors ← ancestors(i) in N_i(r)
older_siblings ← older_siblings(i) in N_i(r)
dump ← ∅
for bp ∈ vertical changepoints do
  if bp ∈ ancestors then
    dump ← dump ∪ ancestors(bp)
  else
    dump ← dump ∪ descendants(bp) ∪ bp
  end if
end for
for bp ∈ horizontal changepoints do
  if bp ∈ (older_siblings ∪ ancestors) then
    dump ← dump ∪ older_siblings(bp)
  else
    dump ← dump ∪ younger_siblings(bp) ∪ bp
  end if
end for
N_i^T(r) ← delete(dump_posts ∪ descendants(dump_posts))
from N_i(r)
```

V. NEIGHBOURHOOD COLOURING AND PRUNING

A. Colouring

Even if the structure contains some important information about the type of conversation, there is still some ambiguity left, and two similar neighbourhoods can represent very different types of conversation. We can easily reduce this ambiguity by assigning colours to vertices, which allows us to identify some relevant property of the post.

In particular, we assign the following colours:

- *Red*: ego post.
- *Orange*: other posts written by the author of ego. It allows to identify re-entries (when the same author participates several times in the discussion [4])
- *Yellow*: parent of ego post and other posts written by the same author. It allows to identify, for instance, debates between the ego author (red and orange) and this other author (yellow). We have observed this phenomena in our data, and it is often the cause of long chains.
- *White*: root post. Differentiating the root post from the rest has been proven by previous research on online discussions. For instance, some types of users seem to get more replies than others when they initiate a thread ([5], [6]). Also, in terms of preferential attachment, root posts usually get more replies than the non-roots ([7], [8]).
- *Black*: none of the above.

Since some posts might correspond to several colours, we perform a sequential assignment of colours given by `black, orange, red, yellow, white`.

We do not claim this choice of labels to be of universal. Indeed, other labellings might also give interesting results since they would look at conversations from new points of view (e.g.: a colour for leaf vertices, colours according to the post length, or colours to represent the sentiment of the post).

B. Pruning

At this point, we can still find structures the only difference of which is that one of them has more leafs hanging from some of the nodes. Thus, we prune every neighbourhood by leaving a maximum of two consecutive siblings of the same colour and where neither of them has any children. The reason of setting the limit to two is that it is the minimum necessary to distinguish between *zero*, *one* and *more than one* consecutive occurrences. In other words, we consider that the difference between one and zero replies to a post is relevant, but that five and six replies do not make any difference worth being represented. Moreover, this choice encourage smaller structures, which allows the computation of isomorphisms (Section VI-A) even in real time.

VI. Application to Reddit forums

Our data consists of two different forums of Reddit, namely the Podemos dataset [3] and the Game of Thrones dataset[4]. 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 on October 18th and 19th, 2014. Nowadays, its members use it mainly to share and discuss about political news. The Game of Thrones forum is a casual discussion forum about the Game of Thrones TV series. The Podemos dataset contains 75,000 posts corresponding to 4,262 threads written by 9,160 users between April 25th and October 20th, 2014. The Game of Thrones dataset contains 75,000 posts corresponding to 5,862 threads written by 18,907 users.

A. Neighbourhood extraction

As a previous step for all further analysis, we extracted the structure-based ($r = 2$), the order-based ($r = 2, n = 4$) and the time-based ($r = 2$) neighbourhood around every post for the two forums.

Given a type of neighbourhood (structure-based, order-based or time-based) we extract the neighbourhood around every post of a forum as follows. A new label is assigned to every neighbourhood structure the first time it is detected. For every post, we extract its neighbourhood, colour and prune it (Sections III and IV). Then we check whether it is isomorphic to some of the n neighbourhoods that we have already seen. If this is the case, then the current neighbourhood is given the same label than the neighbourhood that we already saw. Otherwise we create a new label for the current neighbourhood.

[3]https://www.reddit.com/r/podemos

[4]https://www.reddit.com/r/gameofthrones

The set of all neighbourhoods $\mathcal{N}_i(r)$, $\mathcal{N}_i^O(r,n)$ and $\mathcal{N}_i^T(r)$ for the two forums constitutes our *dictionary*.

Once the six extractions are finished, we merge the dictionaries into a global one and sort the labels by frequency of the neighbourhood (1 for the most frequent and so forth). Figure 5 shows some frequent neighbourhoods that will be discussed later. In the worst case, where every post has a unique neighbourhood, every extracted neighbourhood should be compared against every neighbourhood that already appears in the dictionary before concluding that it needs a new entry. In this case, the computational cost of this operation is $\mathcal{O}(n^2)$ where n is the number of posts. In practice, however, the bigger the number of posts processed, the less likely we go through all the dictionary without finding an isomorphism. Moreover, if we keep the dictionary sorted by frequency, a large number of neighbourhoods will find an isomorphism among the first neighbours of the dictionary.

B. Comparing neighbourhood methods

In this section we compare different aspects of our three classes of neighbourhoods, namely (a) the size and frequency distribution of the neighbours for each class, (b) the discrepancies between methods when they extract the neighbourhood of a same post, and (c) the neighbourhood census obtained for each class in the same forum.

1) Size and frequency distribution: We detected in the the whole dataset 3,791 different structure-based neighbourhoods, 174 different order-based and 1,488 different time-based, which means that the structure-based captures a much wider set of structures. Of course, this is not necessarily an advantage since these extra structures may be just spurious and only occur a very small number of times. Indeed, Figure 2 shows that around half of the time-based neighbourhoods (41%) and the structure-based neighbourhoods (60%) occur only once, while the number of neighbourhoods that appear more than 100 times (among 150,000 posts) is similar for the three types of neighbourhood. To cover 95% of the posts we need 623 structure-based neighbourhoods, 40 order-based and 155 time-based.

The main reason why the frequency of the order-based neighbourhoods decays faster is that the most frequent neighbourhood (id 1) is much more frequent (upper-left red point in the left plot of figure 2) than the most frequent time-based neighbourhood. This is the main handicap of the order-based neighbourhood since, as we will see later, it avoids finding richer structures.

2) Discrepancies: The divergent number of neighbourhoods between structure-based, order-based and times-based suggests that some neighbourhoods, for instance, time-based, might be systematically mapped to the same neighbourhoods in order-based. Or similarly, that some structure-based neighbourhoods are systematically mapped into the same time-based neighbourhoods. Figure 3 gives a general overview of these mappings. The left figure shows, for every post, its time-based neighbourhood and its order-based neighbourhood,

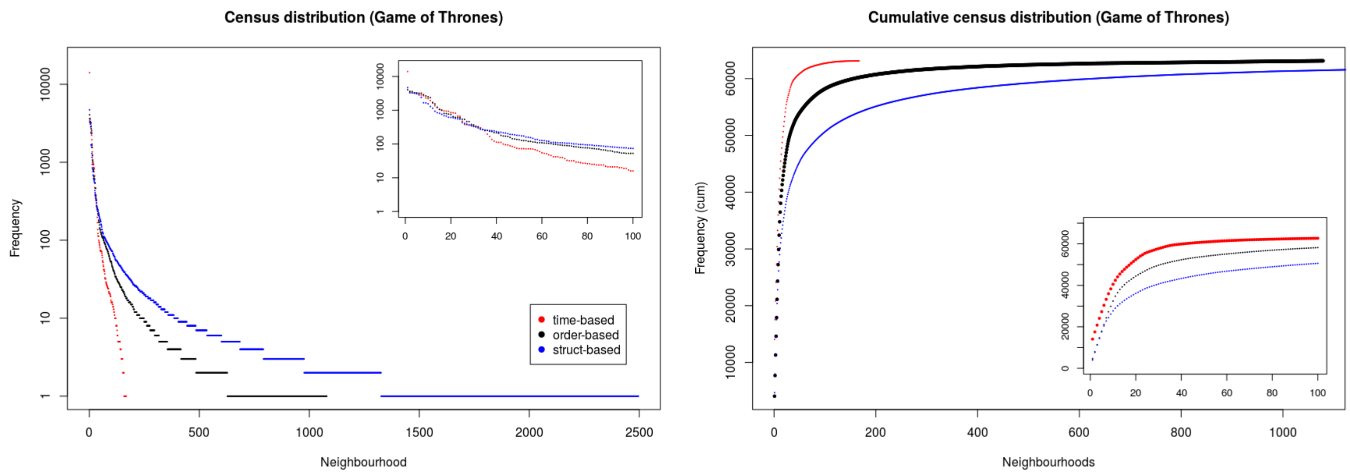

Fig. 2. Comparison of census distributions with order-based and time-based neighbourhoods.

struct-based		order-based		time-based	
POD	GOT	POD	GOT	POD	GOT
7 (9.6)	7 (7.4)	1 (25.2)	1 (25.2)	4 (9.0)	4 (6.4)
3 (16.5)	9 (12.6)	3 (32.1)	3 (32.1)	8 (17.2)	8 (12.2)
2 (23.1)	3 (17.8)	2 (38.8)	2 (38.8)	2 (25.3)	2 (17.9)
20 (27.8)	11 (22.9)	13 (45.3)	13 (45.3)	3 (31.2)	9 (23.2)
4 (31.1)	5 (27.8)	4 (50.1)	4 (50.1)	10 (36.6)	14(28.4)
13 (34.1)	2 (32.3)	18 (54.7)	18 (54.7)	14 (41.8)	11(33.5)
6 (36.9)	12 (36.1)	6 (58.9)	6 (58.9)	6 (45.8)	10(38.5)
5 (39.4)	27 (38.8)	12 (65.5)	10 (62.8)	15 (48.9)	5 (43.1)
15 (41.9)	4 (41.4)	16 (68.5)	12 (65.5)	19 (51.5)	6 (47.4)
12 (43.7)	6 (43.9)	5 (70.6)	16 (68.1)	5 (53.9)	3 (51.3)
21 (45.3)	21 (45.9)	17 (72.8)	5 (70.7)	7 (55.9)	25(55.1)
30 (46.9)	15 (47.7)	23 (75.5)	17 (72.8)	16 (57.6)	15(57.8)
16 (48.4)	13 (49.2)	31 (76.1)	23 (74.5)	36 (59.4)	12(60.2)
39 (49.8)	20 (50.6)	19 (77.5)	31 (76.1)	25 (61.2)	7 (62.2)
44 (51.2)	30 (51.9)	22 (79.0)	19 (77.5)	21 (62.6)	21(63.9)
8 (52.4)	22 (53.1)	24 (80.3)	22 (78.9)	34 (63.9)	19(65.5)
22 (53.3)	43 (54.2)	26 (81.5)	24 (80.3)	38 (65.1)	17(66.7)
51 (54.2)	8 (55.2)	29 (82.7)	26 (81.5)	23 (66.4)	36(68.0)

TABLE I
FREQUENCE THIS IS AN EXAMPLE

while the right figure shows the time-based and the structure-based. The points in the diagonal correspond to posts that have been assigned the same neighbourhood by the two compared methods.

The diagonal of the time-based / order-based plot contains 36% of the posts. The diagonal of the structure-based / time-based plot contains 40% of the posts. For the order-based / struct-based (not shown) there are 41% of the posts in the diagonal. In general, the structures with a bigger agreement are among the most frequent. This is comprehensible since the most frequent tend to be also the most simple structures with less room for discrepancy. There is a 30.75% of the posts that are assigned the same neighbourhood structure by the three methods.

Rather than looking at where the methods agree, it is even more interesting to see where they disagree, since this will reveal whether one of the methods is incapable of detecting some interesting conversational structure. In Figure 3 some horizontals lines can be spotted specially in the lower part of the plots. These correspond to posts with different time-based neighbourhoods that are assigned the same order-based neighbourhood (left), and different time-based neighbourhoods that are assigned the same order-based neighbourhood (right). Looking for the order-based neighbourhoods that have a correspondence with a larger set of time-based neighbourhoods, we detect that these correspond to neighbourhoods 1 (77 different time-based) and 50 (57 different time-based), 17 and 18. Looking for the time-based neighbourhoods that have a correspondence with a larger set of structure-based neighbourhoods, we detect that these are neighbourhoods 10, 4, 8, 19, 14.

In retrospective, these results are not surprising. What we see is simply that structures that are often captured by one method collapse into another single structure when another method with a smaller space of structures is used. Yet, the three methods mostly agree when they find a frequent neighbourhood. But this might be due to the fact that simpler motifs tend to be in shorter threads, where there is no room for two neighbourhood methods to detect a different structure.

It is clear from the last sections that the order-based does not do a good job on capturing all the richness of the conversational structures. There are two other reasons to choose the time-neighbourhood over the structure based. First, the time-based has less spurious structures, covering 95% of posts with 155 neighbourhoods, against the 623 neighbours necessary to cover that amount with the structure-based. Second, the time-based neighbourhood has more grounded criteria to decide where the neighbourhood ends, while the structure-based has no criteria at all.

C. Conversation-based clustering of users

In this section, we cluster users in the Podemos dataset based on the neighbourhoods of their posts. Our intuition is that some users like participating in some kind of discussion

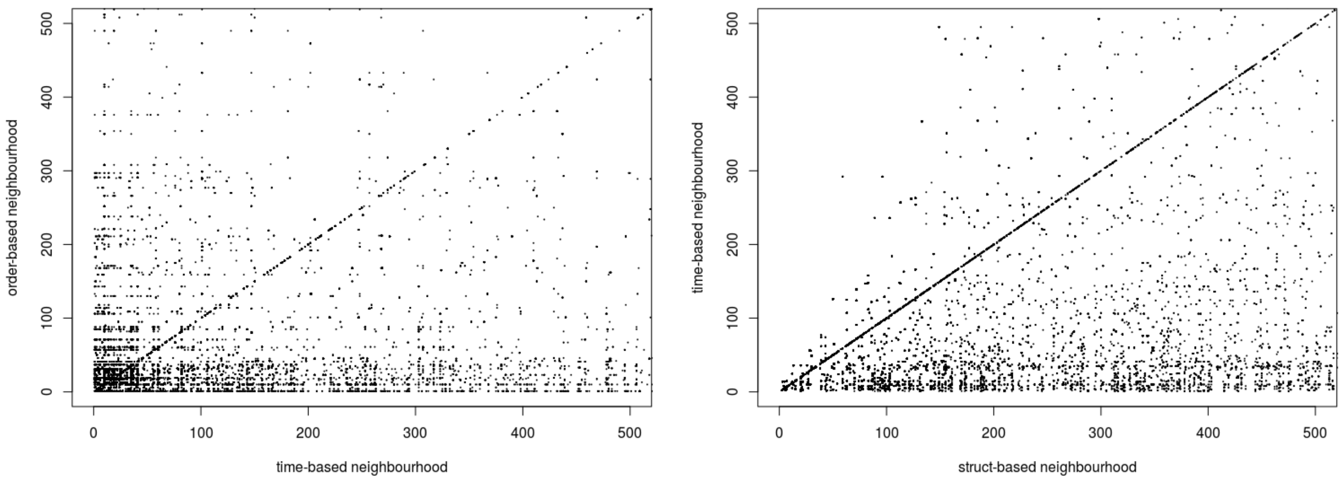

Fig. 3. Neighbourhood of posts from the point of view of order-based and time-based. Neighbourhoods with the same label in the order-based and the time-based axes are isomorphic. The points in the diagonal correspond to posts to whom the two methods assign exactly the same neighbourhood. Horizontal rows of points correspond to posts that are assigned many different time-based neighbourhoods but the same single order-based neighbourhood.

rather than other. Certainly, most part of the information necessary to understand the nature of a discussion in in its textual content. 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 context, we turn our attention towards the structure of the discussions, which can also contain some information. We work under the hypothesis that the structural-neighbourhoods in which a user post in embedded reflect the kind of conversation in that part of the thread.

Our dataset contains 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 dictionary, 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.

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 $k = 1, ...20$ 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.

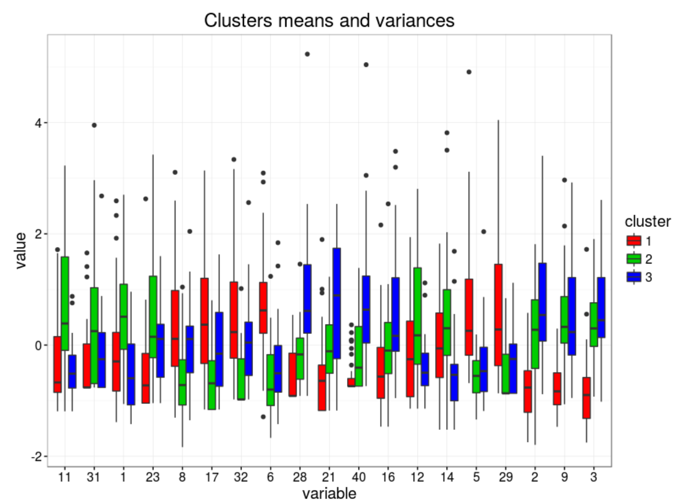

Fig. 4. Whiskers plots in the most relevant features (time-based)

Since the set of feature dimension is relatively large, we show in Figure 4 how the users in a cluster are distributed for each feature. The features shown are the ones that have an bigger absolute value (in mean) for each cluster.

- The green cluster (attraction for X, X, repulsion XXX) correspond to users that...
- The red cluster...
- The blue cluster...

VII. Conclusions

The goal of is paper was to shed some light on what would be a proper method to characterise the discussion in which a post is embedded. For that, we proposed three different kinds of neighbourhood applicable to tree structures by which conversation threads are represented. We named this three types of neighbourhood structure-based, order-based and time-based. While the first one is a mere extension of the classical concept of neighbourhood in a graph, the aim of the other two is to explicitly consider time when deciding which are the neighbours of a given post.

We showed the limitations of the order-based to capture the structure of conversations. We think this finding is relevant because, in some contexts, order is more relevant than real time, and assuming that posts (or whatever vertices represent) arrive in homogeneous intervals gives excellent results [8].

We also showed that the main contribution of the time-based neighbourhood with respect to the most basic structure-based is that, thanks to the use of changepoints to set limits of the neighbourhoods, the space of neighbourhoods is considerably reduced and with less occasional neighbourhoods that occur only once. Besides, it prevents neighbourhoods from being too big, even though some post-pruning is still necessary.

We analysed the time ....

the it is common in the literature to use the order rather than the real time. The

We introduced three different a method to characterise conversations of users in online threads. Due to the tree nature of online threads, traditional patters 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.

VIII. Future work

In this paper we focused on the comparative analysis between three kind of neighbourhoods in order to demonstrate that the time-based neighbourhoods are able to capture relevant structures that other definitions of neighbourhood cannot. We also showed that it can be used to characterise users according to the kind of conversation in which they participate.

We can see many applications of this kind of neighbourhood.

- Caracterise reactions to user posts: instead of capturing the neighbours all around a post, we can limit we neighbourhood to the descendants. This would allow to cluster users from a slightly (but maybe important) point of view.
- The boundaries of our time-based neighbourhood are decided by the changepoint algorithm PELT. How sensible is time-based neighbourhood to the choice of the algorithm?
- Merge more neighbourhoods with sociological criteria.
- Use different coulours to represent different phenomena.

References

[1] A. McCallum, X. Wang, and A. Corrada-Emmanuel, “Topic and role discovery in social networks with experiments on enron and academic email,” *Journal of Artificial Intelligence Research*, vol. 30, no. 1, pp. 249–272, 2007. [Online]. Available: http://www.aaai.org/Papers/JAIR/Vol30/JAIR-3007.pdf

[2] R. Bhatt and K. Barman, “Global Dynamics of Online Group Conversations,” in *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[3] R. Killick, P. Fearnhead, and I. Eckely, “Optimal detection of changepoints with a linear computational cost,” *Journal of the American Statistical Association*, pp. 1590–1598, 2012.

[4] L. Backstrom, J. Kleinberg, L. Lee, and C. Danescu-Niculescu-Mizil, “Characterizing and Curating Conversation Threads: Expansion, Focus, Volume, Re-entry,” in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, 2013, pp. 13–22.

[5] I. Himelboim, E. Gleave, and M. Smith, “Discussion catalysts in online political discussions: Content importers and conversation starters,” *Journal of Computer-Mediated Communication*, vol. 14, no. 4, pp. 771–789, jul 2009.

[6] A. Lumbreras, J. Lanagan, J. Velcin, and B. Jouve, “Analyse des rôles dans les communautés virtuelles : définitions et premières expérimentations sur IMDb,” in *Modèles et Analyses Réseau : Approches Mathématiques et Informatiques (MARAMI)*, 2013, pp. 1–12.

[7] V. Gómez, H. J. Kappen, and A. Kaltenbrunner, “Modeling the structure and evolution of discussion cascades,” in *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, 2010, pp. 181–190.

[8] V. Gómez, H. J. Kappen, N. Litvak, and A. Kaltenbrunner, “A likelihood-based framework for the analysis of discussion threads,” *World Wide Web*, p. 31, apr 2012.

[9] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, “Knowledge sharing and yahoo answers,” in *Proceeding of the 17th international conference on World Wide Web - WWW '08*. New York, New York, USA: ACM Press, 2008, p. 665. [Online]. Available: http://dl.acm.org/citation.cfm?id=1367587 http://portal.acm.org/citation.cfm?doid=1367497.1367587

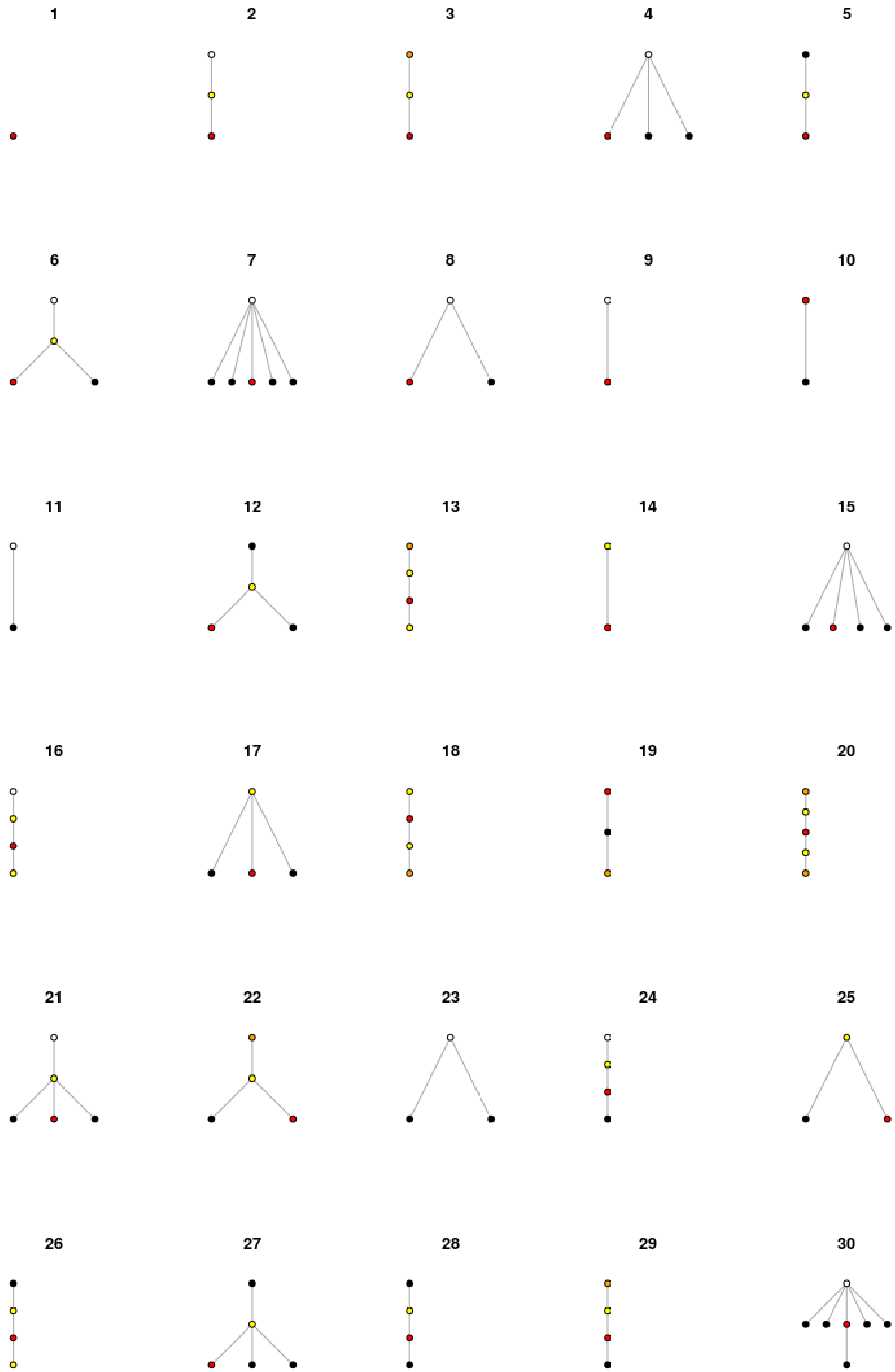

Fig. 5. Neighbourhoods reference list