Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp196-201)

# Text Mining for Customer Satisfaction Monitoring

NICHOLAS EVANGELOPOULOS
Department of Information Technology and Decision Sciences
University of North Texas
P.O. Box 305249 Denton TX 76203-5249
UNITED STATES
http://www.coba.unt.edu/itds/faculty/evangelopoulos/evangelopoulos.htm

*Abstract:* - In this paper we lay the necessary groundwork towards Quality Control in Customer Relationship Management, using free text customer feedback as the only source of data. In a scheme that follows the general principles of Case-Based Reasoning, dubbed here Case-Based Free Text Evaluation, a small subset of documents with customer comments is evaluated by human experts to obtain customer satisfaction ratings. The ratings of the remaining documents are estimated automatically. Now the entire document collection can be resampled to generate control charts that monitor customer satisfaction. As an illustration of this framework we are using viewers' comments submitted to the Internet Movie Database (IMDb) Web site after they watched a recent popular film.

*Key-Words:* - Text Mining, Customer Relationship Management, Case Based Reasoning, Text Clustering

## 1 Introduction

Modern Customer Relationship Management (CRM) is often faced with the time-consuming task of sifting through large volumes of customer feedback comments which represent an underexploited source of business intelligence. The main difficulty here is that text data need to be converted to some numerical format, so that well-established statistical quality control tools can take over.

The purpose of this paper is to discuss the feasibility of Customer Satisfaction Monitoring using established Quality Control tools, when the only data available are in free text format. The proposed solution is to use humans to evaluate a relatively small set of text documents against some desirable rating standard, then produce rating estimates for the rest of the documents automatically.

## 2 Free Text Evaluation

Our approach to free text evaluation will follow a methodology related to Case-Based Reasoning, will use Vector Space Model arguments to calculate similarity between documents, and a *k*-means text clustering to determine the initial case base.

### 2.1 Vector Space Model (VSM)

The Vector Space Model (VSM), a ranking model that ranks individual documents against a query, was originally introduced by Gerald Salton and his associates [10], [11], and has been involved in a large part of the Information Retrieval research. VSM calculates similarity between two documents represented as vectors of unique terms, by considering the cosine of the angle that is formed between the two vectors,

$$sim\left(\vec{d},\vec{q}\right) = \frac{\vec{d}\cdot\vec{q}}{\left\|\vec{d}\right\|\cdot\left\|\vec{q}\right\|}. \qquad (1)$$

Various improvements to the basic VSM have been introduced by researchers, such as term stemming [7], [9], or Latent Semantic Indexing (LSI) [4]. Based on these of quantifying document similarity, Text Classifiers have been widely used in document categorization (indexing). Methods for construction of document classifiers include inductive learning techniques [3], probabilistic and decision tree classifiers [5], genetic algorithms, neural networks [13], taxonomies [8], statistical clustering [2], and K-Nearest Neighbor [12, pp. 102-103], [6].

### 2.2 Case-Based Reasoning (CBR)

Case-based reasoning (CBR) is an Artificial Intelligence framework that functions through the usage of previous cases. New cases are compared to a case-base and checked for similarity. As the CBR system is utilized, new cases will be added to the case-base, building upon the general knowledge component found in the center of the CBR cycle, which follows conceptual steps referred to as *retrieve, reuse, revise,* and *retain* [1].

### 2.3 *K*-Means Text Clustering (KMTC)

Since our CBFTE approach will compare each not-yet-rated customer comment to its closest match in the Case Base, it is important to select the initial members of the Case Base not randomly, but in an optimized way that maximizes the average similarity of all cases to their closest match. The reason why we need our cases to be very close to their best matches is because, when two documents are very similar, their ratings are expected to be similar also, however if the documents are dissimilar they could have ratings that are very different, but they could also have ratings that are very similar, since there are more than one possible ways for customers to express a certain level of satisfaction. It is, therefore, desirable to identify an initial set of documents (training set) by forming document clusters and selecting an equal number of documents from each cluster to participate in the training set.

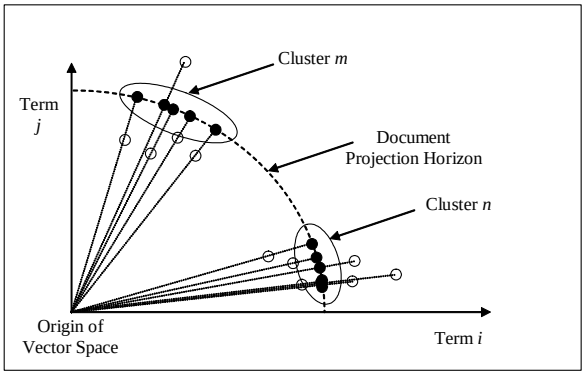


Fig. 1. Clusters of documents in the Vector Space.

*Algorithm 1:*

For $m$ = 1 to $M$, where $M$ is a moderately large number;

1. Select a randomly chosen set of k seed documents that define k initial clusters and calculate their term concatenations $q_i$, $i$ = 1,…, $k$.
2. Perform $k$ queries using qi as the query documents. For each document $d_j$ in the collection, $j$=1,…,$n$, consider the similarities $Sim(d_j, q_i)$ and assign $d_j$ to the cluster whose term concatenation $q_i$ is nearest. Use heuristics to deal with orphaned documents and empty clusters. Recalculate the term concatenations for the cluster receiving the new document and for the cluster losing the document.
3. Repeat Step 2 until no more assignments take place.
4. Calculate the Overall Similarity of the final clustering arrangement and save the clustering arrangement having the maximum Overall Similarity.

Next $m$.

The heuristics mentioned in step 2 are explained below.

*Heuristic KMTC.1:*

If a document is orphaned (= without a cluster to belong to) it will join the least populated cluster. Ties will be resolved by random assignment.

*Heuristic KMTC.2:*

If a cluster is left empty, it will be joined by a document that deserves to be in its current cluster the least. Ties will be resolved by random assignment.

Figure 1 illustrates two such clusters populated with documents with a high degree of within-cluster similarity. Because our Vector Space representation of documents uses the cosine of the angle between vectors as a measure of similarity, the usual n-dimensional Cartesian space with the rectangular coordinates gives way to a system of normalized polar coordinates where radius lengths do not matter, only angles do. Essentially all documents get projected on an n-dimensional spherical surface (a hyper-sphere) marked on Figure 1 as the document projection horizon. VSM's view of the document clusters then becomes similar to an observer's view of the starry night sky. Some stars (or planets, or even galaxies for that matter, since they all look so similar) appear to form clusters, but are they really close to each other? Well, stars that appear to participate in star clusters, for one, usually are! In the results section we will investigate what happens with document clusters.

### 2.4 CBFTE Algorithm

In view of the preceding discussion, we summarize the procedures followed by a Case-Based Free Text Evaluation engine in the following algorithm:

*Algorithm 2:*

1. Removal of Unique Documents: Separate all unique documents from the collection.
2. KMTC: Perform $M$ runs of $K$-Means Text Clustering (*Algorithm 1*) and choose the clustering solution with the highest overall similarity.

3. Construction of the Case Base: Have a human expert manually rate l representatives of each cluster. Store the *kl* rated cases in the case base.
4. Case-Based Free Text Evaluation: For each of the remaining $N - kl$ documents, find the best match in the case base, and produce a corresponding rating estimate.
5. Acceptance Sampling: Sample $s$ of the estimated ratings and ask the human expert to rate them independently. Learning: Expand the case base by storing the verified ratings.
6. Repeat step 5 until all verified ratings are equal to the corresponding estimated ratings. Then accept the remaining $N - kl - s$ unverified estimates and stop.

## 3 A Customer Satisfaction Example

We will now discuss an example with free text data coming in the form of customer feedback. The data set under consideration will include short comments expressing how satisfied customers were with a certain product or service. Corresponding ratings in numerical form will also be used in order to evaluate the performance of our automatic rating algorithm.

### 3.1 Methods

We surveyed movie viewers' comments on a major box office movie that was released in 2005. These viewers were users of the Internet Movie Database (IMDb), a popular Web Forum related to movies. Their movie feedback comments were posted during a certain 30-day period of 2005. Users posted their star-rating (on a 1-10 Likert scale). The distribution of the submitted ratings, based on a sample of about 50,000 users who volunteered to vote electronically during the aforementioned 30-day period, is shown in Fig. 2.

During that same period, about 2,100 written comments were submitted by registered users, together with their Likert-scale star-ratings. Those comments included a short comment (typically one line) summarizing their impression, and a longer comment (typically 1 paragraph to 1 page) elaborating on their feelings and opinions regarding the movie. About 900 of those registered users' postings originated in the United States and the remaining 1,200 in a large number of other countries. The fact that the comments were posted in English served, of course, as a source of bias: For instance, 250 comments came from the United Kingdom, 97 from Canada, 60 from Australia, whereas Germany contributed 21 postings, France 10, Greece 6, and Austria 3. Nevertheless, countries such as Argentina, Mexico, Turkey, Egypt, or Thailand, were still represented in the sample. The most serious imbalance in the sample was the gender bias: The 2,100 users who posted comments included only 200 women. This was probably due to the nature of the particular movie which, being an action-adventure, apparently had many more male than female viewers who felt strongly about the movie and decided to submit comments about it. To some extent this is not only a bias of the sample, but also a reflection of the gender distribution in the population of interest (i.e., the viewers of the movie).

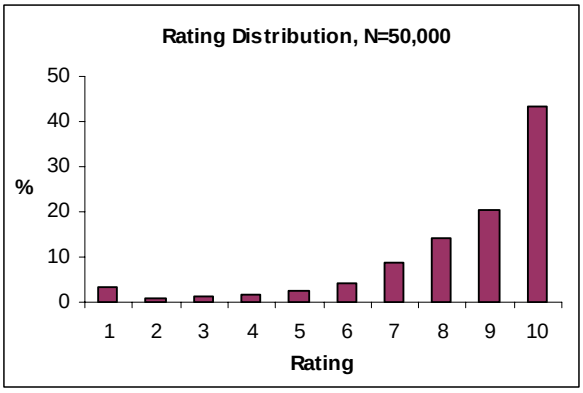


Fig. 2. Distribution of ratings submitted by 50,000 users. Information courtesy of *The Internet Movie Database* (http://www.imdb.com). Used with permission.

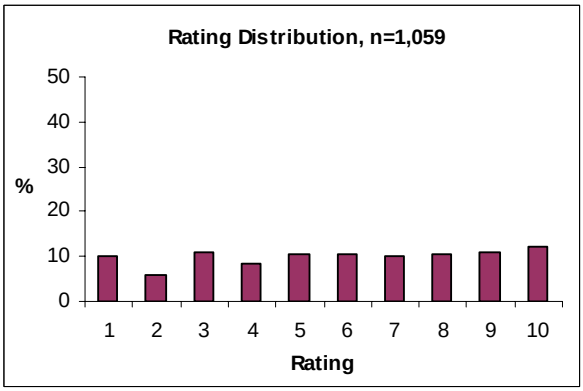


Fig. 3. Distribution of 1,059 sampled ratings.

For the purposes of this study, 1,059 short comments were sampled, using a sampling scheme that employed approximate separate sampling, i.e., picked an about equal number of comments from each rating level. The number of comments

participating in the sample that came from each rating level are shown in Fig. 3.

### 3.2 FTE Algorithm Implementation

As part of our free text evaluation approach, a dictionary of all non-trivial terms included in the document collection was compiled. The size of the dictionary varied with the number of comments under consideration. For example, a small subset of 260 short comments had 418 different terms after being filtered with 530 stopwords. Another sample of 520 short comments had 732 different terms, whereas the full sample of 1059 short comments had 1228 different terms. Although the size of the dictionary grows bigger as the number of documents in the collection goes up, it's interesting to notice that the top 20 terms, after we sort the three dictionaries by descending term frequency, are almost identical: 18 out of 20 terms were present in all three dictionaries as part of the top 20 list, and the remaining 2 turned up in spots 21-25. Another interesting observation is the number of unique terms (i.e., terms that appear in only one document). The first dictionary had 317 unique terms out of 418, the second had 541 out of 732, and the third 833 out of 1228. For very large document collections, the number of terms in the dictionary is expected to approach or even exceed the 10,000 to 20,000 words that are commonly used in the English language. The 20 terms with the highest frequencies included judgment expressions such as *good*, *best*, *better*, *great*, *bad*, and *disappointing*. They also included the word *movie*, words from the film's title and the director's name. This second category is unrelated to the evaluative nature of these short comments. Subsequently, a new stopword list with 537 terms was applied. We started with a standard stoplist, removed evaluative terms such as bad and good, then added 7 non-evaluative terms that were specific to this set of documents.

For the purposes of calculating similarities between documents, binary weights were used, since they have been known to produce slightly better results when they operate on short documents. Utilization of a Porter Stemmer yielded only marginal difference in the results, so no stemming was applied to the document terms.

After the dictionary customization, *k*-means text clustering was applied to the set of documents, using a variety of number of clusters *k* values, as well as a number of clustering runs *M*, so that an optimal clustering solution could be picked. One representative from each cluster was then considered to have a known rating value (i.e., as if it was now rated by a human expert) and the remaining documents had their ratings estimated using a Best Match approach, i.e., without any adaptation (modification) of the estimated ratings.

## 4 Results

After estimating the "unknown" ratings in the way that was described in section 3, the percentage of documents having an estimated rating that was exactly equal to the "true" rating was obtained. Fig. 4 shows this rating accuracy for a collection of $N$=260 user comments and for a varying number of clusters that ranged from $k$=5 to $k$=120. The rating accuracy starts around 10% and reaches approx. 55%. For each value of $k$, $M$=40 iterations were applied so that the best clustering solution could be determined.

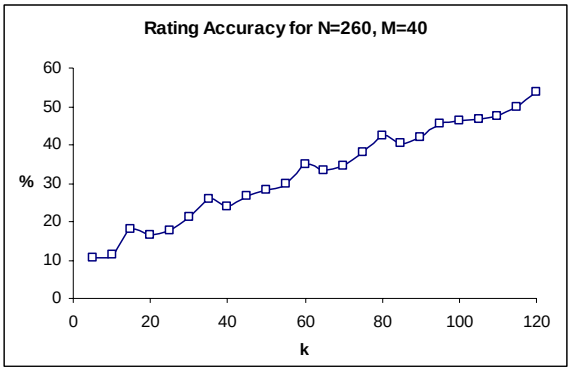


Fig. 4. Rating accuracy for $N$=260, $M$=40.

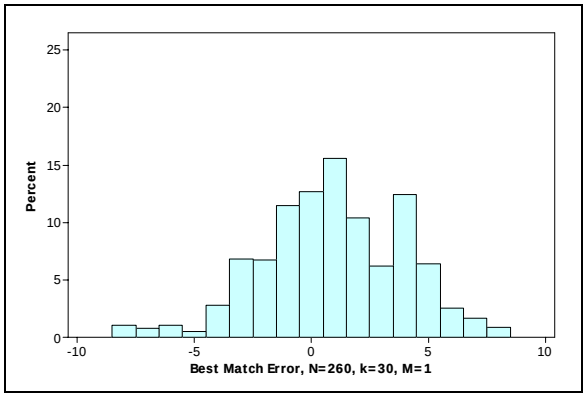


Fig. 5. Rating error distribution for $N$=260, $k$=30, $M$=1.

While it is important to achieve high accuracy rate, it is also interesting to see the entire distribution of the rating error, defined here as the discrepancy between the estimated and the observed (=reported)

rating. Fig. 5 shows this distribution for the selected case of $N$=260, $k$=30, applying only $M$=1 iteration during the search for the optimal clustering solution, but applying 10 repetitions of the algorithm so that a representative and smooth error distribution could be obtained. From the shape of the distribution we see that the errors are quite spread, and that could be due to the somewhat small number of clusters, or the minimal number of clusters considered before the final clustering solution could be selected.

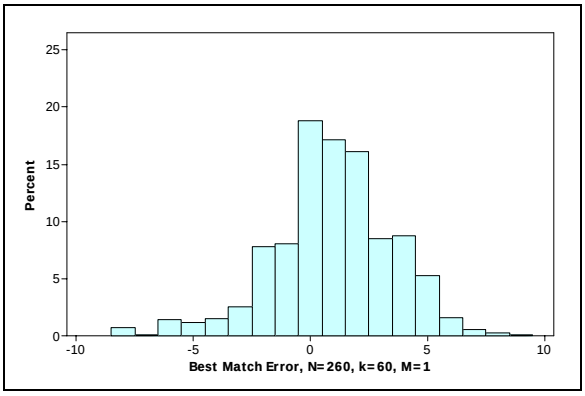


Fig. 6. Rating error distribution for $N$=260, $k$=60, $M$=1.

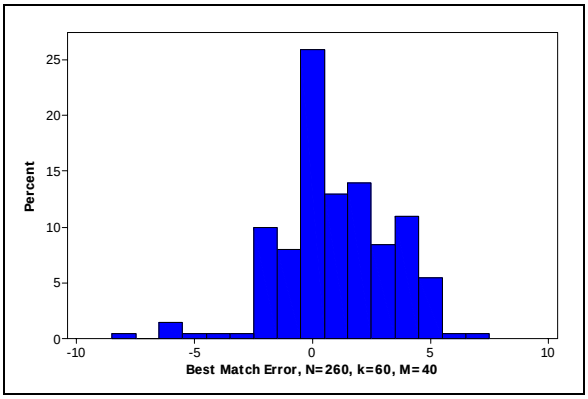


Fig. 7. Rating error distribution for $N$=260, $k$=60, $M$=40.

Figure 6 explores the effect of the number of clusters $k$, by examining the error distribution when $N$ and $M$ are kept constant to 260 and 1 respectively, but $k$ changes from 30 to 60. We observe that the errors really come a little closer to zero, although not by a lot. Figure 7 keeps the new value of $k$ and increases the repetitions $M$ from 1 to 40. Comparing figures 6 and 7 it is quite obvious that considering a large number of clustering solutions before a final clustering arrangement can be selected appears to be very beneficial. As the repetitions $M$ increase from 1, to 10, to 40, the Mean Absolute Error (MAE) decreases from 2.08, to 2.056, to 1.905, respectively.

Finally, to investigate the effect of the size of the document collection on the rating error, Figure 8, where $N$=520, $k$=60, and $M$=1, can be compared against $N$=260, $k$=60, and $M$=1, respectively, in Fig. 6. Clearly, Fig. 8 presents an error distribution that is more spread, away from zero.

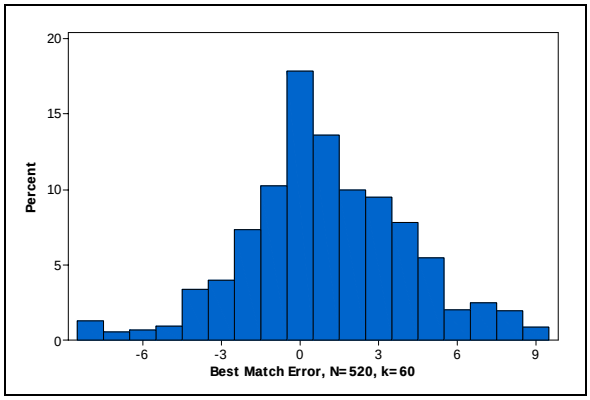


Fig. 8. Rating error distribution for $N$=520, $k$=60, $M$=1.

## 5 Discussion

After an overall evaluation of the rating error distributions presented in the results section, we might claim that free text evaluation following the Case-Based approach that was presented in this paper is not without merit, but we would have to admit that it's not without problems, either. Customers can be very creative in the ways they express their feelings and their satisfaction. They can use rare words, slang, make obscure points, be sarcastic, or play word games, just to name a few. Still, with moderately large numbers of documents $k$ that are processed by humans and fairly large numbers of clustering solution repetitions $M$ that are processed by computers, the rating estimation errors can be kept within tight ranges. As a future direction, it is interesting to examine the possible improvement in rating accuracy or the possible reduction in rating estimation error when longer, more descriptive comments are used, as opposed to the short, summarizing tag lines that were employed here.

## 6 Conclusion

In this paper we brought together a number of text mining tools in order to create a framework for free text evaluation. Our methodology was applied to a

collection of short customer feedback comments that expressed satisfaction or dissatisfaction over a recently released film, where the users also provided quantitative information expressing the same satisfaction in the form of a rating score. Our example illustrated that it is actually possible for a human rater to pay the cost of manually rating a subset of customer feedback comments and then enjoy the benefit of having the rest of them automatically rated without a great loss of rating accuracy. This way, a basis for further quantitative analysis and, therefore, more efficient customer relationship management can be established.

*References:*


[1] Aamodt, Agnar; Enric Plaza (1994), Case-based reasoning; Foundational issues, methodological variations, and system approaches, *AI Communications*, Vol. 7, No.1, 1994, pp. 39-59.

[2] Anick, P and Vaithyanathan, S, Exploiting Clustering And Phrases For Context-Based Information Retrieval, *SIGIR*, 1997.

[3] Cohen, W. W. and Singer, Y, Context-Sensitive Learning Methods for Text Categorization, *SIGIR'96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996, pp. 307-315.

[4] Deerwester, S., Dumais, T., Furnas, W., Landauer, K., and Harshman, R., Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, Vol. 41, 1990, pp.391-407.

[5] Lewis, D.D. and Ringuette, M, Comparison of Two Learning Algorithms for Text Categorization, *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

[6] Li, Fan, and Yang, Yiming, A Loss Function Analysis for Classification Methods in Text Categorization, *The Twentieth International Conference on Machine Learning (ICML'03)*, 2003, pp. 472-479.

[7] Porter, M. F, An Algorithm for Suffix Stripping, *Program*, Vol. 14, No. 3, 1980, pp. 130-37.

[8] Raghavan, P, Information Retrieval Algorithms: A Survey, *Symposium on Discrete Algorithms*, ACM-SIAM, 1997.

[9] Riloff, E, Little words can make a big difference for text classification, In *Proc. 18th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1995, pp. 130-136.

[10] Salton, G., and M. E. Lesk, Computer Evaluation of Indexing and Text Processing, *J. Association for Computing Machinery*, Vol. 15, No. 1, 1968, pp.8-36.

[11] Salton, G., and C. Buckley, Term-Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management*, Vol. 24, No. 5, 1988, pp.513-23.

[12] Weiss, S. M, and N. Indurkhya, *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann, San Francisco, 1998.

[13] Wiener, E., Pedersen, J.O. and Weigend, A.S, A Neural Network Approach to Topic Spotting, *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.