Using a Data Mining Approach: Spam Detection on Facebook

M. Soiraya, S. Thanalerdmongkol, C. Chantrapornchai
Department of Computing, Faculty of Science
Silpakorn University, Thailand, 73000

ABSTRACT

In this work, we present a social network spam detection application based on texts. Particularly, we tested on the Facebook spam. We develop an application to test the prototype of Facebook spam detection. The features for checking spams are the number of keywords, the average number of words, the text length, the number of links. The data mining model using the decision tree J48 is created using Weka [1]**.** The methodology can be extended to include other attributes. The prototype application demonstrates the real use of the Facebook application.

General Terms

Data Mining, Web Application, Social Networking, Decision Tree.

Keywords

Social network;Spam dection;Data minig; Facebook application.

1. INTRODUCTION

Data mining has been used in many applications recently [2]. It has been used for classification, identification, prediction, finding association etc. It is also combined to existing applications such as inventory/stock forecasting, image classification, spam classification and so on.

In this work, we study the potential application on detecting spams on social network using data mining. We particularly develop a prototype system that detects the spam on the Facebook. The application is running on the server where the Facebook request is rerouted to it to process the spam checking. We use the decision tree model J48 for classification. Sample 150 posts are trained and 75 posts and tested on the model. The features used are the number of words, the length of the post, and the number of the links with the recall rate of 66%.

Currently, there are several methods that detect spam patterns using data mining [3].

1. Anamaly detection. This is a way to detect an abnormal behavior among all typical case data.

2. Associative learning. It is like Amazon book suggestion. Typical users that have a behavior may perform the other more behaviors.

3. Cluster detection. The data is clustered in a group using some similarity threshold or criteria.

4. Classification. If we know the classification beforehand, we may categorize the given data into classes.

5. Regression. It is a prediction model. Based on the history data, the future behavior may be predicted using the model.

There are many works that study spam detections in various ways. For example, the work in [4] [5] [6] studied spam emails and images in emails. Some works inspect on the spam images only like [7] which use the decision tree method. Wang et.al. used image feature filtering to detect image spam [8] Many works until now focused on the web spam [9] [10]. The web link was investigated as the web spam or not [11]. In [12], improper web access detection was presented. The work demonstrated the usage of the proxy server with the blacklist, whitelist, keyword blocking. Jin, Lin, Luo and Han presented the framework called SocialSpamGuard. It is a spam detection system for social networks. The framework models the media network as a graph. The features are inspected such as image content, texts, and social network features to indicate the spam behaviours [13].

In [14],the framework for spam detection is also presented. The work is based on the HITS web link method and the bipartite graphs. The scores are calculated and semi-unsupervised learning was used to model. Also, in [15] social spam detection is proposed with six features such as plagiarism, valid link, number of advertisements, unrelated tags, tag spams, contents of sources etc.

Typical data mining technique for spam detections are as following [16].

1. Keyword search. This considers the documents that match the query best. Usually, TFIDF is calculated.
2. Linked-based ranking. The approach ranks the links. The popular algorithm is Pagerank or HITS.

In the above literature, there are also other features that are useful for counting such as term frequency, inverse document frequency, hyperlinks to documents etc. Normal data mining consist of two major steps: model construction where the model is created using a training set data and model testing where the test set is used to test the accuracy of the model.

To construct the model, the features of the problem need to be investigated thoroughly. Then the classification algorithm is used to derive the model. In this paper, we use the decision tree J48 as a prototype model for the classification. The selected attributes are the number of keywords, the number of links, the length of the post, the average number of words in a post. The relationships of the attributes may further be investigated in the future. The goal of this work is to only demonstrate the use of data mining model in detecting spams in the Facebook application.

The paper is organized as follows. The next section presents some backgrounds and theory in data mining. Section 3 presents our methodology. Section 4 presents the model results and application architecture. Section 5 concludes the paper.

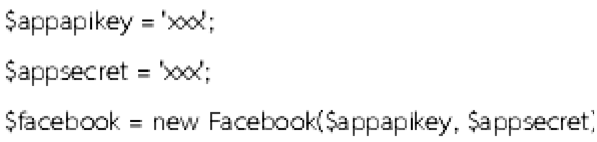
2. BACKGROUNDS

We present parts of the theories and technology used in the work.

2.1 Facebook API

In the development, we rely on the facebook APIs for acquiring user data. The more details of the APIs may be found at http://developers.facebook.com/.

The Facebook APIs are web services of Facebook which can be called by PHP client library. The important technology used is *Facebook Graph API*. It is related to objects and connections to Facebook including authentication. Every object has a unique ID which can be used to be contacted. For a particular application on Facebook, the application key and application secret need to be created.



```
$appapikey = 'xxx';

$appsecret = 'xxx';

$facebook = new Facebook($appapikey, $appsecret)
```

Facebook graph APIs can be accessed using http://graph.facebook.com/ID/CONNECTION_TYPE. After processing, the data is transferred back to the application. Besides, the Facebook graph APIs has special objects such as *me*, http://graph.facebook.com/me where it will return information about the current object such as authorization, reading, searching, publishing, deleting, analytics privileges.

2. 2 Web Filtering

Typical web filtering uses the approaches such as

1. Blacklist/Whitelist. The blacklist is the list that is not safe to be accessed while the whitelist is the list that can be allowed to access.
2. Keyword blocking. The keywords are specified in a prohibited list. When the keywords are detected in the page, the web access is rejected immediately.

2.3 Word Segmentation

In English, the word segmentation is obvious by using separators such as space and period. The proper noun begins with the capital letter. However, in Thai language, there is no space between words and no separators. The proper noun cannot be noticed by the uppercase characters. It is more difficult to do word segment to extract the keywords. In Thai, to do word segmentation, there are several ways [17].

1. Thai dictionary. This is a typical way. The extracted string is matched with the words in the dictionary. The accuracy of the segmentation depends on the dictionary selected. Two approaches have been used: longest matching and maximal matching. The longest matching finds the matched words with the longest number of characters in the dictionary. In the maximal matching, the possible ways to create the segments in a sentence are extracted and it will select the way that the minimal number of segmentation.
2. Statistical reference. The approach uses the corpus of phases. The corpus is created for a purpose of the application area.

2.4 Stop Word Removal

This step is the remove unimportant words that are not useful in the context. In English, it would be articles, prepositions, adverbs, exclamations, pronouns, etc. In Thai, also, the stop words are in the same manner. However, it is more difficult to detect due to the nature of the segmentation mentioned previously. Again, they need to be removed as much as possible before the keyword extraction and indexing process to reduce errors in the latter step.

2.5 Measurement

In the work, we use precision and recall to measure the accuracy of the model. Given True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN),

Precision is defined as

$$\frac{TP}{(TP+FP)}$$

Recall is defined as

$$\frac{TP}{(TP+FN)}$$

3. METHODOLOGY AND DESIGN

The work uses the data mining technique to create a model for filtering. In this section, first, we present the step to create the model. Then, we present the system architecture and flow diagram.

3.1 Spam Checking and Data Mining

Given the URL, it is searched in the blacklist database. If not found, the post text will be used next. The words are extracted from the post. The keywords are checked against the keyword database (which is extracted from 100 training posts). The ratio of the keywords and all the words in the post is calculated. If it is less than 0.06. This ratio is based on the inspection of the training words in 100 training posts in Figure 1. We then feed the post words into the model to classify it as a spam or not.

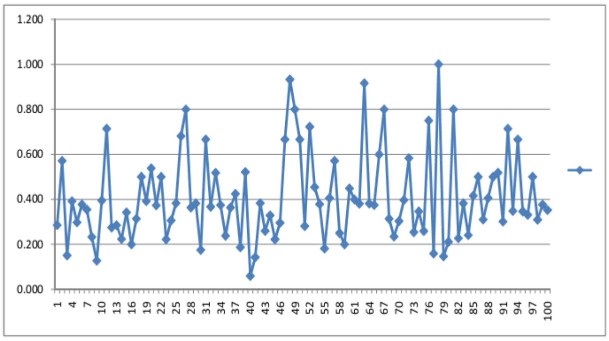


Fig. 1 : The ratio of the keywords and the total words in the sample posts.

There are preprocessing needed after extracting the post text. Swath [17] is used to chop words in Thai with the help of separators and dictionary. After that, the stopping words are eliminated to maintain only words that should be useful for detection.

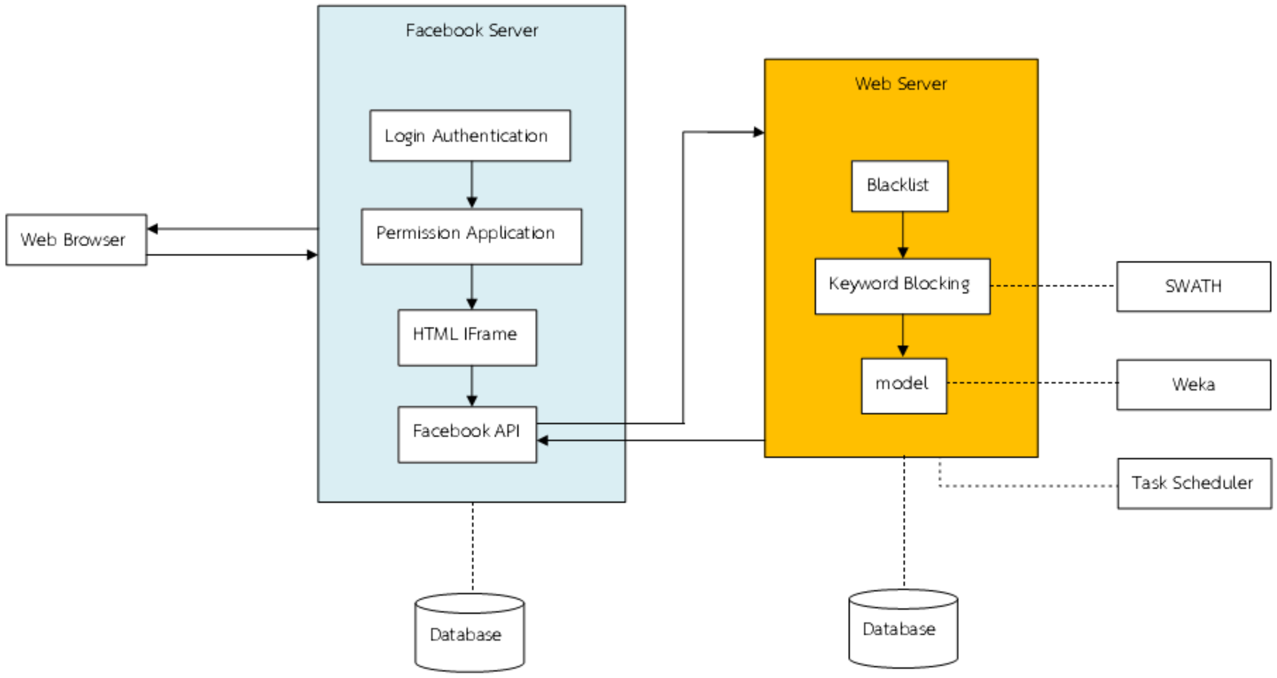


Fig. 2: The system architecture.

To feed into the model, first, the number of links is counted. Second, the number of words is counted and next, the length of the post after removing the stop words is calculated. At last, the average number of words in the post and the number of keywords are counted.

3.2 System Architecture

Figure 2 presents the architecture of the whole application. We denote Facebook server is between the user and application. It requires the login authentication for using Facebook. The permission application is the part where the user privileges are specified. HTML frame is the one which extracts the page to be displayed on Facebook application. Facebook API is the collection of functions which make our pages connect to Facebook modules.

Web server is our part which performs user information checking. Particularly, it performs the spam checking. It contains the blacklist, keyword blocking process, and our data mining model. The keyword blocking process needs *swath* [17] to segment words from the text in Thai. Weka [1] is used for data mining model and task scheduler is the application that sends the program to run periodically.

3.3 Sequence diagram

Figure 3 shows the interaction of the whole components. The user accesses the page via web browser. Our work is the run as an application on Facebook which needs the user to login and allow the permission to access their information. After the user gives the permission, IFrame extracts the user page from Canvas URL. Next, the web server calls the Facebook API to access the user data. The data are filtered using blacklist and keyword blocking where the database of the list are accessed. The keyword blocking uses *swath* to the segmentation and then the words are checked. Also, in the next step, the data mining model is invoked in the server if the spam is not found in the both previous step. If the spam is detected by the model, the blacklist may be updated. If the spam is found, the spammer list is created. Task scheduler invokes the checking of the user data periodically.

4. EXPERIMENTS AND APPLICATION

For testing the model, we divide the data into 2 sets. There are 150 posts for training which contains 75 normal posts and 75 spam posts. The testing set is 100 posts which contains 50 normal posts and 50 spam posts. The training data is fed into J48 model in Weka. Figure 4 shows the results after training.

From the tree, on the leaves, YES node is the spam post and NO node is the normal post.

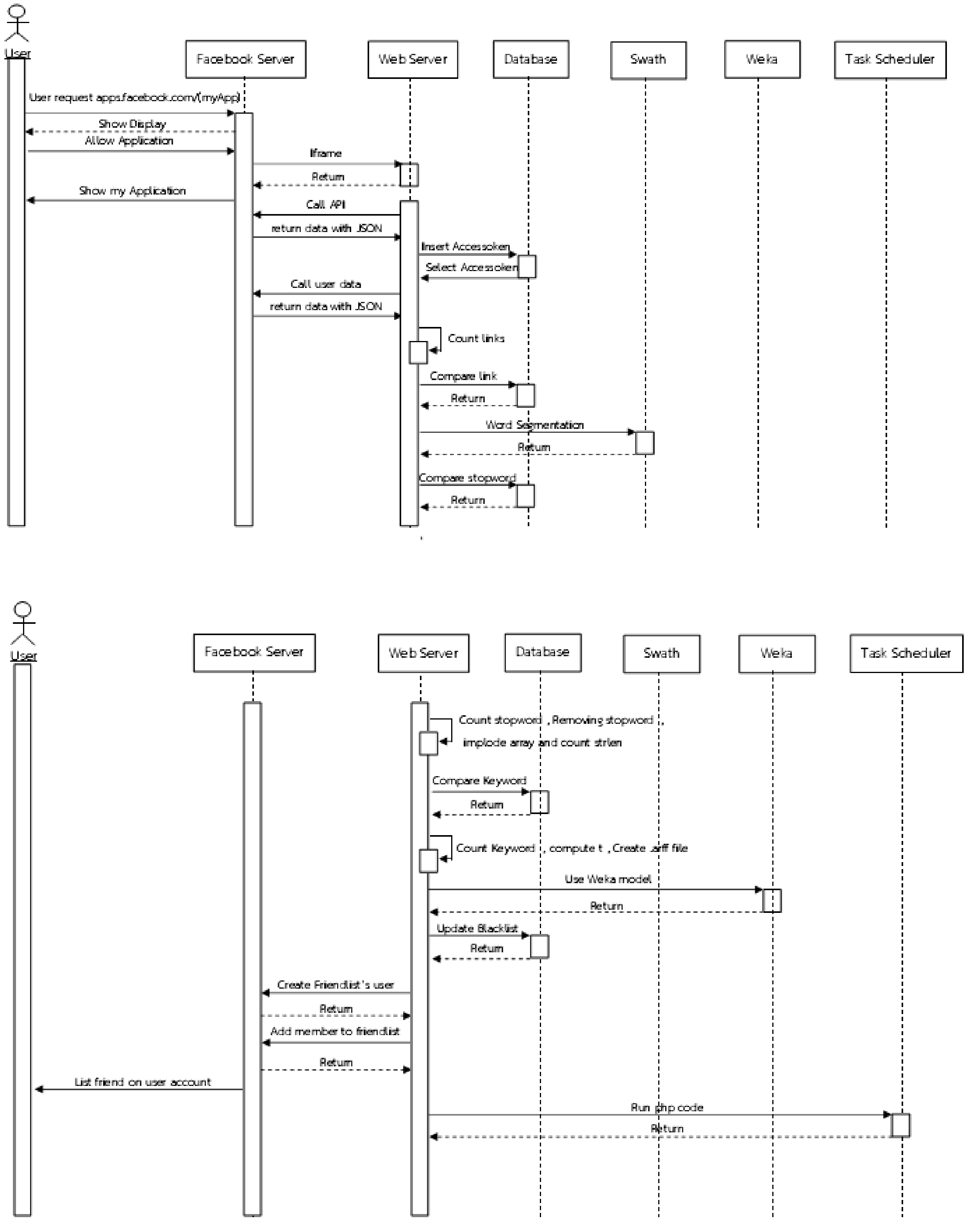
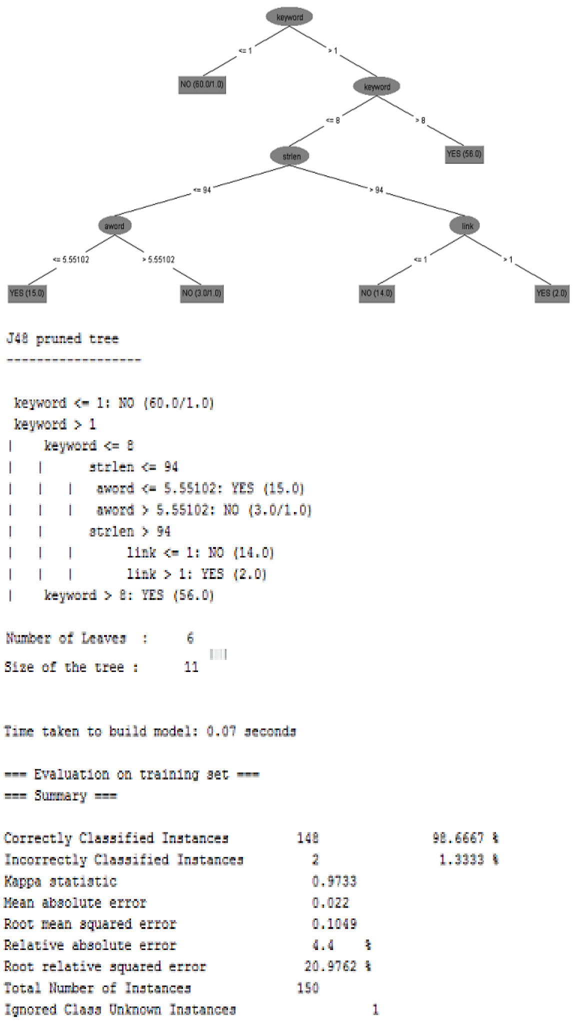


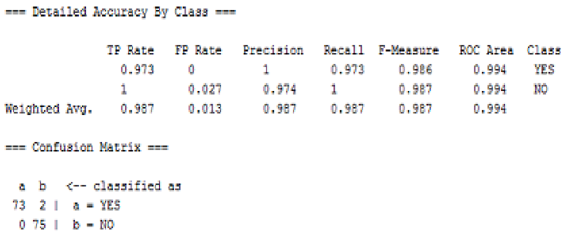
Fig. 3: Sequence diagram.



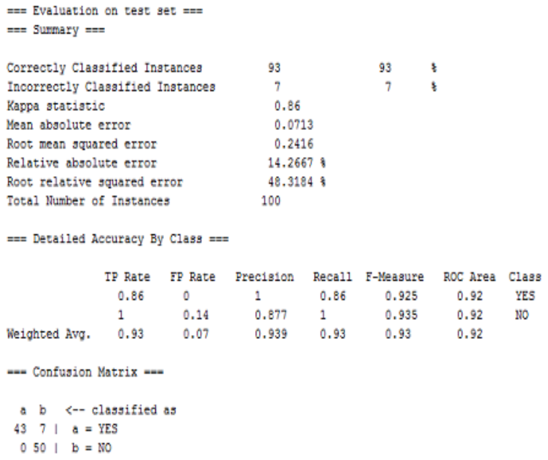
```
J48 pruned tree
------------------

 keyword <= 1: NO (60.0/1.0)
 keyword > 1
|    keyword <= 8
|   |      strlen <= 94
|   |   |   aword <= 5.55102: YES (15.0)
|   |   |   aword > 5.55102: NO (3.0/1.0)
|   |      strlen > 94
|   |   |       link <= 1: NO (14.0)
|   |   |       link > 1: YES (2.0)
|    keyword > 8: YES (56.0)

Number of Leaves  :     6

Size of the tree :      11


Time taken to build model: 0.07 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         148               98.6667 %
Incorrectly Classified Instances         2                1.3333 %
Kappa statistic                          0.9733
Mean absolute error                      0.022
Root mean squared error                  0.1049
Relative absolute error                  4.4     %
Root relative squared error             20.9762 %
Total Number of Instances              150
Ignored Class Unknown Instances                  1
```

Fig. 4: Decision tree.

For the testing data, we obtain the results as in Figure 5.



```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.973     0          1         0.973     0.986      0.994    YES
                 1         0.027      0.974     1         0.987      0.994    NO
Weighted Avg.    0.987     0.013      0.987     0.987     0.987      0.994

=== Confusion Matrix ===

  a  b   <-- classified as
 73  2 |  a = YES
  0 75 |  b = NO
```



```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          93               93      %
Incorrectly Classified Instances         7                7      %
Kappa statistic                          0.86
Mean absolute error                      0.0713
Root mean squared error                  0.2416
Relative absolute error                 14.2667 %
Root relative squared error             48.3184 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.86      0          1         0.86      0.925      0.92     YES
                 1         0.14       0.877     1         0.935      0.92     NO
Weighted Avg.    0.93      0.07       0.939     0.93      0.93       0.92

=== Confusion Matrix ===

  a  b   <-- classified as
 43  7 |  a = YES
  0 50 |  b = NO
```

Fig. 5: Results of testing data.

Table 1 summarizes the results. The training gives the higher precision and recall rates than that of the testing data.

Table 1: Accuracy results.

Data set	Precision	Recall	F-measure
Training	100%	86%	92%
Testing	61%	66%	63%

Figure 6 shows the application example. In Figure 7, the user needs to allow the application to access Facebook data. After that the application is running as a background process while pages are extracted from Facebook.

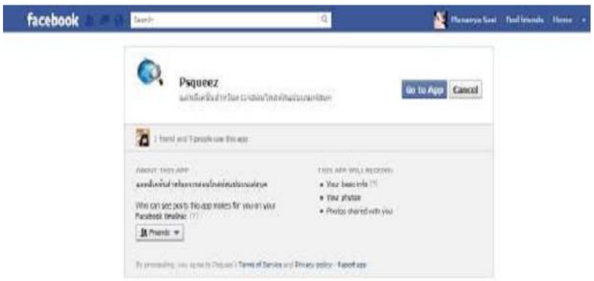
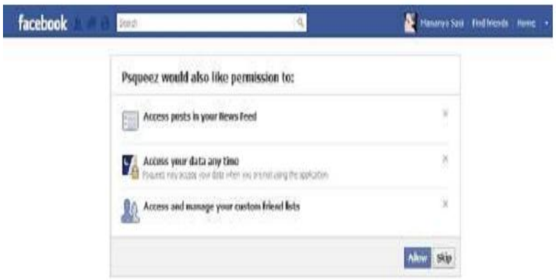


Fig. 6: Application interface.



Psqueez would also like permission to:

Access posts in your News Feed

Access your data any time

Access and manage your custom friend lists

Allow Skip

Fig. 7: Allowing the application to access user data.

Figure 8 shows the detection of the spam. Once detected, the program puts the friends who gave the post in the list called SqueezeApp.

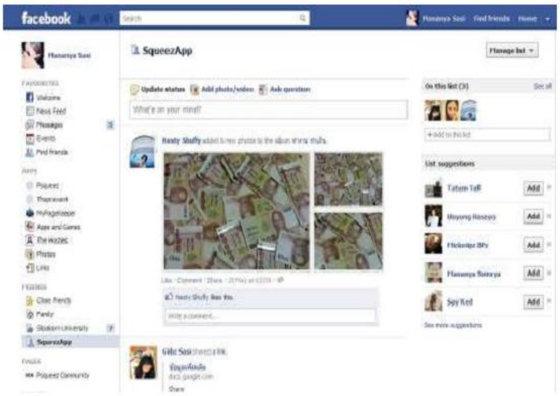


Fig. 8: Allowing the application to access user data.

5. CONCLUSION

The paper demonstrates the Facebook application using data mining to detect the spam. For the spam detection, the blacklist, keyword blocking are applied first. Then, the data mining model is used to detect spams further. The features used in the process is the number of links, the number of words, the length of posts etc. The precision and recall rates achieved is around 61%-63% due to the small training sets. This can be improved if more training data are available. The model is included in the Facebook application. The system integration is demonstrated where the user page can be redirected to the web server for spam checking first.

6. REFERENCES

[1] "Weka 3: Data Mining Software in Java," University of Waikato, [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed 25 June 2012].

[2] X. Amatriain, A. Jaimes, N. Oliver and J. Pujol, "Data Mining Methods for Recommender Systems," in *Recommender Systems Handbook*, F. Ricci, Ed., Springer Science+Business Media, 2011, pp. 39-71.

[3] Brian, "Five main methods of detecting patterns in data mining," [Online]. Available: http://legallysociable.com/2012/04/05/five-main-methods-of-detecting-patterns-in-data-mining/. [Accessed 30 June 2012].

[4] P. Hayati and V. Potdar, "Evaluation of spam detection and prevention frameworks for email and image spam: a state of art," in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services (iiWAS)*, 2008.

[5] "Mail-SeCure Image-Based Spam Treatment," 2009.

[6] Y. Sawaya and Y. Miyake, "A Study of Spam Mail Detection System before. Receiving the Message Body," in *Joint Workshop on Information Security Cryptography and Information Security Conference System*, 2009.

[7] S. Krasser, Y. Tang, J. Gould, D. Alperovitch, P. Judge and S. Krasser, "Identifying Image Spam based on Header and File Properties using C4.5 Decision Trees and Support Vector Machine Learning," in *IAW 2007*, 2007.

[8] Z. Wang, W. Josephson and Q. L. M. Charikar, "Filtering Image Spam with Near-Duplicate Detection," in *CEAS*, 2007.

[9] N. Spirin and J. Han, "Survey on Web Spam Detection:Principles and Algorithms," 2012.

[10] A. Benczúr, I. Bíró, K. Csalogány and T. Sarlós, "Web spam detection via commercial intent analysis," Alberta, Canada,,, 2007.

[11] L. Becchetti, C. Castillo, D. Donato and R. a. L. S. Baeza-YATES, "Link analysis for Web spam detection," *ACM Transactions on the Web (TWEB)*, vol. 2, no. 1, February 2008.

[12] K. Suttirut and C. Phongpensri, "Improper Web Access Protection Technique Based on Proxy Cache Server," in *National Conference on Computer Science and Engineering*, Bangkok, Thailand, 2007.

[13] X. Jin, C. X. Lin, J. Luo and J. Han, "SocialSpamGuard:A Data Mining-Based Spam Detection System for Socia lMedia Networks," 2011.

[14] M. Bosma, E. Meij and W. Weerkamp, "A Framework For Unsupervised Spam Detection In Social Networking Sites," 2012.

[15] B. Markines, C. Cattuto and F. Menczer, "Social Spam Detection," in *AIRWeb*, 2009.

[16] J. Pei, B. Zhou, Z. Tang and D. Huang, *Data Mining Techniques for Spam Detection.*

[17] P. Charoenpornsawat, "Software: SWATH - Thai Word Segmentation," [Online]. Available: http://www.cs.cmu.edu/~paisarn/software.html. [Accessed 10 July 2012].

[18] J. W. Seifert, "Data Mining: An Overview," Congressional Research Service, 2004.