

# SocConnect: Intelligent Social Networks Aggregator

Yuan Wang<sup>1</sup>, Jie Zhang<sup>2</sup>, and Julita Vassileva<sup>1</sup>

<sup>1</sup>Department of Computer Science, University of Saskatchewan, Canada

<sup>2</sup>School of Computer Engineering, Nanyang Technological University, Singapore

<sup>1</sup>{yuw193, jiv}@cs.usask.ca <sup>2</sup>zhangj@ntu.edu.sg

**Abstract.** We have developed a dashboard application called “SocConnect” for integrating social data from different social networking sites (e.g. Facebook, Twitter), which allows users to create personalized social and semantic contexts for their social data. Users can blend their friends across different social networking sites and group them in different ways. They can also rate friends and/or their activities as favourite, neutral or disliked. Machine learning can be usefully applied in predicting the interest level of users in their social network activities, thus helping them deal with cognitive overload.

## 1 Introduction

Social Networking Sites (SNSs) have changed how people communicate: nowadays, people spend more time on SNSs than ever, and prefer communication via SNSs over emails. Current SNSs have the limitation of poor user data interoperability [1]. User content, online activities, and friends are scattered over different places. It becomes increasingly inconvenient for users to manage their social data and constantly check many sites to keep track of all recent updates. People may also keep different accounts on the same SNS in order to protect their privacy or other purposes. In addition, users are often overwhelmed by the huge amount of social data, especially friends’ activities. There have been many attempts to create social network aggregators that integrate a user’s accounts on different SNSs. Based on their platforms, social aggregators can be classified as web-based and desktop applications. In web-based aggregators, users need to register and create a new account for the aggregator, and provide their SNSs accounts information to the aggregator. In desktop aggregators, which have been emerging on mobile platforms recently, users do not need to create an account. Based on their functions, social aggregators can be divided into three groups: write-only, read-only, and write and read. Write-only and read-only aggregators usually are lightweight and web-based. They allow users to publish the same status update to multiple SNSs. However, these applications do not allow users to blend or group their friends from different places; therefore the updates appear scattered, out of context.

In contrast, we take a read-only desktop-based approach for integrating the user’s social data on different SNSs, and that allows users to organize and to create personalized contexts for their social data. We also provide personalized

recommendation of friends' activities from different SNSs. While many SNSs deploy algorithms based on the analysis of social network structure to recommend new friends to the user, there haven't been many approaches to recommend contents on SNSs. One such approach is SoNARS, which combines results from collaborative filtering and content-based recommendation [2]. A Twitter desktop client application called TalkingPuffin ([talkingpuffin.org](http://talkingpuffin.org)) allows users to remove "noise" (uninteresting updates) by manually muting users, retweets from specific users or certain applications. Our approach provides content-based recommendations by applying machine learning techniques on previous rated by the user social activities.

## 2 SocConnect Dashboard

The SocConnect dashboard [3] is a client application communicating with a server that processes the data and generates recommendations. It retrieves the information about the user's friends and their activities (streams) using the APIs of different SNSs. SocConnect provides three functional categories, managing friends, rating friends and activities, and recommendation of activities.

The first functional category, managing friends contains two functions: *blending friends* and *grouping friends*. Blending friends allows the user to merge the different accounts of a given friend across two or more SNSs, and to create a single blended friend account for this friend in the dashboard. The second function is to group friends. Users can put their friends, both individual SNS accounts and integrated accounts into named groups. This function allows users to express the context and semantics of friendships, which could be the shared characteristics, interests or activities between friends. The second functional category, *rating friends and activities*, allows users to rate friends or friends' activities as favourite or disliked. The favourite activities are bookmarked, and be revisited easily.

The third functional category, *personal recommendations* automatically provides recommendation of activities that may be interesting based on the previous ratings and the information about friend groups. A content-based approach is used to predict possible favourite or disliked activities by the user. The activities that were rated by the user in the past are fed to machine learning algorithm which is trained to predict the level of liking that the user may have of an incoming activity. Both textual (vector space) and non-textual features of the activities are used for learning to predict which activities will be found interesting by the user. The following non-textual are used: the actor (the friend who originated the activity), the actor type (if the friend is marked as favourite, or disliked), the activity type (upload photo album, share link, upload a photo, upload a video, status update, application use, reply, retweet), the SNS that is the source of the activity (Twitter or Facebook), the application that is the source of the activity (e.g. Farmville or ForSquare), and user rating of the activity. Based on these features the machine learning algorithm is trained to predict the level of liking of future incoming activities, which are classified into 5 possible levels varying from *strong dislike* to *strong like*. The incoming activities in the combined

stream of the integrated SNSs is displayed in a way that highlights the recommended activities (see [3] for details).

We experimented with four different machine learning algorithms in WEKA to see which one will generate the best predictions. These algorithms (methods) were: Decision Tree, Radial Basis Function (RBF), Naive Bayesian Network and SVM (Support Vector Machine). We compared also the quality of recommendations using only textual data, only non-textual data and a combination of both textual and non-textual data. We found that when using only non-textual features, although the performance difference among the algorithms is not significant, support vector machine (SVM) provides the best performance, and it correctly classifies 69.9% of instances in the testing data. Using only the textual features, the performance is not very good (72% for BayesNet, 75% for SVM, 77% for Decision Tree, and 84.5 for RBF). Combining textual and non-textual features yields the best performance overall (75% for BayesNet, 80% for SVM and Decision Tree, 81.4% for RBF). Therefore, even though more computationally expensive, using the text features is worthwhile.

### 3 Conclusions and Future Work

We have created a novel desktop based SNSs integrator, called “SocConnect”. It allows users to blend their friends across different SNSs, to group them and to rate their friends and activities. This integrator is used as a platform for generating personalized recommendations of activities for the user, to help users deal with the information overload. We have investigated different machine learning methods to learn about implicit user preferences and generate predictions and found one, RBF that looks particularly promising. In our future work we will explore more deeply the importance of different features of SNS activities. We will conduct user studies with the SocConnect prototype to test its usability. We will also work on expanding its functionality to “read and write” rather than just “read”.

### 4 References

1. F. Carmagnola, F. Venero, and P. Grillo, “Sonars: A social networks-based algorithm for social recommender systems,” in *Proc. 17<sup>th</sup> International Conference on User Modeling, Adaptation, and Personalization*, 2009.
2. G. Ereteo, M. Buffa, F. Gandon, M. Leitzelman, and F. Limpens, “Leveraging social data with semantics,” in *Proc. of the W3C Workshop on the Future of Social Networking*, 2009.
3. Y. Wang, J. Zhang, and J. Vassileva, “SocConnect: A user-centric approach for social networking sites integration,” in *Proc. International Conference on Intelligent User Interface (IUI) Workshop on User Data Interoperability in the Social Web*, 2010.
4. Webster, A., Vassileva, J.: Personal relations in online communities. in *Proc. Adaptive Hypermedia and Adaptive Web-Based Systems*, Dublin, Ireland, Springer LNCS (2006), 223-233.