

Predicting the Importance of Newsfeed Posts and Social Network Friends

Tim Paek, Michael Gamon, Scott Counts, David Maxwell Chickering, Aman Dhesi*

Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
{timpak|mgamon|counts|dmax}@microsoft.com

Indian Institute of Technology Kanpur*
Kanpur 208016
Uttar Pradesh, India
amand@iitk.ac.in

Abstract

As users of social networking websites expand their network of friends, they are often flooded with newsfeed posts and status updates, most of which they consider to be “unimportant” and not newsworthy. In order to better understand how people judge the importance of their newsfeed, we conducted a study in which Facebook users were asked to rate the importance of their newsfeed posts as well as their friends. We learned classifiers of newsfeed and friend importance to identify predictive sets of features related to social media properties, the message text, and shared background information. For classifying friend importance, the best performing model achieved 85% accuracy and 25% error reduction. By leveraging this model for classifying newsfeed posts, the best newsfeed classifier achieved 64% accuracy and 27% error reduction.

Introduction

According to market research (Morgan Stanley, 2009) social networking is a global phenomenon; Facebook alone has over 350 million active users with 137% year-to-year growth. Indeed, over the last 3 years, users spent more global Internet minutes on Facebook than any other website. As more people join social networking sites, and users expand their network of friends, they are often confronted with a triage problem: their user accounts are flooded with newsfeed posts and status updates, most of which they consider to be “unimportant” and not newsworthy (as we demonstrate later in our data analysis). In this paper, we explore to what extent we can accurately predict users’ perceived importance of newsfeed posts and of their friends. We employ machine learning to not only learn classifiers for newsfeed posts and friends, but also to gain insight into the kinds of features related to social media properties, the message text, and shared background information that are indicative of importance. Such models and insight could be used to develop intelligent user interfaces that filter or re-rank newsfeeds.

This paper consists of four sections. First, we provide background on social media and related research. Second, we describe a study in which Facebook users were asked to rate the importance of their newsfeed posts and friends.

Third, we delineate all the features we engineered, and relate the results of model selection experiments in which we learned support vector machine (SVM) classifiers using different combinations of features. Finally, we discuss the results with an eye towards future research, including what benefit might be possible with personalization.

Background

Most content in popular social media websites takes the form of status updates or posts that are contributed by users and subsequently pushed out to others who are friends or followers of that user. Facebook utilizes this concept by allowing users to post text status updates, as well as to share links, photos, and videos. Once posted, this content is pushed to the newsfeeds of friends in the post sender’s social network, where it is presented in reverse chronological order. Based on usage statistics from Facebook¹, a rough estimate shows that the typical Facebook user receives well over 1,000 items per week from 130 friends. Despite an average of 55 minutes per day spent on the site, given the sheer number of items and chronological presentation, users are likely to miss some potentially interesting content.

This highlights the need for better tools to surface the most important newsfeed posts. Facebook itself has implemented a system for distinguishing the more important or interesting content (the “News Feed”) from the stream of all content (“Live Feed”). The details of the algorithm for identifying News Feed content are not publicly known, but it appears to use a heuristic approach that includes metrics like what type of content was posted (e.g., a status update, link, photo) and how many comments it has received². The system does not appear to take into account the message text or any historical information such as how frequently users have corresponded with the post sender. Furthermore, there is relatively little functionality in Facebook to help users triage feed content explicitly. Other than gross level setting like blocking and hiding (specifying friends and applications you do not want to receive content from) and specifying individuals from whom you would like to see more content, there is no functionality for nuanced content triage such as

¹ <http://www.facebook.com/press/info.php?statistics>

² <http://www.facebook.com/help/?page=408#/help/?faq=16162>

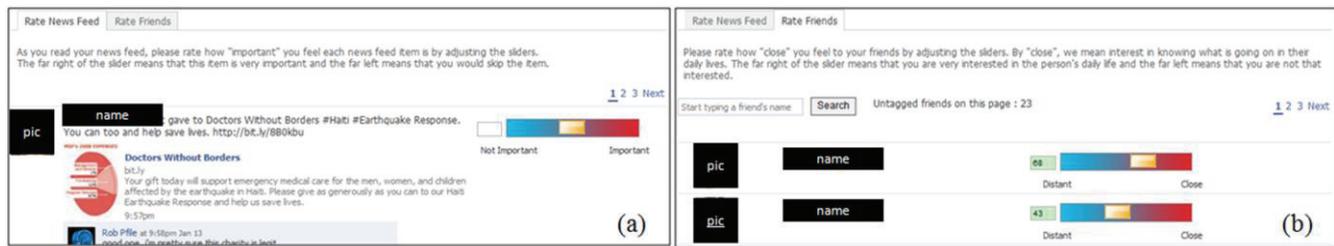


Figure 1: Screenshots of the Newsfeed Tagger Facebook application showing how participants rated (a) news feed posts and (b) friends.

preferentially weighting content based on keywords, or otherwise helping users rank their feed content.

Related Research

Given that Facebook data is not entirely public, little research has examined methods for content ranking in Facebook. However, several efforts have demonstrated the predictive qualities of Facebook data. First, in terms of leveraging social media to uncover relationships, Gilbert and Karahalios (2009) showed that properties such as the number of intimacy words exchanged between two users on their Facebook walls and days since their last communication, can predict “tie strength” (Granovetter, 1973), or the strength of the relationship between any two users, with moderate to strong accuracy. In our study, although we had users rate the “importance” of their friends, it was in terms of how interested they were in knowing about their daily activities. Because it is possible to have weak tie strength *and* a strong interest in knowing about a friend’s daily life (e.g., a boss), our focus here is more about news and less about social relationships, though we consider all such variables to be useful features for classification.

In addition to predicting tie strength, Facebook has been analyzed statistically to better understand a variety of properties of users and their behavior. For example, new Facebook users’ photo posting behavior can be predicted by the photo posting behavior of friends in their network (Burke et al., 2009). The number of friends has been shown to have a curvilinear relationship with the social attractiveness and extraversion of the user (Tong et al., 2008). And Sun et al. (2009) demonstrated that information diffuses in small chains of users that may then merge, rather than starting at a single point.

In terms of triaging content more broadly, research in the email domain has demonstrated a number of benefits to leveraging social metadata. For example, Venolia et al. (2001) highlight a variety of social attributes of email that contribute to perceived importance, including whether it was addressed directly to the user as well as the relationship of the sender to the user (e.g., whether the email came from a manager). Given the potential usefulness of social metadata, prototype systems such as DriftCatcher (Locker, 2002), Bifrost (Balter & Sidner, 2002), and SNARF (Neustadter et al., 2005) have incorporated social relationship information when organizing and presenting email to the user in order to facilitate triage. Finally, of notable relevance, Horvitz et al.

(1999) demonstrated that machine learning could be leveraged to rank email content for near-automated triage.

In summary, the sheer number of posts most users see in Facebook highlights the need for better content ranking. While relatively little research exists specifically on ranking social network feed content, prior work has demonstrated that user behavior and Facebook content are predictive of a number of phenomena, and thus are good candidates on which to train statistical models for classification. From work in the email domain, we know that both social metadata and machine learning have been successfully leveraged to help triage incoming content. Here, we take a similar approach in what, to our knowledge, is the first research to apply machine learning to build predictive models of newsfeed importance, which in turn can be used to build interfaces that help users triage their flood of posts.

User Study

In order to obtain importance ratings for newsfeed posts and friends, we conducted a user study. We recruited 24 participants through an email solicitation sent to our organization. Participants were required to be active Facebook users who checked their newsfeed on a daily basis. All participants had at least 200 friends in their social network. They were also financially compensated for their involvement.

Data Collection Method

Participants were asked to download a Facebook application we developed called “Newsfeed Tagger” under Facebook’s Terms of Service agreement. As shown in Figure 1, the application consists of two tabs: one to *Rate News Feed* and another to *Rate Friends*. For newsfeed posts, the application retrieved and displayed posts using the same markup language style as the newsfeed on the Facebook home page (Figure 1(a)). Participants were instructed to rate the importance of each post using a slider next to the post, where “the far right of the slider means that this item is very important and the far left means that you would skip the item.” The sliders provided a continuous value from 0 to 100. For rating friends, participants received a list of friends in their network ranked according to a simple heuristic that took into account the last time users interacted with that friend and how frequently. As shown in Figure 1(b), because users

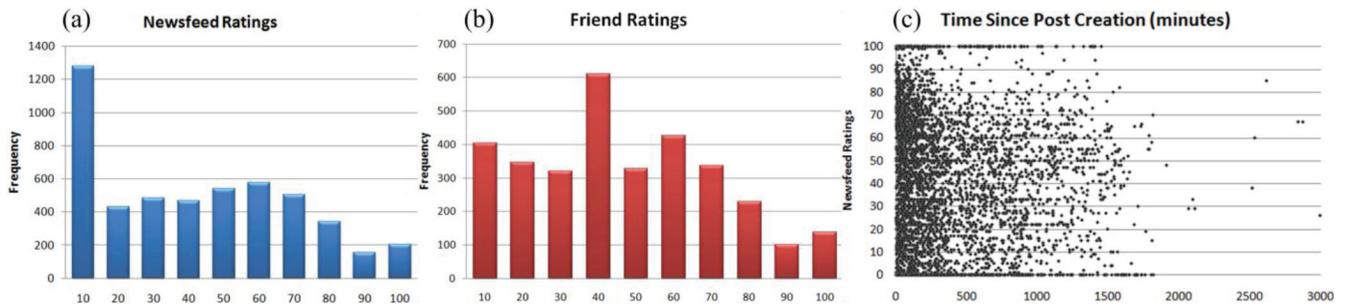


Figure 2: (a) Newsfeed ratings histogram; (b) Friend ratings histogram; (c) Scatter plot of time since post creation by newsfeed rating.

had over 200 friends, we also included a search box so that users could find friends. Participants were instructed to use the adjacent sliders to rate how “close” they were to the friend, where closeness was defined as “interest in knowing what is going on in their daily lives”. Because many of the participants found it onerous to rate all their friends, we asked them to rate at least 100 friends.

Participants were asked to do the rating every day for a full business week. Because we allowed participants to submit their ratings at their own leisure, not all participants actively rated their newsfeed and friends. In all, we received 4989 newsfeed ratings and 4238 friend ratings. Upon initiating the study, we downloaded whatever information was programmatically available for the participant’s Facebook account through the beta version of the Facebook Open Stream API per the Terms of Service agreement. Because participants had extensive social networks, we did not download information about all the friends in their networks but only those they remembered enough to rate in the *Rate Friends* tab. Because participants rated friends who had not sent posts during the week of the study, and not all poster senders were rated by the participants, only 3241 out of the 4989 posts (65%) had ratings for the sender, along with other downloaded information. We used this smaller dataset for model selection so that we could compare the effects of using different sets of features.

Data Analysis

In order to validate the need for newsfeed triage, we first examined descriptive statistics for the ratings. Figure 2(a) displays a histogram of all the newsfeed ratings. The mode of the ratings was 0 – hence the large spike in the left of the histogram. The average rating was 37.3 and the median 36. Note that ratings greater than 80 comprised the two smallest bins in the histogram. $\frac{3}{4}$ of the ratings were below 60. In short, the descriptive statistics demonstrate that most participants regarded the majority of newsfeed they received to be unimportant, though participants varied in their rating distributions, as we revisit later.

Figure 2(b) displays a histogram of the friend ratings. Similar to the newsfeed ratings, the two smallest bins consist of ratings 80 and above. The mode was 0 and $\frac{3}{4}$ of the ratings were below 60. The average friend rating was 42.4 and the median was 40. Hence, our participants

considered the majority of their friends to be people for whom they had little to moderate interest in knowing about their daily affairs. This does not include the friends they could not remember.

Finally, because Facebook utilizes reverse chronological ordering of the newsfeed, we assessed to what extent timeliness, or urgency, was correlated with the ratings. In other words, we investigated whether participants considered the most recent newsfeed posts to be the most important. Figure 2(c) shows a scatter plot where the x-axis represents the time since the post was created in minutes and the y-axis represents the newsfeed rating. Note that instead of a left leaning slope the scatter plot shows more of a vertical column; indeed, the Pearson correlation ($r=.01$) was not statistically significant. In short, for our participants, reverse chronological ordering did not suffice to surface the most important newsfeed posts.

While many Facebook users would have suspected much of the data analysis reported in this section, no prior research has, to our knowledge, provided any such empirical validation.

Model Selection Experiments

We conducted model selection experiments with two goals in mind: first, we sought to identify what kinds of features were predictive of the perceived importance of newsfeed posts and friends, and second, we sought to attain the maximum classification accuracy possible on the data. Given the successful track record of linear kernel SVM classifiers in the area of text classification (Joachim, 1998), and the fact that they can be trained relatively quickly over a very large number of features (e.g., n-grams), we decided to learn linear SVM classifiers using the Sequential Minimal Optimization (SMO) algorithm (Platt, 1999). For performance reasons, we discretized the values of the continuous predictor variables into 5 bins containing roughly the same number of cases in each bin. For our primary target variable, newsfeed rating, which is also continuous, we split the ratings into 2 bins, *Important* and *Not Important*, for several reasons. First, we intended to employ the models as a type of spam filter, which is typically binary. Second, finer-grained classification would have been difficult given the size of our dataset (3241 cases). Furthermore, although we could have set the target

variable threshold to the midpoint of the sliders (i.e., 50), given the skewed histogram in Figure 2(b) we decided to use the median rating (i.e., 35) instead. This allowed us to avoid modeling complications due to unbalanced classes.

Feature Engineering

Having downloaded all programmatically available content from participants' Facebook accounts, we engineered features from three types of information: social media properties, the message text and corpus, and shared background information.

Social media properties. Social media properties included any properties related to the newsfeed post and sender, excluding the actual text. In particular, we extracted: Whether the post was a wall post or feed post; Whether the post contained photos, links, and/or videos; Total number of comments by everyone; Total number of comments by friends (including multiple comments); Total number of comments by distinct friends; Total number of likes by everyone; Total number of likes by friends; Time elapsed since the post was created; Total number of words exchanged between the user and the sender on their respective walls (including comments); Total number of posts from the user to the sender; Total number of posts from the sender to the user; Time since the first exchange; Time since the most recent exchange; Total number of photos in which both the user and sender are tagged together; Total number of photos the user has of the friend and vice versa; Total number of friends overlapping in their respective networks.

For every post, we also had lists of Facebook account IDs that had provided comments, likes, etc. We created a set of features based on knowing the importance rating of the account IDs; in particular, the maximum friend rating of people who posted comments, put likes, or are otherwise tagged in photos. The intuition here is that even if users do not find the post content to be important, it may become important if someone they know and track with great interest commented on it. For mutual friends between the user and sender, we also extracted the maximum, minimum and average friend rating, along with its variance.

Message text and corpus. For text analysis features, we looked at two sources: the post and the corpus of all posts exchanged between the user and the sender. Because Facebook maintains only the most recent posts, for the corpus we were only able to retrieve posts up to roughly 2-3 months prior to the date of retrieval. In order to capture the linguistic content of the post and corpus, we extracted both n -gram features, with n ranging from 1 to 3, and features based on the Linguistic Inquiry and Word Count dictionary (LIWC, Pennebaker et al., 2007). N -gram features had binary values depending on whether the n -gram was present or not, whereas the LIWC features

consisted of counts. Binary features that were observed 3 times or less in the corpus were eliminated. The LIWC features correspond to the counts of words in a text belonging to each of 80 categories in the LIWC dictionary. Note that Gilbert and Karahalios (2009) did not utilize any n -gram features and only looked at 13 emotion and intimacy related LIWC categories: Positive Emotion, Negative Emotion, Family, Friends, Home Sexual, Swears, Work, Leisure, Money, Body, Religion and Health. Given our focus on news, we decided to include all other categories, such as Insight (e.g., "think", "know"), Assent (e.g., "agree", "OK") and Fillers (e.g., "you know", "I mean").

In addition, we also extracted a number of other text-oriented features from the post and corpus: Whether there were embedded URLs; Total number of stop words; Ratio of stop words to total words; Ratio of non-punctuation, non-alphanumeric characters to total characters; Sum and average of $tf.idf$ (term-frequency \times inverse document frequency) of all words in a post or corpus, where $tf.idf$ scores were computed on all the posts in the entire dataset; Sum and average of $tf.idf$ of all words in a post or corpus, where $tf.idf$ scores were computed on Wikipedia; Delta of the previous two $tf.idf$ measures; Message length in tokens and characters.

Shared background information. Finally, for every participant and rated friend, we compared shared background information in terms of the following self-disclosed categories: Affiliations, Hometowns, Religion, Political Views, Current Location, Activities, Stated Interests, Music, Television, Movies, Books, Pre-College Education, College and Post-College Education, and "About Me" Profile. For each of these categories, after removing category-specific stop words (e.g., "high school" for Pre-College Education), we extracted the number of common words as well as the percent overlap.

Experimental Setup

All of our model selection experiments were conducted in the following manner. First, the dataset was split into five folds of training and test data. The training set of the first fold was then utilized to tune the optimal classifier parameter settings, as measured on the test set of the first fold. These settings were then used to learn SVM classifiers on the training files for the remaining four folds. Evaluation was performed on the test sets of the four folds. We conducted a grid search on the first fold to determine optimal values for the SVM cost parameter c , which trades off training error against model complexity (Joachims, 2002). For feature reduction of binary features, even after imposing a count cutoff, the number of n -gram features was in the tens of thousands. So, we reduced the number of features to the top 3K features in terms of log likelihood ratios (Dunning, 1993) as determined on the training set.

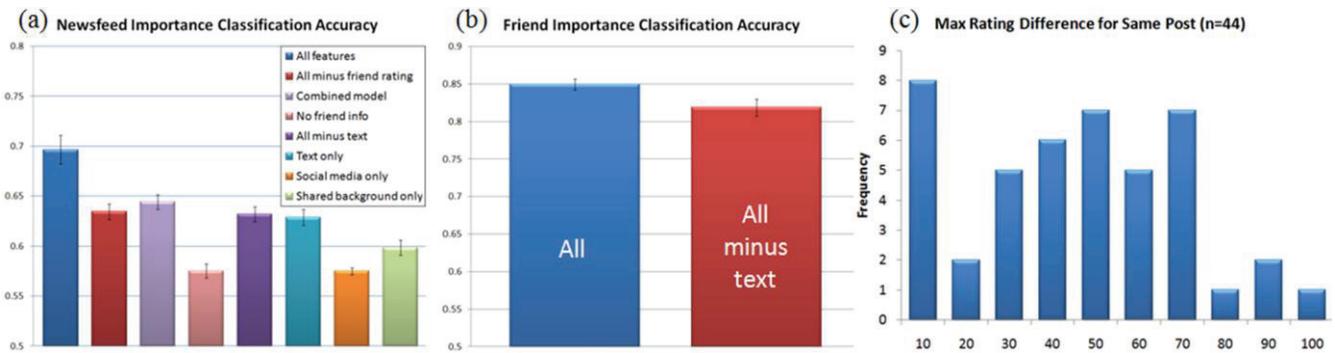


Figure 3: (a) Newsfeed importance classification; (b) Friend importance classification; (c) Histogram of the maximum rating differences between participants for the same newsfeed post. Error bars represent standard errors about the mean.

Results

For newsfeed importance, the first model we learned was a classifier using all of the features, including the friend rating. Because we do not have access to the friend rating in a deployed setting and can only infer it, the performance of this model provides an upper bound for classification accuracy. The baseline accuracy, based on predicting the majority class, is 51.3%. As shown in Figure 3(a), using all features achieved the highest classification accuracy at 69.7% with 37.7% relative reduction in error rate. In analyzing the top 50 selected features ranked by their learned SVM weights, we observed a number of findings: First, not surprisingly, friend rating was the top feature, though because friend rating (as a predictor variable) was discretized into 5 bins, only the top and bottom bins were selected. We computed the Pearson correlation to measure the relationship between newsfeed rating and friend rating, and found it to be statistically significant ($r=0.38$, $p(\text{two-tail}) < .01$). Second, we found that the majority (34/50) of the top features were message text and corpus features. Examples included the count of LIWC Home words, the count of LIWC Ingestion words, and having a trigram consisting of “with” and two capitalized words (i.e., Proper Name). Note that because some n-gram features (e.g., “beer”) are also incorporated into LIWC counts (e.g., Ingest) the SVM algorithm splits the weight between the features. Hence, the top message features may be more predictive than what is indicated by their SVM weights, and some redundant message features that are very predictive may be missing from the top 50. Finally, in the rest of the features (16/50), we found social media properties, such as the average friend rating for mutual friends, and shared background information, such as the number of common words about Music.

Having confirmed a significant correlation between newsfeed rating and friend rating, we explored how well we could predict friend importance. Given that having a very high or low friend rating was an important predictor of newsfeed rating, we learned a multi-class classifier consisting of an SVM to predict the top 10% of the friend ratings (ratings > 77), an SVM to predict the bottom 10% (ratings < 7), and an SVM to predict the middle 80%. For

classification, the multi-class classifier utilizes the prediction of the SVM with the highest class probability. As shown in Figure 3(b), compared to the majority class baseline of 80.3%, the multi-class classifier achieved an accuracy of 85.0%, a 24.9% relative error reduction. Inspecting the top 50 selected features of the 3 SVM classifiers, the majority were corpus features (i.e., features based on analyzing all exchanged messages between the participant and sender). This again highlights the importance of textual features, even for friend importance. Indeed, as shown in Figure 3(b), if we remove all message text and corpus features (“All minus text”), then the accuracy dips to 81.9%, which is close to the baseline (but statistically different by McNemar’s test, $p < .01$).

Turning back to newsfeed classification, if we remove friend rating and all of its related features, such as the maximum friend rating of people who commented on a post, the accuracy drops to 63.4% (see “All minus friend rating” in Figure 3(a)). However, just because we will not have access to the friend rating of a sender in a deployed system does not mean that we cannot infer and incorporate it as a feature. As such, we integrated the multi-class classifier for friend importance into a classifier for newsfeed importance as follows:

$$p(\text{news}|\vec{f}) = \sum_i p(\text{news}|\vec{f}, \text{friend}_i) \cdot p(\text{friend}_i|\vec{f})$$

where $p(\text{friend}_i)$ is a normalized probability iterating over how likely it is that the sender of the newsfeed post is a top 10%, middle 80%, or bottom 10% rated friend. This combined model achieved a slightly higher accuracy at 64.4% (a 26.6% error reduction), but was not statistically different than the “All minus friend rating” model

Looking at other combinations of features, if we have no knowledge at all about the sender – i.e., if we remove all social media properties about the sender, all corpus related message features and all shared background features, then the accuracy drops dramatically to 57.5% (see “No friend info”). If instead we remove just the message text and corpus features (“All minus text”), then the accuracy is comparable to “All minus friend rating” at 63.2%. Finally, we investigated the contribution of the feature types just by themselves. As shown in the rightmost side of Figure 3(a), the “text alone” model was not statistically worse than any

of our other previous models, but was significantly better than both “Social media only” and “Background only” ($p < .01$). This again highlights the importance of message text and corpus features.

Discussion and Future Research

Overall, the best classification performance of any model is the combined model at 64% accuracy. Even if we had the friend ratings for all post senders, the upper bound is near 70%, which may be sufficient for re-ranking for newsfeed posts but is probably not good enough for triage. In fact, we tried to evaluate our models with respect to ranking (precision/recall), but discovered that most of the participants did not rate their newsfeeds in distinct sessions. As such, we could not accurately identify what newsfeed posts were available to participants at the time of rating, which we would need to know for ranking.

In our experiments using different combination of features, and in perusing the top selected features for the best performing models, textual features consistently stood out. In fact, a classifier using just text features was statistically no different than one which added other types of features. Even for predicting friend rating, corpus features contributed significantly to accuracy. This suggests that for prediction focused on news, having access to the message text and corpus features is vital.

In terms of limitations, our results are limited by the relatively small size of the data. Because of privacy restrictions, we were not able to collect data at a larger scale. However, we plan to conduct another user study, one where participants will rate their newsfeed at distinct times of the day so that we can evaluate ranking performance.

As for future research, one very promising direction is personalization. In our data analysis, we noticed 44 cases in which more than 1 participant had rated the same newsfeed post. We computed the difference between the highest rating and the lowest rating for identical posts (40 cases had only 2 ratings). Figure 3(c) displays a histogram of the differences in newsfeed ratings. If users generally find the same kinds of newsfeed posts to be important or unimportant, we would expect to see a heavily skewed distribution leaning towards the left. Here, we see that 36/44 (82%) of the cases differ in rating by more than 10 points with the average and median difference being roughly 41 points. In short, it was not unusual for two participants to give very different ratings to the same post, suggesting that importance ratings can be quite subjective, although it is hard to generalize with only 44 cases. As such, we decided to explore whether we could improve classification accuracy through personalization. In our data, only one participant had about 400 or more newsfeed ratings. For this user, we learned the “All minus friend rating” model. Compared to a majority class baseline of 53.6%, the personalized classifier achieved 69.6% accuracy, a 34.6% error reduction. With such an auspicious result, as we continue to collect more data, we plan to conduct more personalization experiments.

In this paper, we provided empirical validation of the need for triaging newsfeeds and moving beyond the standard reverse chronological ordering of posts. Having engineered a large set of programmatically available features for predicting newsfeed and friend importance, we conducted classification experiments using different combination of features and identified predictive features. To our knowledge, this research constitutes the first of what is likely to be many papers on triaging newsfeeds.

Acknowledgements

We thank Roger Booth for assistance in automating LIWC feature generation.

References

- Balter, O., & Sidner, C. (2002). Bifrost Inbox Organizer: Giving users control over the inbox. In *Proc. Nordic CHI 2002*, ACM Press.
- Burke, M., Marlow, C., and Lento, T. (2009). Feed me: Motivating newcomer contribution in social network sites. In *Proc. CHI 09*, ACM Press.
- Gilbert, E., & Karahalios, K. (2009). Predicting Tie Strength with Social Media. In *Proc. CHI '09*, ACM Press.
- Granovetter, M. (1973). The Strength of Weak Ties. *American Journal of Sociology*, 78 (6), pp. 1360-1380.
- Horvitz, E., Jacobs, A., and Hovel, D. (1999). Attention-Sensitive Alerting. In *Proc. UAI-99*, 305-313.
- Joachims, T. (1998): Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of ECML-1998*, 137-142.
- Joachims, T. (2002). Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms. Kluwer/Springer.
- Locker, A. (2002). Understanding Implicit Social Context in Electronic Communication, MIT Master's Thesis.
- Morgan Stanley (2009). “Mobile Internet Report” published Dec. 15, 2009.
- Neustadter, C., Brush, A., Smith, M., & Fisher, D. (2005). The Social Network and Relationship Finder: Social Sorting for Email Triage. In *Proc. CEAS 2005*.
- Pennebaker, J. D., Chung, C. K., Ireland, M., Gonzales A., and Booth, R. J. (2007). The LIWC2007 Application. <http://www.liwc.net/liwcdescription.php>.
- Platt, J. (1999): Fast training of SVMs using sequential minimal optimization. In: B. Schoelkopf, C. Burges and A. Smola (eds.) *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, 185-208.
- Tong, S.T., van der Heide, B., Langwell, L., & Walther, J. (2008). Too Much of a Good Thing? The Relationship Between Number of Friends and Interpersonal Impressions on Facebook. *J. of Computer-Mediated Communication*, 13(3), 531-549.
- Venolia, G., Dabbish, L., Cadiz, J., & Gupta, A. (2001). Supporting Email Workflow, Microsoft Technical Report TR-2001-88.