Large Scale Homophily Analysis in Twitter Using a Twixonomy

Stefano Faralli, Giovanni Stilo and Paola Velardi
Sapienza University of Rome
Dipartimento di Informatica
{faralli,stilo,velardi}@di.uniroma1.it

Abstract

In this paper we perform a large-scale homophily analysis on Twitter using a hierarchical representation of users' interests which we call a Twixonomy. In order to build a population, community, or single-user Twixonomy we first associate "topical" friends in users' friendship lists (i.e. friends representing an interest rather than a social relation between peers) with Wikipedia categories. A word-sense disambiguation algorithm is used to select the appropriate wikipage for each topical friend. Starting from the set of wikipages representing "primitive" interests, we extract all paths connecting these pages with topmost Wikipedia category nodes, and we then prune the resulting graph G efficiently so as to induce a direct acyclic graph. This graph is the Twixonomy. Then, to analyze homophily, we compare different methods to detect communities in a peer friends Twitter network, and then for each community we compute the degree of homophily on the basis of a measure of pairwise semantic similarity. We show that the Twixonomy provides a means for describing users' interests in a compact and readable way and allows for a fine-grained homophily analysis. Furthermore, we show that mid-low level categories in the Twixonomy represent the best balance between informativeness and compactness of the representation.

Introduction

In this paper we analyze the role of homophily - defined as the tendency of individuals to befriend other individuals sharing the same interest - in Twitter communities. The study of homophily in social networks has been conducted either with the objective detecting communities in friendship networks and then measuring their interest similarity, or of clustering individuals according to their interest similarity and then assessing the contribution of homophily to the formation of clusters. In the former case, users' interests can be inferred from those of the community they belong to [Zamal *et al.*, 2012] [Bhattacharya *et al.*, 2014]. In the latter case, homophily is used to predict the likelihood of a user of becoming a member of a community [Colleoni *et al.*, 2014] or of bonding with another user [Kang and Lerman, 2012; Yuan *et al.*, 2014]. Both objectives are of interest for companies and policy makers seeking to determine the potential addressees of specific campaigns and to analyze the dynamics of interest formation and sharing among peers. However, we observe that the majority of approaches to the study of homophily are based on a characterization of users' interests that incurs in at least one of the following problems, when applied to large social networks: i) variability in time due to the unstable behavior of users ii) computational complexity caused by the need to process the content of millions of messages iii) poor ability to synthesize the thousands of extracted interest labels. Our contribution is to reduce the impact of all these problems, by exploiting friendship information, which is readily available in users' profiles, is more stable than other highly dynamic features, and notably, can be represented at selectable levels of generality, by connecting authoritative friends with Wikipedia categories. We call this hierarchical representation of users' interests a "Twixonomy". The paper is organized as follows: Section 2 is dedicated to the state of the art, Section 3 briefly describes the datasets and tools used in this study, Section 4 presents our algorithm for creating the Twixonomy and Section 5 analyzes the degree of homophily of Twitter communities, detected using three different approaches. Finally, Section 6 is dedicated to concluding remarks and future work.

Related Work

A first problem in homophily analysis is to properly model the notions of "being friends" and "sharing an interest". Being friends implies the existence of a social relation among peers (e.g. family, friends, colleagues): in social networks this has commonly been modeled by mutual-follow and/or mutual-mention relations [Barbieri *et al.*, 2014; Colleoni *et al.*, 2014]. The notion of interest, instead, has been represented in literature in a variety of ways. Textual features are the most common way to model users' interests: in Zamal *et al.* [2012] textual features are extracted from tweets using SVM, while Colleoni *et al.* [2014] use both information from users' profiles and textual features. In Kim *et al.* [2010] it is shown that words extracted from the titles of Twitter lists can represent latent characteristics of the users in the respective lists. In Kapanipathi *et al.* [2014] named entities are extracted from tweets, then, Wikipedia categories, named *primitive in-*

terests, are associated to each named entity. To select a reduced number of higher-level categories, named *hierarchical interests*, spreading of activation [Anderson, 1968] is used on the Wikipedia graph, where active nodes are initially the set of primitive interests. Note that, despite their name, hierarchical interests are not hierarchically ordered. Similarly to us, Bhattacharya *et al.* [2014] try to infer users' interests at a large scale. Their system, named Who Likes What, was the first system capable of inferring Twitter users' interests at the scale of millions of users. First, the topical expertise of popular Twitter users is learned from the names and descriptions of Twitter lists in which such users actively participate. Then, the interests of the users who subscribe to at least 3 expert users are transitively inferred. By doing this, Who Likes What can infer the interests of around 30 millions users. Evaluation is performed at a much smaller scale by manually comparing extracted interests with those declared in a number of users' bios, and by using human feedback from 10 evaluators. The evaluators commented that the inferred interests, even though useful, are sometimes too general: on the other hand, given the large and unstructured nature of the extracted interests (over 36 thousand distinct topics), generating labels at the right level of granularity is not straightforward.

Using textual features extracted from users' communications seems a natural way for modeling their interests. However, this information source has several drawbacks when applied to big data, such as the set of Twitter users. First, it is computationally very demanding to process millions of tweets in real time (about 500 million tweets per day in 2014); secondly, messages are very short and often uninformative: a better approach would be to analyze the grand total of all tweets sent by each user, but this would be even more demanding; third, unless we are addressing a specific community (like e.g. the members of a political party, as in Colleoni *et al.* [2014]), the number of topics grows quickly and it is very hard to make sense of them, or even to evaluate their quality. Another drawback of textual features is their volatility over time: as shown in Pal and Counts [2011], there is no evidence that a user's tweets can be characterized by temporally stable features, the only exception being topical authorities, who are more focused in their messages. In Barbieri *et al.* [2014] the authors argue that users' interests can also be implicitly represented by the authoritative nodes they are linked to, via topical links. This information is available in user profiles and does not require additional textual processing. Furthermore, in Myers and Leskovec [2014] it is shown that a small number of users with the highest indegree (i.e., celebrities) account for over 50% of follows and unfollows variability while "common" users tend to be rather stable in their relationships. Topical friends are therefore both stable and readily accessible indicators of a user's interest. As a means for systematically analyzing homophily in large networks, however, this information is difficult to interpret and sparse, as in the case with lexical features.

In what follows, we present a method for efficiently analyzing homophily in a large Twitter network (we use the full Twitter network in 2009 and a smaller network of New York Twitter users in 2014) using a "Twixonomy", a taxonomy of interests derived by matching users' friends with Wikipedia pages. Given a mutual-follow-mutual-mention friendship network, a graph clustering algorithm, and the Twixonomy, the objective of our analysis is *to measure the semantic similarity of cluster members, in order to determine the contribution of homophily to the formation of communities*.

Data and resources

For our study we use the following resources: i) **The Twitter 2009 network:** The authors in Kwak *et al.* [2010] crawled and released the entire Twitter network as of July 2009. Since Twitter data are no longer available to researches, this remains the largest available snapshot of Twitter, with 41 million user profiles and 1.47 billion social relations. Even though things might have changed in Twitter since 2009 - the number of users has grown to 500 million - our purpose in this paper is to demonstrate the scalability of our algorithms on a very large sample of users; ii) **The Twitter 2014 NewYork network:** On June 2014 we crawled a sample of about 100 thousand New York Twitter users starting from a seed of 3800 users who twitted more than 20 times in New York[1]; iii)**Babelfy:** Babelfy [Moro *et al.*, 2014] is a graph-based word-sense disambiguation (WSD) algorithm. It is based on a loose identification of candidate meanings coupled with a densest subgraph heuristic which selects high-coherence semantic interpretations. Babelfy disambiguates all nominal and named entity mentions occurring within a text, using the BabelNet semantic network [Navigli and Ponzetto, 2012]. BabelNet is a very large multilingual knowledge base, obtained from the automatic integration of Wikipedia and WordNet. Babelfy has been shown to obtain state-of-the-art performances in standard WSD benchmarks and challenges. Both BabelNet and Babelfy are available on-line[2]; iv) **The Wikipedia Graph:** We created the Wikipedia graph from the Wikipedia dump in 2009 and 2014 (for consistency with the two Twitter population datasets). The Wikipedia graph is the basis from which we infer the Twixonomy for each of the two Twitter populations.

The Twixonomy

This Section describes the algorithm for obtaining the Twixonomy starting from a Twitter population P. First, we extract from users' profiles the set F of users, followed by at least one user in P. Note that the sets P and F are different, though possibly overlapping: for Twitter 2009, since this is the complete Twitter population, we have $F \subset P$ and for the NY population we have $|F| \gg |P|$. We generate the Twixonomy from the set F, as follows (refer also to the pseudo-code shown in Algorithm 1): **a) Identify topical nodes**. For every user $u \in F$ the objective is to identify a corresponding wikipage in Wikipedia, if there is one. We define "topical users" as those u for which one such corresponding wikipage exists [3]. Obtaining a correspondence between a user screen name and a Wikipedia category e.g. @britneyspears $\rightarrow$ Britney Spears, is not trivial for a number of reasons. Firstly,

[1] the details of the geolocalization algorithm are omitted for the sake of space and because they are outside the scope of the paper

[2] http://babelnet.org/

[3] This definition differs slightly from that in Barbieri *et al.* [2014], however it seems equally intuitive

Algorithm 1 Build Twixonomy.

Input: F = twitter users followed by at least one member of the initial Twitter population P
CG: top category hierarchy from Wikipedia.
Output: a DAG taxonomy where: i) leaf nodes are wikipages associated to Twitter "topical" users, and the remaining nodes are Wikipedia categories; ii) edges are one of three kinds: <super-category , category>, <category , wikipage>, <wikipage , Twitter "topical" user>.

```
1: G = empty directed graph
2: for each twitter u:F do
3:    u.senses = ∅;
4:    u.profile = Twitter.getProfile(u);
5:    senses = BabelNetSenses(u.profile.name);
6:    if |senses|==1 then
7:       u.senses = senses
8:    else
9:       target = u.profile.name;
10:      context = {
            u.profile.name,
            u.profile.statusline,
            u.profile.location
         };
11:      u.senses = Babelfy.getSenses(target,context);
12:   end if
13:   for each sense ∈ u.senses do
14:      G.addEdge( sense , u.profile.screenName );
15:      for each edge ∈ path(sense,CG) do
16:         G.addEdge(edge);
17:      end for
18:   end for
19: end for
20: TWIXONOMY= removeCycles(G);
21: return TWIXONOMY;
```

Algorithm 2 Remove Cycles.

Input: a directed GRAPH G
Output: a DIRECTED ACYCLIC GRAPH (DAG)

```
1: while (VC = detectCycle(G)) <> ∅ do
2:    G' = G[VC] (vertex-induced subgraph of G)
3:    cyc = getOneCycle(G')
4:    break the cycle cyc on G;
5: end while
6: return G
```

Twitter names do not directly correspond to Wikipedia page names, and secondly, many pages can be associated to a named entity, for example: Britney (person), Britney (album), Britney (Busted song), Britney ("For the Record" documentary), etc. We perform joint name resolution and disambiguation (in case of multiple corresponding nodes) using Babelfy [Moro *et al.*, 2014], which disambiguates a textual input against BabelNet senses[4]. For any user and screen-name, e.g. @britneyspears, we first retrieve from the corresponding Twitter profile the fields $name$, $line - status$, and $location$, e.g. "Britney Spears", "Its Britney ...", "Los Angeles, CA". Then, we retrieve all BabelNet senses associated to the $name$ field (lines 4-5 of Algorithm 1) and, if there are multiple senses, we submit to Babelfy the sentence generated by concatenating these strings, e.g. "Britney Spears It's Britney ... Los Angeles, CA". Finally, we retrieve the disambiguated sense(s) that Babelfy has associated to the string $name$. These steps are shown in lines 5-12 of Algorithm 1. With reference to our previous example, the sense $Britney\ (person)$ is returned. Note that in many cases there are no senses corresponding to a Twitter $name$ field, since most users in F are common users (as to be expected). In some cases, however, a match exists but is missed, e.g @pinballwizard (i.e. pinball wizard, whose $name$ field is again the non splitted $pinballwizard$). To increase the recall we use a name splitting heuristics when no BabelNet senses are retrieved from the $name$ field (this step is omitted in Algorithm 1 for the sake of brevity); **b) Build the Twixonomy**. Let's denote with T the set of wikipages associated with topical users in F: these represent the "leaf nodes" [5] of the Twixonomy. Note that, after disambiguation, there is one leaf node (i.e. a wikipage) for each topical user in T. Furthermore, every node $t \in T$ is associated with the number of users in P who follow t. We then consider in the Wikipedia graph all the nodes that can be reached starting from any $t \in T$ and traversing the graph up to one of the 22 Wikipedia top categories[6], i.e. Art, Agriculture, Concepts, etc (these steps are shown in lines 13-17 of Algorithm 1). The resulting graph G, even when starting from a relatively small population P (like the NY-Twitter 2014), is still very large (since T can be quite large), and furthermore has a high number of cycles[7], e.g. Economics lists $\rightarrow$ Business lists $\rightarrow$ Economics lists. To obtain a DAG (directed acyclic graph), i.e. our final Twixonomy, we need to remove cycles. There are several algorithms for identifying simple cycles in graphs, such as those described in Tiernan [1970] Tarjan [1972], Johnson [1975] J.L.Szwarcfiter and P.E.Lauer [1974]. In practice however, all these algorithms have a high administrative cost in terms of time and memory, therefore we define an optimized iterative algorithm. Our procedure to remove cycles is summarized in Algorithm 2. In line 1, the $detectCycle$ procedure is iteratively applied on a graph G. This procedure, based on Kahn's topological ordering algorithm [Kahn, 1962], returns the set of nodes VC in G belonging to at least one cycle. This is obtained by ordering the nodes of the directed graph G and identifying cases for which topological ordering is not possible because there is a cycle. This step has a complexity of $O(V+E)$ [Kahn, 1962]. Then (line 2) we consider the vertex-induced subgraph G' of the set VC, and we apply the $getOneCycle$ procedure. This procedure, again based on topological ordering, returns the first encountered cycle in G', which is subsequently broken in G (lines 3-4). Steps 1-4 are iterated on the reduced graph, until no more cycles are found. Overall, the worst case complexity is $O((V+E)*C)$, where C is the number of cycles in G. Even though the worst case complexity of Algorithm 2 is the same as for Johnson's algorithm [Johnson, 1975], an op-

[4] BabelNet senses are mapped to Wikipedia pages

[5] hereafter we define these nodes interchangeably as as topical nodes, leaf nodes, primitive interests or wikipages

[6] http://en.wikipedia.org/wiki/Category:Main_topic_classifications

[7] http://en.wikipedia.org/wiki/Wikipedia:Dump_reports/Category_cycles

Table 1: Network statistics.

	Twitter 2009	NY-Twitter 2014
#users (P)	40,171,624	101,362
#topical users (T)	1,787,909	736,929
% of users described by at least one topic	66%	**99%**
Average ambiguity of topical users before disambiguation	5.27	5.33

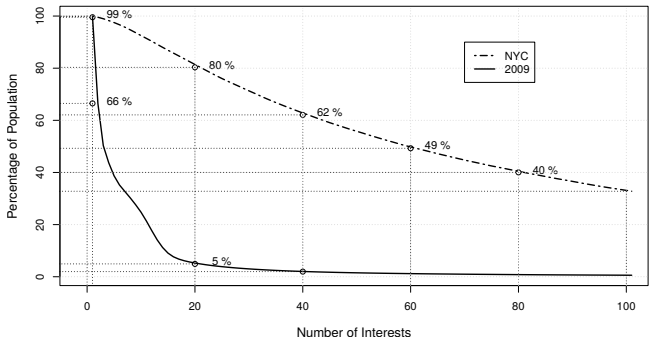

Figure 1: Coverage as a function of the number of detected topics per user (Twitter 2009 and NY-Twitter 2014).

timized use of computational resources derives from the fact that in general $G' \ll G$, and that topological ordering has reduced memory requirements compared to "classical" cycle detection algorithms. In practice, on the very large Wikipedia graph obtained when starting from the Twitter 2009 population, the algorithm was able to remove all cycles in 12 hours, while all the previously cited cycle detection algorithms either saturated the memory or could not return a solution after six days when using a mid-high level desktop computer. Table 1 shows some network statistics. In the Twitter 2009 dataset, we identified 1,8 million topical users and in the NY-Twitter 2014 dataset over 700 thousand topical users, even though the initial population P is two orders of magnitude smaller than for Twitter 2009. Figure 1 shows the coverage of the Twitter 2009 and NY-Twitter 2014 populations as a function of the number of expressed interests. The two populations are rather different in the following respects: in the 2009 dataset 66% of the population P is described by at least one topic (and related categories), while e.g. 5% is described by at least 20 topics. Instead, **99%** of NY-Twitter 2014 is described by at least one topic and 80% has at least 20 topics. NewYorkers are considerably more connected compared to the "older" 2009 network, both because rapidly increasing connections is a general trend in the Twitter graph, and because this is a tendency of NY citizens[8].

Concerning coverage, Figure 1 favorably compares with the results in Bhattacharya *et al.* [2014], where the authors mention that their coverage is 77% on a network sample which also dates from 2014. In their system, however, interests are *induced* from those of expert users to whom a user is connected, rather than mentioned *explicitly* in a user's profile, therefore in principle our methodology is also more reliable. We note that to further improve coverage we could use a

[8]http://www.statista.com/statistics/322947/facebook-fans-twitter-followers-of-new-york-knicks/

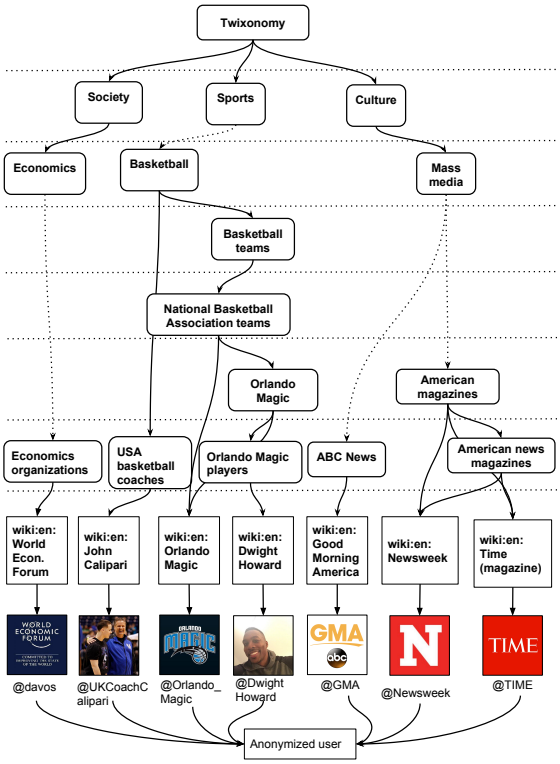

Figure 2: Example of Twixonomy for a single user.

method similar to Bhattacharya *et al.* [2014], by (cautiously) inferring additional interests for a user on the basis of his/her peer friends. On the other hand, as will be discussed later in Section 4, homophily is a significant but not pervasive phenomenon in Twitter, therefore the assumption on which Bhattacharya *et al.* [2014] base their algorithm is not fully proven. The last line in Table 1 also shows that the initial ambiguity of topical users' names was rather high (5.27 for Twitter 2009 and 5.33 for NY-Twitter 2014). Though Babelfy has been extensively evaluated in Moro *et al.* [2014], we manually evaluated a sample of 200 ambiguous user names for which a wikipage was selected by Babelfy, and 200 names for which no correspondence was found, achieving an F-measure of 0.82. To improve precision, in a similar manner to what we proposed for coverage, topical users' peer friends profiles could be used to provide Babelfy with more context [9]. Table 2 provides some Twixonomy statistics, such as the number of nodes $|V|$ and edges $|E|$ before and after removing cycles, and the max depth of the extracted Twixonomy. We can see that, even starting from very different population sizes, the two Taxonomies are of the same order of magnitude. Note that employing the same method as the one illustrated in Algorithm 1 we can build a single-user or a community Twixonomy. For example, Figure 2 shows the Twixonomy of a single "common" user with 7 topical friends in his/her friendship list. In the Figure, Wikipages are the leaf nodes of the Twixonomy, and the other nodes are Wikipedia categories layered by generality level. The Figure shows (as we

[9]this is left to future work

Table 2: Twixonomy statistics.

	Twitter 2009	NY-Twitter 2014
$\|V_G\|$ before pruning	3,146,851	1,542,924
$\|E_G\|$ before pruning	5,628,750	3,397,353
$\|V_T\|$ in pruned Twixonomy	2,195,441	1,038,205
$\|E_T\|$ in pruned Twixonomy	3,202,959	1,863,286
Max depth of Twixonomy	15	15

further discuss in Section 4) that mid-low categories are the most representative of a user's interests since, as the distance between a wikipage and a hypernym node increases, the semantic relatedness decreases. In the example the categories *Economics, Basketball* and *Mass Media* could be chosen to summarize all the user's primitive interests.

Homophily Analysis

In this Section we perform a homophily analysis of Twitter communities in order to identify the relations among users' similarity, strength of their ties and type of shared interests. To analyze homophily, we first provide a Twixonomy-based definition of users similarity, then we analyze the role of homophily in communities, by comparing the average community members similarity with that of randomly selected users. Figure 3 illustrates the advantage of using a hierarchy of users' interests to determine their pairwise similarity: let's consider two pairs of friends $p_1(a, b)$ and $p_2(c, d)$, such that each pair shares two topics out of three. The pair-wise topic similarity of p_1 and p_2 is the same, however, as shown in Figure 3, users in p_1 are more similar to each other than those in p_2, because the two non-matching topics (T^3, T^4) belong to the same 1-hop (L_1) category, while for users in p_2 the two non-matching topics (T^8, T^9) have a common category only at level L_{k-1}, therefore their semantic distance is higher. To measure pairwise semantic similarity we define the following formula, in line with Thiagarajan *et al.* [2008]:

$$(1) Sem(A,B) = \frac{\sum_{i=1}^{n_k} w\left(d_{A_i^k}\right) \times w\left(d_{B_i^k}\right)}{\sqrt{\sum_{i=1}^{n_k}\left(w(d_{A_i^k})\right)^2} \times \sqrt{\sum_{i=1}^{n_k}\left(w(d_{B_i^k})\right)^2}}$$

In the formula, A and B are the semantic vectors associated to users a and b , A_i^k, $i = 1 \dots n_k$ is the i-th boolean argument of A and is non-zero if the Twixonomy of a includes the node c_i^k of the population's Twixonomy. The index $k = 0 \dots K$ is the generalization level ($k = 0$ indicates Wikipages, as shown in Figure 2), that we also denote as L_k, and $n_k = |V_k|$ is the total number of nodes in the Twixonomy up to L_k. Furthermore, $d_{A_i^k} = k$ is the length of the minimum path connecting c_i^k with a leaf node[10]. Finally $w(d) = \beta \cdot e^{-\alpha \cdot (d+1)}$ is a weight function with exponential decay, where we empirically set $\beta = 2$ and $\alpha = 0.5$. In formula (1), non-zero terms in the numerator are those for which $A_i^k = B_i^k$, however the contribution of a match exponentially decays with the distance k of matching categories from leaf nodes. To identify communities, we first extract a mutual follow mutual mention network MFM from each of the two Twitter populations P. The hunch is, in line with previous studies [Barbieri *et al.*, 2014; Colleoni *et al.*, 2014], that if two users follow each other or mention each other, this can be considered a peer friendship relation. Then, we detect communities in MFM using three alternative community detection methods: 1) **Infomap** [Rosvall and Bergstrom, 2008], which is based on the the flow probability of Random Walks; 2) A variant of **K-core** decomposition [Seidman, 1983] [11]. This algorithm is set out to derive denser clusters than Infomap; 3) **Ego networks**: star clusters obtained from selected users with high degree.

[10] note that leaf nodes matches have $d = 0$

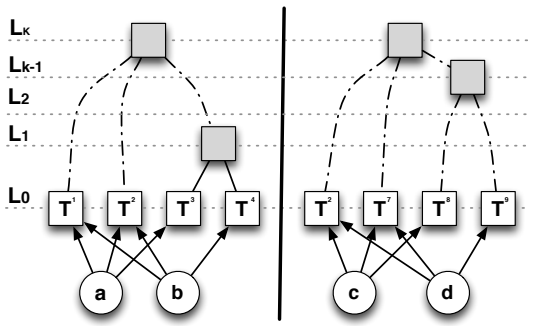

Figure 3: Exploiting the Twixonomy to measure pair-wise users similarity: the semantic distance between non-matching topics T^3, T^4 is lower than for T^8, T^9.

To evaluate homophily we compute, for each clustering algorithm, the average of the average cluster semantic similarity $avg^2(Sem(A, B))$ using, again, three methods: 1) **Clique**: For each cluster we compute the average similarity between all pairs of cluster members; 2) **Connected**: For each cluster we compute the average similarity only for those pairs sharing a link in the MFM network; 3) **Random**: To obtain a reference null model we shuffle user profiles among all clusters obtained with any of the three clustering approaches. In this way we obtain synthetic clusters that follow the same distribution of the original ones, but with random members' interests. In our analysis, we consider only clusters in which the number of members is between 50 and 1000 [12], since these represent "interesting" communities in a real-world setting. For the sake of space here we can only present a summary of our results, limited to the Twitter 2009 network[13]. Figure 4 is obtained by considering the whole set of clusters jointly extracted by the three clustering methods (Infomap, K-core and Ego), and then computing $avg^2(Sem(A, B))$ as a function of the maximum considered generality level L_k in the Twixonomy, for each of the three strategies: Clique, Connected, and Random. The Figure shows that: i) the homophily level of Clique and Connected users is significantly higher than that of Random users, and Connected members

[11] We remove from the original graph G the in-nest most core obtained from the K-core decomposition [Seidman, 1983]. Then we then re-run the procedure on the remaining graph G' as described in Valari *et al.* [2012] until $G' \neq \{\emptyset\}$.

[12] spanning around 150, the Durban's number http://en.wikipedia.org/wiki/Dunbar's_number

[13] the findings are more or less the same for NY-Twitter

are more homophilous; ii) a "saturation" effect is observed as the generality level grows above levels 6-7, a trend which is to be expected, also given what we said in Section 3 concerning upper Wikipedia categories. Figure 5 provides a comparison of the three clustering methods: we now compute separately, for each clustering method, the homophily considering only Connected users. In addition, for each generality level L_k we *only* compute in formula (1) the matches between categories at the same level k, denoting this measure as $(Sem_k(A,B))$ [14]. The Figure demonstrates that the major contribution to similarity is provided by the categories at generality levels 2-3, where homophily is up to 0.45 for K-core. Furthermore, Figures 4 and 5 together show that a higher homophily can be observed among *connected members of K-core clusters*. Finally, Figure 6 plots, for K-core Connected users, the homophily degree as a function of clusters density. It shows that, while homophily is indeed a significant phenomenon in Twitter communities, it is not pervasive: as shown by the bold trend line, cluster density is one of the parameters that positively influence the homophily degree (this was also demonstrated by the superiority of a dense-clustering method such as K-core, observed in Figure 5), though possibly not the only one. Therefore, as we already remarked, inferring a user's preferences on the basis of those of his/her friends is not a fully reliable strategy. Our last experiment analyzes the relations between homophily and semantic categories. For a selected number of highly populated mid-low level categories c in the Twixonomy, we consider all users $P_c \in P$ with an interest in c, as a proxy of a "semantic" community. We then measure the $avg(Sem(A,B))$ of Connected users within P_c. The results are summarized in Figure 7, where bars are the average similarity values of Connected members in each semantic category. The semantic similarity, in agreement with previous findings, is computed taking into account only matches up to the second level. The dashed line represents the average homophily of Connected members with $k \leq 2$, which is 0.26 as shown in Figure 4. Indeed, homophily also depends to some degree on the interests that characterize a community. For example, people interested in education (*Schoolteachers*) and *Fashion* are more homophilous, while those supporting political leaders (*Current National Leaders*) and *Women's Organizations* have a lesser tendency to befriend other users with the same interests. This is an interesting finding which, of course, requires further investigation.

Concluding remarks and future work

We described a novel method for analyzing homophily in large social networks based on a hierarchical representation of users' interests, that we called Twixonomy. A Twixonomy can be induced for single users, communities, and populations, thereby providing material for a variety of demographic analyses. We applied the Twixonomy to the study of homophily in two large Twitter populations, leading to a number of interesting findings. The advantage of our method is twofold: first, users' interests can be expressed in a compact, tunable, and readable way, as opposed to methods that derive thousands of different topics; second, we rely only on interests explicitly expressed by the users themselves, rather than on interests inferred from other connected users, which

[14]Formula (1) instead cumulates all matches from L_0 to L_k

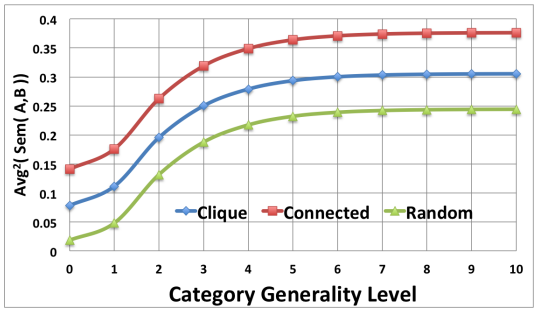

Figure 4: Average homophily in communities, using different methods to compute member similarity.

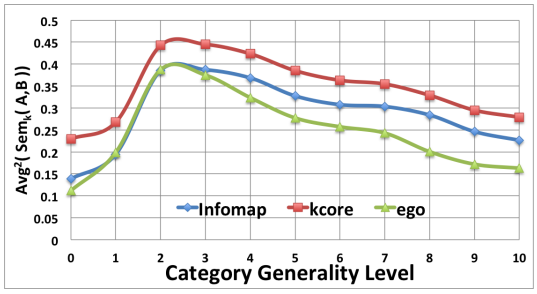

Figure 5: Homophily of "Connected" community members, using different Community Detection Algorithms.

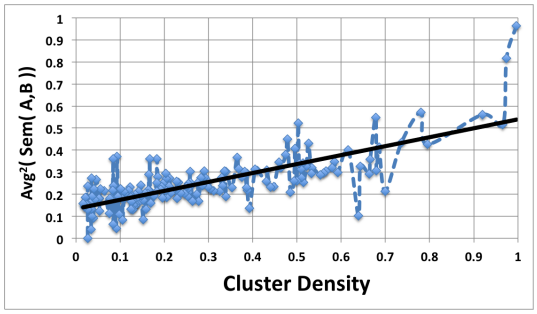

Figure 6: Homophily of "Connected" members in K-Core clusters, as a function of cluster density.

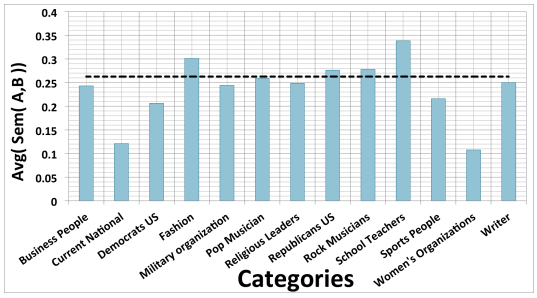

Figure 7: Homophily in selected Twixonomy Categories.

is a less reliable strategy. Our work can be extended in many ways: the quality and coverage of the Twixonomy can be further improved by exploiting the network structure both to increase precision of Twitter name sense disambiguation and coverage of users; a more systematic analysis of the best generalization level to describe users' interests can be conducted; pruning strategies to delete less meaningful Wikipedia hypernymy relations in the Twixonomy can be devised, and finally, the study of parameters (or possibly, semantic categories) that induce higher homophily can be analyzed in more detail.

References

H.E. Anderson. Fire spread and flame shape. *Fire Technology*, 4(1):51–58, 1968.

N. Barbieri, G. Manco, and F. Bonchi. Who to follow and why: Link prediction with explanations. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 2014)*, New York City, USA, 2014.

Parantapa Bhattacharya, Muhammad Bilal Zafar, Niloy Ganguly, Saptarshi Ghosh, and Krishna P. Gummadi. Inferring user interests in the twitter social network. In *Proceedings of the ACM Conference on Recommender Systems*, RecSys '14, pages 357–360, 2014.

Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of Communication*, 64:2, 2014.

J.L.Szwarcfiter and P.E.Lauer. Finding the elementary cycles of a directed graph in o(n + m) per cycle. Technical report, Univ. of Newcastle upon Tyne, Newcastle upon Tyne, England, 1974.

Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.

A. B. Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562, 1962.

Jeon Hyung Kang and Kristina Lerman. Using lists to measure homophily on twitter. Technical Report WS-12-09, AAAI Technical Report, 2012.

Pavan Kapanipathi, Prateek Jain, Chitra Venkataramani, and Amit Sheth. User interests identification on twitter using a hierarchical knowledge base. In *The Semantic Web: Trends and Challenges*, volume 8465, pages 99–113. Springer International Publishing, 2014.

Dongwoo Kim, Yohan Jo, Il-Chul Moon, and Alice Oh. Analysis of twitter lists as a potential source for discovering latent characteristics of users. In *CHI 2010 Workshop on Microblogging*, 2010.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the international conference on World wide web*, pages 591–600. ACM, 2010.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244, 2014.

Seth A. Myers and Jure Leskovec. The bursty dynamics of the twitter information network. *CoRR*, abs/1403.2732, 2014.

Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the 4th ACM conference on Web Search and Data Mining (WSDM)*, 2011.

Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160, 1972.

Rajesh Thiagarajan, Geetha Manjunath, Markus Stumptner, Rajesh Thiagarajan, Geetha Manjunath, and Markus Stumptner. M.: Computing semantic similarity using ontologies. Technical report, Hewlett-Packard, 2008.

James C. Tiernan. An efficient search algorithm to find the elementary circuits of a graph. *Communication of the ACM*, 13(12):722–726, 1970.

Elena Valari, Maria Kontaki, and ApostolosN. Papadopoulos. Discovery of top-k dense subgraphs in dynamic graph collections. In Anastasia Ailamaki and Shawn Bowers, editors, *Scientific and Statistical Database Management*, volume 7338 of *Lecture Notes in Computer Science*, pages 213–230. Springer Berlin Heidelberg, 2012.

Guangchao Yuan, Pradeep K. Murukannaiah, Zhe Zhang, and Munindar P. Singh. Exploiting sentiment homophily for link prediction. In *Proceedings of RecSys14*, 2014.

Faiyaz Al Zamal, Wendy Liu, and Ruths Derek. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2012.