NSERC Application Summaries and their Level of Funding

Introduction

The Natural Sciences and Engineering Research Council of Canada (NSERC) distributes hundreds of millions of dollars in awards to tens of thousands of applicants annually. With so much money at stake, the NSERC review process is under constant scrutiny. Some criticisms of the process include; that the process itself is too expensive, that it does not produce quality research, and that it is biased towards a researcher’s reputation rather than the content of their proposal. If these claims are accurate, it could mean that NSERC is implementing a high cost, low return review process. A process that, most concerning, may turn out to be entirely moot. This begs the questions; does the content of an NSERC funding application matter? More specifically, does word usage in application summaries predict grant size? Applying text classification methods to the available data may uncover insights into the distribution of these awards, and perhaps shed light on these criticisms.

Literature Review

One of the most outspoken critics of centralized research funding systems is Alexander Berezin. His 1998 review1 calls such funding systems self-serving old boys’ clubs. He disputes the common belief that governments under-fund research. In fact, he claims the problem to be one of over-funding trivial research. He calls for a simplification of the funding system, suggesting we fund researchers and not proposals, even if it results in lower average funding levels.

A review paper by Richard Gordon and Bryan Poulin (2009)2 suggests that the cost of the NSERC peer review process itself is simply too great. Their calculations suggests that awarding every single qualified researcher a baseline grant, is in fact cheaper than the cost of running the grant review system itself. Their concluding suggestion was to do away with the review process entirely and simply support all eligible faculty members who ask for funds.

In 2013, Jean-Michel Fortin and David Currie3 concluded that scientific impact is only weakly positively related to the amount of NSERC funding received. Further to that, they found that impact per dollar was lower for large grant holders. They suggest that we target a diverse array of funding strategies, prioritizing many smaller researcher projects over large ones.

In their own ways, each of these papers suggests that the funding application review process is unnecessary. These conclusions were reached indirectly, by measuring research outcomes and then attributing them back to the funding review system. Only by analyzing the actual content present in NSERC applications can we directly infer the effect of the review process.

Dataset

The data is provided by the Government of Canada on the Open Data Canada website under the heading “NSERC’s Awards Data”. The target data consists of ten csv files distributed by year (2005-2014). Datasets are available as far back as 1991, however, application summary data was not collected until 2005. Across 2005-2014 there are roughly 240,000 records. Each record represents a unique application for research funding. Datasets contain 35 fields (36 in 2013 and 2014), only two of which are of primary interest; “ApplicationSummary” and “AwardAmount”. Other fields such as “FiscalYear”, “Institution- Établissement”, “ProvinceEN” may be of secondary interest depending on the results of the primary analysis.

“AwardAmount” is an unsigned integer and represents the amount of funding, in Canadian Dollars, which each research application received. There are a large range of funding amounts, from a single dollar, to sums as large as eight figures. This is a full-participation field as there are no missing values in the dataset.

“ApplicationSummary” is a text field. It is a brief summary of the applicant’s proposed research outline written in language that the public can understand. The field has a maximum capacity of 1500 characters. Not every record contains a summary. The text “No summary – Aucun sommaire” will be interpreted as Null.

Approach

**Step 1**

Load, Consolidate, and Explore the Data

**Step 3**

Apply NLP Techniques

**Step 2**

Label each Record with a Classifier

**Step 4**

Build and Evaluate Text Classifiers

Step 1: Load, Consolidate, and Explore the Data

Load data into R directly from the Open Data Canada website. Identify cases of “No summary – Aucun sommaire” in the “ApplicationSummary” field as Null. Align the data schemas of all ten data sets by removing the two additional (irrelevant) fields in the 2013 and 2014 datasets. Combine the ten data sets into a single master set. Filter out records with null values in the “ApplicationSummary” field. Explore the distribution of the “AwardAmount” data. Identify measures of central tendency and measures of spread. Carry this information onto Step 2.

Step 2: Label Each Record with a Classifier

Guided by information in Step 1, use “AwardsAmount” to determine a suitable number of bins, and their appropriate ranges, to use as classifiers. As an example, “Small”, “Medium”, and “Large” awards. Bin the data, attaching a new field to the master set.

Step 3: Apply NLP Techniques

Use R’s text classification packages to implement word tokenization and normalization. Perform operations on “ApplicationSummary” such as; removing spaces, removing numbers, removing stop words, removing punctuation, removing white space, and converting characters to lowercase. Generate a document term matrix for use in Step 4, the implementation of classification techniques.

Step 4: Build and Evaluate Text Classifiers

Split the data into training and test sets. Using the data prepared in Steps 2 and 3, build text classification models from the training data using multiple text classification techniques (leading techniques include Naïve Bayes and Support Vector Machines). Use the models and the test data set to create the predicted data set. Generate confusion matrices to evaluate the predictions of each model. Compare accuracy, precision, recall, etc. across models to determine the optimal method.

Revisions

Two major revisions to the approach were implemented after exploring the data. The first is the filtering out of Application Summaries written in French. NLP and text classification algorithms operate in one language at a time. It would be exceeding complex to split, separately process, and subsequently reintegrated the data into a single model. Secondly, the distribution of the data was found to be heavily skewed. Feedback obtained regarding this issue was to determine where the distribution begins to skew and to eliminate data points above that amount. The result is 67,172 rows of data.

Pre-Results

Both SVM and Naïve Bayes text classification methods were explored in an attempt to find the most appropriate fit for the dataset. It became clear that the Naïve Bayes method was superior in this specific instance. This was due a number of factors.

First, building the SVM model consumed more resources and took longer to process than Naïve Bayes. When running Naïve Bayes, one can build two separate test and training document term matrices. The training matrix can be used to train the model, and the test matrix can be used to evaluate it. By splitting the data before creating the matrices, each one is relatively small. In contrast, SVM requires the test and training data to be derived from the same matrix. One must create the document term matrix first, and then split the data into training and test sets. Due to the nature of creating document term matrices, one matrix derived from 100% of the data is orders of magnitude larger than the sum of two matrices derived separately (from 65% and 35% of the data). When attempting to build a matrix from 100% of the data, every iteration of SVM models simply caused R to become unresponsive. In an attempt to work around this issue, only a single matrix was created using the train data (65%). As a consequence, the model would have to be tested using the same data it was trained with.

Even with this compromise, R could not build a model from a 99% sparsity matrix (as was built using Naïve Bayes). After several failed attempts at various levels, a sparsity set at 70% resulted in a model being built. This means that a word had to appear in 30% of the documents in order to make it into the matrix. This was not desirable, but was necessary to obtain an SVM model. Unsurprisingly, the accuracy of the model was only 39% (with chance being 33%). This was much less than the Naïve Bayes model (detailed evaluation measures will be provided in the Results Section). Considering the model was tested using the same data it was derived from, this result is even less encouraging.

In summary, the size of the data set as well as the availability of computer resources made SVM text classification a challenge. It quickly became apparent that the Naïve Bayes model is the most appropriate method to use in this instance. SVM will not be considered further.

Results

After selecting Naïve Bayes as the preferred classification method, a model was built using 65% of the data. The model was then tested using the subsequent 35% of the data. Figure 1 shows the overall accuracy of the model to be 48.4%. This is significantly better (p < 0.001) than the baseline chance or “No Information Rate” of 33.7%. Further analysis, however, shows that not all classification levels are performing equally. While precision rates are similar, the recall rate for “Medium” ($23,000 – $30,999) awards is much less than that of “Small” (< $23,000) and “Large” Awards (> $31,000). This led me to believe that re-binning the data into 2 factors, Small and Large, might result in a stronger model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Actual | | |  |
|  | 3 Factor | < $23,000 | $23,000- $30,999 | > $31,000 | Precision |
| Predicted | < $23,000 | 4986 | 2910 | 2250 | 0.491425192 |
| $23,000- $30,999 | 1205 | 2001 | 1164 | 0.457894737 |
| > $31,000 | 1727 | 2882 | 4387 | 0.487661183 |
|  | Recall | 0.629704 | 0.256769 | 0.562364 | **48.4% (33.7%)** |
|  | Kappa = 0.225 | |  |  | Accur. (Chance) |

Figure 1 – Confusion Matrix for 3 Factor Naïve Bayes Model

The classification bins were reworked into “Small” and “Large” bins, from “Small”, “Medium” and “Large”. Figures 2 and 3 show the relatively equal distribution of each of these breakdowns. This is by design. The only reason the bars are not exactly equal is because many repeated award amounts exist at the bin cut off values. Each instance of the repeated values falls into only one bin. The new threshold for Small/Large values was $26,000.

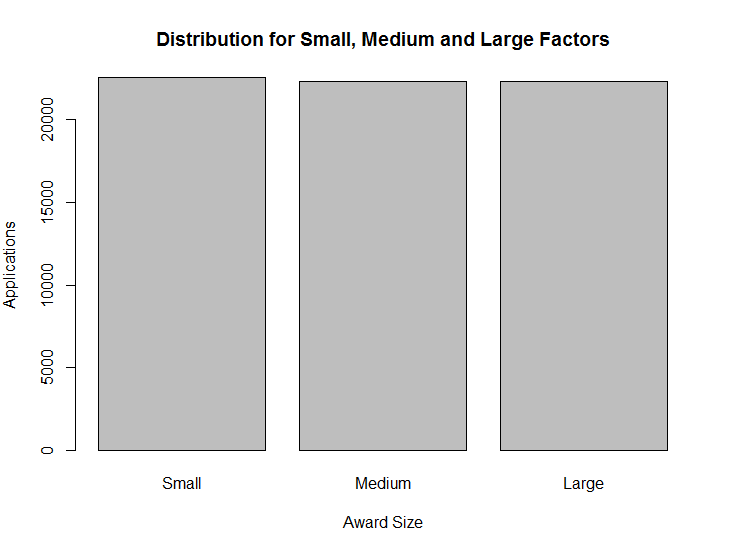


Figure 2 – Distribution for 3 Factor Analysis

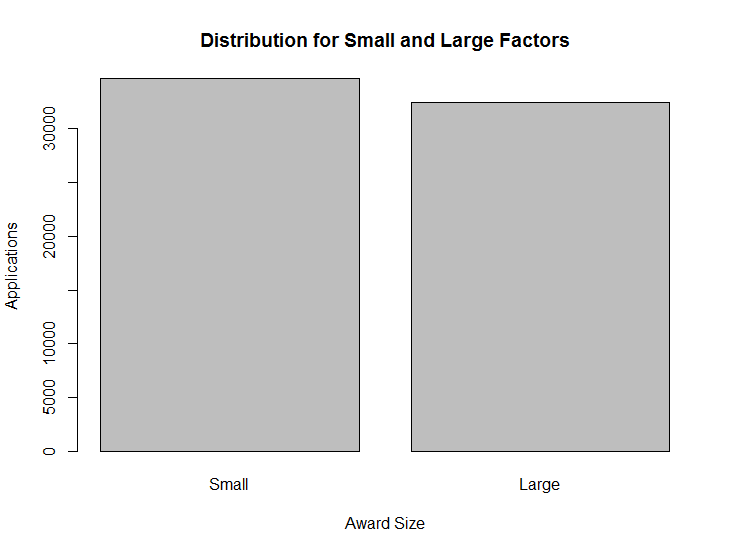


Figure 3 – Distribution for 3 Factor Analysis

The new 2 Factor Naïve Bayes model was constructed using identical test and training data, with the only difference being the new award amount classification split. This model is also significant at p < 0.001 with an overall accuracy of 66.0%, albeit compared to a No Information Rate of 51.6%. Figure 4 shows us that the recall rate across factors is much more balanced as compared to the previous model. Since these models have different baseline chance values, (No Information Rates), they must be compared using balanced accuracy rates. Using this statistic as a comparison, the 2 Factor model outperforms the 3 Factor model with a balanced accuracy rate of 65.9% compared to 61.2%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Actual | |  |
|  | 2 Factor | < $25,999 | > $26,000 | Precision |
| Predicted | < $25,999 | 8498 | 4364 | 0.660705956 |
| > $26,000 | 3621 | 7028 | 0.659968072 |
|  | Recall | 0.701213 | 0.616924 | **66.0% (51.6%)** |
|  | Kappa = 0.319 | |  | Accur. (Chance) |

Figure 4 – Confusion Matrix for 2 Factor Naïve Bayes Model

Summary

The relatively large size of the NSERC data compared to the available computer resources resulted in Naïve Bayes being select over Support Vector Machines as the preferred method of classification. Further to that, the NSERC data is more easily model by splitting the data, relatively equally, into Small and Large awards. Since the model significantly predicts outcomes at a better rate than chance, we can conclude that the words researchers use in their application summaries can be useful in predicting the outcome of their funding amount.

This result is very useful in two ways. First it shows that what a researcher proposes in their application summary does indeed have an effect on the size of the funding they will receive. This means that despite the arguments laid out by critics, the effectiveness and usefulness of the current NSERC selection process does contain merit. Additionally, we only tested two classification techniques. There are many more techniques which were not tested, which could result in stronger predictive models. Further to that, there are also additional revisions that could be applied to the existing Naïve Bayes model. The feedback and revision techniques applied here are surely amateur in nature. I am confident that more improvements can be made to the existing model which would further support the importance of the researcehrs application summaries.

The second way this model can be of use is in a more practical fashion. If this model were to be built into a functional application, aspiring researchers could compose their application summaries, feed them into the model, and determine if their proposal would secure small or large funding amounts. Should their proposal fall into the small category, they could then rewrite their summaries in an attempt to secure larger awards.

Returning to our original research questions; yes, the content of an NSERC funding application does matter. Specifically, the words contained in a research application can be used to reasonably predict the size of the grant to be awarded.

References

# 1 - Berezin A (1998) “The Perils of Centralized Research Funding Systems” *Knowledge, Technology &*

# *Policy. 11(3), 5-26*

# 2 - Gordon R, Poulin BJ (2009) “Cost of the NSERC Science Grant Peer Review System Exceeds the Cost of

# Giving Every Qualified Researcher a Baseline Grant” *Accountability in Research. 16, 13-40*

# 3 - Fortin J-M, Currie DJ (2013) “Big Science vs. Little Science: How Scientific Impact Scales with Funding”

# *PLoS ONE 8(6), e65263. Doi:10.1371/ journal.pone.0065263*