Predicting Award Prices of First Price Sealed Bid Procurement Auctions

Fabian Blasch

Supervisor: Dr. Katharina Fenz

July 18, 2022

Contents

1 Introduction

2 Data

All the information about the procurement contracts, is obtainable through the bid tab archive on the official website of the Colorado Department of Transportation. The information is provided in PDF documents. In each of those documents the following information of the respective auction is provided.

- A table listing all submitted bids, including a unique identifier for each of the participating bidders

- A contract description

- An engineer's estimate

- The contract ID

- The letting date

- Either the amount of time given to complete all the contractual obligations, or a completion due date

- The county in which the contract is to be completed in

For illustrative purposes, Figure 1 displays an example of a bid tab, in particular the second page, which contains the vendor ranking as well as the contract description and the remaining information listed above.

| Colorado Department Of Transportation | | | | Printed On: | 11/17/2015 |
| Vendor Ranking | | | | | Page 1 of 1 |

Letting No:	20151112	Contract ID:	C19868	Project(s): STU1211-084	
Letting Date:	November 12, 2015	Region:	1		
Letting Time:	10:00 AM	Contract Time:	260 WORKING DAYS	Counties:	JEFFERSON, REGION 1

Contract Description:
SH121(WADSWORTH)-HIGHLAND DR-10TH AVE-JEFFERSON CO

THIS PROJECT IS LOCATED ON WADSWORTH BETWEEN HIGHLAND AND 10TH.

CONSTRUCTION WILL INCLUDE A FULL CONSTRUCTION WITH WIDENING OF ONE LANE IN BOTH DIRECTIONS, AND A MULTI MODAL TRAIL ON BOTH SIDES. THE MAINLINE PAVING WILL BE CONCRETE. THE WORK ALSO INCLUDES A CONCRETE BOX CULVERT NEAR HIGHLAND TO CARRY LAKEWOOD GULCH UNDER WADSWORTH.

CDOT WILL ONLY BE ACCEPTING ELECTRONIC BIDS FOR THIS PROJECT. PLEASE CONTACT BID EXPRESS CUSTOMER SERVICE AT 1-888-352-2439 TO OBTAIN AN ACCOUNT IF NECESSARY.

Rank	Vendor ID	Vendor Name	Total Bid	Percent Of Low Bid	Percent Of Estimate
0	-EST-	Engineer's Estimate	$9,821,027.20	91.58%	100.00%
1	870A	SEMA CONSTRUCTION, INC.	$10,723,550.00	100.00%	109.19%
2	884A	HAMON INFRASTRUCTURE, INC.	$10,817,000.00	100.87%	110.14%
3	1275A	CASTLE ROCK CONSTRUCTION COMPANY OF COLORADO, LLC	$10,817,845.03	100.88%	110.15%
4	065A	CONCRETE WORKS OF COLORADO INCORPORATED	$11,614,565.78	108.31%	118.26%
5	232A	AMERICAN CIVIL CONSTRUCTORS, INC. dba ACC Mountain West	$12,338,888.00	115.06%	125.64%

Figure 1: Bid Tab Example

2.1 Scraping

In order to obtain all the archived bid tabs, the html code of the website was first examined using a google chrome extension called SelectorGadget. This tool allows one to identify html nodes, that website contents are associated with. In the case of the bid tab archive, the html node carrying the links to the individual bid tabs is "<td a>". Once this html node is discovered and the consistency across different years in the archive is ensured, the download is easily achieved by looping over the links and downloading the respective PDFs. The hyperlink extraction was performed utilizing *rvest*, by Wickham (2022). For the remaining steps in the data extraction process, a distinction will be made for text based information and tabular data.

2.1.1 Text Based Information

The structure of the text based information allows us to filter the individual parts via regular expressions. Especially, for the letting data, the contract ID, and the county this required no further data cleaning steps. Unfortunately, this is not the case for the contract time and the contract description.

The contract time was not as straightforward to obtain, since the way it is reported is inconsistent across documents. Most of the time, it is reported as working days until all contractual obligations have to be fullfilled. Seldom, however, the bid tab contains a completion date instead. Accordingly, to achieve consistency across documents all completion dates were converted to contract time. This was achieved by first adding 60 days to the letting date, as this is the number of days that the Cdot reports as the expected time between the letting date and the start of the work on site. Then, the difference in days between the completion date and the starting date were computed. As, said difference is only supposed to contain working days the following holidays as well as all weekends were substracted from the difference betwen starting date and completion date.

- New Year's Day

- Dr. Martin Luther King, Jr. Day

- President's Day

- Memorial Day

- Juneteenth

- Independence Day

- Labor Day

- Frances Xavier Cabrini Day

- Veterans Day

- Thanksgiving

- Christmas

The computation was executed utilizing the R package *bizdays*, Freitas (2022). The package enables the user to generate custom calenders. The difference in starting and completion date was therefore easily calculated by setting up a custom calender with the holidays listed above as well as all saturdays and sundays. Then using this calender, the difference between two dates will only take working days into account. The only remaining text based information is the contract description. So far, none of the text based information required extensive preprocessing to obtain variables that can be represented in a tabular format. In the case of the contract description this is not the case. In order to convert the contract description into a format that may be represented in a table, the descriptions were first tokenized. Tokenization refers to splitting the input text into single unique words, i.e., splitting the sentences on spaces and removing all forms of punctuation. The result is then a vector of tokens. Said tokens were then scanned for spelling mistakes utilizing the R package *hunspell* (Ooms, 2020). Once the misspelled words were corrected, stopwords were removed from the list of tokens. Stopwords are words that have no inherent signal associated with their use, examples for such words in the english language would be "a", "is" and "the". In natural language processing there is not necessarily one list of stopwords, depending on the context different libraries of stopwords may be used to remove as much noise as possible from textual data while leaving the signal associated with a series of words in tact. In the case of this thesis, a combination of five different libraries of stopwords was used. All of those libraries, "snowball" , "stopwords-iso", "smart", "marimo" and "nltk" are available through the R package *stopwords*, by Benoit et al. (2021). After filtering out the stopwords, the remaining words were then stemmed. Stemming refers to the process in which a word is reduced to it's root. This means, that words that carry an identical signal are reduced to the same shortest common substring. Consider the following three

words, replacing, replaced, replacement. All those words carry the information that something needs replacement. The language specific circumstances that determin the affixes are not relevant for the information extraction and thus all the aforementioned words are shortened to "replac" Silge and Robinson (2017). After stemming, to remove any remaining misspelled words and also for potential removal of unwanted information, all stemmed words were written to an excel file and checked manually. Given that all stopwords were already removed and the remaining words were reduced to their stem, this was a very feasible task, resulting in a file with around 2000 words to check. Below we observe the top 40 most frequent words that result from our text mining endeavours.
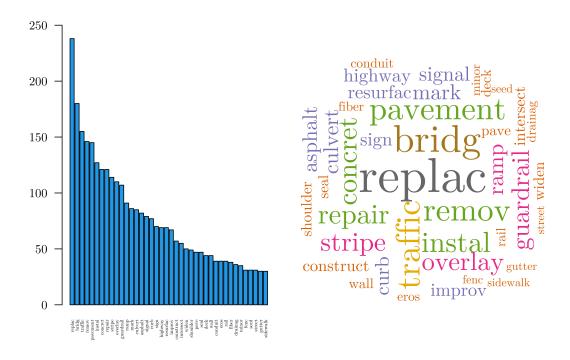

Figure 2: Top 40 Stemmed Description Words

2.1.2 Tabular Data

As displayed in Figure 1, the table in each of the auctions' PDFs contains information on the submited bids, the bidders' identity and an engineer's estimate. To extract the

table containing this information, the package *tabulizer* by Leeper ([2018](#)). This package provides bindings for the Tabula PDF extractor, written in Java. In particular, two functions of the library were combined to write a wrapper for the table extraction. First the function *extract_tables()* was used to attempt automized table detection and subsequent extraction. Unfortunately, however, there are quite a few cases in which the automatic table detection failed because some auctions only have one or two bidders. The resulting tables that summarize those auctions have very few rows and thus the automatic detection does not recognize them as tables. Accordingly, if the output of *extract_tables()* is empty, the implemented wrapper calls *extract_areas()*. This function allows the user to specify an area via the R plot pain to specify where exactly the table is located. Once the location is passed manually, which was necessary for around 10-15% of cases, the extraction worked as intended. To finally, obtain the tables the wrapper was used to loop over the PDFs.

References

Benoit, K., Muhr, D., & Watanabe, K. (2021). *Stopwords: Multilingual stopword lists* [R package version 2.3].

Freitas, W. (2022). *Bizdays: Business days calculations and utilities* [R package version 1.0.11].

Leeper, T. J. (2018). *Tabulizer: Bindings for tabula pdf table extractor library* [R package version 0.2.1].

Ooms, J. (2020). *Hunspell: High-performance stemmer, tokenizer, and spell checker* [R package version 3.0.1].

Silge, J., & Robinson, D. (2017). *Text mining with r: A tidy approach* (1st). O'Reilly Media, Inc.

Wickham, H. (2022). *Rvest: Easily harvest (scrape) web pages.*