Mining potential ideas in crowdsourcing applications

Thanh-Cong Dinh', Hyerim Bae[1]*, and Joonsoo Bae[2]

Department of Industrial Engineering, Pusan National University, South Korea
[2] Department of Industrial and Information Systems Engineering, Chonbuk National University, South Korea
cong.dinh@pusan.ac.kr, hrbae@pusan.ac.kr, jsbae@chonbuk.ac.kr

Abstract. Crowdsourcing applications require decision makers to deal with extremely large amounts of information in order to isolate the relatively small number of potential ideas. Tools that can facilitate idea classification, therefore, are essential. In this light, the present study developed a methodology that can help decision makers mine ideas having the potential to be used for New Product Development (NPD). Because ideas are textual information, this paper proposes a text mining technique that automatically retrieves data from crowdsourcing applications. That data is then transformed into numerical values that are based on a set of measurements. Finally, an online mode logistic regression (OLR) algorithm is used to predict the probability of being potential of ideas.

Keywords: crowdsourcing, idea mining, text mining, logistic regression

1 Introduction and related work

The term "crowdsourcing" was first used by Jeff Howe to indicate the act of outsourcing a task traditionally performed by an employee or contractor to a large undefined group of people (i.e., a crowd) [1]. Crowdsourcing applications (e.g. Dell's IdeaStorm [2], Starbuck's MyStarbucksIdea [3], etc.) encourage individuals to generate ideas for New Product development (NPD). However, there are drawbacks associated with crowdsourcing, such as participants' lack of expertise in the subject area and the correspondingly poor quality of solutions. Therefore, before even considering new and interesting ideas, firms have to carefully screen them. And unfortunately, the great mass of unstructured data (e.g. textual information) generated by such applications is very difficult to screen or evaluate, not to mention access. This fact is reflected in the relative paucity of case studies on, for example, Dell's IdeaStorm [4-6]. Studies on technical idea mining [7, 8] and new product idea screening [9, 10] have been published, but these have not considered the crowdsourcing environment. The motivation of the present study was to formulate a new method for mining potential NPD ideas in crowdsourcing applications. For this purpose, we developed a novel, text-mining- and prediction-model-based approach to save decision makers time in screening masses of ideas and help them determine

* Corresponding author

which ideas are potentially implementable for NPD. Our work has three contributions. First, a combined text mining/computational linguistics technique [11] for extraction of useful data from crowdsourcing applications; second, a set of measurements for evaluation of the ideas contained in that data; third, an online mode logistic regression (OLR) algorithm for prediction of the probability of being potential of ideas in consideration of the above set of measurements.

This paper is organized as follows: Section 2 introduces our methodology, and Section 3 discusses the pertinent experimentation and results; lastly, Section 4 draws conclusions and anticipates future work.

2 Methodology for mining ideas

Fig. 1 schematizes the proposed three-step procedure for mining of ideas from crowdsourcing applications. In the first step, a text mining method extracts important terms from textual data, which terms, in the second step, are evaluated by numerical methods. The purpose of this process is to represent a given idea with a set of measurements instead of with textual information. To this end, the following two main modules are proposed.

- Terms Evaluation (TE) calculates the Relevance scores for important terms. First, the text (e.g. title and body) of a target idea is retrieved from its web page. It is then examined to search for Request Terms (RTs) are the request phrases employed by users. RTs can concern accessories for specific devices, new products, or a recent-technology options. Known Terms (KTs) also are extracted from the idea text, and are defined as phrases that are familiar to the firm (e.g. product codes, product types, etc.).
- Interest Evaluation (IE) measures crowd interest in ideas. First, crowd information on a target idea is retrieved from the relevant web page. Then, five measurements of interest — Vote, User type, Diversity, Concern and Expert's interest — are calculated.

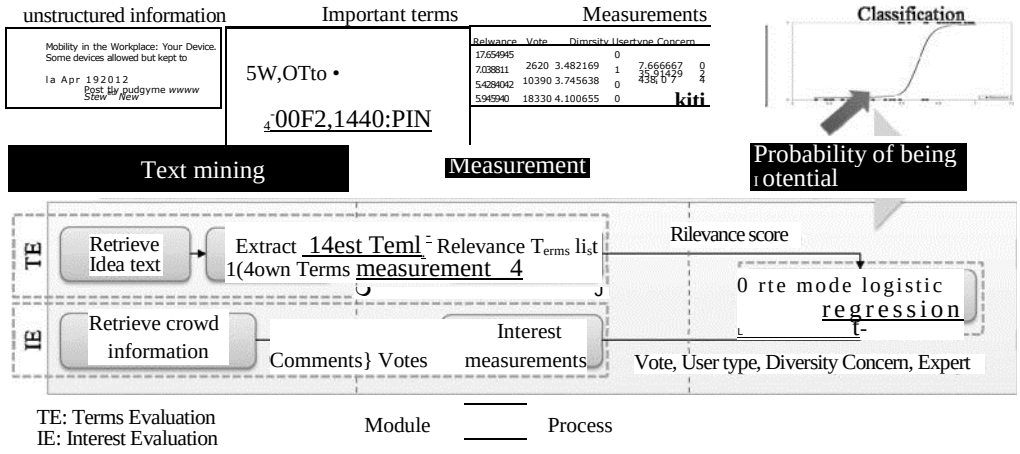


Fig. 1. Overall procedure of the proposed approach

Finally, an online mode logistic regression (OLR) algorithm is applied to calculate the probability of being potential in consideration of both relevance and interest measurements. Our proposed measurements are explained as follows.

2.1 Relevance measurement

In order to reflect the interest level of an RT, we define a function tr(RT) representing its weight value for a recent trend. To retrieve *tr(RT),* a scoring method based on Google Insight for Search [12] is proposed. According to Huang *et al.* [6], when the scope of change of an idea is relatively low it potentially can be implemented. Here, for a KT, we define a function sp(KT) to represent the scope of change. It is to be noted that the weight value reflecting the scope of change of a KT is pre-defined by decision makers. Additionally, well-balanced measurement *b(V)* was adapted from Thorleuchter [7].

Definition 1 (Relevance score): Let *rel(I)* be the relevance score of an idea *I* with terms list *V(a,* A, where *a* and g are RTs and KTs, respectively. Accordingly, we have

$$rel(I) = \sum_{k=1}^{lal} tr(a_k) \times \sum_{k=1}^{(} E \times \text{in } (1.72 + b(V)) \qquad \textbf{(1)}$$

2.2 Interest measurements

In this section, for an idea I, we define five measurements of crowd interest, as listed in Table **1.**

Table 1. Five measurements of crowd intereset in idea *I*

Def.	Description	Function
Vote	Return the number of users interested in idea *I*	vote(1)
User type [5]	Return the binary value where value 1 indicate s the user is a serial user; otherwise, common users receive value 0.	type(1)
Diversity	Return the degree of diversity of users who co mmented on idea *I,* where *n* is the number of d istinct users who commented.	$div(I) = \begin{vmatrix} -E\, p1 \log_2 p_i & n \\ -1 & n<1 \end{vmatrix}$
Concern	Return the crowd's level of interest in idea *I* o ver time. Suppose that idea *I* receives its first c omment on date *fd,* its last comment on date *e d,* and there are *n* comments.	$con(I) = \begin{matrix} (ed - fd)I(n - 1) & n > 1 \\ -1 & n\_<1 \end{matrix}$
Expert's i nterest	Return the number of comments from Dell's e xperts.	epr(1)

2.3 Probability of being potential

We have already presented, herein, methods for measuring the relevance of and interest in an idea. The final task is to estimate an idea's probability of being potential. Utilized for this purpose is a binary classifier with which the value 1 represents "Potential," and the value 0, "Not potential". Logistic regression is a well-known technique to do this task. We employed the stochastic gradient descent (SGD) rule which was introduced in [13]. However, because SGD was designed for batch learning, we extended its applicability to online mode learning by combining it with the weighted majority mechanism. Fig. 2 illustrates our proposed OLR.

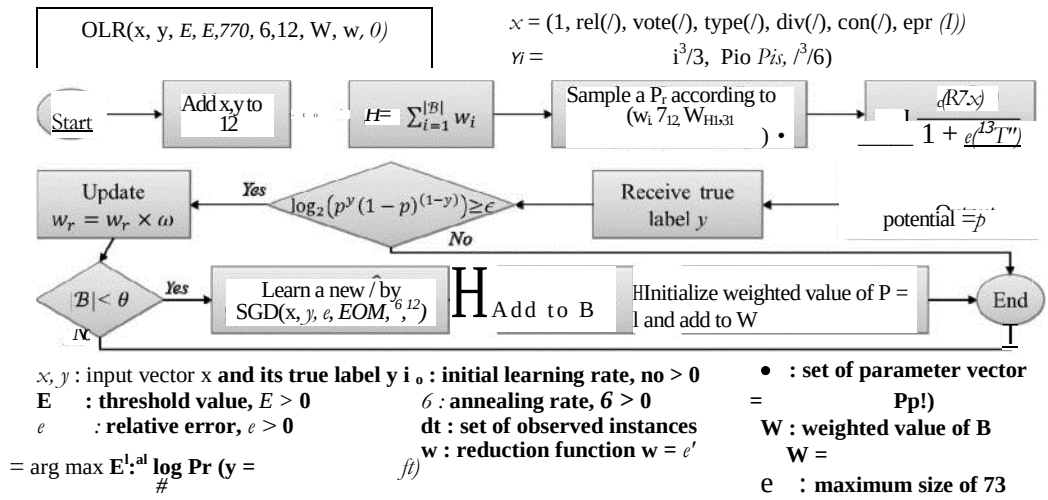
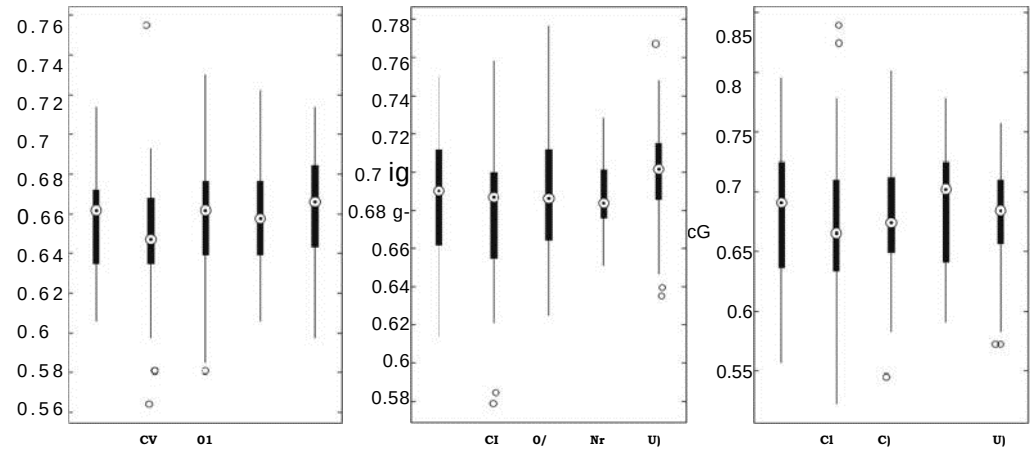


Fig. 2. Proposed OLR algorithm

3 Experiments

For our experimentation, 242 ideas relating to new products were collected from Dell's IdeaStorm. Seventy-seven (77) ideas were marked "Implemented". 48 "Partially Implemented", 7 "Under Review", 29 "Not Planned", and 81 "Archived". Note that in this study, the "Implemented", "Partially Implemented" and "Under Review" ideas were classified as "Potential" and the others as "Not potential". Validation of the proposed text mining method was performed by means of a survey. Over 50% of the observations were found to agree with the results of the RT and KT extractions, whereas around 19.7% did not agree.

Next, we varied the threshold value *e,* parameter a in the reduction function *co,* and *e* which is the maximum size of the set of parameters in order to determine how the result of the OLR changes depends on those values. In all, we had 45 combination sets to examine, and for each set we ran 30 trials. Each trial consisted of 241 idea instances (one instance being sampled randomly to train the first parameter vector). Fig. 3 illustrates results for the best value combinations.



Set No. (1) = 0.5, a = 2, = 80 (2) e = 0.5, a = 3, *8*= 30 *(3)c* = 0.7, a = 2, *0*= 80 **(4) = 1,** a = 2, 8= 80 **(5) e =** 1, a = 3, /9 = 80

Fig. 3. Box plot of accuracy, precision and recall of proposed prediction model

The performance of the proposed OLR algorithm was compared with traditional logistic regression (TLR) and the well-known support vector machine (SVM) algorithm (see the Table 2 results). For TLR and SVM, we used RapidMiner for cross-validation purposes. For OLR, we chose the results from an experiment on set number 5 *(E= 1, a*= 3, *0* = 80).

Table 2. Comparison of TLR, SVM and OLR

Method	Learning type	Accuracy	Precision	Recall
TLR	Batch	0.7410.16	0.7210.00	0.7410.27
SVM	Batch	0.6910.16	0.7110.00	0.5610.26
OLR	Online mode	0.6610.03	0.7010.03	0.6810.05

As the Table 2 data indicates, even though the ORL performance was inferior to SVM and TLR, it was acceptable. As SVM and TLR are batch-learning algorithms, they perform well with training set of labeled data. However, their performances for future instances of labeled data could not be guaranteed. By contrast, our OLR can update its parameter vectors in order to deal with future instances. This is an indication of the flexibility and adaptability of the OLR algorithm.

4 Conclusions and future work

The proposed methodology provides two main advantages to a company that chooses to use crowdsourcing for New Product Development (NPD). First, the company can save time in screening the masses of ideas typically generated from a crowdsourcing application. Second, the proposed online mode logistic regression (OLR) algorithm can function as a recommendation agent. Each time a decision is made, the algorithm

learns a new parameter vector. Unlike Traditional Logistic Regression (TLR), OLR does not require a batch training sample but only a single training sample. And as time varies, the set of learning parameter vectors may be changed to adapt to new decisions.

For further research, full crowdsourcing application data should be tested in order to achieve a better understanding of the behavior of both crowds and decision makers. Moreover, the text mining method, in order to extract information more accurately, needs to be subject to more rules. The proposed OLR must also be improved to provide better quality. Future work might focus on either of the following two goals:

(1) combine text mining with sentiment analysis to analyze comments (comments can add useful information to ideas, and therefore relevance scores might vary with time);

(2) derive a set of measurements that can more accurately represent the behavior of decision makers.

Acknowledgement

This work was supported by a National Research Foundation of Korea grant (No. 2012R1A2008335) funded by the Korean Government.

References

1. The Rise of Crowdsourcing, http://www.wired.com/wired/archive/14.06/crowds.html
2. Dell's IdeaStorm, http://www.ideastorm.com
3. Starbucks's My Starbucks Idea, http://mystarbucksidea.force.com
4. Di Gangi, P.M., Wasko, M.: Steal my idea! Organizational adoption of user innovations from a user innovation community: A case study of Dell IdeaStorm. Decision Support Systems, 48(1), 303--312 (2009)
5. Bayus, B.: Crowdsourcing and Individual Creativity Over Time: The Detrimental Effects of Past Success. SSRN eLibrary (2010)
6. Huang, Y., Singh, P.V., Srinivasan, K.: Crowdsourcing New Product Ideas Under Consumer Learning. SSRN eLibrary (2011)
7. Thorleuchter, D.: Finding New Technological Ideas and Inventions with Text Mining and Technique Philosophy. In: Data Analysis, Machine Learning and Applications, C. Preisach, *et al.*, (eds.) 2008, pp. 413--420. Springer, Berlin Heidelberg (2008)
8. Thorleuchter, D., den Poel, D.V., Prinzie, A.: Mining ideas from textual information. Expert Systems with Applications, 37(10), 7182--7188 (2010)
9. Mahmood, M.A., Sullivan G.L.: Designing an expert consultation system to screen new product ideas: A consumer product application. Expert Systems with Applications, 5(1-2), 87--101 (1992)
10. Chan, S.L., Ip, W.H.: A Scorecard-Markov model for new product screening decisions. Industrial Management & Data Systems, 110(7), 971--992 (2010)
11. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the penn treebank. Comput. Linguist.,19(2), 313--330 (1993)
12. Google Insights for Search, http://www.google.com/insights/search
13. Carpenter, B.: Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. Technical report, Alias-i (2008)