Towards a High-level and Re-targetable Toolkit for Social Media Mining

Jinjing Ma
Deparment of Computer Science
University of North Carolina at Chapel Hill
jinjingm@cs.unc.edu

Abstract

The research of social-media mining focuses on using the appropriate features extracted from social-media data-sources, and getting desired social inferences. However, the iteration of choosing appropriate features and algorithms can lead to tedious programming work, even when machine-learning toolkits are employed. This is because existing toolkits provide general and hence low-level data formats, thus translation of original social media data is needed for each data source such as email and Facebook. Moreover, the toolkits offer different sets of functions, which can require the translation task for each data source to be performed for multiple toolkits. Furthermore, they do not directly support common social-media derived-features, which are computed using machine learning algorithms from more basic features. In this paper we present a high-level and re-targetable toolkit focusing on response correlation and prediction area. This toolkit named SoMMinT is capable of not only retargeting data source and low-level toolkit but also layering derived-feature extraction. When implemented with SoMMinT, two case studies of email response time prediction and YahooAnswers answer quality correlation share about 64% code, 63% classes, and 100% interfaces.

1 Introduction

By discovering patterns in large data sets, data mining can extract meaningful interior relations in original unprocessed data. As a subfield of data mining, social media mining can be described in a three-dimensional design space constructed from source data, features extracted, and desired inferences. In this area, the source data consists of the information generated by online users' communications, including those with general recipients on the community-type sites such as Facebook, Twitter, Q&A sites, and those with specific recipients as in email. With determined data sources, the features to be processed with mining algorithms are extracted from both users' dynamic interactions (e.g. messaging, posting, reading, "share", and "like") and user's online relationship (e.g. friendship and "following"). A feature can be described as a basic feature that is extracted from source data directly such as the number of responses, or a super feature that is derived from basic features using mining algorithms, such as clustered tags. As the desired outcome of social media mining, social inference is usually quantitative analysis results or prediction models, such as message responsiveness prediction and online community evolution.

Social media mining requires tedious programming work to achieve the best outcome, due to the iterative feature and mining algorithm selection process. The situation is even worse when the iterative process is applied repeatedly by different researchers. For example, J. Arguello's and M. Burke's works on newsgroups[1, 2, 3] are implemented separately. This is particularly frustrating when researchers want to test the repeatability of other's work, use past work as a baseline, or apply aspects of other's work on their choice of data source since all the process needs to re-implemented.

A solution to redundant programming is to use toolkits to abstract common parts among studies, then toolkits can be employed for mining algorithms. However, existing machine learning toolkits are low-level and offer different sets of function. We take two examples of Weka and Mallet to illustrate. First, Weka and Mallet both have distinct input formats, so translation of social-media data is required each time for different data sources. Second, Weka offers algorithms for mining data with multi-type fields, but it does not mine textual data very well. On the contrary, Mallet excels in textual mining, by offering topic modeling, but it is incompatible with multi-type-field data. So neither Weka nor Mallet can serve all needs of social media mining alone since social media data is diverse. Thus, the researchers would have to combine or switch between toolkits for optimized outcomes. But this remains a problem because of the distinct data format requirements. It means converting one toolkit's output to fit another toolkit's input format is necessary.

Since researchers would not have enough time or energy to try all possible alternatives, the unpleasant programming process might suppress the flexibility of social media mining research in two ways. First, the variety of data sources, features and mining algorithms can be limited, because translating the data format among different source data sets and toolkits is tedious. For example, among all existing response research mentioned above, only Y. Wang, M. Burke, and R.E. Kraut's work on Facebook status data[4] have applied topic modeling to discover the contents' topics. Other works only took predefined topic into account. Second, the tedious programming process can reject the layering of features/mining algorithms: some features currently used are derived from original data directly. Layered features can be helpful, and we take an ongoing response prediction study on StackOverflow for example. In StackOverflow, each question is assigned with several tags indicating the topic. However, the number of tags is enormous, and the semantics of tags often overlap. Thus, semantic tag groups extracted from original tags would be a better feature to use.

Therefore, a high-level, re-targetable toolkit is needed to integrate general social media mining research process. In this paper, we present a toolkit named SoMMinT allowing data sources switching, multiple low-level toolkits retargeting, and feature layering. Currently, we are focusing on response time/quality research, including correlation and prediction. In the following sections, we will use two cases, email response time prediction and YahooAnswers answer quality correlation, for motivating, illustrating, and evaluating various aspects of SoMMinT.

In Section 2, we introduce related research efforts on which SoMMinT is built. These include general social media mining research, two case studies, and existing machine learning toolkits. The main features of the toolkit are presented in Section 3. Section 4 evaluates the performance of the toolkit applied in the cases from Section 2. Section 5 summarizes the contribution and future steps of our work.

2 Related Work

2.1 General social media mining research

Social media mining has various sub-fields depending on the desired social inferences, including users' interests, response time/quality of messaging, and other aspects. For example: L. Backstrom and J. Leskovec's work predicts and recommends friendship on Facebook based on supervised random walks [5]; M. Maia's team clustered users sharing behavioral pattern based on user interactions [6]; R.W. White's team combined interests of those who have visited the same pages [7]; and research on Flickr provided individual content recommendation and personalized search based on the social correlations among its users [9, 8].

Among all the sub-fields of social media mining, we focus on response studies. Current response studies mainly focus on the correlation of conversation features and response measurements (quantity, quality, and speed). There are several important examples: J. Teevan's team analyzed the factors of the three measurements of response on Facebook status data with topic controlled [10]; Another research performed on Facebook status focused the effect of gender and modeled topic on response number [4]; J. Arguello's and M. Burke's research on newsgroups analyzed topic and rhetoric factors' correlation with responses [1, 2, 3]; and Y. M. Kalman and S. Rafaeli's study profiled email response time [11].

While social media mining has two types of desired inferences: correlation and prediction, the published response research is only about correlation. The gap in the response-prediction field is being filled by my research group in University of North Carolina at Chapel Hill (UNC-CH) with the work in email and StackOverflow response time prediction. In the following part we will use two cases from UNC ongoing and published research to illustrate response prediction and correlation respectively.

2.2 Case Study

2.2.1 Case 1: Email response prediction

This case is derived from the ongoing work of the UNC-CH group; some researchers in our group are exploring prediction model for email response time. Temporally we are focusing on predicting response existence and the first response time based on the initiating message[1]. Figure 1 shows the current research process.

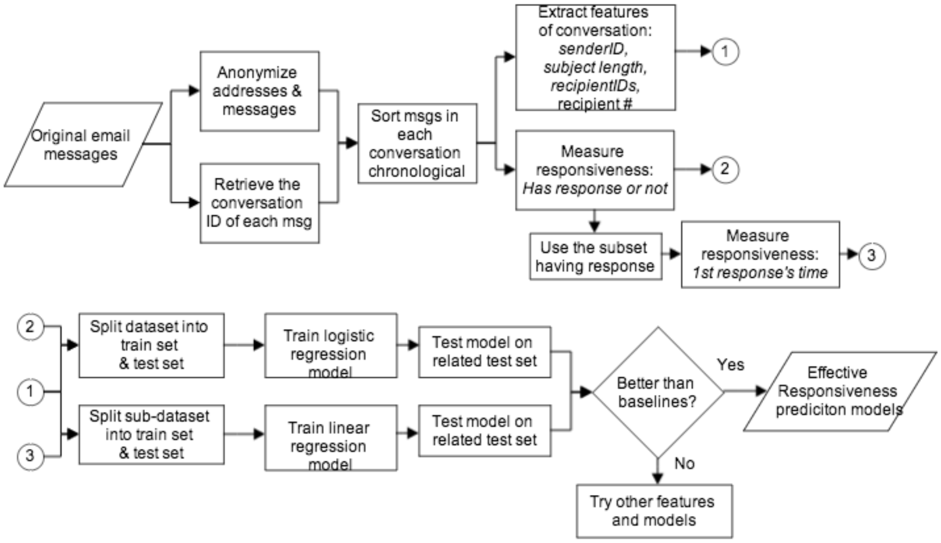
Original email messages
Anonymize addresses & messages
Retrieve the conversation ID of each msg
Sort msgs in each conversation chronological
Extract features of conversation: senderID, subject length, recipientIDs, recipient #
1
Measure responsiveness: Has response or not
2
Use the subset having response
Measure responsiveness: 1st response's time
3
2
1
3
Split dataset into train set & test set
Split sub-dataset into train set & test set
Train logistic regression model
Train linear regression model
Test model on related test set
Test model on related test set
Better than baselines?
Yes
Effective Responsiveness prediciton models
No
Try other features and models

Figure 1: Email response time prediction process

[1] The work can be found at https://bitbucket.org/jbartel/recipientprediction/

The source data is generated by users' communication through email. After anonymizing messages/users and reconstructing conversations, conversation features and response time measurements are extracted as input of mining algorithms. Then, Logistic regression model for predicting response existence and linear regression for first response time are trained separately. If the trained models have satisfactory performance on test set, we then have the expected outcome of effective email response prediction models. Otherwise, we would iteratively try alternative features, measurements, and models.

2.2.2 Case 2: YahooAnswers answer quality correlation

This case is derived from F.M. Harper and his colleagues' work of analyzing factors affecting answer quality across common Q&A sites, including Google Answers, Library Reference, AllExperts, YahooAnswers, and Live QnA [12]. Here we only use the part of YahooAnswers for case study with process, which is illustrated in Figure 2.

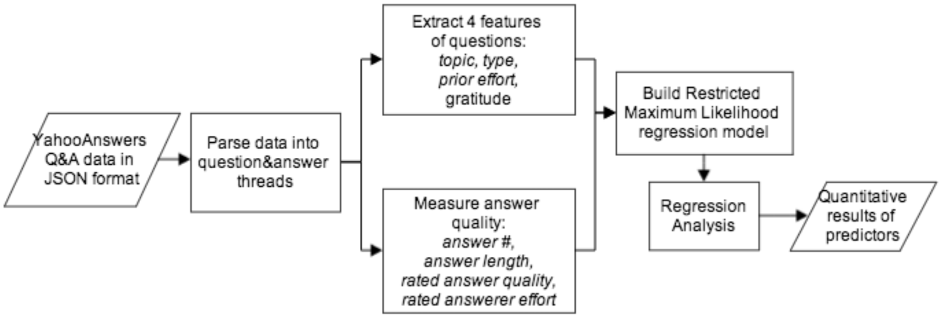
YahooAnswers Q&A data in JSON format
Parse data into question&answer threads
Extract 4 features of questions: *topic, type, prior effort,* gratitude
Measure answer quality: *answer #, answer length, rated answer quality, rated answerer effort*
Build Restricted Maximum Likelihood regression model
Regression Analysis
Quantitative results of predictors

Figure 2: Process of YahooAnswers answer quality factoring

In this case, user online Q&A interactions generate the source data. Then it is parsed into questioning and answering threads (conversations), extracted from which are question features and answer quality measurements. Then, a restricted maximum likelihood regression model is built to analyze the correlation between features and measurements to find outstanding factors of answer quality.

These cases do not only illustrate the process of correlation and prediction research, but also demonstrate that certain tasks are repeated in different efforts. As we mentioned in the introduction section, redundant programming can be reduced by using machine learning toolkits that abstract common parts of mining algorithms. Below we discuss existing toolkits and how they can be applied to both general social media mining and the two cases.

2.3 Existing Machine Learning Toolkits

Existing machine-learning toolkits can be used in the process of social media mining as shown in Figure 3. These low-level toolkits take features extracted from social media data as independent variables and response measurements as dependent variables, then perform mining algorithms on researchers' choice. Based on the evaluation on chosen models also offered by these toolkits, researchers can decide whether alternative features and algorithms are needed. In this part, we introduce two widely used toolkits: Weka and Mallet.

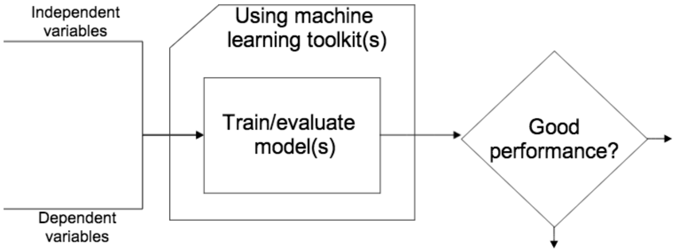

Figure 3: Application of existing machine learning toolkits

2.3.1 Weka

Weka is a Java-based package supporting common machine learning algorithms on numeric, nominal, date, and string data fields[13]. Thus, it fits many requirements of social media mining research, including response prediction/analysis. In the cases introduced in Section 2.2, both nominal (email sender ID and recipient IDs, all four question features) and numeric (email subject length and recipient number) features are used.

However, Weka has particular formatted input, called ARFF or XRFF for sparse data matrix. This would require the researchers to convert and save their data in required format, which would cost them extra time to learn details about the particular format and API. Moreover, though Weka supports textual data by using filters to map words to vectors, it doesn't support natural language processing very well because of apostrophe conflict exist between the text and ARFF formatting and lack of topic modeling algorithms mining semantics of text.

2.3.2 Mallet

Mallet is a Java-based toolkit specifically for mining text, including document classification, clustering, topic modeling, and statistical natural language processing [14]. Mallet supports input as original text files or a formatted single file.

Among the text mining algorithms, topic modeling is the most common used in social media mining. Topic modeling is a type of statistical process for detecting the abstract ”topics” in a set of documents. For example, word ”sunshine”, ”bikini”, ”sea”, and ”swim” would be categorized with ”nature”, ”clothing”, and ”sports” separately, but they might be considered together as an abstract topic ”beach” in certain context. Topic modeling aims at finding such inner correlation on abstract topics among documents.

An approach of modeling abstract topics is Latent Dirichlet allocation (LDA). LDA can be regarded as a technique of dimensionality reduction. It is a hierarchical Bayesian model converting each document into a probability vector over a collection of abstract topics, where each topic is represented by a set of words and their frequency subjecting Dirichlet distribution [15]. In our case studies, for example, LDA could have been used to mine topics in emails and check if a question' topic matches its category in YahooAnswers data.

Although Mallet excels Weka in text mining, it only supports textual data. Considering the variety of data field type of social media data, like answer number and recipient IDs mentioned above, neither of the toolkits applied alone can achieve the best possible outcome for social media mining.

2.4 Discussion

Machine learning toolkits can be applied in the two cases for model training and evaluating part as shown in Figure 3. However, both of the two cases require tedious programming work of the iterative feature and mining algorithm selection process, with the input-formatting limitation of existing toolkits mentioned above. While the process for even a single case is tedious, the problem exacerbates when different researchers repeat it. For examples, the redundant translation work on the same data source of newsgroups is implemented separately by researchers [1, 2]. Another example is that the same response measurement, response number, is repeatedly extracted among several response research from Section 2.1.

Moreover, although existing toolkits have complementary function sets, no previous response effort has used multiple toolkits. This might be because each low-level requires a unique input format, which causes extra translation tasks.

Higher-level abstraction of common parts among different research efforts can reduce the redundancy by encouraging more code sharing. The abstraction of the two cases is shown in Figure 4. First, the original data of messages are sorted into conversations, from which features and response measurements are extracted to be input of some mining algorithms. Depending on the desired inference (correlation or prediction), the models are trained and evaluated on corresponding sub-dataset (the whole data set, or train/test sets). If the trained model does not produce a satisfactory outcome of prediction model or quantitative analysis, alternative features/measurements and algorithms would be tried iteratively.

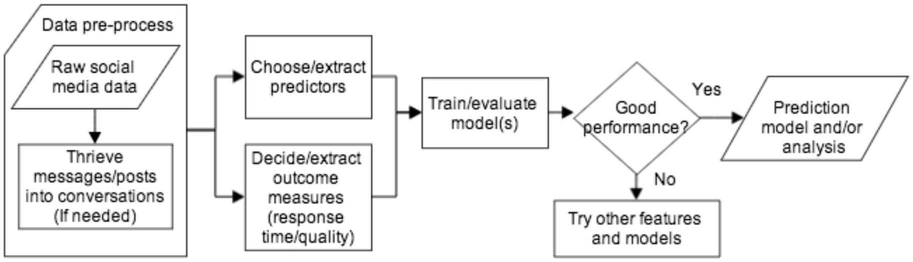
Data pre-process
Raw social media data
Thrieve messages/posts into conversations (If needed)
Choose/extract predictors
Decide/extract outcome measures (response time/quality)
Train/evaluate model(s)
Good performance?
Yes
No
Prediction model and/or analysis
Try other features and models

Figure 4: Common process of response prediction/analysis

Based on the abstraction in Figure 4, a high-level, re-targetable toolkit is presented in the next section, allowing data sources interchange and multiple low-level toolkits retargeting.

3 A New Framework

Such a high-level, re-targetable toolkit, SoMMinT, is presented in this section. Our long-term perspective is to build a toolkit integrating general social media mining process to allowing switching data sources and retargeting multiple low-level toolkits. But in this paper, we narrowly focus on integrating the two cases above to produce a toolkit for response prediction and correlation. In social media mining design space of social media data source, features, and desired social inferences, we only target data source and features. Figure 5 models the process of using this high-level toolkit's different modules at stages. First, the researchers need to decide desired social inference (response correlation or prediction) and data source, then choose a corresponding parser to preprocess the source data. In each iteration, after choosing features and mining algorithms to be applied, they can choose suitable feature extracting rules. In the loop, the researchers just choose the suitable models (super feature extracting rules in the diagram) to extract features derived from the more basic ones and for prediction/correlation. After the model evaluation module is called, they can decide to repeat this process or accept the outcome.

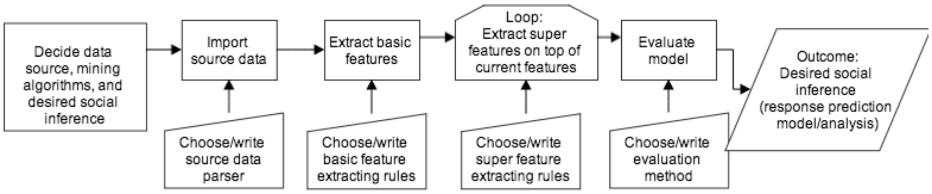
Decide data source, mining algorithms, and desired social inference
Import source data
Extract basic features
Loop: Extract super features on top of current features
Evaluate model
Outcome: Desired social inference (response prediction model/analysis)
Choose/write source data parser
Choose/write basic feature extracting rules
Choose/write super feature extracting rules
Choose/write evaluation method

Figure 5: Process of using this high-level toolkit for response prediction/analysis

In this section, we have shown that this toolkit is more high-level than existing general purpose toolkits. Next, we will explain how data source interchange a low-level toolkit retargeting are realized.

3.1 Data source interchange

As shown in Figure 6.(a), the difficulty of social media data source switching is caused by repeated translation between each pair of source data and toolkit, especially when researchers try to use multiple toolkits. There would be $n * m$ translations, if we have n data sources and m low-level toolkits. This problem can be solved by a uniformed inter-stage data format. In this case there would be $n + m$ translations. The solution is shown in Figure 6.(b): different source data is transformed to the inter-stage data, which would be translated to input format of low-level toolkits.

The assistive data format is called “ThreadDataSet”. A ThreadDataSet is a collection of “ThreadData” representing a conversation. A ThreadData contains a chronological sequence of “MessageData” standing for a message or a post. For email, a ThreadData is an email conversation consisting of email messages. While for YahooAnswers data, a ThreadData is a discussion of a certain question, when the question and each answer becomes a MessageData. Different source data for response research are parsed into the uniform data of ThreadDataSet. Then, a ThreadDataSet is converted to a “IntermediateDataSet”. “IntermediateDataSet” is an interface of wrapped data for certain machine-learning toolkit format. For example, IntermediateDataSet for Weka contains data in Weka format. Figure 6.(b) shows the transformed translation process between source data and low-level toolkits.

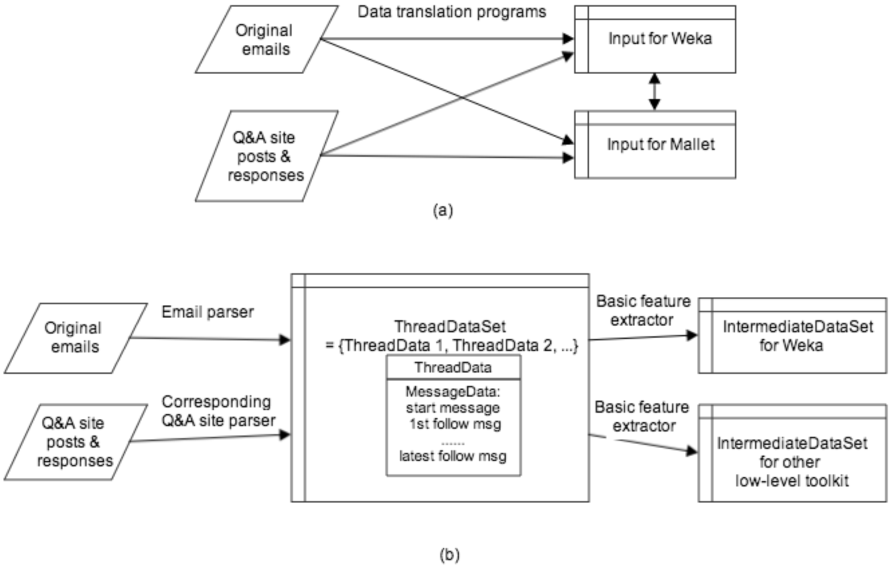
Original emails
Data translation programs
Input for Weka
Q&A site posts & responses
Input for Mallet
(a)
Original emails
Email parser
ThreadDataSet = {ThreadData 1, ThreadData 2, ...}
ThreadData
MessageData: start message 1st follow msg ...... latest follow msg
Basic feature extractor
IntermediateDataSet for Weka
Q&A site posts & responses
Corresponding Q&A site parser
Basic feature extractor
IntermediateDataSet for other low-level toolkit
(b)

Figure 6: Translation between source data and low-level toolkits: (a) without SoMMinT (b) with SoMMinT

Figure 7 shows the detailed ThreadDataSet-to-IntermediateDataSet conversion process. The process is based on a "BasicFeatureExtractor" and a set of "BasicFeatureRule"s. A BasicFeatureRule is a function taking a ThreadData as input, then produces the desired "basic feature" (a feature that can be extracted directly from a conversation, such as the number of responses). The BasicFeatureExtractor organizes a set of BasicFeatureRules, using them to convert a ThreadDataSet to a specified IntermediateDataSet saving all features produced by BasicFeatureRules.

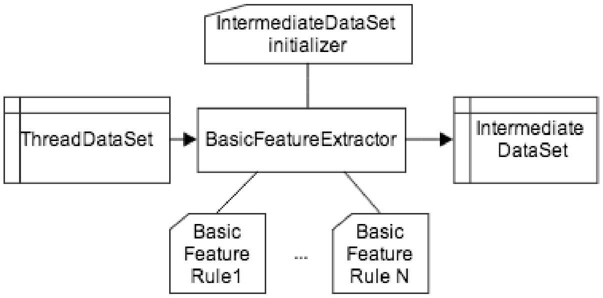

Figure 7: Framework of Basic feature extraction

Since SoMMinT focuses on response prediction/correlation, only instances of conversation are integrated as ThreadDataSet. When other social media mining areas are accommodated in this toolkit, ThreadDataSet and BasicFeatureExtractor might not be suitable. An example is users' friendship evolution prediction. Such link prediction would require a uniform data structure of graph, and we might need a different corresponding basic feature extracting process. In feature extensions, we can switch ThreadDataSet and BasicFeatureExtractor to other appropriate module without affecting other parts of this toolkit.

3.2 Low-level toolkits retargeting

To allow the low-level machine learning toolkits retargeting, we define a intermediate data interface "IntermediateDataSet". On IntermediateDataSet we can extract features that cannot be derived from original data directly, which is defined as "super feature". The extracting process is performed using "SuperFeatureExtractor" by employing a set of "SuperFeatureRule". Similar to BasicFeatureRule, a SuperFeatureRule is also a function consuming an instance in IntermediateDataSet and output a super feature value. But a SuperFeatureRule contains a machine-learning model offered by the low-level toolkits, such as a classifier from Weka or a topic model from Mallet. SuperFeatureRule offers interface to train wrapped models with a given IntermediateDataSet before being employed in the super feature extracting process. The final modeling step is also performed by SuperFueatureExtractor, except in only one SuperFeatureRule trained with response measurement is used. The process is shown in Figure 8.

As shown in Figure 8, low-level machine learning toolkits retargeting is realized by SuperFeatureRule, where each mining algorithm from these toolkits is wrapped. Thus SuperFeatureExtractor can switch or combine the functions of different toolkits simply by calling corresponding SuperFeatureRules.

We have shown above how to solve the problem of data source interchange and low-level machine learning toolkit retargeting. For response prediction and correlation, we have described two cases respectively and a toolkit abstracting their commonalities. In the following part, we will address how the two cases are implemented with the toolkit and evaluate the code sharing that results from the implementation.

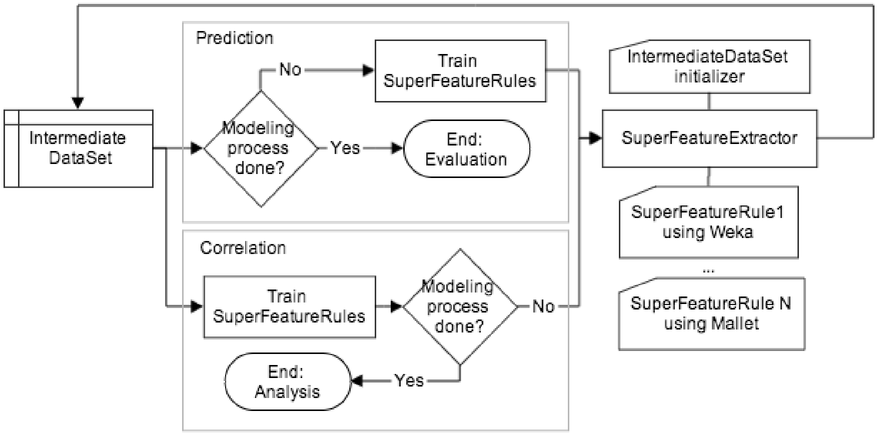

Figure 8: Framework of layered super feature extraction

4 Implementation and Evaluation

Implementation of the two cases with SoMMinT follows their procedures. In Case 1, we apply a toolkit Weka for Logistic regression and linear regression models. In Case 2, the human-judged features and measurements are considered as external files to be integrated into model inputs; the restricted maximum likelihood regression model is simplified as logistic regression model (it is still a maximum likelihood model) because necessary information to build the original model was not provided in their paper.

Table 1 shows that about 64% code, 63% classes, and 100% interfaces used in these cases are shared, and the researchers only need to write 61 lines for YahooAnswers answer quality correlation and 65 lines for email response prediction using provided Java API. Also, it only takes few lines for each case to switch data source and low level toolkit.

	Code line #	Class #	Interface #
Email response time prediction	1106	31	8
YahooAnswers answer quality correlation	1161	30	8
Code shared	725	19	8

Table 1: Performance of the framework on the two cases

The two case studies only employed part of SoMMinT, which offers more general integration with 2644 lines, 82 classes, and 12 interfaces in total. The number of lines is counted by a tool called Metrics.

5 Conclusion and Future Work

This paper presented a high-level and re-targetable toolkit named SoMMinT for social media mining research. SoMMinT integrates multiple conversation-type social media data as well as different machine learning toolkits. Thus, the redundant work of translating is reduced, which is not only between social media data and input formats of low-level toolkits but also among the input formats. Therefore, data source interchange and low-level machine learning toolkits retargeting become simple with this high-level toolkit. When employed on two research of email response time prediction and YahooAnswers answer quality correlation, this toolkit promotes program sharing to 64% code, 63% classes, and 100% interfaces.

Currently SoMMinT builds on the two case studies in this paper. Future work would be performed in the following steps to generalize it. First, more machine learning toolkits would be integrated to cover more mining algorithms such as collaborative filtering. Second, parsers of additional types of data sources, such as Facebook and StackOverflow, would be supported. Then, based on these two steps, more cases of response research process would be implemented with this toolkit to make it more applicable in this area. After that, since SoMMintT is extendible, we would implement different modules like intermediate data set to fit the needs of other areas, such as users' grouping and user interest prediction.

6 Acknowledgement

Thanks to Prasun Dewan and Jay Aikat for the insightful comments, as well as Jacob Bartel and Preeti Arunapuram for detailing their work for case study.

References

[1] Jaime Arguello, Brian S. Butler, Elisabeth Joyce, Robert Kraut, Kimberly S. Ling, Carolyn Rosé, and Xiaoqing Wang. Talk to me: Foundations for successful individual-group interactions in online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 959–968, New York, NY, USA, 2006. ACM.

[2] Moira Burke and Robert Kraut. Mind your p's and q's: When politeness helps and hurts in online communities. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3195–3200, New York, NY, USA, 2008. ACM.

[3] Moira, Elisabeth Joyce, Tackjin Kim, Vivek An, and Robert Kraut. Introductions and requests: Rhetorical strategies that elicit response in online communities. In *C&T '07: Third International Conference on Communities & Technologies 2007, East*, pages 21–40, 2007.

[4] Yi-Chia Wang, Moira Burke, and Robert E. Kraut. Gender, topic, and audience response: An analysis of user-generated content on facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 31–34, New York, NY, USA, 2013. ACM.

[5] Lars Backstrom and Jure Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[6] Marcelo Maia, Jussara Almeida, and Virgílio Almeida. Identifying user behavior in online social networks. In *Proceedings of the 1st Workshop on Social Network Systems*, SocialNets '08, pages 1–6, New York, NY, USA, 2008. ACM.

[7] Ryen W. White, Peter Bailey, and Liwei Chen. Predicting user interests from contextual information. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 363–370, New York, NY, USA, 2009. ACM.

[8] Dongyuan Lu and Qiudan Li. Personalized search on flickr based on searcher's preference prediction. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 81–82, New York, NY, USA, 2011. ACM.

[9] Ralf Schenkel, Tom Crecelius, Mouna Kacimi, Thomas Neumann 0001, Josiane Xavier Parreira, Marc Spaniol, and Gerhard Weikum. Social wisdom for search and recommendation. *IEEE Data Eng. Bull.*, 31(2):40–49, 2008.

[10] Jaime Teevan, Meredith Ringel Morris, and Katrina Panovich. Factors affecting response quantity, quality, and speed for questions asked via social network status messages. In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press, 2011.

[11] Y.M. Kalman and S. Rafaeli. Email chronemics: Unobtrusive profiling of response times. In *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 108b–108b, Jan 2005.

[12] F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. Predictors of answer quality in online q&amp;a sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 865–874, New York, NY, USA, 2008. ACM.

[13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[14] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.