Towards the Design of a Platform for Abuse Detection in OSNs using Multimedial Data Analysis

Thomas Vanhove Philip Leroux Tim Wauters Filip De Turck
Department of Information Technology, Ghent University - iMinds
Gaston Crommenlaan 8/201, 9050 Gent, Belgium
Email: thomas.vanhove@intec.ugent.be

***Abstract*—Online social networks (OSNs) are becoming increasingly popular every day. The vast amount of data created by users and their actions yields interesting opportunities, both socially and economically. Unfortunately, these online communities are prone to abuse and inappropriate behaviour such as cyber bullying. For victims, this kind of behaviour can lead to depression and other severe problems. However, due to the huge amount of users and data it is impossible to manually check all content posted on the social network. We propose a pluggable architecture with reusable components, able to quickly detect harmful content. The platform uses text-, image-, audio- and video-based analysis modules to detect inappropriate content or high risk behaviour. Domain services aggregate this data and flag user profiles if necessary. Social network moderators need only check the validity of the flagged profiles. This paper reports upon key requirements of the platform, the architectural components and important challenges.**

I. Introduction

Social networks are a direct consequence of the introduction of Web 2.0 applications on the Internet. With the focus shift to more user-driven content, the construction of online communities became possible. The popularity of social networks, such as Facebook and Twitter, has grown ever since they were created and they are still continuing to expand worldwide. In Europe for example, 38% of all people confirm to have a profile on any social network. For the category 16- to 24-year-olds, the percentage rises to a staggering 80% [1].

The data created by users and their actions yields many opportunities to learn about their online and real world interests [2]. In turn, this leads to new powerful forms of advertising, often personalized based on known interests [3]. Unfortunately, as in the real world, some users tend to misbehave and abuse online communities. Examples are identity theft, inappropriate sexual or racist behaviour and cyber bullying. For the victims, these actions lead to serious consequences like depression and in some cases suicidal tendencies [4]. Research has shown that at least 11% of all youngsters, between the age of 12 and 18, have been a victim of cyber bullying [5]. Moreover, due to the amount of users and data on a social network it is impossible to manually check all content posted on the site. A platform able to analyze the social network feed and user behaviour would be an enormous step forward. By flagging suspicious and high risk profiles, the platform would aid the social network moderators, after which they could personally check the validity of such profiles.

AMiCA [1] is a research project with the ultimate goal of developing a platform able to trace harmful content in an automatic way by mining relevant website sources, and collect, analyse, and integrate large amounts of subjective information using text, image, audio and video analysis. In this article we will formulate an answer to three main research questions: what are the requirements for such an analysis platform? Based on these requirements, how does the architecture of the platform look? Finally, what are the important implementation challenges?

The remainder of this paper is organized as follows: in Section II we review similar research for analyzing online social behaviour. Section III defines the requirements for the monitoring platform. The proposed architecture is described in Section IV and in Section V we elaborate on workflow scenarios. Section VI identifies important implementation challenges and Section VII concludes this paper.

II. Related Work

Cyber bullying is one of the most researched problems on social networks [6]. The academic world however has not yet reached consensus about the definition of cyber bullying. Smith et al. define it as an aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against a victim who cannot easily defend him or herself [7]. This definition is based on a well-accepted definition of bullying by Olweus [8] and will therefore also be used in this paper.

Next to cyber bullying, issues like inappropriate sexual behaviour are also afflicting online communities. Research has led to models for the detection of these cases [9], [10], [11]. Companies such as Mollom provide a web service using these detection modules to scan for spam and inappropriate content. These models have nonetheless never been combined into one platform for the analysis of social network feeds.

The large amount of data the platform would need to process from these feeds might trigger a need for a more intelligent internal data representation like ontologies. Ontologies describe specific domains of knowledge through individuals, their categorization and relationships [12]. In addition to defining a domain into a machine-readable format, they also allow reasoning over the saved information based on the

[1] http://www.amicaproject.be

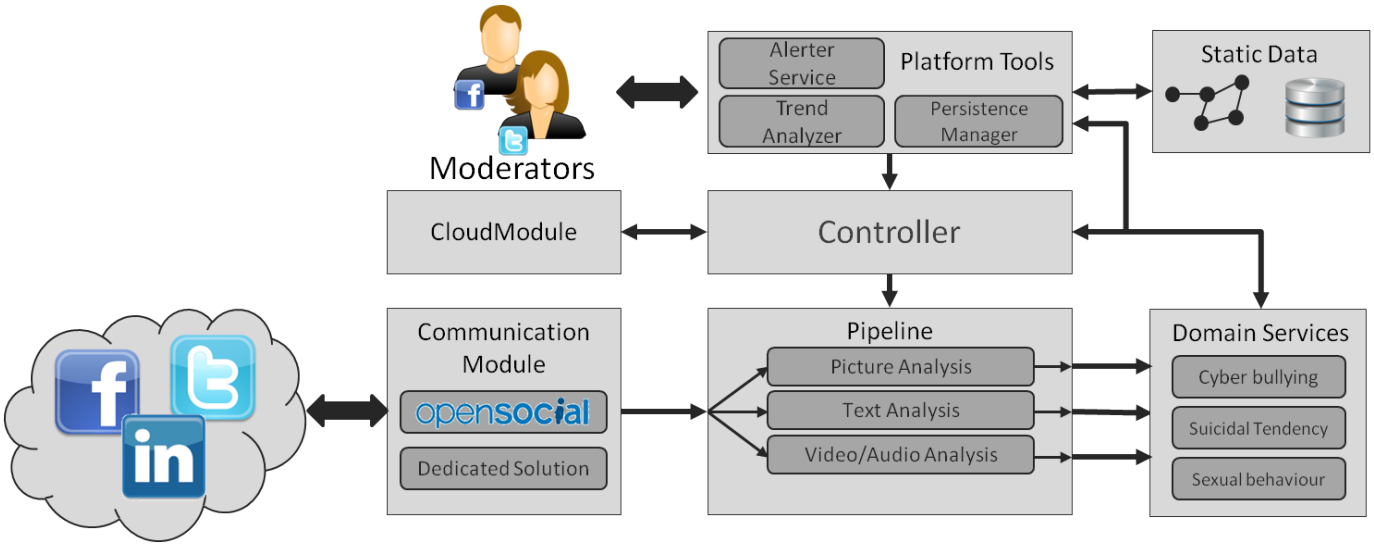


Fig. 1: Overview of the high-level architecture of the social network monitoring platform consisting of a communication module providing input for the analyzing pipeline which provides information for the domain services. Both are managed by a controller while the platform tools provide additional functionality.

defined concepts, properties and relations. They have already been succesfully used in content recommender systems [13] as well as in the research domain of social networks [14].

III. Platform requirements

Time will play a crucial role for the detection of certain harmful content. For instance, the detection of suicidal tendencies is more time critical than the detection of inappropriate language. Therefore, services should be manageable in terms of execution time (i.e., instant, overnight, on a weekly basis, etc.) and fluctuations in the amount of data generated on the social network feed depending on the time of day. Performance, scalability and dynamic management of the components are hence important factors for the platform architecture. Secondly, to ensure the platform is future-oriented, it needs to be extendable. This way the platform can be extended with new use cases or modules for the analysis of data. Some of these modules will also be used in different scenarios for different goals and will therefore need to be reusable.

Deploying the platform in a cloud environment can achieve the performance and scalability when needed and in a similar manner the platform itself can be provided to social network moderators using the cloud. The impact on the side of the social network would thus be very minimal as the software would run off-site in the cloud while access through a web browser eliminates any issues related to on-site installation. However, because the social network data will be off-site, the security and privacy aspects of this data become main requirements.

Two possible deployment strategies exist for the platform in the cloud: Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS). SaaS is a model where software is delivered through the cloud and typically accessed through a web browser. In the PaaS model a computing platform is provided together with tools and libraries for development. The moderators of the social network would not only have access to the platform through a webservice, but would also be able to extend the platform with own software. Supporting such a deployment strategy would involve creating a library for developers and a screening process for externally developed applications.

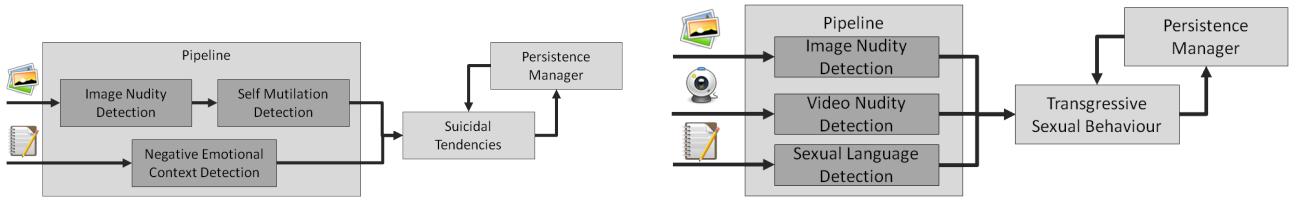
IV. Architecture

Figure 1 shows the proposed architecture schematically. It consists of five main components. Three are responsible for handling and analysing the social network data, i.e. the communication module, the pipeline and the domain services, and two components are general, the controller and platform tools. Required storage of additional or profile-based data is stored in the static data component. The following section details these key architectural components.

A. Communication module

The communication module is responsible for mining the data from one or multiple social networks. The data of interest ranges from text-based information to image, video and audio files. To retrieve this data, a dedicated solution can be developed for the connection with the social network. However, as every social network has its own interface for communication, developing a dedicated solution for every social network is labor-intensive.

The OpenSocial Application Programming Interface (API) [15] provides tools for developing applications across different social networks, but it might not be able to access as much data as the dedicated solution. The lack of data leads to a less accurate and advanced analysis of the social network, which in turn leads to a less trustworthy detection of harmful content.



(a) Detection of suicidal tendencies

(b) Detection of transgressive sexual behaviour

Fig. 2: Example workflows of the platform

B. Pipeline

The communication module passes the data to the pipeline, where it will be assigned to the different analysis modules according to different categories such as text, image, audio and video files. These modules are deployed in parallel and scaled on distributed systems (e.g. cloud environments) to ensure the performance of the system. This is possible because the analysis modules are considered to be use case independent components: they run their algorithm based on the available input and turn it over to the domain services. The deployment and scaling of the modules in cloud environments and overall management will be handled by the controller, discussed in Section IV-F. To ensure the extendability of the pipeline, all the modules comply with an agreed upon common interface.

C. Domain services

Domain services make a decision concerning one specific case of abuse based on the data aggregated from different analysis modules. Figure 1 shows three specific domain services: detection of cyber bullying, inappropriate sexually suggestive behaviour and detection of suicidal tendencies. The domain services are however not limited to these examples as they also conform to a common interface, again ensuring the extendability of the platform.

Similar to the analysis modules, the domain services are deployed and scaled by the controller. They are also executed in parallel and independently of each other, often on distributed systems, again to guarantee the performance of the entire platform.

D. Platform tools

Additional functionality and a user interface will be provided through a variety of platform tools. Examples of platform tools are the trend analyzer and alerter service. The trend analyzer uses recent history and current feed input to discover trends in online social behaviour of a user or a group of users. The alerter services will raise an alert for the moderators of the social network when a domain service discovers a possible case of abuse. These tools will have access to any output of the domain services and to the saved static data.

Some of the platform tools will also provide input for the controller. Consider the case where the trend analyzer discovers an increase in suicidal tendencies following the suicide of a celebrity. This information is passed on to the controller, which could then decide to dedicate more resources to the domain service for suicidal tendencies detection and scale the analysis modules accordingly. This will result in a faster discovery of any user with suicidal tendencies for which an alert will be raised by the alerter service.

E. Static data

Selective profile information about users, discovered trends and raised alerts will be saved by the platform. A persistence manager handles access to this data. On the one hand this avoids conflicts like redundancy, on the other hand this shields the structure of the data from the platform tools and domain services. As mentioned above, there might exist a need for ontologies based on the semantic data available on the social network. However, as the persistence manager protects the representation of the data, the platform will be able to handle other structures too (e.g. SQL database).

F. Controller

The controller is responsible for managing the domain services and the analysis modules in the pipeline. These modules and services are deployed in cloud environments (e.g. Amazon EC2) by the controller in order to satisfy the performance requirement. To achieve this goal the controller communicates with those environments through widely available APIs (e.g. Amazon EC2 API Tools) or dedicated solutions.

Aside from controlling the deployment of modules, the controller is also an entity controlling the execution of workflows. Workflows result from the close connection of the communication module, the analysis modules in the pipeline, the domain services and the data moving through these components. Section V furter explains the concept.

V. Workflow scenarios

In this section we elaborate on two illustrative scenarios to clarify the workflow concept in the platform. Figure 2 shows two example workflows. In Figure 2a the domain service uses the input of image analysis, text analysis and any relevant static data in the platform for the detection of suicidal tendencies. The image analysis consists of two consecutive analysis modules. The first module detects a certain level of nudity in a picture while the second module checks for any signs of self-injury. This scenario shows how one analysis module can act as a filter for others. As most self-injury occurs on parts of the body that are usually covered, the nudity detection module filters out less likely pictures. In

the text analysis, negative emotional context is detected, such as sadness. Combined with information retrieved from the user's profile or from a previous analysis, a decision is made concerning the suicidal tendency of the user. This information is saved and, if required, used by the platform tools.

Figure 2b illustrates a workflow for the detection of transgressive sexual behaviour. Three analysis modules work in parallel on three categories of multimedial data. Images and videos are scanned for nudity and text is examined for sexual language. The domain service combines the results and decides whether to flag the profile or not. Note that the image analysis module is the same module as in the previous workflow.

VI. Implementation challenges

In this section we identify the main implementation challenges of the proposed platform architecture

A. Aggregation of analysis results

Domain services aggregate data from several data analysis modules in the pipeline. Consider the scenario depicted in Figure 2a where the domain service uses the output of a text- and an image analysis module. Combining these factors could give a strong indication towards the suicidal tendencies of a specific user. Be that as it may, what would be the influence of either factor on the final decision? Furthermore, in a realistic use case the domain service reasons about even more different analysis results, making the combination all the more challenging. Commonly used techniques for reasoning about data are neural networks, rule-based systems, decision trees and collaborative filtering.

B. Scaling domain services

Because of the close connection between the analysis modules and domain services in the workflow, scaling the domain services is a tedious process. Once a domain service is scaled upwards, it will analyze the same amount of data in a shorter timeframe. This data is provided by one or more analysis modules from the pipeline and so as to optimize the workflow, these modules also need to scale up. If the modules are not scaled up, the domain service will not work optimally as not enough analyzed data will be available. A similar problem occurs when a domain service is scaled down and the analysis modules also need to be scaled down. To prevent such problems, an efficient resource management needs to be implemented in the controller.

C. Selective profile data storage

Social networks contain vast amounts of data and saving all this data in the platform is impossible and inefficient. Selecting and saving the useful data is therefore a very important challenge, especially concerning the privacy of social network users. In addition, thoughtful decisions should be made as to when stored data becomes irrelevant and thus may be deleted.

VII. Conclusions

In this paper we proposed an architecture of a platform for analyzing online social network behaviour with the ultimate goal of tracing harmful content in an automatic way. The pluggable architecture consists of several components based on predetermined requirements: performance, scalability, reusability and extendability. Analysis modules detect inappropriate content and high risk behaviour after which domain services accumulate these results and flag user profiles if necessary. With this platform, moderators of social networks will be able to quickly and accurately scan the network feed and intervene if necessary. Future work will consist of evaluating the performance of the proposed platform, and study in depth the three mentioned challenges.

Acknowledgment

The AMiCA (Automatic Monitoring for Cyberspace Applications) project is funded by IWT (Institute for the Promotion of Innovation through Science and Technology in Flanders).

References

[1] Eurostat, "Individuals using the internet for participating in social networks," December 2012.

[2] C. Haythornthwaite, "Social networks and internet connectivity effects," *Information, Communication & Society*, vol. 8, pp. 126–147, June 2005.

[3] G. Vickery and S. Wunsch-Vincent, *Participative Web and User-Created Content: Web 2.0, Wikis and Social Networking*. OECD Publications, OECDpublishing, October 2007.

[4] K. Fisher, "Girl commits suicide after being cyber bullied." ABC4 Article, October 2012. http://goo.gl/8hFGz.

[5] H. Vandebosch and K. Van Cleemput, "Cyberbullying among youngsters: profiles of bullies and victims," *New Media Society*, vol. 11, pp. 1349–1371, December 2009.

[6] R. S. Tokunaga, "Following you home from school: A critical review and synthesis of research on cyberbullying victimization," *Computers in Human Behavior*, vol. 26, no. 3, pp. 277–287, 2010.

[7] P. K. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell, and N. Tippett, "Cyberbullying: its nature and impact in secondary school pupils," *Journal of Child Psychology and Psychiatry*, vol. 49, no. 4, pp. 376–385, 2008.

[8] D. Olweus, *Bullying at School: What We Know and What We Can Do*. Understanding Children's Worlds, Wiley, 1993.

[9] C. Peersman, F. Vaassen, V. Van Asch, and W. Daelemans, "Conversation level constraints on pedophile detection in chat rooms," in *CLEF (Online Working Notes/Labs/Workshop)* (P. Forner, J. Karlgren, and C. Womser-Hacker, eds.), 2012.

[10] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proceedings of International AAAI Conference on Weblogs and Social Media, Workshop Social Mobile Web*, 2011.

[11] M. Dadvar and F. de Jong, "Cyberbullying detection: a step toward a safer internet yard," in *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pp. 121–126, ACM, 2012.

[12] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[13] B. Adrian, L. Sauermann, and T. Roth-berghofer, "Contag: A semantic tag recommendation system," in *Proceedings of ISemantics 07*, pp. 297–304, JUCS, 2007.

[14] P. Mika, "Flink: Semantic web technology for the extraction and analysis of social networks," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, no. 2-3, pp. 211–223, 2005.

[15] J. Mitchell-Wong, R. Kowalczyk, A. Roshelova, B. Joy, and H. Tsai, "Opensocial: From social networks to social ecosystem," in *Digital EcoSystems and Technologies Conference, 2007. DEST '07. Inaugural IEEE-IES*, pp. 361 –366, February 2007.