Classifying Idea Adoptability in Online Communities

Guijia He, Jongbum Baik, and Soowon Lee

***Abstract*—**Recently many corporations actively adopt consumers' ideas in order to improve the quality of their service and products. Nevertheless, it is a huge work for company staffs to select useful ones from tens of thousands ideas. Hence how to predict whether an idea would be adopted and how to efficiently and correctly extract useful ideas, are important issues. In this research, we propose a method to classify ideas into two classes, adopted and rejected, by using text mining techniques. Several experiments are conducted using various algorithms and features in order to find an appropriate way to handle the idea classification problem. Experimental results show that the proposed classification method can obtain performance as high as 82% using F1-Measure and thereby it can help company staffs to improve work efficiency to some extent.

***Keywords*—**My Starbucks Idea, Idea Prediction, Idea Adoptability, Idea Classification.

I. Introduction

WITH the development of web 2.0, Internet users can write and share their information easily and freely. As a result, the amount of information increases sharply. How to extract useful information from such huge data became an important issue, especially for marketing and business. In marketing domain, more and more companies tend to make and adjust strategies based on consumer's behavior [1]. Some of them gather and analyze voice of customer data like product reviews to find consumer's favors. Meanwhile some others build idea-posting websites and actively inspire consumers to create and share their ideas. Subsequently, companies can improve the quality of their service and products by adopting good ideas and suggestions. One of these idea-posting websites in the case is *MyStarbucksIdea* [2], *MSI* for short.

Since early 2008, *MSI* has been built for Starbucks to gather ideas and feedback from their consumers. On *MSI*, any user can post his/her ideas covering one of fifteen categories including drink, food, services and so on. And meanwhile other users can comment and vote on the ideas. Starbucks staffs select and review some of ideas according to company's requires. If an idea is useful, it would be adopted and launched in the real world; otherwise it would be rejected. Generally, the more votes that an idea got, the more popular it is. However, the popularity just represents consumer's favors but not the company's [3]. That means even if an idea got a plenty of votes, it may not be adopted. And in contrast, the idea with even no votes may still be adopted. Therefore it is hard to collect useful ideas for the company only according to the number of votes, and Starbucks staffs have to judge usefulness based on contents of ideas. Unfortunately, it is almost an impossible work for them to review tens of thousands of ideas. As a result, only a few ideas are reviewed.

In order to recommend the ideas with high probability of adoption, Lee et al. built an innovative idea recommendation system using term-based and non-term-based features [4]. Their system can get high precision when only a few ideas are recommended. However, the performance sharply decreases along with the increase of the number of recommendation. Moreover, the number of top ideas recommended by the system is limited and it may not be helpful enough for Starbucks staffs. In order to solve the problems, to save the cost of review and to find out more 'hidden' useful ideas, we attempt to automatically classify ideas into two classes, adopted and rejected, based on text mining techniques.

Following the introduction, Section Ⅱ describes the procedure of text preprocessing and feature extraction. The classification method is represented in Section Ⅲ. Section Ⅳ analyzes experimental results in detail. Finally, Section Ⅴ concludes our work and suggests some future work.

II. Text processing

In this research, we suppose whether an idea is adopted by Starbucks mainly depends on the content of the idea rather than its popularity. Hence we attempt to classify ideas based on contents of ideas using text mining techniques. Differing from news articles, contents of ideas on *MSI* contain many noises and errors. In the text processing step, we conduct a preprocessing procedure to handle the noises so as to extract high-quality features from processed texts.

A. Preprocessing

Since content of ideas is typed by users with arbitrary words, it is so 'dirty' that we have to conduct a preprocessing

Guijia He is with the School of Computer Science & Engineering, Soongsil University, Seoul, Republic of Korea, (e-mail: twofirst@hotmail.com).

Jongbum Baik is with the School of Computer Science & Engineering, Soongsil University, Seoul, Republic of Korea, (e-mail: jongdal100@gmail.com).

Soowon Lee is with the School of Computer Science & Engineering, Soongsil University, Seoul, Republic of Korea, (corresponding author's e-mail: swlee@ssu.ac.kr).

This work was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013 (No. 13-912-03-003) and supported by Mid-career Researcher Program through NRF grant funded by the MEST (No. 2013016948) and by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2013(Grants No. C01203290100400718).

procedure before feature extraction. As shown in Fig. 1, we first clean them by removing useless parts such as URLs, emoticons and specific symbols. For convenience, Internet users often use abbreviations instead of long-length words. These words are usually meaningful, so we detect abbreviations and restore them back to their full spellings. Since *MSI* is a worldwide idea-posting website, some words may have different spellings even though their meanings are the same. It may be due to the differences between British English and American English. Similarly we find them out and unify them. Besides that, some words are wrong due to users' mistakes. Here we use an online spelling correction tool provided by Google to correct the mistakes [5]. In order to avoid over-correcting, some proper nouns related to Starbucks are reserved according to a list of products and materials crawled from the Starbucks menu.

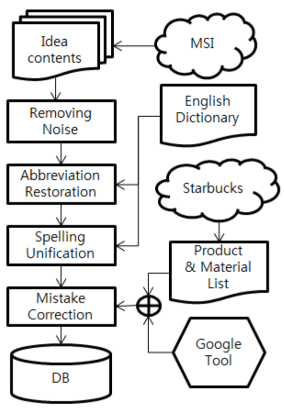

Fig. 1 Preprocessing Flow

B. Feature Extraction

In sentiment analysis studies, there are some arguments on whether unigram or bigram is better [6], [7], while some other studies declared high order n-gram is much better in large scale datasets [8]. We conjecture that the difference of performance is due to the distance between polar words and features. However idea classification is different from sentiment analysis because in idea classification, no words explicitly show that its appearance would influence the adoptability of the idea. As a result, it is a problem to choose an appropriate policy of n-gram so as to extract high-quality features. Unfortunately, there are still no reports about whether different n-gram policies can affect results of idea classification. In order to analyze the influence of n-gram and to seek a better way to extract features, in this research, we extract features using three policies, unigram, bigram, and combination of unigram and bigram.

Concretely, an idea is represented by a bag of words. Subsequently we remove the meaningless words named stop-words and then stem the remaining words using Porter's stemming algorithm [9]. The stemmed words are used as features of our unigram policy. For bigram, we first detect sentences of an idea and then extract bigram features from each sentence. Notice that we do not stem the bigram features and do not remove stop words in order to reserve their original information. The combination of unigram and bigram is built by merging both unigram features and bigram features.

Table I shows differences between the features selected by the three policies. Notice that the unigram features miss an important phrase, "fill up", due to the filter of stop-words. In addition, unigram contains more general features like "<get>", while bigram includes more distinct features like "<got, a>". In contrast, the combination of unigram and bigram retains the complete information.

TABLE I
EXTRACTED FEATURES BASED ON DIFFERENT N-GRAM POLICY

Example: I got a free drink to fill up.	
N-gram Policy	Extracted Features
Unigram	<get>, <free>, <drink>, <fill>
Bigram	<#, I>, <I, got>, <got, a>, <a, free>, <free, drink>, <drink, to>, <to, fill>, <fill, up>, <up, .>, <., #>
Unigram+ Bigram	<get>, <free>, <drink>, <fill>, <#, I>, <I, got>,<got, a>, <a, free>, <free, drink>, <drink, to>, <to, fill>, <fill, up>, <up, .>, <., #>

[a]# stands for a symbol at the beginning or at the end of a sentence.

III. EXPERIMENT

A. Dataset

On *MSI*, there are tens of thousands of ideas covering fifteen categories in total. We crawled the ideas posted before July 2012 and only use the ones reviewed by Starbucks staffs. In order to seek appropriate datasets for classification, five datasets are built. The ideas from three popular categories are used to build three single-category datasets named 'Single-Coffee', 'Single-Food' and 'Single-Card' respectively. Then we merge them together to form the fourth dataset, 'Multi-Three'. As the fifth dataset, 'Multi-All' contains all of ideas from all fifteen categories. Our purpose is to compare performance between those single-category datasets and multi-category datasets so as to find out the most appropriate datasets for idea classification. Furthermore, each dataset is separated into five samples to avoid over-fitting and to check whether our classifiers can handle different data. Average results of the five samples are used to evaluate the performance of classification.

B. Classifiers

To test whether ideas can be correctly classified into adopted or rejected, we train several different classifiers using a popular tool named WEKA [10]. The classifiers include Naïve Bayes, SVM, and Logistic. Naïve Bayes is a simple but effective method based on probability theory. By assuming features are conditionally independent from a given class, Naïve Bayes aims to find the class that can maximize the likelihood of a given idea [11]. In contrast, SVM is a non-probabilistic binary linear classifier, by which instances are presented as points in space. The basic idea of SVM is to find a hyper plane that can divide the two classes as wide as possible. In Lee's research [4], Logistic classifier gets the best result in their idea recommendation system. Here we use Logistic to check whether it can handle idea classification problem and whether it can get desired performance as well. Furthermore, we use it as the baseline and compare it with the other two classifiers.

TABLE II
PERFORMANCE COMPARISON BETWEEN SINGLE-CATEGORY AND MULTI-CATEGORY DATASET

Dataset	Feature Type	Weight Type	Feature Num	Classifier	P(Adopted)	R(Adopted)	F1-Measure(Adopted)
Single-Coffee	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.822	0.917	0.866
Single-Food	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.786	0.880	0.828
Single-Card	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.769	0.924	0.839
Multi-Three	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.677	0.897	0.771
Multi-All	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.658	0.842	0.739

TABLE III
CLASSIFIER COMPARISON IN DIFFERENT DATASETS

Dataset	Feature Type	Weight Type	Feature Num	Classifier	P(Adopted)	R(Adopted)	F1-Measure(Adopted)
Single-Coffee	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.822	0.917	0.866
Single-Coffee	Unigram + Bigram	Term Frequency	100	SVM	0.748	0.928	0.827
Single-Coffee	Unigram + Bigram	Term Frequency	100	Logistic	0.735	0.869	0.796
Single-Food	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.786	0.880	0.828
Single-Food	Unigram + Bigram	Term Frequency	100	SVM	0.745	0.876	0.802
Single-Food	Unigram + Bigram	Term Frequency	100	Logistic	0.798	0.764	0.779
Single-Card	Unigram + Bigram	Term Frequency	100	Naïve Bayes	0.769	0.924	0.839
Single-Card	Unigram + Bigram	Term Frequency	100	SVM	0.755	0.983	0.854
Single-Card	Unigram + Bigram	Term Frequency	100	Logistic	0.671	0.845	0.747

IV. RESULTS ANALYSIS

A. *N-gram Comparison*

Since the features used in our classification are extracted from contents of ideas, they would be different by using different n-gram policies such as unigram, bigram and the combination of unigram and bigram. In order to analyze whether there exist performance differences among the three policies, we use the same classifier (Naive Bayes) and the same number of features (top 500 by feature selection) for each dataset.

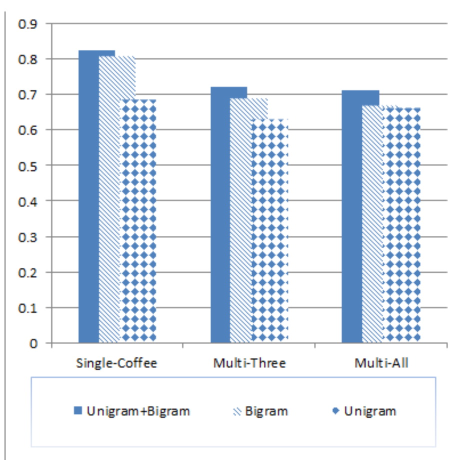

Fig. 2 Performance comparison with different n-gram policies

The result in Fig. 2 shows that the combination of unigram and bigram outperforms the others in all three datasets. We find that the performance increases obviously after adding the bigram features. This may be because there are some meaningful phrases in the contents of ideas, e.g. "bring back" in "Please bring back Frappuccino Happy Hour!" and "free drink" in "free drink with cup purchase". When the unigram policy is used, phases are split into single word and hence lost their original meanings particularly after removing stop-words. As a result, unigram got the worst results in all three datasets.

B. *Single-category vs. Multi-category*

Since there are fifteen categories in total on *MSI*, a problem is what data would be appropriate for idea classification, the ones from a single-category or the ones from a multi-category? Similarly, we use the same configuration and implement classification with different datasets. The classification results are shown in Table Ⅱ. Even though the Multi-Three dataset is composed of three single-category datasets, however the performance does not increase along with the increase of the number of training data. On the contrary, the precision and F1-measure of Multi-Three is much lower than the three single ones'. And the performance of Multi-All is even worse. This implies that there may exist some domain-based distinct features that can distinguish adoptability only in its category but would fail to handle other categories. The results suggest that idea classification should be implemented using single-category data rather than multi-category data.

C. *Classifier Comparison*

In this part, we aim to find an appropriate classifier for idea classification. Since Logistic classifier gets the best result in Lee's research [4], here we use the results of Logistic as the baseline and compare them with the results of Naïve Bayes and SVM. According to the results shown in Table Ⅲ, Naïve Bayes provides best performance compared with the other classifiers. On the contrary, Logistic gets the worst results. This is different from Lee's report. Because in Lee's research, they use a lot of statistical features beyond text terms, and Logistic classifier hence performs others. However in our experiment, all of features are extracted from text, and in contrast, the classifiers that are good at text classification such as Naïve Bayes and SVM, can get desirable results.

We notice that the best results come from the Single-Coffee dataset. This may be because Single-Coffee is the most popular category on *MSI* and it includes sufficient data to train a good classification model. With the dataset, we get precision as 82.2%, recall as 91.7%, and F1-measure as 86.6%. That means if an idea is classified as adopted by our method, it would be really adopted by Starbucks company with the probability as high as 82%. Meanwhile more than 91% useful ideas can be discovered. Moreover, the desirable results imply that whether an idea would be adopted or not, mainly due to its content. Therefore text mining techniques can be used to handle the idea classification problem to some extent. Furthermore, our method can help Starbucks staffs to improve work efficiency and to detect more excellent ideas.

V. Conclusion and Future Work

In this paper, we applied text mining techniques to idea classification for idea-posting websites like *MyStarbucksIdea*. By classifying ideas into two classes, adopted and rejected, our method can help Starbucks staffs to improve their work efficiency and find out more useful ideas. We summary our experimental results as follows: 1) contents of ideas play a very important role in idea classification problem so that text mining techniques can be used. 2) Using single-category data can yield better performance compared with multi-category data because different categories may include different issues. 3) Extracting features based on combination of unigram and bigram policy can retain more useful information. 4) The classifiers, that can well handle text classification like Naïve Bayes, are reasonably good at idea classification.

For the future work, we intend to analyze related and similar ideas to extract their common topics. With the topics, we can further analyze the differences of interests between the company and consumers. Beyond contents of ideas, the comments from other users may be considered by Starbucks staffs more or less. Therefore we also plan to analyze how users' opinion would affect the adoption of an idea by using sentiment analysis.

References

[1] J. Hauser, G.J. Tellis, A. Griffin, "Research on innovation: A review and agenda for marketing science," Marketing Science, 25(6), 687-717, 2006. http://dx.doi.org/10.1287/mksc.1050.0144

[2] http://mystarbucksidea.force.com/.

[3] H. Lee, S. Seo, "What Determines an Agreeable and Adoptable Idea? A Study of User Ideas on MyStarbucksIdea.com," in *Proc. 46th Hawaii International Conf. on System Sciences.*

[4] H. Lee, K. Choi, D. Yoo, Y. Suh, G. He, and S. Lee, "The more the worse? Mining valuable ideas with sentiment analysis for idea recommendation," in *Proc. of* the Pacific Asia Conference on Information Systems (PACIS).

[5] https://code.google.com/p/google-api-spelling-java/

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proc. of EMNLP-02 Conf. on Empirical Methods in Natural Language Processing*, Philadelphia, pp. 79-86.

[7] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the OMG," in *Proc. 5th International AAAI Conf. on Weblogs and Social Media*.

[8] H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," in *Proc. of 21th National Conf. on Artificial Intelligence*, 2006.

[9] T. Brants, "Natural Language Processing in Information Retrieval," in *Proc. of 14th Meeting of Computational Linguistics in the Netherlands, 2003*.

[10] L. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical Machine Learning Tools and Techniques with Java Implementations," 1999.

[11] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," in CS224N Project Report, Stanford, 2009.