International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195216

Predicting Early Reviews for Effective Product Marketing on E-Commerce Websites

F. Femila[1], S. Janakipriya[2], R. B. Nivetha Sruthi[2], S. Rohini[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India

[2]UG Scholar, Department of Computer Science and Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

The percentage of purchasing products by the user has been increased drastically through web. Users even have the facility of sharing their thoughts about the particular product on web in the form of reviews, blogs, comments etc. Many users read review information given on web to take decisions for buying products. Some users may give the reviews for hyping the sale of the product or to decrease the sale. This may confuse the customers who rely on the reviews to buy a product. So, there is a need to find the honest reviews and remove fake reviews that are added by malicious or fraud user. The proposed system comes up with the solution for this problem. Leading events has been used to find the time interval between the reviews. The proposed system mines the active periods such as leading sessions to accurately locate the hierarchical fraud. These leading sessions can be useful for detecting the local anomaly instead of global anomaly of product reviews. After this to analyze the rating, reviews and hierarchy of the product we examine three facts, they are rating based facts, review based facts and hierarchy facts. In addition, we propose an optimization-based aggregation method to integrate all the facts for fraud detection. The evaluations of this optimization are done on synthetic dataset that are collected. The classified and summarized product review information helps web users to understand review contents easily in a short time.

Keywords : Arduino, Wi-Fi (ESP 8266), Load cell, Database System

I. INTRODUCTION

The emergence of e-commerce websites has enabled users to publish or share purchase experiences by posting product reviews, which usually contain useful opinions, comments and feedback towards a product. As such, a majority of customers will read online reviews before making an informed purchase decision. It has been reported about 71% of global online shoppers read online reviews before purchasing a product. Product reviews, especially the early reviews (i.e., the reviews posted in the early stage of a product), have a high impact on subsequent product sales. We call the users who posted the early reviews early reviewers. Although early reviewers contribute only a small proportion of reviews, their opinions can determine the success or failure of new products and services. It is important for companies to identify early reviewers since their feedbacks can help companies to adjust marketing strategies and improve product designs, which can eventually lead to the success of their new products. For this reason, early reviewers become the emphasis to monitor and attract at the early promotion stage of a company. The pivotal role of early reviews has attracted extensive attention from marketing practitioners to

induce consumer purchase intentions. For example, Amazon, one of the largest e-commerce company in the world, has advocated the Early Reviewer Program, which helps to acquire early reviews on products that have few or no reviews. With this program, Amazon shoppers can learn more about products and make smarter buying decisions. As another related program, Amazon Vine2 invites the most trusted reviewers on Amazon to post opinions about new and pre-release items to help their fellow customers make informed purchase decisions. Based on the above discussions, we can see that early reviewers are extremely important for product marketing. Thus, in this paper, we take the initiative to study the behaviour characteristics of early reviewers through their posted reviews on representative e-commerce platforms, e.g., Amazon and Yelp. We aim to conduct effective analysis and make accurate prediction on early reviewers. This problem is strongly related to the adoption of innovations. In a generalized view, review posting process can be considered as an adoption of innovations3, which is a theory that seeks to explain how, why, and at what rate new ideas and technology spread. The analysis and detection of early adopters in the diffusion of innovations have attracted much attention from the research community.

II. LITERATURE SURVEY

[1] Ghose and Ipeirotis (2006) proposed two ranking mechanismsfor ranking product reviews: a consumer-oriented ranking mechanismthat ranks the reviews according to their expected helpfulness, and amanufacturer-oriented ranking mechanism that ranks them accordingto their expected effect on sales. They used econometric analysis with
text mining to make their ranking work

[2] Wu et al. (2013) carried out an analysis on both seller and customer reviews. Before purchasingany item, customers go through various things, such as customerreviews, seller reviews, and price comparison with othermarketplaces. The authors used all these parameters to determine thewillingness to pay of customers using a conceptual model.

[3] Li et al. (2013) analyzed content-based and source-based reviewfeaturesthat directly influence product review helpfulness. It was alsofound that customer-written reviews that were less abstract in contentand highly comprehensible result in higher helpfulness

[4] Lee and Shin (2014) investigated whether the quality of reviews affectsthe evaluations of the reviewers and the e-commerce website itself.They conducted pilot tests prior to the main experiment. Theparticipants were asked questions such as (a) how frequently they useonline shoppingmalls, and (b) if they had ever used the target product.They investigated (a) how the reader's acceptance depends on the qualityof online product reviews and (b)when such effects aremore or lesslikely to occur. Their findings indicated that participants' intention topurchase the product increases with positive high-quality reviews asopposed to low-quality ones.

[5] Huang et al. (2015) examined message length togetherwith aspectsof review patterns and reviewer characteristics for their joint effects onreview helpfulness. They found that the message length in terms ofword count has a threshold in its effects on review helpfulness. Beyondthis threshold, its effect diminishes significantly or becomes near nonexistent.

[6] Allahbakhsh et al. (2015) proposed a set of algorithms for robustcomputation of product rating scores and reviewer trust ranks.They harvested user feedback from social rating systems. Social ratingsystems collect and aggregate opinions (experience of using a service,purchasing a product, or hiring a person that is shared with other communitymembers, in order to help them judge an

item or a person thatthey have no direct experience with) to build a rating score or level of
trust worthiness for items and people.

[7] Chua and Banerjee (2016) found a relation between helpfulness andreview sentiment, helpfulness and product type, and helpfulness andinformation quality. Review sentiment was classified in three categories: favorable, unfavorable, and mixed. The products were categorizedas search products and experience products. The information qualityhas three major dimensions: comprehensibility, specificity, and reliability.

[8] Qazi et al. (2016) explainedwhy somereviews aremore helpfulcompared to others. As the helpfulness of online reviews helps theonline web user to select the best product, they read several reviewsof that product and finally conclude whether the review was helpfulor not.

III. ALGORITHM

An unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing and automated related works.

3.1 PORTER STEMMER ALGORITHM

The porter stemmer algorithm is a process for Removing suffixes by automatic means is an operation which is especially useful in the field of information retrieval. In a typical IR environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abstract. Ignoring the issue of precisely where the words originate, we can say that a document is represented by a vector of words, or *terms*. Terms with a common stem will usually have similar meanings, for example:

CONNECT
CONNECTED
CONNECTING
CONNECTION

IV. CONNECTIONS

Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

Usually is desired that only inflectional morphemes are removed (those corresponding to declinations, conjugations etc) not also derivational morphemes (which corresponds to different parts of speech).Porter algorithm does not fulfill this. One can make his own set of rules(for any language).Porters stemmer advantage is its simplicity and speed.

Assuming that one is not making use of a stem dictionary, and that the purpose of the task is to improve IR performance, the suffix stripping program will usually be given an explicit list of suffixes, and, with each suffix, the criterion under which it may be removed from a word to leave a valid stem.

Perhaps the best criterion for removing suffixes from two words W1 and W2 to produce a single stem S, is to say that we do so if there appears to be no difference between the two statements `a document is about W1' and `a document is about W2'. So if W1=`CONNECTION' and W2=`CONNECTIONS' it seems very reasonable to conflate them to a single stem. But if W1=`RELATE' and W2=`RELATIVITY' it seems perhaps unreasonable, especially if the document collection is concerned with theoretical physics. (It should perhaps be added that RELATE and RELATIVITY *are* conflated together in the algorithm described here.) Between these two extremes there is a continuum of different cases, and

given two terms W1 and W2, there will be some variation in opinion as to whether they should be conflated, just as there is with deciding the relevance of some document to a query. The evaluation of the worth of a suffix stripping system is correspondingly difficult.

The success rate for the suffix stripping will be significantly less than 100% irrespective of how the process is evaluated. For example, if SAND and SANDER get conflated, so most probably will WAND and WANDER. The error here is that the -ER of WANDER has been treated as a suffix when in fact it is part of the stem. Equally, a suffix may completely alter the meaning of a word, in which case its removal is unhelpful. PROBE and PROBATE for example, have quite distinct meanings in modern English. (In fact these would not be conflated in our present algorithm.) There comes a stage in the development of a suffix stripping program where the addition of more rules to increase the performance in one area of the vocabulary causes an equal degradation of performance elsewhere. Unless this phenomenon is noticed in time, it is very easy for the program to become much more complex than is really necessary. It is also easy to give undue emphasis to cases which appear to be important, but which turn to be rather rare. For example, cases in which the root of a word changes with the addition of a suffix, as in DECEIVE/DECEPTION, RESUME RESUMPTION, INDEX/INDICES occur much more rarely in real vocabularies than one might at first suppose. In view of the error rate that must in any case be expected, it did not seem worthwhile to try and cope with these cases.

3.2 THE ALGORITHM

To present the suffix stripping algorithm in its entirety we will need a few definitions.

A *consonant* in a word is a letter other than A, E, I, O or U, and other than Y preceded by a consonant. (The fact that the term `consonant' is defined to some extent in terms of itself does not make it ambiguous.) So in TOY the consonants are T and Y, and in SYZYGY they are S, Z and G. If a letter is not a consonant it is a *vowel*.

A consonant will be denoted by c, a vowel by v. A list ccc... of length greater than 0 will be denoted by C, and a list vvv... of length greater than 0 will be denoted by V. Any word, or part of a word, therefore has one of the four forms:

CVCV ... C
CVCV ... V
VCVC ... C
VCVC ... V

These may all be represented by the single form

[C]VCVC ... [V]

where the square brackets denote arbitrary presence of their contents. Using $(VC)^m$ to denote VC repeated m times, this may again be written as

$[C](VC)^m[V]$.

m will be called the *measure* of any word or word part when represented in this form. The case m = 0 covers the null word. Here are some examples:

m=0 TR, EE, TREE, Y, BY.
m=1 TROUBLE, OATS, TREES, IVY.
m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

The *rules* for removing a suffix will be given in the form

(condition) S1 -> S2

This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2. The condition is usually given in terms of m, e.g.

(m> 1) EMENT ->

Here S1 is `EMENT' and S2 is null. This would map REPLACEMENT to REPLAC, since REPLAC is a word part for which m = 2.

The `condition' part may also contain the following:

*S - the stem ends with S (and similarly for the other letters).

v - the stem contains a vowel.

*d - the stem ends with a double consonant (e.g. -TT, -SS).

*o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

And the condition part may also contain expressions with *and*, *or* and *not*, so that

(m>1 and (*S or *T))

tests for a stem with m>1 ending in S or T, while

(*d and not (*L or *S or *Z))

tests for a stem ending with a double consonant other than L, S or Z. Elaborate conditions like this are required only rarely.

In a set of rules written beneath each other, only one is obeyed, and this will be the one with the longest matching S1 for the given word. For example, with

SSES -> SS
IES -> I
SS -> SS
S ->

(here the conditions are all null) CARESSES maps to CARESS since SSES is the longest match for S1. Equally CARESS maps to CARESS (S1=`SS') and CARES to CARE (S1=`S').

1) 3.3 STEPS

2) In the rules below, examples of their application, successful or otherwise, are given on the right in lower case. The algorithm now follows:

3)

4) Step 1a

5)

SSES -> SS caresses -> caress
IES -> I ponies ->poni
ties ->ti
SS -> SS caress -> caress
S -> cats -> cat

Step 1b

(m>0) EED -> EE feed ->feed
agreed ->agree
(*v*) ED -> plastered ->plaster
bled ->bled
(*v*) ING ->motoring ->motor
sing ->sing

If the second or third of the rules in Step 1b is successful, the following is done:

AT -> ATE conflat(ed) -> conflate
BL -> BLE troubl(ed) >trouble
IZ -> IZE siz(ed) >size(*d and not (*L or *S or *Z))-> single letter
hopp(ing) -> hop
tann(ed) -> tan
fall(ing) -> fall
hiss(ing) -> hiss
fizz(ed) -> fizz
(m=1 and *o) -> E fail(ing) -> fail
fil(ing) -> file

The rule to map to a single letter causes the removal of one of the double letter pair. The -E is put back on -AT, -BL and -IZ, so that the suffixes -ATE, -BLE and -IZE can be recognised later. This E may be removed in step 4.

Step 1c

(*v*) Y -> I happy ->happi

sky ->sky

Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

6) Step 2

(m>0) ATIONAL -> ATE relational -> relate
(m>0)TIONAL -> TION conditional -> condition
rational -> rational
(m>0) ENCI -> ENCEvalenci -> valence
(m>0) ANCI -> ANCE hesitanci->hesitance
(m>0) IZER ->IZE digitizer ->digitize
(m>0) ABLI ->ABLE conformabli->conformable
(m>0) ALLI ->AL radicalli->radical
(m>0) ENTLI ->ENT differentli->different
(m>0) ELI ->E vileli ->vile
(m>0) OUSLI ->OUS analogousli->analogous
(m>0) IZATION -> IZE vietnamization->vietnamize
(m>0) ATION ->ATE predication ->predicate
(m>0) ATOR -> ATE operator ->operate
(m>0) ALISM -> AL feudalism ->feudal
(m>0) IVENESS -> IVE decisiveness ->decisive
(m>0) FULNESS ->FUL hopefulness ->hopeful
(m>0) OUSNESS -> OUS callousness ->callous
(m>0) ALITI -> AL formaliti ->formal
(m>0) IVITI -> IVE sensitiviti ->sensitive
(m>0) BILITI ->BLE sensibiliti ->sensible

The test for the string S1 can be made fast by doing a program switch on the penultimate letter of the word being tested. This gives a fairly even breakdown of the possible values of the string S1. It will be seen in fact that the S1-strings in step 2 are presented here in the alphabetical order of their penultimate letter. Similar techniques may be applied in the other steps.

7) Step 3

(m>0) ICATE -> IC triplicate ->triplic
(m>0) ATIVE -> formative -> form
(m>0) ALIZE -> AL formalize -> formal
(m>0) ICITI -> IC electricity -> electric
(m>0) ICAL -> IC electrical -> electric
(m>0) FUL -> hopeful -> hope

8) Step 4

(m>1) AL -> revival ->reviv
(m>1) ANCE -> allowance -> allow
(m>1) ENCE -> inference -> infer
(m>1) ER -> airliner ->airlin
(m>1) IC ->gyroscopic ->gyroscop
(m>1) ABLE -> adjustable -> adjust
(m>1) IBLE -> defensible ->defens
(m>1) ANT -> irritant ->irrit
(m>1) EMENT -> replacement ->replac
(m>1) MENT -> adjustment -> adjust
(m>1) ENT -> dependent -> depend
(m>1 and (*S or *T)) ION -> adoption -> adopt
(m>1) OU -> homologous -> homolog
(m>1) ISM -> communism ->commun
(m>1) ATE -> activate ->activ
(m>1) ITI ->angularity -> angular
(m>1) OUS ->homologous -> homolog
(m>1) IVE -> effective -> effect
(m>1) IZE ->bowdlerize ->bowdler
goodness -> good

The suffixes are now removed. All that remains is a little tidying up.

9) Step 5a

(m>1) E -> probate ->probat
rate -> rate
(m=1 and not *o) E -> cease ->ceas

10) Step 5b

(m> 1 and *d and *L) ->
Single letter
controll ->control
roll->roll

The algorithm is careful not to remove a suffix when the stem is too short, the length of the stem being given by its measure, m. There is no linguistic basis for this approach. It was merely observed that m

could be used quite effectively to help decide whether or not it was wise to take off a suffix. For example, in the following two lists:

list A	list B
RELATE	DERIVATE
PROBATE	ACTIVATE
CONFLATE	DEMONSTRATE
PIRATE	NECESSITATE
PRELATE	RENOVATE

-ATE is removed from the list B words, but not from the list A words. This means that the pairs DERIVATE/DERIVE, ACTIVATE/ACTIVE, DEMONSTRATE/DEMONS- TRABLE, NECESSITATE/NECESSITOUS, will conflate together. The fact that no attempt is made to identify prefixes can make the results look rather inconsistent. Thus PRELATE does not lose the -ATE, but ARCHPRELATE becomes ARCHPREL. In practice this does not matter too much, because the presence of the prefix decreases the probability of an erroneous conflation.

Complex suffixes are removed bit by bit in the different steps. Thus GENERALIZATIONS is stripped to GENERALIZATION (Step 1), then to GENERALIZE (Step 2), then to GENERAL (Step 3), and then to GENER (Step 4). OSCILLATORS is stripped to OSCILLATOR (Step 1), then to OSCILLATE (Step 2), then to OSCILL (Step 4), and then to OSCIL (Step 5). In a vocabulary of 10,000 words, the reduction in size of the stem was distributed among the steps as follows:

Suffix stripping of a vocabulary of 10,000 words

Number of words reduced in step 1: 3597
" 2: 766
" 3: 327
" 4: 2424
" 5: 1373
Number of words not reduced: 3650

The resulting vocabulary of stems contained 6370 distinct entries. Thus the suffix stripping process reduced the size of the vocabulary by about one third.

V. METHOD

This section gives a brief description about the methodologies used in the proposed system.

4.1 DATA MINING

Data Mining is the discovery of knowledge of analyzing enormous set of data; by extracting the meaning of the data and then predicting the future trends and also helps companies to take sound decisions, based on knowledge and information. Data mining software is one of a number of analytical tools for analyzing data.

4.2 DATA FLOOD

The current technological trends inexorably lead to data flood. More data is generated from banking, telecom, and other business transactions. More data is generated from scientific experiments in astronomy, space explorations, biology, high-energy physics, etc. More data is created on the web, especially in text, image, and other multimedia format.

4.3 WEB MINING

Web Contents Mining and Web Usage Mining. Web Contents Mining can be described as the automatic search and retrieval of information and resources available from millions of sites and on-line databases though search engines / web spiders. Web Usage Mining can be described as the discovery and analysis of Web mining can be broadly defined as discovery

and analysis of useful information from the World Wide Web. Based on the different emphasis and different ways to obtain information, web mining can be divided into two major partsuser access patterns, through the mining of log files and associated data from a particular Web site.

4.4 CONTENT MINING

Web content mining is the mining, extraction and integration of useful data, information and knowledge from Web page content. The heterogeneity and the lack of structure that permits much of the ever-expanding information sources on the World Wide Web, such as hypertext documents, makes automated discovery, organization, and search and indexing tools of the Internet and the World Wide Web such as Lycos, Alta Vista, WebCrawler, ALIWEB, MetaCrawler, and others provide some comfort to users, but they do not generally provide structural information nor categorize, filter, or interpret documents.

VI. IMPLEMENTATION

The paradigm for implementing the proposed system is discussed below.

Pre-processing

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues.

Mining foremostevents

The Application fraud is usually happens in Foremost Events, therefore identifying fraud Product is actually to detect fraud within foremost events of Products. Specifically, we first propose a simple yet effective algorithm to identify the foremost events of each Product based on its historical usage records. Then, with the analysis of products ranking behaviours, we find that the fraudulent products often have different usage patterns in each foremost events compared with normal Products.

Foremost events

There are two main steps for mining Foremost Events.

(i) We need to discover Foremost events from the products historical Usage records.

(ii) We need to merge adjacent events for constructing foremost event records.

Usage facts

A Foremost session is composed of several foremost events. Therefore, we should first analyze the basic characteristics of leading events for extracting fraud evidences. By analyzing the Product's historical usage records, we observe that Product's usage behaviours in a foremost event always satisfy a specific ranking pattern, which consists of three different ranking phases, namely rising phase, maintaining phase and recession phase.

Grade facts

The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences. Specifically, after an product has been published, it can be rated by any user who downloaded it. Indeed, user rating is one of the most important features of products advertisement. The product which has higher rating may attract more users to access and can also be ranked higher in the leader board. Thus,

rating manipulation is also an important perspective of fraud.

Evaluation facts

Besides ratings, most of the Product stores also allow users to write some textual comments as product reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular products. Indeed, review manipulation is one of the most important perspectives of product Usage facts. Specifically, before downloading or purchasing a new product, users often firstly read its historical reviews to ease their decision making, and a product contains more positive reviews may attract more users to access. Therefore, imposters often post fake reviews in the foremost sessions of a specific product in order to inflate the product uses, and thus proper the product's ranking position in the leader board. Although some previous works on review spam detection have been reported in recent years, the problems of detecting the local anomaly of reviews in the leading sessions and capturing them as evidences for ranking fraud detection are still under-explored.

Facts aggregation

After extracting three types of fraud evidences, the next challenge is how to combine them for ranking fraud detection. Indeed, there are many ranking and evidence aggregation methods in the literature, such as permutation based models, score based models. However, some of these methods focus learning a global ranking for all candidates. This is not proper for detecting ranking fraud for new products. Other methods are based on supervised learning techniques, which depend on the labeled training data and are hard to be exploited. Instead, we propose an unsupervised approach based on fraud similarity to combine these evidences. The combined evidences provides the best and the fraudelent product details.

Product recommendation

The recommendation process is very helpful to the mobile user to choose best apps and to avoid fraud products before to buy. The recommendation process compares the evidence aggregated result with the leading session better products.

VII.RESULTS

An important part of our information-gathering behaviour has always been to find out what other people think. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others. The sudden eruption of activity in the area of opinion mining and sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in text, has thus occurred at least in part as a direct response to the surge of interest in new systems that deal directly with opinions as a first-class object.Here we used visual studio and SQL server 2005 for predicting early reviews in e-commerce websites.

DATA

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases.

Fig.1 Shows the dataset which is used

sno	appname	review	reviewdt
23805	Panasonic	An excellent prod...	12/6/2018
23806	Mi	Picture Colour ar...	5/15/2018
23807	TCL	A fantastic TV at ...	10/1/2018
23808	iFFALCON	very bad custom...	7/7/2018
23809	iFFALCON	TV is quite good ...	12/17/2018

Fig.1 Dataset

STEMMING

There are several types of stemming algorithms which differ in respect to performance and accuracy and how certain stemming obstacles are overcome.

A simple stemmer looks up the inflected form in a look up table. The advantages of this approach are that it is simple, fast, and easily handles exceptions. The disadvantages are that all inflected forms must be explicitly listed in the table: new or unfamiliar words are not handled, even if they are perfectly regular (e.g. cats ~ cat), and the table may be large. For languages with simple morphology, like English, table sizes are modest, but highly inflected languages like Turkish may have hundreds of potential inflected forms for each root.A lookup approach may use preliminary part-of-speech tagging to avoid overstemming.

Stemming

transid	appname	review	reviewdt	rating
23814	Samsung	tv good but.. so...	3/12/2019	3
23842	Toshiba	bad customer s...	1/23/2019	1
23871	Toshiba	picture colour n...	1/23/2019	2
23881	Marq	pls dont buy pro...	7/7/2018	3
23896	Samsung	bad customer s...	1/20/2018	1
23907	Samsung	fantastic tv affo...	1/31/2018	2

Fig.2 Shows the stemming process

Fig.2 Stemming process

AGGREGATE RESULT

After the data is collected , the mean value for the following data is found then it undergoes foremost mining, the usage facts are found .Grade facts are calculated, stemming process is done and finally the aggregate result is given.

Fig.3 Shows the final output

Final Aggregate

	appname
▶	Panasonic
	Samsung
	TCL
*	

Fig.3 Output

7. CONCLUSION

Here, we have studied the novel task of early reviewer characterization and prediction on real-world online review datasets. Our empirical analysis strengthens a series of theoretical conclusions from sociology and economics. We found that (1) an early reviewer tends to assign a higher average rating score; and (2) an early reviewer tends to post more helpful reviews. Our experiments also indicate that early reviewers' ratings and their received helpfulness scores are likely to influence product popularity at a later stage. We have adopted a competition-based viewpoint to model the review posting process, and developed a margin based embedding ranking model (MERM) for predicting early reviewers in a cold-start setting. In our current work, the review content is not considered.

In the future, we will explore effective ways in incorporating review content into our early reviewer prediction model. Also, we have not studied the communication channel and social network structure in diffusion of innovations partly due to the difficulty in obtaining the relevant information from our review data. We will look into other sources of data such as Flixster in which social networks can be

extracted and carry out more insightful analysis. Currently, we focus on the analysis and prediction of early reviewers, while there remains an important issue to address, i.e., how to improve product marketing with the identified early reviewers. We will investigate this task with real e-commerce cases in collaborationwith e-commerce companies in the future.

VIII. REFERENCES

[1]. X. Rong and Q. Mei, "Diffusion of innovations revisited: from social network to innovation network," in CIKM, 2013, pp. 499- 508.

[2]. I. Mele, F. Bonchi, and A. Gionis, "The early-adopter graph and its application to web-page recommendation," in CIKM, 2012, pp. 1682-1686.

[3]. Y.-F. Chen, "Herd behavior in purchasing books online," Comput-ers in Human Behavior, vol. 24(5), pp. 1977-1992, 2008.

[4]. Banerjee, "A simple model of herd behaviour," Quarterly Journal of Economics, vol. 107, pp. 797-817, 1992.

[5]. A. S. E, "Studies of independence and conformity: I. a minority of one against a unanimous majority," Psychological monographs: General and applied, vol. 70(9), p. 1, 1956.

[6]. T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estima-tion of word representations in vector space," in ICLR, 2013.

[7]. A. Bordes, N. Usunier, A. Garc´ıa-Duran,´ J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in NIPS, 2013, pp. 2787-2795.

[8]. A. S. E, "Studies of independence and conformity: I. a minority of one against a unanimous majority," Psychological monographs: General and applied, vol. 70(9), p. 1, 1956.

[9]. M. L. S. D. X. W. L. S. Mingliang Chen, Qingguo Ma, "The neural and psychological basis of herding in purchasing books online: an event-related potential study," Cyberpsychology, Behavior, and Social Networking, vol. 13(3), pp. 321-328, 2010.

[10]. V. G. D. W. Shih-Lun Tseng, Shuya Lu, "The effect of herding behavior on online review voting participation," in AMCIS, 2017.

[11]. S. M. Mudambi and D. Schuff, "What makes a helpful online review? a study of customer reviews on amazon.com," in MIS Quarterly, 2010, pp. 185-200.

[12]. J. J. Mc Auley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products." in KDD, 2015, pp. 785-794.

[13]. E. Gilbert and K. Karahalios, "Understanding deja reviewers." in CSCW, 2010, pp. 225-228.

[14]. E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in CIKM, 2010, pp. 939-948.

[15]. C. Wang and D. M. Blei, "Collaborative topic modeling for recom-mending scientific articles," in SIGKDD, 2011, pp. 448-456.

[16]. R. Herbrich, T. Minka, and T. Graepel, "Trueskill: A bayesian skill rating system," in NIPS, 2006, pp. 569-576.

[17]. J. Liu, Y.-I. Song, and C.-Y. Lin, "Competition-based user expertise score estimation," in SIGIR, 2011, pp. 425-434.

[18]. Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in ICML, 2014, pp. 1188-1196.

[19]. Y. B. Xavier Glorot, "Understanding the difficulty of training deep feedforward neural networks," in AISTATS, 2010, pp. 249-256.

[20]. R.A.Bradley and M.E.Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," in Biometrika, 1952, pp. 324-345.

Cite this article as :

F. Femila, S. Janakipriya, R. B. Nivetha Sruthi, S. Rohini, "Predicting Early Reviews for Effective Product Marketing on E-Commerce Websites ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 2, pp. 290-300, March-April 2019. Available at doi :

https://doi.org/10.32628/CSEIT195216

Journal URL : http://ijsrcseit.com/CSEIT195216