

Exploring User Feedback of a E-Learning System: A Text Mining Approach

Wen-Bin Yu¹ and Ronaldo Luna²

¹Department of Business and Information Technology,
Missouri University of Science and Technology, Rolla, Missouri, USA
yuwen@mst.edu

²Department of Civil, Architectural and Environmental Engineering,
Missouri University of Science and Technology, Rolla, Missouri, USA
rluna@mst.edu

Abstract. Given a collection of survey comments evaluating an E-learning system, the text mining technique is applied to discovering and extracting knowledge from the comments. The purpose is to categorize the comments into several groups in an attempt to identify key criticisms or praises from students using the E-learning system. This study is able to assist the evaluators of the E-learning system to obtain the summarized key terms of major “concerns” without going through potentially huge amount of survey data.

Keywords: E-learning, Text Mining, Survey.

1 Introduction

A GIS learning system [6] was funded by the National Science Foundation (NSF) to introduce GIS to undergraduate students enrolling in a typical civil engineering program. The system consists of five modules with specialties in environmental, geo-technical, surveying, transportation, and water resources, since these topics are standard topics in civil engineering programs nationwide. This GIS learning system enables faculty to bring practical applications to the classroom in ways which are not possible through traditional instruction.

Surveys were conducted a week after each class using the GIS Learning system in a lab environment to evaluate effectiveness of instruction and the learning system itself. Each survey contained both quantitative and qualitative questions. Quantitative questions (i.e., multiple choices, true/false questions) were easy to be analyzed since statistical and scientific techniques can be applied directly to the results. However, the qualitative part including open-ended questions requires human efforts to read all feedbacks from students in order to come up with conclusions.

Text mining is a useful technique which is able to handle unstructured data such as textual data. Therefore, this research aims to apply text mining to the survey comments from students in an effort to assist the evaluation team with the analysis of qualitative data.

2 Literature Review

Text mining is a process which attempts to discover and extract hidden knowledge from unstructured text and present it to users in a concise form semiautomatically. It is more art rather than science, and can be defined broadly. Some concepts in computer science, such as information extraction (IE), information retrieval (IR), natural language processing (NLP), and data mining (DM) are also applied in text mining. Text mining optionally inherits some of their techniques to deal with specific problems in each step throughout the entire process.

Text mining process starts from pre-processing which transforms unstructured data into the most structured, meaningful representation of the document collection. The most common way to structure documents is to convert them from qualitative format into quantitative format. Mathematical and statistical methods are useful for term selection and quantitative representation of the documents. Vector-space approach is the conventional model which serves these purposes. Then, the core mining process analyzes the set of documents from the quantitative representation. DM techniques, for instance, classification, clustering, decision trees, or neural networks, can be employed here to extract relations, trend as well as patterns from the document collection, and present the results based on users' requirements.

2.1 Pre-processing

The goal of the pre-processing stage is to create a quantitative representation for the documents. Starting unstructured or partially structured, a document is then proceeded to enrich structure, and finally ended up with the most advanced and meaning-representing features which will be used for the core mining process. Vector-space approach is used to convert and represent the document in quantitative format. The final representation of the collection of documents will be in a term-document frequency matrix.

The resulted matrix is generally sparse and will become much sparse when the size of document collection increases since few terms contain in any single document. Also, only hundreds of documents can yield thousands of terms. Huge computing time and space are required for the analysis. Therefore, reducing dimensions of the matrix can improve performance significantly. Singular Value Decomposition (SVD) is a popular technique to deal with dimensional reduction. It projects the sparse, high-dimensional matrix into smaller dimensional space.

In addition, another way to improve retrieval performance of the analysis is to apply weighting methods. According to Berry and Browne [2], the performance refers to the ability to retrieve relevant information while dismiss irrelevant information. Each element of the matrix ($a_{i,j}$) can be applied weighting and represented as

$$a_{i,j} = L_{i,j}G_iD_j,$$

where $L_{i,j}$ is the frequency weight for term i occurring in document j , G_i is the term weight for term i in the collection, and D_j is a document normalization factor indicating whether document j is normalized.

This equation was originally applied from information retrieval for search engine where longer documents have a better chance to contain terms matching the query than the shorter ones. Therefore, the document normalization factor was included to equalize the length of the document vectors from documents which vary in length [7]. Since this study focused on text mining and the lengths of the documents in the collection were not varied, the third factor was unnecessary and ignored by replacing the variable with 1. Then, the final equation is

$$a_{i,j} = L_{i,j}G_i$$

Defining the appropriate weighting depends on characteristics of the document collection. The frequency weights and term weights are popular weighting schemes which are described in more detail in the following subsections.

2.2 Frequency Weights

They are functions of the term frequency ($L_{i,j}$). This factor measures the frequency of occurrence of the terms in the document by using a term frequency (TF). Common methods include binary and logarithm. Three common weighting schemes are shown below where $f_{i,j}$ represents the original frequency of term i appears in document j .

- Binary: $L_{i,j} = \begin{cases} 1 & \text{if term } i \text{ is in document } j \\ 0 & \text{otherwise} \end{cases}$
- Logarithm: $L_{i,j} = \log_2(f_{i,j} + 1)$
- None: $L_{i,j} = f_{i,j}$

Sometimes, a term is repeated in a document for a lot of time; thus, it reflects high frequency in the document collection as a whole even though it appears in only one document. To reduce the effect from the repetitive terms, Binary and Logarithm can be applied to the term frequency. The Binary method takes no repetitive effect into account while Logarithm reduces the effect, but still maintains it in some degree. Therefore, the Logarithm is a method in between Binary and None. Moreover, taking log of the raw term frequency reduces effects of large differences in frequencies [4].

According to Berry and Browne [2], the selection of appropriate weighting methods depends on the vocabulary or word usage patterns for the collection. The simple term frequency or none weighting term frequency is sufficient for collection containing general vocabularies (e.g., popular magazines, encyclopedias). Binary term frequency works well when the term list is short (e.g., the vocabularies are controlled).

2.3 Term Weights

Term weights take word count in the document into account. Common methods are:

- Entropy: $G_i = 1 + \sum_j \frac{p_{i,j} \log_2(p_{i,j})}{\log_2(n)}$
- GF-IDF: $G_i = (\sum_j f_{i,j}) / \sum_j X(f_{i,j})$
- IDF: $G_i = \log(n / \sum_j X(f_{i,j}))$
- Normal: $G_i = 1 / \sqrt{\sum_j f_{i,j}^2}$
- None: $G_i = 1$

where $f_{i,j}$ represents the original frequency of term i appears in document j , n is number of documents in the collection, $p_i = f_{i,j} / \sum_j f_{i,j}$ and,

$$X(f_{i,j}) = \begin{cases} 1 & \text{if term } i \text{ is in document } j \\ 0 & \text{otherwise} \end{cases}$$

The choice for an appropriate term weight depends on the state of the document collection, or how often the collection is likely to change. This weighting scheme responds to new vocabulary and affects all rows of the matrix. Thus, it is useful when updating of new vocabulary is acceptable or rare such as static collections whereas it is disregarded when updating needs to be avoided by using none weighting. All of the formulas emphasize words that occur in few documents whereas give less weight to terms appearing frequently or in many documents in the document collection. Entropy takes the distribution of terms over documents into account. Normal is the proportion of times the words occurring in the collection.

2.4 Core Mining: Clustering

The stage inherits analysis methods from data mining such as classification, decision trees, and clustering. Since the goal was to cluster comments into several clusters without pre-defined categories, this research only focuses on clustering.

Feldman and Sanger [5] defined clustering as an unsupervised process which classifies unlabeled objects into meaningful groups called clusters without any prior information or pre-defined categories. The labels associated with the groups of objects are obtained from the data.

Clustering determines the features which better describe objects in the set, intra-cluster similarity, while distinguish objects in the set from the collection, inter-cluster dissimilarity [1]. Intra-cluster similarity measures a raw frequency of a term k_i inside a document d_j , aka TF factor. Inter-cluster dissimilarity measures the inverse of the frequency of a term k_i among the documents in the collection, aka inverse document frequency or IDF factor. Term weights, which were introduced in the previous section were derived from this theory. IDF weighting focuses on inter-cluster dissimilarity and tries to reduce the effect when the terms appearing in many documents are not useful for distinguishing documents. The product of TF and IDF (TF-IDF) was proposed as a reasonable measure which tries to balance the two effects, intra-cluster similarity and inter-cluster dissimilarity.

According to Feldman and Sanger [5] clustering can have different characteristics. It can be flat if it produces disjoint clusters, or can be hierarchical if the resulted clusters are nested. Clustering will be hard if every object belongs to exactly one cluster, whereas it will be soft when each object may belong to more than one cluster and have a fractional degree of membership in each cluster. There are three common types of clustering algorithms which are agglomerative, divisive, and shuffling. Starting with each object in a separate cluster, the agglomerative algorithm merges clusters until the criterion for stop is met. By contrast, the divisive algorithm starts with all objects stored in one cluster, and then split it into clusters until stopping criterion is satisfied. The shuffling algorithm redistributes objects into clusters.

The most commonly used clustering techniques are k-Means method and Expectation-Maximization (EM) [5]. Both of them are spatial clustering techniques. However, k-Means is hard, flat, and shuffling while EM is soft, flat, and probabilistic. Unlike k-Means, EM is scalable and allows clusters to be of arbitrary size and shape [3].

3 Experiment and Preliminary Results

The goal of this research was to extract knowledge from surveys conducted after students had accomplished each lab, performing on the GIS learning system. Feedbacks from open-ended questions in the surveys were taken as documents. Thus, each document of the collection to be analyzed in this research contained a few sentences. The nature of the data was not too general since the questions were based on the GIS labs. This led the answers possibly contain jargons as well as specific terms used in the learning system. On the other hand, students could freely express their opinions in the answers which could be anything and then made vocabulary uncontrolled. SAS Enterprise Miner was a tool being used throughout the research.

3.1 Pre-processing

The original data was gathered electronically as texts and no data type conversion was needed. However, it was prepared into a suitable format for feeding into the text miner program during preparatory processing. A CSV file was constructed where rows contained a list of documents and columns represented the author (i.e., the name of the student) of each text and the text itself. The blank cell showed the missing comment from the student.

After the input data file was prepared, the model for text mining was created as a diagram in SAS Enterprise Miner, displayed in Figure 1. The left node was an Input Data node where the data file was imported into and the right node was a Text Miner node where text mining process would be performed to explore information in the document collection. Both nodes were made connected via a line. The direction of the arrow represented the flow of data.



Fig. 1. A Model for Text Mining

The input data was fed into the text mining process. SAS Enterprise Miner took care of tasks automatically based on parameter settings.

A synonym list has been modified from the default provided by SAS Enterprise Miner. A part of the list is shown in Figure 2.

	TERM	PARENT	CATEGORY
1	application	software	noun
2	computer	software	noun
3	program	application	noun
4	gi	gis	noun
5	hands on	hands-on	adj
6	hands-on	interactive	adj
7	interactive	practical	adj
8	borrow sites	borrow site	noun
9	borrow sites	site	noun
10	borrow site	site	noun

Fig. 2. A Part of the Modified Synonym List

Computer, program, application, and software were defined as synonyms. Moreover, the list handled misspellings by defining the correct spelling as the parent of each misspelled word. For example, the fourth and fifth rows of the list in Figure 4 were created since students sometimes spelled 'gis' to 'gi' or forgot a hyphen in 'hands-on'.

Also, a term-document frequency matrix was derived based on the parameters set for the Text Miner node. Table 1 shows some of key parameter settings.

Table 1. Parameter Settings for Term-Document Frequency Matrix Conversion Stage of Pre-Processing

Property	Value
Compute SVD	Yes
SVD Resolution	High
Max SVD Dimensions	100
Scale SVD Dimensions	No
Frequency weighting	Log
Term Weight	Entropy

To improve performance, SVD was computed with high resolution. The higher resolution yields more SVD dimensions, which summarizes the data set better while require more computing resources. The number of SVD dimensions should not be too small to lose concepts and should not be too large to keep noise. Dumais [4] performed information retrieval and found that performance increased over the first 100 dimensions, hitting the maximum, and then falling off slowly. Also, the higher number had been tested in this experiment, but yielded no difference in results. Thus, 100 seemed to be a good start for the maximum number of SVD dimensions. Moreover, since the vocabulary of the collection was not too general and not too controlled, it fit in between Binary and None frequency weights. Therefore, Logarithm was an appropriate frequency weight applying here. Furthermore, the three term weighting techniques were tested which were Entropy, GF-IDF, and None.

3.2 Clustering

Clustering technique was applied to cluster comments from students into clusters. Table 2 shows some key parameter settings for this process.

Table 2. Parameter Settings for Clustering of Core Mining Processing

Property	Value
Automatically Cluster	Yes
Exact or Maximum Number	Maximum
Number of Clusters	10
Cluster Algorithm	EXPECTATION-MAXIMIZATION
Descriptive Terms	7
What to Cluster	SVD Dimensions

Automatically cluster was enabled to allow clustering on the data set. The number of clusters was unknown; thus, it was not possible to define the exact number of clusters. The maximum number was set to 10 since the document collection was small and 10 clusters should be sufficient to cover all ideas. Expectation-maximization (EM) clustering technique was being used. The number of descriptive terms was set to 7 which is reasonable for the size of data. Clustering worked on the term-frequency matrix after dimensional reduction (i.e., SVD) had been applied.

After all required parameters were set appropriately, the Text Miner node was run. Finally, the resulted clusters from text mining, with three different term weighting algorithms, were shown in Figures 3, 4, and 5. The survey comments were collected from 52 students who responded to the question “Please list ways in which the lab activity that covered air pollution sources and transport could be improved.”

Clusters				
#	Descriptive Terms	Freq	Percentage	RMS Std.
1		8	0.153846153...	0.0
2	+ instruction, explanation, + good explanation, why, + good, + do, more	9	0.173076923...	0.1470874...
3	could, + site, would, fill, lecture, + have, on	11	0.211538461...	0.1502967...
4	+ easy, over, air, less, lab, + make, + short	16	0.307692307...	0.1516472...
5	answer, no	4	0.076923076...	0.1481622...

Fig. 3. Clusters from Text Mining with Entropy Term Weighting (Log_Entropy)

Clusters				
#	Descriptive Terms	Freq	Percentage	RMS Std.
1		8	0.153846153...	0.0
2	+ step, + show, on	4	0.076923076...	0.1080237...
3	+ site, would, fill, could, + have, on, + make	6	0.115384615...	0.1489361...
4	depth, spend, understand, in, more, + software, + easy	13	0.25	0.1409576...
5	answer, no, + have	3	0.057692307...	0.1161862...
6	do, out of, + complete, + good explanation, + learn, + not, lab	12	0.230769230...	0.1448537...

Fig. 4. Clusters from Text Mining with GF-IDF Term Weighting (Log_GF-IDF)

Clusters				
#	Descriptive Terms	Freq	Percentage	RMS Std.
1	+ not, + step, + short, + show, + easy, why, + make	29	0.557692307...	0.1329294...
2	depth, answer, spend, more, in, + instruction, over	18	0.346153846...	0.1521579...

Fig. 5. Clusters from Text Mining with IDF Term Weighting (Log_IDF)

The three weighting methods produced different clusters with different descriptive terms as we as different frequencies in each cluster. Notice that the total number of comments was 52, but the sums of frequencies from all clusters did not reach 52, 48 for Log_Entropy, 46 for Log_GF-IDF, and 47 for Log_IDF. It suggested that some comments were not able to be classified in any cluster. Those comments were ones considered as outliers. Also, there is a cluster carrying no descriptive term. Not only blank comments, comments containing few key terms were classified in this type of clusters.

From the results, human effort is required in order to construct sentences from the resulted key terms to interpret meanings of each cluster. For example, “easy, over, air, less, lab, make, short” may convey the ideas “Make the Air lab easier, shorter with fewer (less) steps.” Also, words are stemmed, so analysts have to guess which form should be used to construct the sentence. For instance, “easy” can be “easy” or “easier”. The author can either think that the lab is easy, or the lab should be made easier. This variation can change the meaning of the cluster. Moreover, some words which inverse the sentiment of the statement such as ‘not’, ‘never’, and ‘rarely’ are not identified which term is their pair. Thus, interpretation might be misled if the text miners or analysts are not familiar with the domain and document collection. One of the future works for this study is to consult with domain experts to appropriately interpret the meaning of each cluster and to suggest which weighting methods provide the most reasonable clustering outcome.

4 Conclusion

In text mining, the pre-processing stage is very important and dominates the entire process. There are several techniques available to be applied in text mining. Text miners have a chance to adopt or ignore techniques based on the nature of their data sets. From the experiment in applying text mining in survey comments, text mining is able to cluster comments into clusters without pre-defined labels. Attached with each cluster is a set of descriptive terms which summarizes the idea of each one. Analysts are able to read only these descriptive terms, instead of all documents, to obtain the ideas of the entire collection.

Text mining captures only important terms which represent the main focuses or concerns of the document collection as a whole. Some terms which occur only in a few documents will not be included in any specific cluster. The ideas which differ from majority can hardly be captured by text mining. Sometimes, those ideas are important and might be useful since they capture issues which others fail to concern.

Finally, future works may include effective approaches to evaluate the clustering results. Also, sentiment analysis might be helpful for evaluators as well. Classifying the comments by sentiment, not by category, predicts whether each comment from students is positive or negative. This assists system evaluators and developers to understand students’ satisfaction towards the learning system.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern information retrieval*, vol. 463. ACM press, New York (1999)
2. Berry, M.W., Browne, M.: *Understanding search engines: mathematical modeling and text retrieval*. Soc. for Industrial & Applied Math. 8 (1999)
3. Bradley, P.S., Fayyad, U., Reina, C.: *Scaling EM (expectation-maximization) clustering to large databases*. Microsoft Research (1998)

4. Dumais, S.T.: Improving the retrieval of information from external sources. *Behavior Research Methods* 23(2), 229–236 (1991)
5. Feldman, R., Sanger, J.: *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press (2006)
6. Luna, R.: Learn-Civil-GIS. Retrieved March 2010, from Learn-Civil-GIS.org (2007), <http://learn-civil-gis.org>
7. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5), 513–523 (1988)