Soc. Netw. Anal. Min.
DOI 10.1007/s13278-012-0074-8

ORIGINAL ARTICLE

Using dynamic community detection to identify trends in user-generated content

Rémy Cazabet · Hideaki Takeda · Masahiro Hamasaki · Frédéric Amblard

Received: 9 February 2012 / Revised: 21 May 2012 / Accepted: 28 May 2012

Abstract In this paper, we present a new solution for trend detection in user-generated content, and more particularly Web 2.0 social networks. Whereas some propositions have been published in this domain recently, we have chosen a new approach based on network analysis. We first create an evolving network of terms, which is an abstraction of the complete network, and then run a dynamic community detection algorithm on this evolving network. In order to be able to detect not only short, bursting events, but also more persistent topics, we test our solution on a social network for which we have information about all published contents for a period of more than 2 years: the Japanese network Nico Nico Douga. After presenting our solution in detail, we present the results on this dataset, notably a statistical analysis of communities' sizes and durations, examples of detected communities, and a typology of the different kinds of trends detected. Finally, we discuss the advantages and disadvantages of this method, as well as its possible applications.

R. Cazabet (✉)
IRIT, University of Toulouse, 15 Rue des Lois,
31000 Toulouse, France
e-mail: cazabet@irit.fr

H. Takeda
National Institute of Informatics (NII), 2-1-2 Hitotsubashi,
Chiyodaku, Tokyo, Japan

M. Hamasaki
National Institute of AIST, JST CREST, 1-18-13 Sotokanda,
Chiyoda-ku, Tokyo, Japan

F. Amblard
IRIT, UT1, University of Social Science, 2 rue du Doyen
Gabriel Marty, 31042 Toulouse, France

1 Introduction

With the emergence of the Web 2.0 and online social networks, user-generated contents are becoming ever more available. In thousands of different websites, millions of users post comments, videos or pictures, comment these contents, share them with different audiences, or make references to them. This tremendous quantity of information constitutes an incredible database in which nearly every kind of content can be found. However, as all this information appears on a horizontal level, all mixed with each other, finding some specific information can become a hard problem. One solution would be to sort this information automatically, to create groups of items—they might be pieces of text, videos, pictures, or any other media—related together on a semantic level. A good example is the Twitter social network. With around 300 millions registered users currently, it is a wonderful place for people around the world to post messages, from details of their everyday life to comments about the world's most important events. People realized quickly that this social network could be used as an information source, and the notion of hashtag was introduced. By using hashtags, a word preceded by a # character, users became able to easily reference their post as belonging to a specific topic. This function, invented by users to answer the problem of finding specific information in a large mass of unrelated topics, was soon fully integrated in the social network and is now used everyday by millions of Twitter's users to find messages related to specific topics. However, this method stays quite simplistic, and has some strong limitations. The lack of official tags and the need for users to voluntarily mark their posts leads do some weaknesses:

Published online: 21 June 2012

- Several distinct hashtags can be used to identify the same event. Therefore, a user searching for one of these tags will obtain as a result only part of the relevant tags
- The same tag can be used for different events. Hashtags need to be short and are frequently acronyms. Therefore, two people in different places can begin to use the same hashtag to refer to different events
- Users need to write the correct tag to each of their relevant posts. For users twitting fast, for example, when a natural disaster occurs or attending a conference, it is easy to misspell of forget the hashtag
- People wanting to post about an event, place, or celebrity have to search for the tags used by other people, with the problems listed above

In many social networks based on the idea of sharing contents, users also have the possibility to identify there contents by using tags. Usually, these tags are keywords, freely attributed by users. The problems are exactly the same as for Twitter hashtags, with several people using different tags for the same topic or, on the contrary, same keywords for different things. Therefore, it appears that relying on these raw keywords or hashtags to identify trends is not reliable enough. It is necessary to use most sophisticated tools in order to find them. Some approaches, like Bhattacharyya (2011), try to find relations between clusters using ontologies, to find words with similar linguistic roots or synonymy relationships. However, these kind of approaches are not the most efficient with real social networks, as users are likely to use names of people, events or places, and sometimes even specifically created words, for which ontologies cannot be of any help. A good method will therefore be able to find related words solely by their usage on the network.

We also think that a key aspect of trend detection in social networks is the dynamic question. To share about an event, users use some keywords. But most events only last for a period of time, so we can think that these keywords will be used only during this period. However, some events can last for a few hours, some others for a few days, months, or years. Most methods to detect trends are based on the notion of keywords' burst, but can we still detect these bursts when some events might last for several months while others will disappear after 1 day? These problems have not been addressed by most existing works on the topic.

Several methods have been proposed in the past few years to extract trends of events from social networks. In the following, we will present several of them, with their weaknesses and strengths, to be able to compare them with our proposition.

The first works related to this problem were on the detection of bursting terms. A bursting term is defined as a term which is extensively used during a limited period of time, far more than before and after this period. For example, in Kleinberg (2003), an infinite state automaton is used to identify such terms in any kind of corpus. The main strength of this method is to be able to quantify the duration of a burst, the analysis being done on a dynamic corpus, and not on consecutive snapshots. Applied on web 2.0 Social networks, such tools will successfully identify terms of interest, but, for one event or trend, numerous bursting terms will be detected.

In the work by Laniado et al. (2010), the aim is to identify important terms on a real, large social network dataset, namely Twitter. By using static analysis on 1-day windows, they were able to identify key terms, some of them presenting a real burst in usage, other ones with a more sustained usage. However, this analysis was done again on single terms, and only Twitter's hashtags. However, the good results of this work show that it is possible and meaningful to identify topics on social network's dataset, despite their noisy nature.

Benhardus et al. (2010) applied the same kind of mechanism, but using all terms in the tweets, and not only unique terms but also bigrams. However, this method does not try to cluster terms related together.

The solution proposed by Li et al. (2008) and applied on Delicious with tags associated with shared URLs allows to group together terms that do not form a digram but appear simultaneously for the same user content. For instance, if we search for bigrams, we will never consider the words "food" and "recipe" as linked, while, by searching for co-occurrences, we will consider them as linked. However, this method just keeps co-occurrences most frequent than a threshold and does not try to merge them. Therefore, it will not be able to put in the same community tags that are used exclusively, like, for example, two spellings of the same name or term, that will never be used simultaneously but belong to the same topic. Furthermore, as pointed by the authors, this mechanism results in a very large number of identified topics (148,000 for a dataset smaller than the one studied in this paper).

Weng et al. (2011) used wavelet transformation is used to detect bursts for single terms. Then, the authors proposed to use similarity of burst patterns to assign a similarity to pairs of terms. In a second step, they create a network in which nodes represent terms and links a similarity in the burst patterns of these terms. A classical community detection on networks algorithm is then used to extract clusters of similar terms. This solution has the advantage, compared with the previous ones, of being able to detect communities of terms including terms that do not necessarily appear simultaneously. However, we can still notice some weaknesses: the community detection algorithm used is a static one; therefore, the detection is run

only on 1-day windows. Therefore, there is no continuity in the communities detected (on their Twitter analysis, no event is detected more than 1 day). We can also notice that the communities contain very few terms (2–3), which is sometimes not enough to understand the meaning of the community. Finally, the method used for clustering does not allow overlap of terms, which can be a problem. For example, they detect a community with terms "Vuvuzela" and "Soccer" (the analysis was made during the soccer world cup), which means that this day, no other community of terms could be detected with the term "soccer".

Two other methods (Sankaranayan et al. 2009; Becker et al. 2011) propose a quite different approach, with the same objectives. The idea here is not to find trends composed of terms, but trends composed of user contents. For instance on Twitter, they will try to aggregate tweets that seem similar, and deduce afterward from this group of trend what is its meaning. The main advantage of these methods is that they are designed to run on-line, which means that each new tweet is automatically assigned an existing trend or create a new one if no trend seems to be related to them. As a counterpart, as indicated by the authors of TwitterStand (2009), there is a problem of trend fragmentation. If, when a new event begins, two tweets on the same topic create two different trends, all other tweets on the same event will be added to one or the other of these clusters, resulting in a fragmentation of the trends. Similarly, a User content can be clustered one and only one time, in solely one community. Finally, TwitterStand has the strong advantage compared with most other methods of being able to detect trends of undefined length. With this method, some trends can last for only 3 days and some others far longer. However, this is implemented by an "ad hoc" method consisting in ending a trend if the time centroid of all its components is older than 3 days. This means that a trend, to continue to exist after this period, must continue to grow exponentially without ever slowing down.

In the following, we will propose a solution that uses a different approach, based on network analysis. Our aim is to provide a method which is efficient enough to study trends on-line, i.e. to detect new trends as soon as they appear on the studied media. We want to propose a general method, therefore not using features specific to a platform like hashtags, but simply co-occurrences of terms. Last but not least, we want to be able to study long-term events as well as short episodic ones, along with their potential evolution along time.

In the first part, we will describe the method in itself. Second, we will present the results on the Nico Nico Douga dataset. Finally, we will conclude by a short discussion on the advantages and disadvantages of this method, as well as its possible applications.

2 The method

Our solution consists in applying dynamic community detection on an evolving network of terms. Methods proposed until now use either a clustering of user contents, a detection of bursting terms, or a static network of terms per day.

On the contrary, what we propose is to detect trends on an abstraction of the network reflecting its evolution. Our basic components are terms extracted from user contents. We create links between these terms to represent correlation of usage between them, and therefore obtain a network of terms, representing the usage of these terms in the studied social network. By updating this network according to the modification of terms' usage on the network, we obtain a dynamic network. Then, we apply a dynamic community detection algorithm, which takes as an input a dynamic network and gives as a result a set of dynamic communities. These communities of terms will be our detected trends. One can then find back the original user contents corresponding to these trends.

2.1 Creation of the dynamic network

What we have initially is a set of events (publications of contents), occurring at known times, and therefore, ordered. In a first step, we will need to transform this set of events into a dynamic network. To do so, the idea is simple: each time a new term appears, it will become a new node of our network. The number of nodes will therefore evolve very slowly after an initialization time during which we will learn all usual words used on the social network. Edges will be created between nodes if these nodes appear frequently together. On a static network, we can consider, for instance, the number of co-occurrences, and consider as linked only nodes for which a number of co-occurrences exceeds a given threshold. In our case, we will consider that a link exists between two words as long as these two words appear together frequently enough during a given period. We define a period P and a frequency F. If a given co-occurrence occurs F times or more in a period of time P, we know that there must be a link between the concerned items during this period. As we are working on an evolving dataset, and that we want to be able to do on-line trend detection, we cannot know, at a given time t, if the two terms will continue to have more than F co-occurrences in the period of time to come. What we do is therefore to consider that a link exists between two terms at a time t if there was at least F co-occurrences between them in the last period of size P. This backward mechanism implies a latency between the time the words begin to appear frequently together and the time a link is created between them. However, this latency depends mostly on F, and,

when a new important trend appears on a social network, terms involved in this trend usually have a frequency so high compared with usual ones that the delay will likely be short.

For this study, we have chosen $F = 100$ and $P = 30$, which means that we need at least 100 co-occurrences of two terms in the last 30 days to create a link between them. We could change these parameters to focus on better-defined communities or, on the contrary, broader ones. Of course, these values are adapted to the studied network, Nico Nico Douga, and should be very different on a network with more publications like Twitter. These values might also be updated if the number of videos published increase strongly during the studied period.

2.2 Using dynamic community detection

Community detection on graphs is a well-known problem with lots of applications. With the growth of interest for large real-world networks during the past few years, it became even more attractive, and dozens of new solutions were proposed. We can cite among the most known the methods proposed by Girvan and Newman (2002), Palla et al. (2005) and Infomap (2007). These algorithms have been used to detect communities on networks of terms (Capocci et al. 2004), but only on static networks. Even more recently, some algorithms have been proposed to detect communities on dynamic networks, like the methods proposed by Aynaud (2010), Mucha (2010) Palla (2007), and Cazabet (2011).

The two first ones use a same idea, which is to create, from a succession of snapshots of the network, a single network, where similar nodes in different snapshots are linked to each other. On these networks, usual algorithms based on modularity optimization can be used, with a generalized version of the modularity. However, these two methods will not be appropriate for our goal, for two reasons. First, they are not adapted to study networks with many step of evolution. In the network studied in this paper, we will have more than 10,000 nodes and more than 50,000 steps of evolution. Therefore, an aggregation in a single network would represent a network of 500 million nodes, which is a very large network. Handling such a network is a challenge in itself, and node of the two techniques are able to run on a graph of this size, on computers we have access too. We also have to remind that, as all snapshots are aggregated in a single large network, it is not possible to simply parallelize the computation by running algorithms on each snapshot independently. Even if we do not consider all steps of evolution, but only one step per day (which implies a latency in the detection of very commented events like, recently, the Japanese earthquake or the death of Steve Jobs), we still obtain, on a 2-year analysis period, a dense network of 7 millions nodes, for a network which is relatively very small compared with Twitter or Flickr. Furthermore, these techniques are not adapted to run on-line, which means that at each step, we need to recompute the whole network. This means that the community detected 1 day might be quite different on another day.

The method proposed by Palla (2007) also works on a sequence of snapshots, but without creating a single graph. The basic idea is, for each step of evolution, to compute its communities with the Clique Percolation Method (CPM) (Palla et al. 2005), and then to recognize in step $n + 1$ the communities present in step n. This works because of the local nature of CPM, which is based on, first, the identification of all cliques of a given size s, followed by the aggregation in a same community of all cliques with $n - 1$ nodes in common. The evolution can then be characterized by several operations: communities can grow or contract, merge or split. Some communities are born and other ones disappear. The method could be adapted to do on-line detection; however, this method tends to be not very efficient, specially on large networks (Fortunato 2009; Navarro and Cazabet 2011) and is very costly in term of computation, and even more in term of memory usage. (at each evolution step, the algorithm will have to compute all existing cliques of a given size on the whole network).

On the contrary, iLCD (Cazabet and Amblard 2011) is an on-line algorithm, with a very low computational cost for each evolution, and was designed to run on social networks. It is based on the idea that each community is an agent on the network, which can integrate or reject nodes. Communities that become similar might also be merged. Finally, communities can be born by a set of nodes strongly linked together that do not belong to an existing community, or die if they do not contain enough nodes. The principle is that the network is not represented as a sequence of snapshots, but as a sequence of network modification. At each network modification, a local computation is made concerning the nodes and communities directly impacted by the modification, which ensure a minimal computational cost adapted to on-line analysis. As our dataset corresponds to the past evolution of a social network, we "replay" the evolution of this network, re-creating the dynamic network of terms from published videos, passing the evolutions of this network to iLCD which gives as an output the current alive communities and, at the end of the dataset, a summary of all detected communities. The whole process takes less than 15 min on an ordinary computer for the 26 months of the dataset (4 million videos, 3 million different terms).

We can note that dynamic community detection is a rather new domain, and other algorithms will certainly be proposed in the future. In the same time, some analysis

methods on these dynamic communities are proposed (Gilbert et al. 2010). If the new algorithms proposed are more efficient than the existing ones, we will be able to use them while keeping the same mechanism of producing an evolving network of terms and therefore improve the efficiency of this solution while keeping the same mechanism.

3 Results

3.1 The dataset

The network on which we decided to apply this method is a Japanese video-sharing network called Nico Nico Douga. Whereas the richest network for trend detection is probably Twitter, there are some limitations to its usage: first, it is nearly impossible to work on the whole network, both because we do not have the technical capabilities and because the API allows only to obtain a subset of the published tweets. But even more importantly, we do not know any available dataset lasting more than a few weeks. Most trend detection done on real networks until now have been done on short periods of time, and, as a consequence, only try to detect bursting, short events. We were interested in detecting both short-time trends and long-time trends, and with complete data for a period of more than 2 years, the Nico Nico Douga dataset was adapted for this purpose.

3.2 Nico Nico Douga presentation

This network offers the same possibilities than YouTube, but with some added features. First, videos can have links to other videos, for instance, if the two videos are related. This possibility is strongly used for some purposes, especially collaborative creation of videos. A study regarding these collaborations, and in particular the case of Hatsune Miku, has been published in (Hamasaki et al. 2008).

Second, and it is the feature used in this paper, videos are given tags by users. Each video on nico nico douga can be tagged by a number of tags varying from 0 to 10. These tags can be very generic ("Video-games", "Music", etc.) or more specific ("kitty", name of a baseball team, name of a video-game, etc). For our analysis, we do not take into account the video in itself, but consider only its tags. What we are interested in is to count the number of co-occurrences of terms, i.e. the number of times two terms appear in the description of the same video. It is similar to what is done on tweets, without the problems of unmeaning words and indivisible words' bigrams. The original data were extracted between February 2007 and May 2009. They concern a little more than 4 million videos, using more than 3 million different tags. However, most of them are used less than ten times. Users share videos on all kind of topics, but among the most popular, we can cite Video games and music videos.

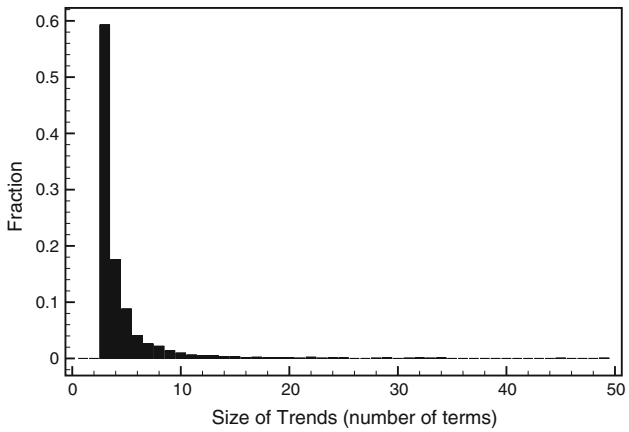

Fig. 1 Distribution of trends size

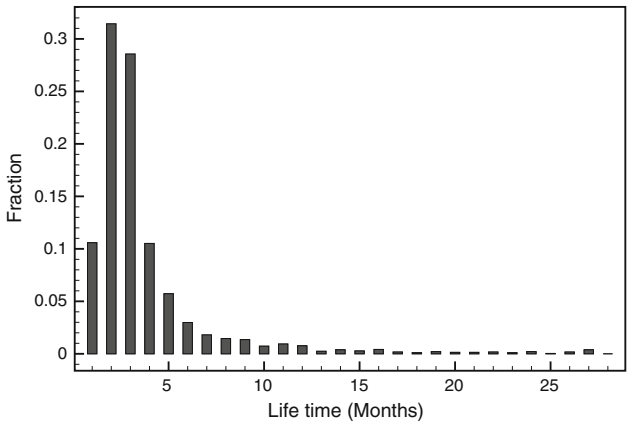

Fig. 2 Distribution of trends duration

The total number of communities detected is 2,865. However, there is a strong disparity between these communities. On a dynamic network, we have to remind that some communities can exist only for a short period of time, while others might exist from the beginning to the end of our analysis. In Figs. 1 and 2, we display the repartition of the size of the communities and of their lifetime.

3.3 Life time

We observe that most communities have a quite limited lifetime, between 2 and 3 months. It means that when a new topic appears, people usually speak about it intensively for 1–2 months, and then the interest in this topic decreases, and it disappears from the list of our communities. As this length is longer than our parameter P (30 days), we know that these short communities are not a bias due to the network creation process, but really correspond to a topic that was popular for a period longer than

30 days, and which became less popular after this period. If we had chosen another value for the parameter F, these communities could last a little longer, but the fewer the value of F, the more noise we have in our results.

There are also many communities of longer lifetime. We can see 378 communities lasting 6 months or more, and 21 communities lasting 2 years or more, while our dataset covers only 2 years and 3 months.

3.4 Size of communities

As we can see in Fig. 1, most communities are very small communities, with less than five nodes. It is not surprising, because for a given topic, users tend to use always the same words. For example, for a video game, users will tag videos corresponding to it with the name of the game, sometimes with two or three slightly different writing, a tag for the game's category, or series, and the tag meaning "video game". On the contrary, there are some communities including a quite large number of nodes. 15 communities have more than 30 nodes. However, all these nodes do not necessarily belong simultaneously to the community. One keyword can appear at one time, disappear, and be replaced later. As a good example, one community with a long life concerns the video games series Final Fantasy. This community includes the names of several episodes of the series, but not necessarily at the same time. For example, when a new game is released, it appears in the community. But after a few weeks, it will disappear, until a new game is released. As this series is very popular in Japan, many videos can be uploaded concerning a previous episode of the series at any time, and some tags therefore appear without any obvious reason.

3.5 Correlation between life time and size

A reasonable hypothesis would stand that communities living longer would have more nodes as a consequence of the phenomenon explained above. Figure 3 shows the correlation between the two values. If, indeed, communities with a short lifetime are mostly small, and if large communities tend to be communities with a long life, there is an important proportion of communities with a long life but including a few nodes. The best example of this is a community alive during the whole dataset, but including only four nodes. These nodes can be translated in "video", "nico-nico video", "nico-nico commentary", and "commentary". These very common tags are used extensively during the whole dataset, frequently together, but do not have any reason to be linked to other tags. Therefore, this community remains stable from the beginning to the end of the dataset.

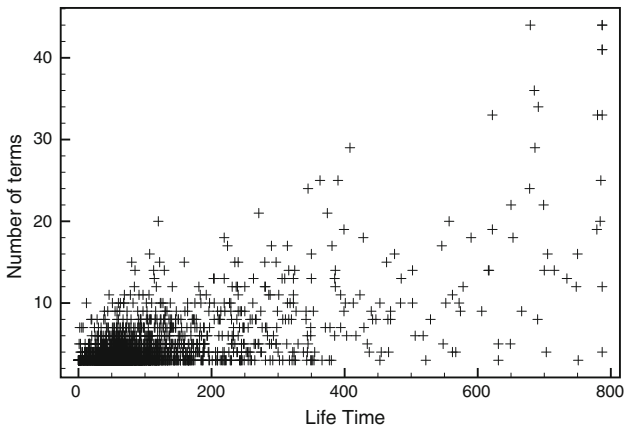

Fig. 3 Correlation between life time and size of trends

3.6 Categories of communities

As we explored the different events detected, we found that they could be classified into different categories. In this section, we describe these different categories and give example of events corresponding to them.

3.6.1 Short events

First, there is the category of short events, related to a new, unique event. Nico Nico Douga is a social network strongly used by young Japanese, who share a lot on video games, animations, and music. Consequently, most topics are related to these kinds of subjects. In Table 1, we give a few examples of these events, for games, identified by their most significant term, which is naturally the name of the game. We also give for them their date of beginning and date of end, and the day the related event—namely, release date of the game in Japan—actually occurs. These data are not exhaustive; they are only a few examples to illustrate the results.

As we can observe, most of the events are created at a date close to the release date of the game. For several of them, the event is created even before the release date of the game. When a game is about to be released, users share

Table 1 Examples of trends detected, related to video-games

Detected event	Creation date	Ending date	Release date
Devil May Cry	12/02/2007	09/08/2008	01/31/2008
Fable 2	12/06/2008	02/03/2009	12/18/2008
GearsOfWar2	10/14/2008	12/29/2008	11/07/2008
Assassin's Creed	01/25/2008	02/26/2008	01/31/2008
Soul Calibur IV	07/07/2008	11/15/2008	07/31/2008
Uncharted	11/11/2007	01/02/2008	11/16/2007

We can observe that their apparition is strongly correlated to the release date of the games

Table 2 Examples of events with long duration

Terms in the event	Creation date	Ending date
Cats, kitty, animal, Nico Nico Cats' videos	03/06/2007	–
J-League, Soccer, Sport	04/13/2007	–
Vocal, arrangement, Dojin Music	01/24/2008	–
Pro Wresling, WWE, WWF, Sport	07/18/2007	10/25/2008

Usually, these events are related to general topics

a lot about the little information they have. When the game is released, players share in-game videos of them playing some tricky part or just to share their favorite part of the game.

3.6.2 Generic topics

A second category concerns long-term, generic events. Users tend to share frequently on some topics, which are not related to a specific unique event. For instance, we gave in Table 2 some of these events, with their beginning and end dates if available.

When no ending date is provided, it means that this event is considered as still alive at the end of the dataset. We see that some of these communities of terms are alive from nearly the beginning (our dataset begin in February 2007) to the end of the dataset, and therefore that people share on these topics continuously. One of these events, WWE, however, stops before the end of the dataset. This could have different meanings, for example, a diminution of interest from users in the topic, as WWE (the most famous wrestling league) appears less frequently in the dataset after the trend ending date.

3.6.3 Repetitive events

A third category we can define is about reappearing events. Some communities are detected a first time, then disappear, come back, disappear again, and so on and so forth. We give some examples in Table 3. For some of them, this behavior is totally normal, and specially interesting, as for Christmas. For some others, this can be considered or not as a bias. We can consider that these events should represent one and only one, long-lasting event, and not several short ones. They are probably split because some of their terms are used together less frequently for a certain period; therefore, we consider that a link does not exist between them anymore, and the disappearance of this link cause the dynamic community to end. When the link reappears, the community will be created again. This can cause the community to disappear several times if its terms' appearance frequency is close to our threshold. This behavior satisfies our requirements in term of real-time event detection: we always know which are the trendiest topics. However, we lose the information about the history of the community. In order to solve this problem, one would want to try to match existing communities to previous, ended ones, or maintain knowledge about dynamic communities likely to be only in an "on ice" mode.

3.6.4 Social network usage terms

All the categories of detected trends presented above were related to a specific topic. Sometimes, it was a very narrow, precise topic (for instance, a game), sometimes a broader one (Jazz, Cats,…), but it was always something not intrinsically related to the social network itself. The same trends could be detected similarly in another social network and simply reflect the interests of users. But we also had the surprise to find some trends composed of terms that were sometimes not explicit in themselves, and we had to understand some particularities of the platform, some habits of its users to understand them. We present some of them in the following.

One of these communities exists during nearly the whole dataset.

- "Entertainment", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10": on Nico Nico Douga, there is a limit to the length of posted videos. But many users want to share longer videos, sometimes complete movies or just long personal films. When doing so, they have to post several videos, and identify them as belonging to the same long one. To do so, they use numbers to identify the number of videos in the series, and identify which number of the series it is. This community results from this behavior.
- Other ones appear several times:

Table 3 Examples of repetitive events

Topic	First apparition	Second apparition	Third apparition
Christmas	10/27/2007 to 01/14/2008	11/07/2008 to 01/19/2009	
Tour de France (annual cycling event)	06/17/2007 to 07/30/2007	06/02/2008 to 12/30/2008	
Jazz	09/18/2007 to 11/13/2007	11/26/2007 to 12/04/2008	03/21/2009 to –
Figure Skating	10/20/2007 to 09/11/2008	10/17/2008 to 01/04/2008	01/15/2008 to –

- "@" ";" "[" "]" "test": a few communities of this kind appear in the dataset in the period of November–December 2008. They all contain the term "test" and a few special characters. When checked on the network, these videos are no longer available, but were probably used for some kind of test
- "Anime", "Video-Games", "Music", "Ranking", "Ask for commentaries": This community appears from the middle of the dataset. It contains the most common terms of the social network, and the word "ranking". To understand it, we have to know that a popular behavior on Nico Nico Douga is to post a video which is a selection of the favorite videos of the user on a given topic. Some users are even famous for posting regularly good quality videos of this kind. The keyword to identify these "best of" videos is the keyword "Ranking".

3.6.5 Sub-events

If most video-games, music groups or topics such as Jazz or base-ball have only one dynamic community, and therefore one trend, some others, that we would tend to categorize as one topic, are represented by several communities with our detection process. This is usually the case for very popular subjects, which lead to a great number of videos of different categories. Two of these topics are the Hatsune-Miku/VOCALOID phenomenon and the Idol Master video game.

Hatsune miku is an interesting phenomenon of user collaboration on social networks (Hamasaki et al. 2008). VOCALOID is the name of a singing synthesizer, a software that, provided with lyrics and a melody, produces a corresponding song. Hatsune miku is the name given to an imaginary character that embodies the singer of songs produced by VOCALOID2, the first version of VOCALOID to become very popular in Japan. This software encountered a great success, and in particular on Nico Nico Douga, on which hundreds of users began to post thousands of videos, sometimes with just a new original song, sometimes by making VOCALOID sing a popular song, sometimes by adding a video-clip illustrating a popular Hatsune Miku song, and so on and so forth. In our dataset, more than 40 trends are detected with the term VOCALOID in it. The longest one begins on 08/19/2008 and continues until the end of the dataset. The beginning matches with the release of VOCALOID2 (08/31/2008). This trend contains many terms, notably the name of other VOCALOID characters, and terms such as "music", "original album", or "Miku-Video". In the same time, there are many other shorter trends containing the term VOCALOID and that are centered on specific topics: for example, one of these trend is centered on VOCAROCK (rock with VOCALOID), another one on "Miku Miku Dance" (software to realize 3D movies of Hatsune miku), and so on and so forth.

3.7 Following a trend evolution

An interesting feature of the trend detection seen as a dynamic community detection approach is the ability to track the evolution of communities, and, therefore, of trends. A dynamic community is composed of nodes, and at any time new nodes can be integrated into the community, or rejected. Therefore, in our trend, we know precisely which terms were used initially, which terms were added at which time and, sometimes, which terms stopped to be used in this context. Figure 4 shows an example of the visualization of the evolution of a trend along time, for the series of video games "Metal Gear Solid". Each horizontal bar represents a term, begins when this term is considered as integrated in the trend, and ends when the algorithm considered that the term no longer belonged to this trend. As we can see, some terms are really representative and therefore belong to the trend for the whole period, while some other words are mostly used for some periods, and therefore appear only episodically in the community. In this example, we see that the community is originally formed around the words "games", "metal gear", and "MGS", an abbreviation of the name of the game. In early 2008, "MGS3" was added. There is no obvious reason for the apparition of this term, which corresponds to a game released in 2004. A possible explanation could be the proximity of release of the next episode of the series. Some times later, the term "Disarmament" appears in the community for a short period. MGS is an infiltration game, in which disarming an opponent is a difficult but rewarding

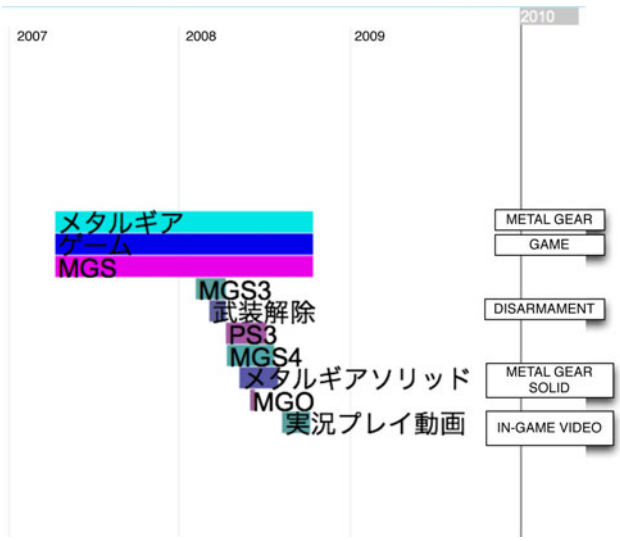

Fig. 4 Evolution of the MGS trend

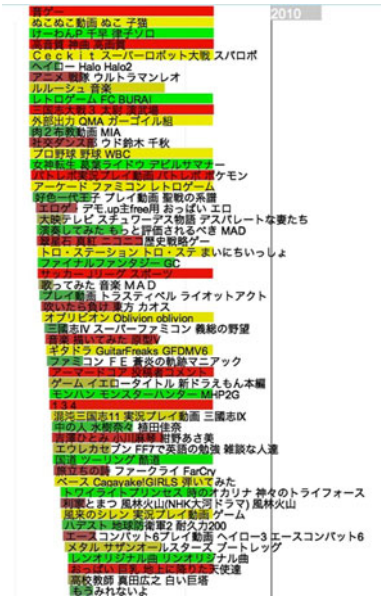

Fig. 5 Visualization of the evolution of trends with ten terms or more. On the horizontal axis, we can see the time, each *horizontal line* represents a community. We can see at a glance the apparition of communities (communities are ordered by age of birth), and the existence of communities with a short life time (*short lines*), longer ones, and some communities lasting until the end of the dataset

act, that users like to share in video. Then, in April 2008, the two terms "PS3" and "MGS4" are added simultaneously. MGS4 is the name of the new opus of the series, released in June 2008, on the video game console "PS3". Shortly after, a new term appears, which is just a new way to write the name of the video game in Japanese. The term "MGO", appearing in June 2008, is an acronym for "Metal Gear Online", the online version of the game, which was part of MGS4, released at the same month. Finally, the last term corresponds to "in-game videos" and is quite common in video-games trends. We can generate automatically this visualization for every trend. For short trends, there are usually few modifications, while, for long ones, there are usually several modifications.

We can, in the same way, visualize the duration time, creation, and extinction of trends. In Fig. 5, we show an example of what it looks like if we keep only trends including more than ten different terms. We can distinguish in a glance between trends lasting during the whole dataset, trends with a short life span, trends alive at a given time, and so on and so forth. (names of the trends are a subset of the terms of the communities).

4 Discussion

The method proposed here has several advantages over the previous methods described in the first part of the paper. First, compared with all other methods based on trends composed of terms, this method is the first one able to put one term in several trends, which is a key point to be able to differentiate between trends with related topics without merging them or ignoring one of them. Second, this method is the first one able to detect both bursting events with a short life and sustained topics of interest. Finally, this method is also the first one able to detect the evolution of a trend, characterized by the integration or the removal of terms from an existing communities. However, this solution also suffers from some weaknesses. We will first describe this weaknesses and which solutions we could use to improve them, then we will discuss the possibility to adapt this algorithm to other networks, which can be quite different like, for example, Twitter.

4.1 Possible improvements

The main one is probably due to the process used to create the dynamic network, which creates latency both in the first detection of new events and in the disappearance of events. We think that, in the future, we could adapt a method as the one proposed by Kleinberg (2003) to improve this detection. As several techniques have been proposed to detect bursting terms, we could adapt them to detect bursting co-occurrences of terms and consider this bursting co-occurrences as our active link of the evolving network of terms.

Another way to improve the solution would be to improve the communities themselves. Whereas most detected communities are indubitably meaningful, results as directly obtained by the described mechanism could be improved to furnish to users an efficient tool to browse and explore existing communities.

- A first improvement could be to be able to detect trends composed of only one or two terms. If the case of a successful trend which can be identified by so few terms is probably rare, simple mechanisms could be added to deal with this possibility.
- With about 500 active trends at the end of the dataset, it is not easy for an user to find the one he is interested in. If we can easily set up a research system (return the trends which contains the word the user is searching for), one would want to class the detected trends by categories. Some static community detection algorithms propose a hierarchical decomposition, which could be used to create these categories. Unfortunately, to our knowledge, no dynamic hierarchical community detection algorithm has been proposed so far. However, while exploring the results, we observed that some terms where belonging to many communities and that these terms were frequently generic terms. For example, the terms that appear in the higher number of

different communities are Video-games, Animation, Music, Need comments, Entertainment, Sports, In-game videos, VOCALOID, PS3, Xbox360. All these tags represent popular topics that are represented by many narrower trends. It is not certain that such phenomenon of category tags would appear similarly on other social networks, like Twitter for example, but it could be a first step for categorization. (It could be particularly efficient to solve the problem of fragmentation of trends, for example, several trends which topics are several aspects of the same game or phenomenon like VOCALOID, for which the term the more characteristic of the common topic is very likely to appears in all trends.)

4.2 Adaptation to other networks

An interesting question about this algorithm is as to we can adapt it on other networks and, in particular, to the richest of them, Twitter. Nico Nico Douga was a network easy to deal with, as what we are working on are tags. With tags, we do not have the problems of bigrams (one unique term composed of two words, like, for instance, "Los Angeles", or "Amy Winehouse"). We also do not have the problem of unmeaning terms, like "and", "the" or possession mark. So, yes, this method is directly applicable to social networks in which user contents are described by keywords or tags, like FlickR or Delicious. But to apply it to Twitter-like networks, or to citation networks (Kas et al. 2003) one would have to make a pre-treatment to users' texts in order to transform them in a set of keywords. the same problem was encountered by other methods like in TwitterStand [twitterStand], which seem to handle it with success using stoplists and other standard solutions of text mining. The remaining problem is inevitably the problem of the size of the networks. We think that the methods that try to cluster directly users' contents will have difficulties to deal with such a large quantity of data in real-time. Methods based on identification and clustering of "bursting" or "interesting" terms seem to be potentially faster. In our case, there are two steps that might cause problems: keeping the network of terms up-to-date, and running the dynamic community detection algorithm. The first point seems not very hard compared with some problems currently handled in data management and is not really specific to this method. The second point depends on the algorithm used and, as explained, the dynamic algorithm has a very low cost, as all modifications of the network only cause local computations. In the current paper, we handle more than 2 years of data with millions of keywords in a few minutes. However, with a large enough dataset, it might be possible that the method takes too long. This method, as the other ones presented before, are not really adapted to detect trends on social networks with long texts, such as blogs. It would nevertheless be possible if one could identify the most important words of the long post, and use them as equivalent to tags. Using TF/IDF scores for example, or removing common words might give some results, but specially designed methods would probably be more efficient.

5 Conclusion

We have shown that the solution of considering a social network as an evolving network of terms on which we can apply dynamic community detection is an original and efficient way to deal with the problem of trend detection in user-generated contents. We successfully identified hundreds of meaningful trends, and furthermore, we have temporal data on them, such as their creation date, their disappearance date, and the date at which new terms were added or removed from them, which was not possible with usual trend detection by clustering methods.

Whereas few dynamic community detection techniques have been proposed yet, we can assume that several ones will be available in the future, and that we will be able to use them to further improve such a tool. A next interesting step would be to adapt this solution using real-time data on a social network, for example Twitter, and to design an interface to allow users to browse the current and past topics, together with the related user contents. But trend detection is not limited to Web 2.0 social networks; we could also use it, for instance, on scientific papers to identify growing fields of science, or more generally on any kind of evolving textual corpus.

References

Aynaud T, Guillaume JL (2010) Long range community detection. In: Intelligence and Security Informatics, 2007 IEEE

Becker H, Naaman M, Gravano L (2011) Beyond trending topics: real-world event identification on Twitter. In: Fifth international AAAI conference on weblogs and social media

Benhardus J (2010) Streaming trend detection in twitter. In: National Science Foundation REU for artificial intelligence, NLP and IR

Bhattacharyya P, Garg A, Wu S (2011) Analysis of user keyword similarity in online social networks. Soc Netw Anal Min 1:143–158. doi:10.1007/s13278-010-0006-4

Capocci A, Servedio V, Caldarelli G, Colaiori F (2004) Communities detection in large networks. In: Leonardi S (ed) Algorithms and models for the web-graph. Lecture notes in computer science. vol 3243. Springer, Berlin, pp 181–187

Cazabet R, Amblard F (2011) Simulate to detect: a multi-agent system for community detection. In: IEEE/WIC/ACM international conference on web intelligence and intelligent agent

technology (WI-IAT), 2011, vol 2, pp 402–408. IEEE, New York

Fortunato S (2009) Community detection in graphs. Physics Reports 486(3–5):75–174

Gilbert F, Simonetto P, Zaidi F, Jourdan F, Bourqui R (2010) Communities and hierarchical structures in dynamic social networks: analysis and visualization. Soc Netw Anal Min 1(2):83–95

Girvan M, Newman M (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821

Hamasaki M, Takeda H, Nishimura T (2008) Network analysis of massively collaborative creation of multimedia contents: case study of hatsune miku videos on nico nico douga. In: Proceeding of the 1st international conference on designing interactive user experiences for TV and video. ACM, New York, pp 165–168

Kas M, Carley K, Carley L (2011) Trends in science networks: understanding structures and statistics of scientific networks. Soc Netw Anal Min 2(2):169–187

Kleinberg J (2003) and hierarchical structure in streams. Data Min Knowl Disc 7(4):373–397

Laniado D, Mika P (2010) Making sense of twitter. In: The Semantic Web—ISWC 2010, pp 470–485

Li X, Guo L, Zhao Y (2008) Tag-based social interest discovery. In: Proceeding of the 17th international conference on World Wide Web, pp 675–684. http://dl.acm.org/citation.cfm?id=1367589

Mucha P, Richardson T, Macon K, Porter M, Onnela J (2010) Community structure in time-dependent, multiscale, and multiplex networks. Science 328(5980):876

Navarro E, Cazabet R (2011) Détection de communautés, étude comparative sur graphes réels. Int J Interact Intell Inf 11(1):77–93

Palla G, Barabási A, Vicsek T (2007) Quantifying social group evolution. Nature 446(7136):664–667

Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):814–818

Rosvall M, Bergstrom C (2007) An information-theoretic framework for resolving community structure in complex networks. Proc Natl Acad Sci USA 104(18):7327

Sankaranarayanan J, Samet H, Teitler B, Lieberman M, Sperling J (2009) Twitterstand: news in tweets. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, pp 42–51. http://dl.acm.org/citation.cfm?id=1653781

Weng J, Yao Y, Leonardi E, Lee F (2011) Event detection in Twitter. In: Fifth international AAAI conference on weblogs and social media