4th Construction Specialty Conference
4e Conférence spécialisée sur la construction

Montréal, Québec
May 29 to June 1, 2013 / *29 mai au 1 juin 2013*

Using Text Mining to Predict Construction Project Cost Overruns

Trefor P. Williams[1]
[1]Center for Advanced Infrastructure and Transportation, Rutgers University

Abstract: Text data from a sample of competitively bid California highway projects have been analyzed and used to make a prediction of a construction projects likely level of cost overrun. A textual description of the project and the text of the two largest project line items were used as input. The text data were converted to numerical attributes using text-mining algorithms. Classification rules were produced using the Ridor classification algorithm. Results of the modeling effort showed that the text data provides useful information for predicting a projects outcome

1. Introduction

Many factors affect construction cost overruns. Possibly exact numerical predictions of the completed costs of a construction project are not possible due to the multitude of factors that can affect a projects final outcome. Possibly there are indications provided at the time of the bidding that can show a construction projects potential to have large cost overruns. Numeric data are available at the time of the bid opening that may provide some information about the projects ultimate cost. The project magnitude, the number of bidders, the relationship between the low bid and the other bidders, the number of project line items and concentration of project costs in particular activities can be calculated at the time of the bid opening. It has now become possible to apply various data mining methods to project data to attempt to make predictions. However, the accuracy of the predictions using numerical data can be affected by a multitude of project events and factors that occur after the bid opening, such as extensive scope changes, constructability problems, design errors and poor project management.

With advances in text and data mining it is now possible to study text data. Possibly the text data available can provide indications of a projects constructability and its likelihood of experiencing increases from the original low bid that can augment numerical indicators. In the bidding documents it is possible to find text that provides a summary of what is to be constructed, its location and the required bid items that must be constructed to complete the project. It is now possible to automatically find indications of the projects nature and its difficulty of construction from text descriptions of a projects type. In this paper the possibility of using text mining to make predictions of project costs will be explored. Text mining is often defined in the context of discovering previously unknown information that is implicit in the text but not immediately obvious (Miner et al. 2012) In this case we are attempting to extract information about the different types of projects conducted and the relationship of the project type and the construction methods used to cost overruns. We will study if project text descriptions contain useful information that can be used to predict the projects level of cost overrun.

1.1 Related Text and Data Mining Research in Construction

There have been several applications of the use of classification and data mining to construction management problems. A prototype system that automatically classifies construction documents

according to project components using data mining techniques was proposed by Caldas et al. (2002). Soibelman and Kim (2002) addressed the need for data mining in the construction industry, and the possibility to identify predictable patterns in construction data that were previously thought to be chaotic. In that study, a prototype knowledge discovery and data mining (KDD) system was developed to find the cause of activity delays from a U.S. Army Corps of Engineer's database called the Resident Management System. Soibelman et al. (2008) have addressed the need to develop additional frameworks that allows the development of data warehouses from complex construction unstructured data and to develop data modeling techniques to analyze common construction data types. Mahfouz (2011) has used the SVM algorithm to classify construction text documents. Existing construction text mining research has focused on methods of classifying documents and extracting information from databases. This paper extends the use of text mining to predictive problems.

2. Bidding Data for Analysis

Data for this analysis was collected at random from the website of the California Department of Transportation. All of the projects were bid in 2006. Data from 309 competitively bid highway projects were collected. The bid opening data and the completed cost after change orders were collected. The project low bids ranged between $99,969 and $1,434,085,935. The range of cost overruns/under runs for these projects was -59.75 to +65.62%. The average project was completed at a price -2.51% less than the original bid amount. The project data collected varied widely in cost magnitude and type of construction. Some projects were maintenance projects while others were major rehabilitations or new construction.

Project	Text Description
1	WIDEN ROADWAY IN PLACER COUNTY IN AUBURN FROM FULWEILER AVENUE TO WILLOW CREEK AVENUE ASPHALT CONCRETE (TYPE A) MODIFY SIGNAL AND LIGHTING
2	REPAIR EROSION AND SPALLS IN LOS ANGELES COUNTY IN COMPTON AT ACACIA AVENUE UNDERCROSSING AND AT COLLEGE OVERHEAD COLUMN CASING TEMPORARY SUPPORT
3	MAINTENANCE VEHICLE PULLOUTS IN SAN MATEO AND SAN FRANCISCO COUNTIES AT VARIOUS LOCATIONS 2SM STRUCTURE EXCAVATION (RETAINING WALL) (TYPE Z-2) (AERIALLY DEPOSITED LEAD) ASPHALT CONCRETE (TYPE A)
4	PLANTING AND IRRIGATION IN SAN LUIS OBISPO COUNTY AT VARIOUS LOCATIONS PLANT ESTABLISHMENT WORK MOBILIZATION

Table 1 Input to the text-mining model

2.1 Description of the Text Data

The text data collected were taken from the bid summary for each project. The bid summary information contains a short description of the project and the work to be performed. This data included the location of the project. Table 1 shows a sample of the project descriptions. To provide additional information, the textual descriptions of the two largest project line items by dollar amount in the bid summary form were appended to the project description for each project. This was done to provide more information that could allow various project types to be differentiated by the data mining algorithms.

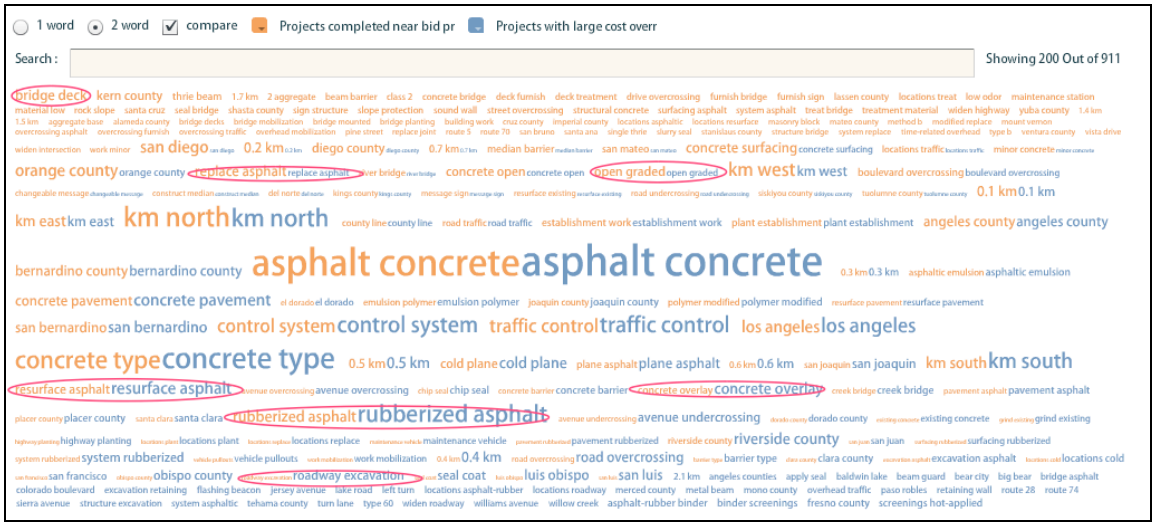

Figure 1: Tag Cloud Showing Comparison of Word and Word Pair Frequency for Projects with Cost Overruns and for Projects Completed Near the Bid Price or with an under run.

Table 2: The Percentage of Selected Word Pairs in the text files

Word Pair	Projects with Low Overruns		Projects with High Overruns	
	Number	% Of Text	Number	% Of Text
Replace asphalt	6	0.33	2	0.14
Open graded	5	0.27	2	0.14
Replace Asphalt	6	0.27	2	0.14
Concrete Overlay	4	0.22	6	0.43
Rubberized Asphalt	8	0.44	14	1.00
Roadway Excavation	1	0.05	4	0.29

2.2 Examination of the Text Using a Tag Cloud

To provide an initial analysis of differences between texts of projects having low or negative cost overruns and those with large cost overruns a comparative tag-cloud was constructed using the Many Eyes web-based data visualization tools developed by IBM (http://www-958.ibm.com/software/analytics/manyeyes/). A tag cloud is a visualization of word frequencies. The available text data were divided into two data sets.

The first data set contains only text from projects that were completed at greater then 5% of the original bid price. The second data set contains text from projects that were completed for less then a 5% cost overrun or for less then the original bid price. Figure 1 shows the comparative tag cloud for frequently occurring word pairs in the two data sets. The size of a word indicates its percentage frequency in the text file. Table 2 provides a more detailed comparison of the word pairs that are circled in the figure. Differences can be observed between the two textual data sets. For instance the word pair “bridge deck” only occurs in projects with low cost overruns and “roadway excavation” occurs in projects with cost overruns much more frequently in projects with high overruns then it does in projects with low overruns or under runs.

Table 2 indicates that projects with certain word pairs, such as “rubberized asphalt” in the textual description are more likely to have large cost overruns. The “rubberized asphalt” word pair encompasses

One percent of the text in the large cost overrun free-form text file while the word pair represents only 0.44% of the text in the small cost overrun file. Possibly these word pairs represent work tasks that are more difficult for contractors to perform within budget constraints, or are less familiar to contractors then more conventional construction techniques.

3 Using Text and Data Mining with the Bidding Data

3.1 The Rapid Miner Software

The models were constructed using Rapid Miner. The Rapid Miner software is an open-source data and text mining system that is widely used to build predictive models. The software incorporates powerful tools for data manipulation, data mining, and text mining. Software for many different types of data mining algorithms are available for experimentation. The Rapid Miner software also includes all of the algorithms from the Weka open source data mining software program (Hall et al. 2009). Rapid Miner uses a building blocks approach that allows models to be developed without the need for extensive programming. The data mining paradigms used in this research were the Ridor classification algorithm developed for Weka and text mining algorithms to manipulate and transform text into a numerical format useable by Ridor. Figure 2 shows the Rapid Miner model constructed to predict the level of cost overrun.

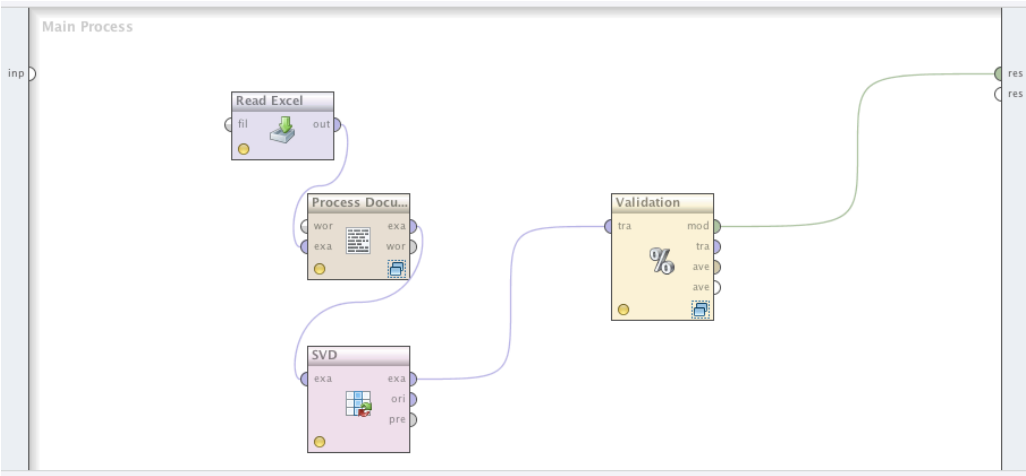

Figure 2 The Rapid Miner Model

3.2 Text Mining

The purpose of text mining is to transform text into numeric attributes that can then be used in data mining algorithms. For each project in the database of California projects, the available text was transformed into a vector that provides a numerical representation of the information in the text. In this case we are attempting to extract information about the different types of projects conducted and the relationship of the project type to cost overruns.

The text for each project must be transformed into a numerical vector that is suitable for use with a data-mining algorithm. There are several steps that are necessary to transform the unstructured text for each project into a standardized numeric vector. These steps are tokenization, stopping, stemming, normalization and vector generation (Weiss et al. 2010). Figure 3 shows the transformations applied to the text data using several text processing functions of the Rapid Miner software (Miner et al. 2012). The data transformation steps of the project description text performed by the Rapid Miner software can be detailed as:

- Transform Cases. Uppercase letters were removed from the text.
- Tokenize. The unstructured text is transformed into a sequence of tokens. Tokens can take on different forms, however in this model the tokens were equivalent to single words. For this model the term frequency was used to scale a tokens value.
- Stemming. In this data transformation related word tokens are normalized into a single form. For example “walking” would be transformed to “walk” (Miner et al. 2012). Stemming has been found to increase the accuracy of classification and clustering data mining algorithms.
- Filter Stopwords Common words like “and” and “but” are removed by removing words on a predefined list.
- Filter Token Length. This filter removes words that are less then three characters long.
- Generate n-gram terms. In this processes, Rapid Miner has been set to allow two word terms to be entered in the term-document matrix that is generated for the text. In other words, token pairs can be combined to produce additional tokens.

After these transformations are completed a matrix of projects and terms is created. After the matrix was created, Singular Value Decomposition, a dimensionality reduction method, was used to transform the matrix of projects and terms into a single numeric value for each project.

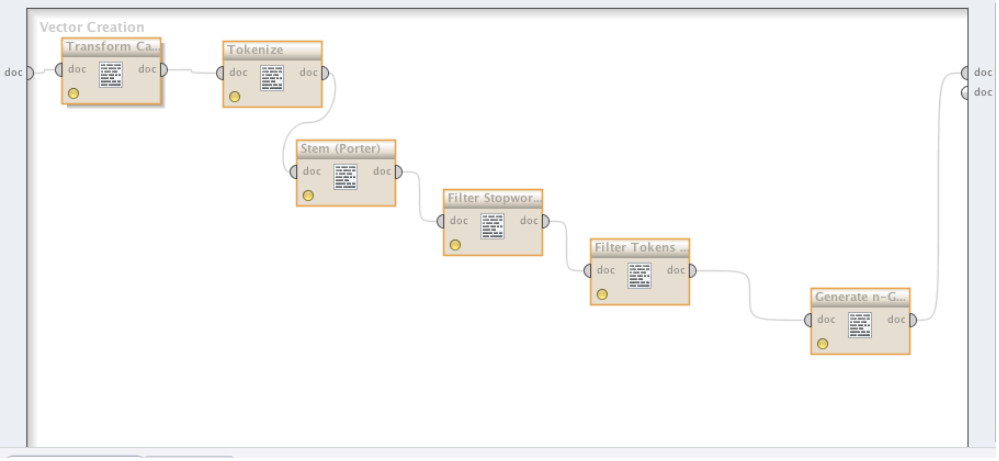

Figure 3 The Text Mining Steps included in the “Process Documents” Block

3.3 The Ridor Classification Algorithm

The transformed text data were used as input to classification models that attempt to predict the level of project cost overrun to be expected at project completion. Classification can be defined as the task of assigning objects to one of several predefined categories (Tan et al. 2006). Here the object the textual description of a highway construction project transformed into a numeric variable and the predefined categories are the level of the completed project cost overrun. Several different classification algorithms are provided in the Weka System that is implemented in the Rapid Miner software. Through experimentation the best-performing algorithm for predicting large project cost overruns were found to be the Ridor algorithm.

3.4 Ridor Algorithm

The Ridor classification algorithm was applied to the transformed text data. The Ridor (Ripple Down Rules) algorithm is a machine learning technique that automatically generates rules from a data set (Gaines and Compton 1995). The Ripple Down Rules algorithm was initially developed to maintain rules for one of the first medical expert systems, and is now routinely used in chemical pathology laboratories to provide interpretive comments to assist doctors in using medical laboratory reports Compton et al.

2006). The Ridor algorithm learns rules with exceptions by generating a default rule. The default or top-level rule is the class of the output that occurs most frequently. Then the algorithm uses incremental reduced-error pruning to find exceptions with the smallest error rate, finding the best exceptions for each exception and iterating (Witten and Frank 2005).

3.5 Bootstrapping

Training and validation of the Ridor models was done using a boot strapping technique. Bootstrapping is a resampling process where a procedure, in this case the generation of the Ridor rules, is performed several times with a different random sampling of the data to produce more accurate predictions. A scheme was employed where seventy percent of the cases are used to train the model and thirty percent are used as test cases. The test cases are selected at random. The model is trained and predictions are produced using the test cases as inputs. This process is repeated 10 times with the training and test cases selected randomly from the data set for each model run. The confusion matrix output (Tables 4 and 5) shows all predictions made by the ten model runs.

4 Predictions Using Text Mining and Classification Rules

4.1 Different Data Sets for Experimentation

Using the Ridor algorithm two different input data sets were used. One hundred forty eight projects (47.90%) had under runs larger then -2.5%. Over 50 projects had under runs larger then -10%. It can be conjectured that many of these projects may have had large under runs due to project scope changes that occurred after the bid opening or some unusual event the modified the nature of the project. Possibly, the text for projects that were reduced in magnitude due to a scope change may provide contradictory information i.e. the initial descriptive text from the bid opening no longer describes the magnitude or type of project that was actually performed. Therefore, two different trimmed data sets were created.

Two data sets were created that excluded all of the cases with cost under runs of greater then -5% and -2.5%. In these models it was attempted to predict three levels of cost overrun based on the input text. The data set including cost overruns up to -5% had 196 cases. The data set that included projects up to a -2.5 % under run had 161 cases. Table 3 shows the range for each overrun/under run category. It indicates that category one is the projects with under runs, category two had medium over runs, and category three had high over runs.

Table 3 Data Set Compositions

Category	Data Set 1	Data Set 2
1	-5%>Under run<0	-2.5%>Under run<0
2	0>Overrun<5%	0>Overrun<5%
3	Overrun >5%	Overrun >5%

5 Modeling Using Rapid Miner

Table 4 shows the predictions made when projects greater then -2.5% composed the input data. This is a confusion matrix where all of the cases would fall on the diagonal axis if all the predictions were correct. The three level predictions produced interesting results particularly in predicting large cost overruns. The class recall shows the percentage of time that a project with a particular level of cost overrun is correctly predicted. The precision indicates the accuracy of the predictions i.e. the percentage of projects where a correct prediction is made.

The predictions made were 35.75% plus or minus 2.68% accurate. However, the model generated interesting results in predicting projects with large overruns. Predicted level 3, the projects with large cost overruns were correctly recalled 54% of the time and had a precision of 35.0%. The precision is percentage of all projects predicted to be large overruns by the Ridor rule that have been correctly labeled. The recall is the percentage of large cost overrun projects that are correctly labeled. Projects near the original bid estimate were poorly predicted. Costs under runs were predicted with a precision of 42%.

Table 4. Prediction using all projects with greater then -2.5% cost overrun

Overrun Level	*True 3*	*True 2*	*True 1*	*Class Precision*
Predicted 3	231	149	285	0.35
Predicted 2	40	14	36	0.16
Predicted 1	160	89	178	0.42
Class Recall	0.54	0.06	0.36	

In order to improve the predictions, a larger data set was used to produce Ridor rules for the cost overruns. The second set included more under run projects up to a -5% under run. Table 5 shows the predictions produced. Figure 4 shows the Ridor Rules for this prediction. Using this data set the predictions made for overrun level three (Large Cost increase) and level 2 (Near). Large projects were predicted correctly 58% of the time.

Table 5 Predictions using all projects with greater then -5% cost overrun

Overrun Level	*True 3*	*True 2*	*True 1*	*Class Precision*
Predicted 3	349	321	137	0.43
Predicted 2	190	192	66	0.43
Predicted 1	65	53	15	0.11
Class Recall	0.58	0.34	0.07	

```
Overrun = 2.0  (196.0/119.0)

        Except (svd_1 <= -0.037672) => Overrun = 3.0  (90.0/32.0) [43.0/11.0]

                Except (svd_1 <= -0.08513) => Overrun = 1.0  (27.0/15.0) [9.0/4.0]

                Except (svd_1 > -0.047001) => Overrun = 1.0  (13.0/8.0) [4.0/2.0]

                Except (svd_1 <= -0.063167) and (svd_1 > -0.073689) and (svd_1 <= -0.071218) =>
                  Overrun = 1.0  (3.0/1.0) [1.0/0.0]

        Except (svd_1 > -0.026924) and (svd_1 <= -0.0225) => Overrun = 3.0  (11.0/1.0) [6.0/2.0]

Total number of rules (incl. the default rule): 6
```

Figure 4 Automatically Generated Ridor Rule

6 Analysis of the results

The output shows that textual descriptions of a construction project can provide an indicator of a projects level of cost overrun. Even with the small data sets used, there were discernable differences in the descriptive text for projects that have higher levels of cost increases and the projects that were completed at or below the original bid amount. The results indicate that a projects likelihood of cost overruns is linked to the nature of the project, and the types of construction methods used.

7 Conclusions and Future Work

The results indicate that textual descriptions about the nature of a project, and project line item text can yield useful information about a projects anticipated level of cost overrun. The results of this initial investigation indicate that additional research into the use of textual information as a predictive tool of project outcomes is warranted and that models should be built using a larger database of projects to determine if prediction accuracy can be improved. Other types of data mining algorithms besides rule-based classifiers can be studied. In addition, the information gleaned from text mining can be combined with additional numeric variables to potentially produce more accurate predictions.

References

Caldas, C., Soibelman L., and Han, J. 2002. Automated classification of construction project documents. *Journal of Computing in Civil Engineering*, 16(4): 234–243.

Compton, P., Peters, L., Edwards, G., and Lavers, T. G. 2006. Experience with Ripple-Down Rules. *Knowledge-Based Systems*, 19: 356-362.

Gaines, B. R., and Compton, P. 1995. Induction of ripple-down rules applied to modeling large databases. *Journal of Intelligent Information Systems*, 5(3): 211-228.

Hall, M., Frank, Holmes, G., Pfahringer B., Reutemann, P., and Witten, I. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11 (1).

Mahfouz, T. 2011. Unstructured Construction Document Classification Model Through Support Vector Machine (SVM). *International Workshop on Computing in Civil Engineering,* ASCE, Miami, FL.

Miner, G., Elder IV, J., Hill, T., Nisbet, R., and Delen, D. (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, Waltham, MA.

Soibelman, L., and Kim, H. (2002). Data preparation process for construction knowledge generation through knowledge discovery in databases. *Journal of Computing in Civil Engineering*, *16*(1): 39-48.

Soibelman, L., Wu, J., Caldas, C., Brilakis,I. and Lin, K. 2008. Management and Analysis of Unstructured Construction Data Types. *Intelligent Computing in Engineering and Architecture* 22(1): 15–27.

Tan, P. N., Steinbach, M., & Kumar, V. 2006. Introduction to Data Mining. Pearson Addison Wesley, Boston.

Weiss, S., Indurkhya, N. and Zhang, T. 2010. *Predictive text mining: a practical guide*. Springer-Verlag, London.

Witten, I., and Frank, E. 2005. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann Publishers, San Francisco.