SmartFund: Predicting Research Outcomes with Machine Learning and Natural Language Processing

Alvin Alaphat
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
aalaphat@nd.edu

Meng Jiang
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
mjiang2@nd.edu

***Abstract*—The mission of the National Science Foundation (NSF) is to promote progress of science. It supports fundamental research and education in science and engineering. It provides funding for researchers to explore a variety of disciplines in thousands of institutions across the United States. The outcomes of such research projects vary widely and with much funding being funneled into these projects, it would be efficient and cost-effective to be able to predict these outcomes so the NSF can make informed decisions on the amount of funding they allocate to future projects. We consider a variety of factors that point to general trends and use a combination of natural language processing techniques (such as topic models and phrase mining) and neural networks to predict how many papers and citations will come from potential research projects seeking funding.**

***Index Terms*—prediction, machine learning, natural language processing**

I. Introduction

The National Science Foundation (NSF) is a United States government agency that supports fundamental research and education in all the non-medical fields of science and engineering. The NSF receives over 50,000 such research proposals each year, and funds about 10,000 of them. The review process moves through three phases over the span of 10 months and puts reviewers through an enduring procedure. Much of this lengthy process results from trying to be objective, so the proposal is analyzed by multiple groups of peer reviewers. To help shorten this process and arrive at an objective decision, reviewers can make use of historic data of past projects and state-of-the-art data science techniques to objectively predict how impactful future projects will be.

In order to help program officers and reviewers, we aim at developing computational methods to predict the outcomes of projects based on multiple features such as their award amount, investigators, university, and abstract. We collect past project data from the NSF website to train machine learning models to understand the relationships between these features and predict the outcomes in a quantitative manner. We will quantify the outcomes in two ways: (1) number of papers that were produced in the project and (2) number of citations the produced papers have. Although in certain cases the quality of papers produced is more informative on the broader impact than the quantity of papers, the number of citations will serve as an indicator of how useful the project is to other researchers. Furthermore, generally more papers will create more impact than no papers at all.

Challenges were met while trying to achieve high predictability in the outcomes. The basic profiling features were not strongly correlated with the outcomes, namely the start year, year of expiration, and award amount. These profiling features alone produced a coefficient of determination (R^2) of .207 with the number of papers and of .110 with the number of citations, respectively, through a three-layer perceptron. Linear models generate much lower coefficient of determination (i.e., .097 and .040). We then had to take a closer look at the text data from the project descriptions, such as the title, abstract, and summary. These textual features must be extracted for training machine learning models. Even after employing multiple natural language processing techniques, we discovered that the neural networks easily met over-fitting, so careful feature selection was needed to address the issue.

To address these challenges, we used natural language processing techniques such as the bag-of-words model [1], TF-IDF techniques [2], phrase mining [3], and Latent Dirichlet Allocation based topic modeling methods [4], [5] to extract textual features that can be correlated with the outcome labels. We investigated feature selection strategies to look for the best performance on the neural network framework of multi-layer perceptron regression.

By employing these additional methods to the initial profiling features, we were able to significantly increase the R^2 score of the model to .348 (relatively +68%) with the number of papers and .188 (relatively +71%) with the number of citations, respectively. Errors (i.e., MAE and RMSE) were also significantly reduced. From the topic modeling results, we discovered that certain topics had higher scores and impacted the model more than others. We could achieve explainability in these topic models through keywords painting as a conceptual image. First, projects with abstracts that mentioned computer science (e.g., "computer", "system", "algorithms", "software") fell under the higher scoring topics on the number of papers produced, compared to certain topic of words (e.g., "ocean", "climate_changes", "species"). Second, projects that mentioned broader and educational impacts (e.g., "collaborative_research", "graduate_student", "undergraduate_student") are in the topics that are highly scored on the number of citations the produced papers have, compared to pure mathematics topics (e.g., “algebraic_geometry”, “partial_differential_equations”, “topology”).

The main contributions are summarized as follows:

- We investigate the correlation between the outcomes of NSF-funded research projects and their many factors used in the review process.
- We develop two supervised learning models, linear regression and multi-layer perceptron, to predict unseen outcomes of projects based on the correlated factors.
- Our approach transforms the project description into latent topics and use the topic distribution as a part of project’s features.
- We enhance the topic discovery and feature extraction using phrase mining. Phrases play as important semantic units in the natural language text. We use a distant training technique to automate the phrase mining.

The rest of this paper is organized standardly as follows: Section 2 presents data description and preliminary analysis with profiling features. Section 3 introduces our approach including textual feature extraction, selection, and regression models. Results are given in Section 4. Section 5 presents related work and Section 6 concludes the work.

II. Preliminary Analysis

A. Data Description

The data we are using has been supplied publicly online by the NSF.[1] It includes information over 200,000 past projects dating back to 1970 including their start and end year, amount of funding awarded, award ID, investigators and their institutions, organization, program, title, abstract, etc. The projects were linked to the papers produced and their citations in the Open Academic Data.[2]

B. Quantifying Outcomes of Projects

We quantify the outcomes of research projects in two ways: (1) number of papers produced and (2) number of citations the produced papers have. Figure 2 presents the count of awarded projects that produce a certain number of papers. The different curves show the distributions (in power law) of awards in different decades. The awards in more recent decades produce more papers. All curves express the same pattern: Most awarded projects produced a small number of papers. So people are interested in predicting which projects go viral (e.g., high #papers, #citations).

C. Statistical Analysis on Profiling Features

Here we conduct statistical analysis (including correlation analysis) on the predictability of project’s “profiling” features such as investigators, award amount, year of effective, and year of expiration. We want demonstrate the needs of additional textual features from project description (using natural language processing techniques).

1) Investigators: Figure 1 shows power-law distributions of #awards per investigator. Most investigators have a small number of awards. We might have two hypotheses (1) an investigator who won more awards would produce more papers and (2) an investigator who won more award would have more citations to the projects. However, the middle figure (#awards vs #papers) and the right figure (#awards vs #citations) do not reflect the patterns clearly.

2) Award amount: Another hypothesis is that the more a project was invested in, the more outcomes the project would make. Figure 4 shows the correlation between the amount of each award (in dollars) and the outcomes (#papers at the top and #citations at the bottom). The distribution of award amount looks like Gaussian on itself with a mean value at around $500,000. However, we cannot find strong correlation between the amount and outcomes in any of the past five decades.

3) Year of effective: Figure 3 presents the trend about the number of awards the NSF has made. The number has been increasing over years. There is a spike in the year of 2009 as a consequence of the 2008 financial crisis. However, as we have seen from Figure 4, more award amount does not lead to more outcomes.

4) Topic examples: All the above profiling features show a coefficient of determination (R^2) of .097 with the number of papers and of .040 with the number citations with the Linear Regression model, respectively. The coefficient of determination are .207 (with number of papers) and .110 (with number of citations), respectively, with a three-layer perceptron to integrate the features.

These features make for a good starting point, but on their own they don’t encapsulate all the factors that go into the proposal review. The coefficient of determination signifies this lack of informative features.

Our idea is to extract features from textual description of the awarded projects. We study a few keywords of research topics before presenting the proposed systematic approach. Figure 6 shows that there was a spike between 2003 and 2012 on “formal_methods” and two peaks (in 1992 and 2014) on “neural_network”; recently, “anomaly_detection”, “knowledge_base”, and “social_media” have become more funded in the Directorate for Computer and Information Science and Engineering (CISE).

III. Proposed Approach

In this section, we present our approach in three parts: (1) extracting features using natural language processing techniques, (2) exploring feature selection strategies, and (3) using three regression models from linear models to shallow neural nets and to deep models.

A. Feature Extraction Techniques

Given project description (title and abstract), we build a pipeline consisting of four types of techniques to extract features of the projects.

First, we use the Bag-of-Words model to generate a unique word vocabulary of terms within each title-abstract corpus [1].

[1] NSF Award Search: https://www.nsf.gov/awardsearch/download.jsp

[2] Open Academic Graph: https://www.openacademic.ai/oag/

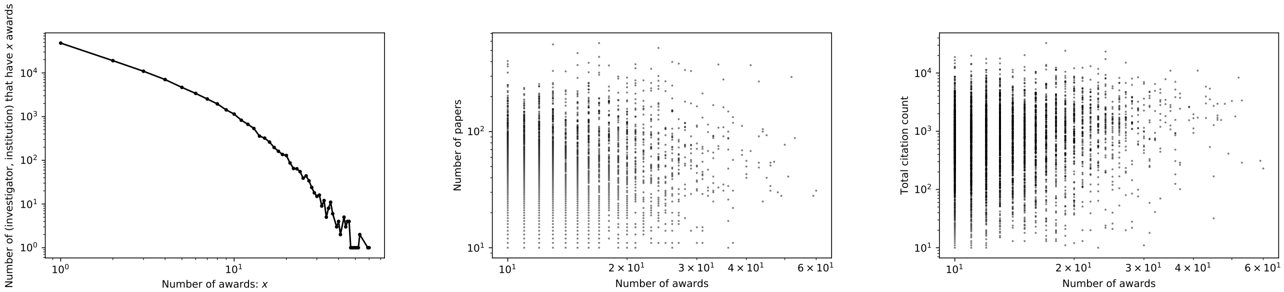

Fig. 1. Left: Distribution of #awards per investigator in log-log scale. Middle: A data point is an investigator's #awards and #papers produced. Right: An investigator's #awards and #citations to the awarded projects.

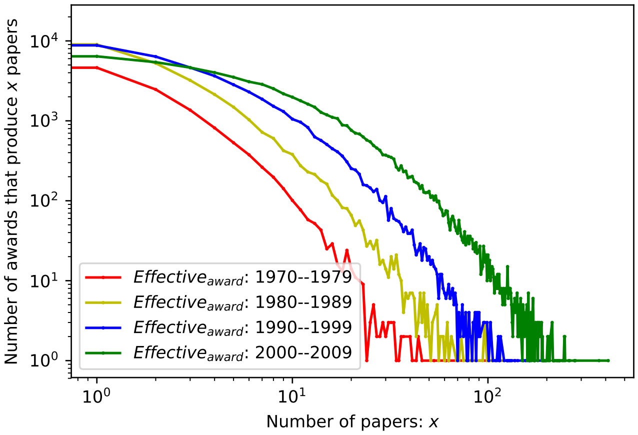

Fig. 2. The number of papers produced by awarded projects in any decade shows power-law distributions.

This bag would generate a point of reference for comparison and analysis with other documents. Stopwords were filtered. Then the most frequent words are "project", "research", "proposed", etc. One issue is that they dominate the representation (i.e., the bag) of documents. Another issue is high dimensionality – we have a huge number of sparse features.

Second, we use topic modeling, specifically Latent Dirichlet Allocation (LDA) [4], [5], to reduce the dimensionality. LDA discovers the word distribution of topics and the topic distributions of documents. The number of topics is much smaller than the size of vocabulary. LDA is a generative probabilistic model for text data. It involves a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is modeled as an infinite mixture over an underlying set of topic probabilities.

Third, we use Term Frequency-Inverse Document Frequency (TF-IDF) to identify the representative words of documents. We observe that without TF-IDF, the topics generated by LDA have big overlaps: the top representative words are often the most frequent words. TF-IDF finds word relevance in collection of documents by calculating an inverse relationship between the word frequency in a document and the percentage of documents the word appears in [2]. Our TF-IDF model calculated a score distribution for word frequencies across each document to label significant terms within each corpus.

We enhanced the LDA model using TF-IDF. It enabled us to use an unsupervised technique that interpreted the TF-IDF documents into a set number of explainable topics that correlated with each corpus. These topics were then used as additional features for the neural network to categorize research projects.

Fourth, we further enhance the topic modeling by upgrading vocabulary from words only to words and *phrases*. AutoPhrase involves finding significant groups of words (i.e., phrases) in documents that are informative or are indicative of a specific topic or concept [3]. Besides informativeness, each of these semantic units are weighted on frequency of occurrence in each document and collocation of tokens in quality phrases occurring with significantly higher probability than expected due to chance. The AutoPhrase model enhanced our bag-of-words and TF-IDF models by introducing phrases that would be collected into a bag-of-phrases increasing the explainability and accuracy of the model by feeding in phrases that hold significance, e.g., "computer_architecture", "physical_chemistry".

B. *Feature Selection Strategies*

The results of topic modeling make lots of sense yet uncertainty. Bag-of-words and bag-of-phrases (2 choices) generate different topics. Different numbers of topics and/or different trials generate different results, too. Suppose we set the number of topics as an integer between 2 and 15, and suppose we make 3 trials. The total number of features is $2 \times \frac{(2+15)\times(15-1)}{2} \times 3 = 714$. We observe that heavy redundancy in using all the features easily lead machine learning models to over-fitting. Feature selection is needed.

One strategy is to choose a particular bag model, a particular number of topics, and a particular trial. The number of features will be the number of topics. It can find the most effective group of topics; however, probably only one of topics in the group is correlated with the labels (i.e., project outcomes).

The other strategy is to select the best ones from all the 714 topics based on their Pearson correlation coefficient (PCC) with the outcomes. We determine the size of selections with expectation of having higher PCC than the profiling features.

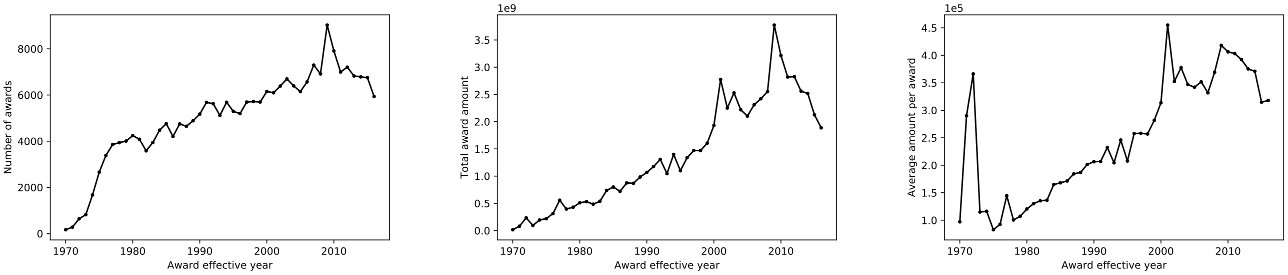

Fig. 3. The trend shows the government is investing more and more research projects over years.

TABLE I

PERFORMANCE ON PREDICTING THE NUMBER OF PAPERS PRODUCED BY AN UNSEEN PROJECT. WE HAVE THE FOLLOWING OBSERVATIONS: FIRST OF ALL, MULTI-LAYER PERCEPTRON (MLP) MODELS PERFORM THE BEST OVER SINGLE-LAYER PERCEPTRON AND LINEAR REGRESSION. SECOND, NEURAL NETWORK MODELS MET THE SERIOUS ISSUE OF OVERFITTING EASILY WITH A FULL SET OF FEATURES. THIRD, PUTTING THE BEST TOPICS TOGETHER CAN PERFORM BETTER THAN CHOOSING A PARTICULAR GROUP OF TOPICS. (FOR MAE AND RMSE, THE SMALLER ERROR, THE BETTER PERFORMANCE; FOR CORRELATION OF DETERMINATION R^2, THE HIGHER CORRELATION, THE BETTER PERFORMANCE.)

Model	**Features**	**MAE (dev)**	**MAE (test)**	**RMSE (dev)**	**RMSE (test)**	R^2 **(dev)**	R^2 **(test)**	
Linear Regression	Profiling	7.714	7.758	13.542	13.760	0.106	0.097	
	+ best Bag-of-Words	7.625	7.676	13.343	13.566	0.132	0.123	4 topics
	+ best Bag-of-Phrases	7.624	7.662	13.285	13.493	0.140	0.132	3 topics
	(All features)	7.402	7.516	12.820	13.148	0.199	0.176	
Single-Layer Perceptron	Profiling	7.054	7.097	12.642	12.800	0.221	0.219	
	+ best BOW	6.724	6.839	11.951	12.264	0.304	0.283	10 topics
	+ best BOP	6.596	6.641	12.073	12.277	0.290	0.281	5 topics
	+ best BOW + best BOP	6.395	6.514	11.663	11.994	0.337	0.314	
	+ top 10 BOW/BOP	6.402	6.946	11.042	12.057	0.406	0.307	overfitting
	(All features)	4.737	11.438	6.581	16.192	0.789	-0.250	overfitting
Multi-Layer Perceptron	Profiling	6.861	6.896	12.733	12.898	0.210	0.207	
	+ best BOW + best BOP	6.348	6.462	11.554	11.948	0.350	0.319	
	+ top 10 BOW/BOP	6.076	6.293	11.243	11.909	0.384	0.324	
	(All features)	6.503	6.914	11.508	12.601	0.355	0.243	
	+ top 10 correlated topics	6.125	**6.261**	11.288	11.727	0.379	0.344	
	+ top 20 correlated topics	6.229	6.371	11.331	11.742	0.374	0.343	
	+ top 30 correlated topics	6.136	6.342	11.071	**11.699**	0.403	**0.348**	

C. Regression Models

We use three types of regression models. First, linear regression is used to find a linear relationship within the data [6]. Our linear model didn't prove to do as well as the neural network but this preliminary method of regression identified somewhat of a linear relationship to the data. Second, we use single-layer perceptron of one layer and 100 neural network units on the layer. It does certain computations to detect latent features in the input data. Third, we use multi-layer perceptron (MLP) of three dense layers (100-50-30 neurons), which was proved to be the most effective [7]. MLP utilizes back propagation technique for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

IV. EXPERIMENTS

In this section, we present experimental settings, results, and our analysis on which learning models, feature extraction techniques, and feature selection strategies can deliver the best performance.

A. Validation and Evaluation Methods

The number of awarded projects is 182,679 in our dataset. We use hold-out validation with 50% for training and 50% for testing. Our approach is used to predict (1) the number of papers produced and (2) the number of citations the produced papers have. Then we use three types of evaluation metrics to present and compare the performance: mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2).

B. Results: Overall Performance

Table I presents the performance of different types of machine learning models and different kinds of features on predicting the number of papers produced by an unseen awarded project. For any machine learning model, the best performance we achieve with textual features carefully extracted and selected is consistently much better than using profiling features only: When the model is linear regression, the test MAE and test RMSE scores are reduced by relatively 3.1% (from 7.758 to 7.516) and by 4.4% (from 13.760 to 13.148), respectively; and the test R^2 score increases relatively by 81.4% (from 0.097 to 0.176). When the model is single-

TABLE II

PERFORMANCE ON PREDICTING THE NUMBER OF CITATIONS THE PRODUCED PAPERS CAN HAVE. WE HAVE THE FOLLOWING OBSERVATIONS: FIRST OF ALL, PERCEPTRON MODELS PERFORM THE BEST OVER LINEAR REGRESSION. SECOND, THE NEURAL NETWORK MODELS (I.E., PERCEPTRONS) MET OVER-FITTING EASILY WITH A FULL SET OF FEATURES. THIRD, PUTTING THE BEST TOPICS TOGETHER CAN PERFORM BETTER THAN CHOOSING A PARTICULAR GROUP OF TOPICS. (FOR MAE AND RMSE, THE SMALLER ERROR, THE BETTER PERFORMANCE; FOR CORRELATION OF DETERMINATION R^2, THE HIGHER CORRELATION, THE BETTER PERFORMANCE.)

Model	**Features**	**MAE (dev)**	**MAE (test)**	**RMSE (dev)**	**RMSE (test)**	R^2 **(dev)**	R^2 **(test)**	
Linear Regression	Profiling	224.2	224.8	503.6	501.1	0.041	0.040	
	(All features)	222.4	225.1	491.9	492.8	0.085	0.071	
Single-Layer Perceptron	Profiling	211.2	212.3	483.9	479.9	0.115	0.119	
	+ best BOW	204.1	205.6	476.1	473.0	0.143	0.144	8 topics
	+ best BOP	206.5	208.0	475.8	472.0	0.144	0.148	8 topics
	(All features)	192.3	299.3	342.5	522.8	0.556	-0.046	overfitting
Multi-Layer Perceptron	Profiling	197.4	198.9	485.2	482.4	0.110	0.110	
	(All features)	204.6	209.9	485.3	492.7	0.109	0.071	
	+ top 30 correlated topics	188.1	**192.4**	455.0	**460.8**	0.217	**0.188**	

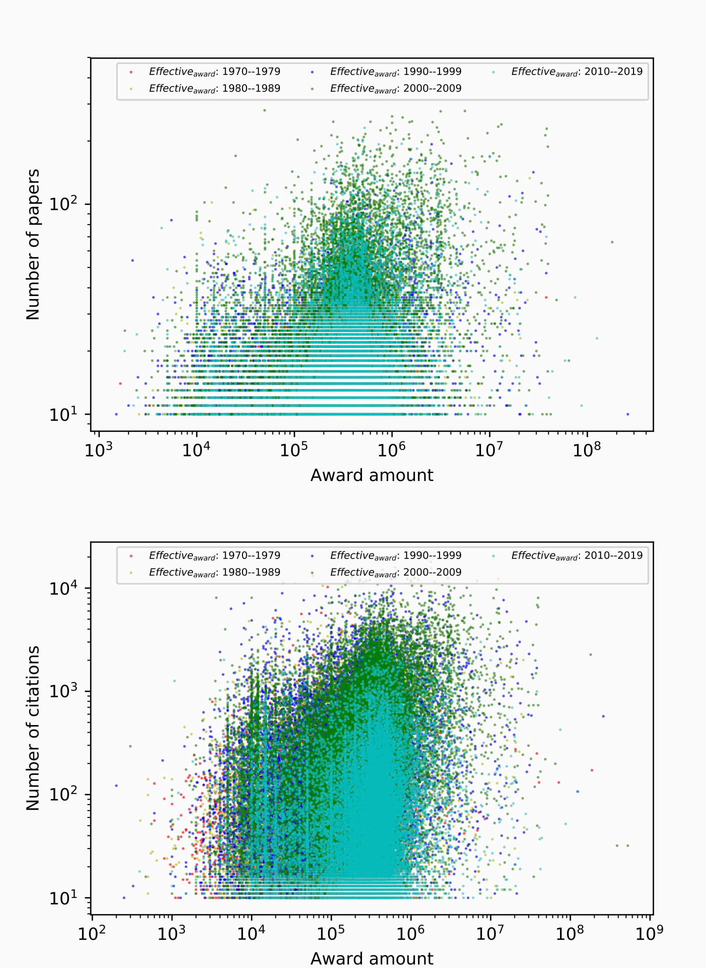

Fig. 4. Correlation analysis between #papers (or #citations) and award amount (in dollars). None of the last five decades shows any strong correlation.

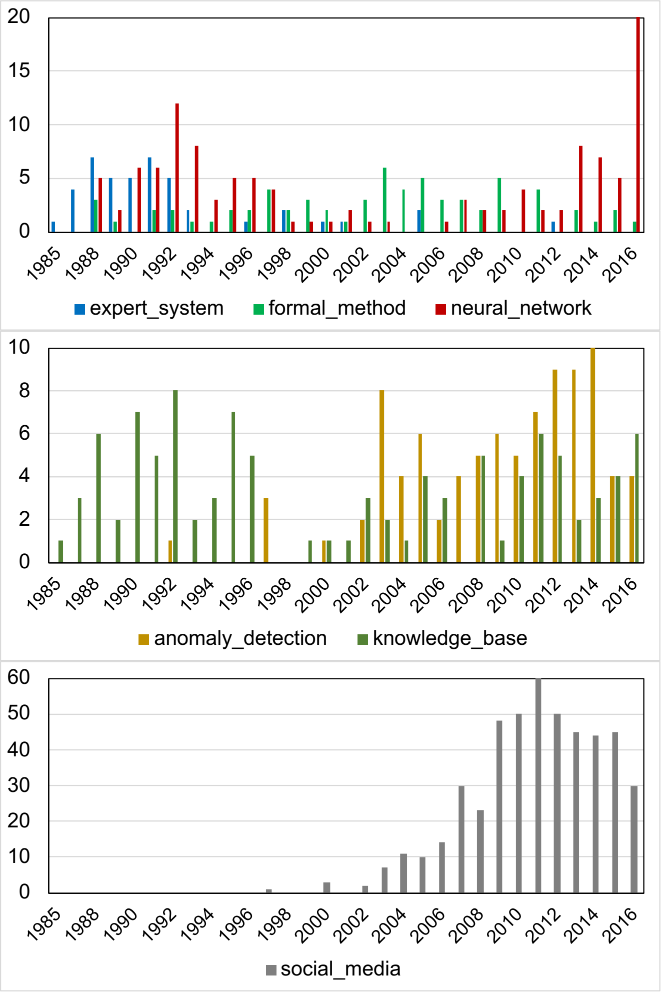

Fig. 5. We investigate the number of awards to six particular keywords. Top: AI ("expert_system", "formal_methods", and "neural_network"). Middle: "anomaly_detection", "knowledge_base". Bottom: "social_media".

layer perceptron with 100 neurons, the test MAE and test RMSE scores are reduced by relatively 8.2% (from 7.097 to 6.514) and by 6.3% (from 12.800 to 11.994), respectively; and the test R^2 score increases relatively by 43.4% (from 0.219 to 0.314). When the model is multi-layer perceptron with 100, 50, and 30 neurons on the three layers, the test MAE and test RMSE scores are reduced by relatively 9.2% (from 6.896 to 6.261) and by 9.3% (from 12.898 to 11.699); and the test R^2 score increases relatively by 68.1% (from 0.207 to 0.348). The best performance is highlighted in bold.

Table II presents the performance on predicting the number

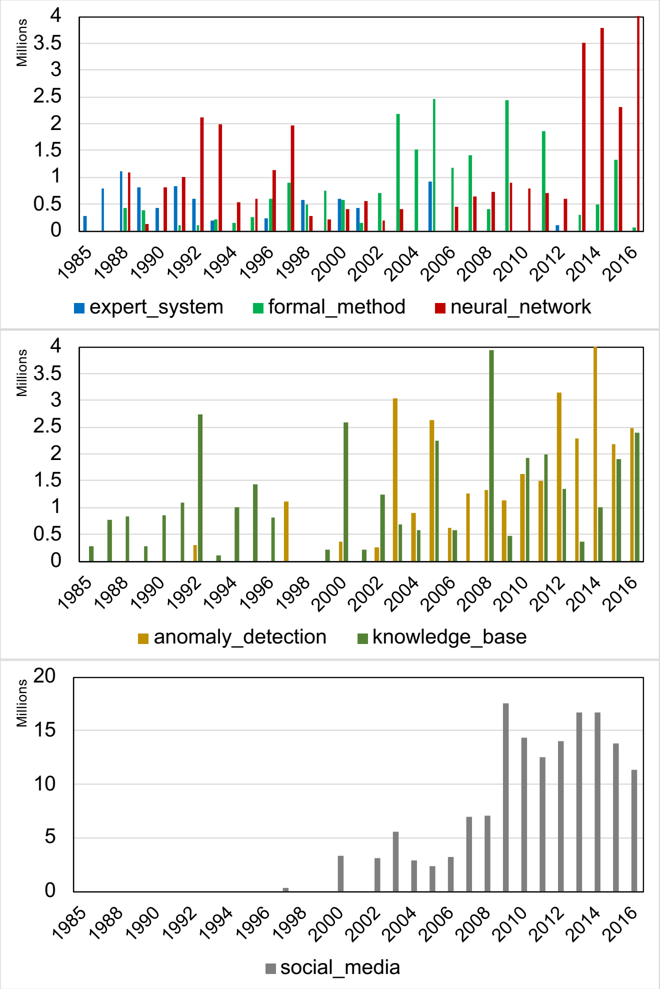

Fig. 6. We investigate the total amount of fund to six particular keywords. Top: AI ("expert_system", "formal_methods", and "neural_network"). Middle: "anomaly_detection", "knowledge_base". Bottom: "social_media".

of citations. For any learning model, the best performance with textual features is consistently much better than using profiling features only: When the model is linear regression, the test R^2 score increases relatively by 77.5% (from 0.040 to 0.071). When the model is single-layer perceptron with 100 neurons, the test R^2 score increases relatively by 24.4% (from 0.119 to 0.148). When the model is multi-layer perceptron with 100, 50, and 30 neurons on the three layers, the test MAE and test RMSE scores are reduced by relatively 3.3% (from 198.9 to 192.4) and by 4.5% (from 482.4 to 460.8); and the test R^2 score increases relatively by 70.9% (from 0.110 to 0.188).

C. Performance Analysis

In this section, we compare the performance from different kinds of topic features, or different machine learning models, or different feature selection strategies. We analyze the results.

1) Topic feature extraction: From Table I we can see that given the particular model (linger regression or percerptron), topic models on bag-of-phrases (with TF-IDF) perform no worse than topic models on bag-of-words (with TF-IDF), and the number of topics on bag-of-phrases can be smaller than that on bag-of-words. With single-layer perceptron, combining the best group of topics from bag-of-words with the best group of topics from bag-of-phrases can perform better than one single group. Table II shows that bag-of-words and bag-of-phrases topic features deliver similar performance. From both tables we can see topic features are complementary with profiling features, which improves the performance significantly.

2) Machine learning models: Generally we can see perceptron performs better than linear regression and multi-layer perceptron performs better than single-layer perceptron, as shown in Tables I and II. Linear regression performs the worst but does not meet overfitting with over 700 topic features; however, perceptron has the problem and the overfitting on single-layer perceptron is significant: the test R^2 score is negative. Using top 10 groups of topics cannot perform better than using only the top 2. Here the "group" means a group of k topics by a particular bag model plus a particular number of topics k plus a particular trial. The reason may also be overfitting. On multi-layer perceptron, using the top 10 groups can perform better than using the top 2. So we conclude that multi-layer perceptron deals with the trade-off between training accuracy and overfitting very well.

3) Feature selection strategies: As multi-layer perceptron consistently performs the best, we study feature selection strategies only on this particular model. As shown in Table I, based on the first strategy (i.e., choosing the best groups of topics), we tried the combination of the best groups in bag-of-words and bag-of-phrases. We also tried the combination of the best 10 groups of topics. The latter performs better with a test R^2 score of 0.324.

Then we explore the second strategy (i.e., selecting the best topics from all groups of topics). We tried the top 10, top 20, and top 30 correlated topics. We observe that all of them can perform better than the best of the first strategy. The best delivers a test R^2 score of 0.348, with a relatively 7.4% improvement. In Table II we can also see that using top 30 correlated topics performs well.

Figure 7 presents the performance of an MLP model with the second strategy, i.e., selecting top 10, 20, 30 most correlated topic features. On the left we present results on predicting the number of papers a project produced. We can see that using the top 10 most correlated topics has the most stable curve – the test RMSE decreases and the test R^2 score increases with the number of training epochs increasing. When we use the top 30, we may have the lowest test RMSE and/or the highest test R^2 score at some point but the curves are not stable.

On the right of Figure 7 we present results on predicting the number of citations the produced papers had. The curve of using the top 30 most correlated topics looks stable. The test RMSE decreases and the test R^2 score increases with the number of training epochs increasing.

D. Most Correlated Topics of Words/Phrases

Table III gives the representative words and phrase for most positively and negatively correlated topics with the outcomes.

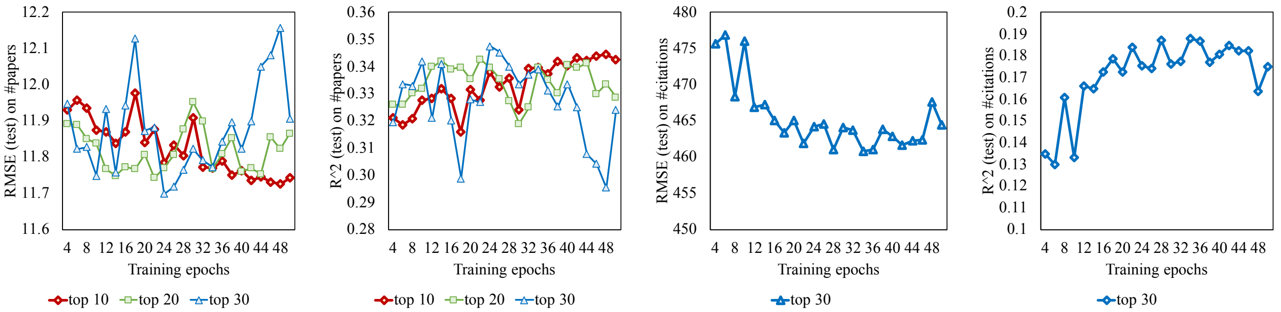

Fig. 7. The left two figures present RMSE and R^2 score on predicting the number of papers produced by projects over the number of training epochs on the multi-layer perceptron (MLP) model. The strategies we investigated are top 10, 20, or 30 of the best topics across the groups. Using the top 10 topics delivers the most stable performance. The right-hand figures present RMSE and R^2 on predicting the number of citations the produced papers have.

TABLE III
REPRESENTATIVE WORDS AND PHRASE FOR THE MOST POSITIVELY OR NEGATIVELY CORRELATED TOPICS.

vs. #papers		
Corr = 0.213	0.195	-0.187
algorithms design software stem performance	design materials data system develop	changes species understanding study evolution
graduate_students		climate_change
vs. #citations		
Corr = 0.098	0.085	-0.084
students data system science develop	design materials data system develop	geometry mathematics equations topology manifolds
collaborative_research		algebraic_geometry

Computer science and/or graduate students-oriented projects could produce more papers than those on studying climate change. Collaborative research projects on systems could have more citations than those on algebraic geometry, mathematics, topology, etc.

V. Related Work

In this section, we review three topics that are related to our study: behavior outcome prediction, topic discovery and feature extraction, and regression models.

A. Outcome Prediction

We have found that we are the only research project to investigate predicting the outcomes of NSF-funded research projects. The motivation of our work is presented in Figure 8. Program directors, managers, and officers are willing to predict the outcomes of proposed research projects based on the proposal applications so that they can allocate funds to promising, successful projects. However, other research projects have also aimed to predict outcomes and measure successes of various processes through similar machine learning techniques. [8] aimed to predicts the risk of projects being downgraded to an unsatisfactory rating through a primarily past ratings and a variety of other factors. [9] addressed the fact that a large number of Big Data projects are failing and attempts to predict the success rate of these papers based on the behavior of the models. [10] was a project that predicts the cost and success of alternative therapy treatments for patients. [11] predicted college student retention rates in online courses based on where they fall on a risk spectrum. [12] predicted the success or failure of a new innovative product early on in its development life-cycle. And, [13] was a similar project in that it attempted to predict the success of software development projects. Recently, representation learning methods are used to estimate or learn the success of teaming and project plans [14], [15].

B. Topic Discovery and Feature Extraction

Many other papers have employed these natural language processing techniques to achieve similar results [16]–[21]. In the case of topic modeling, [22] highlighted using multiple LDA models for individual document categories to create a unified LDA model that can effectively categorize web documents. [23] improved the LDA model by accounting for visual spacing of words in a document to suggest a relationship. [16] introduced the idea of Latent Semantic Indexing or Latent Semantic Analysis which is based on Linear Algebra and uses singular value decomposition on the document-term matrix. Another similar topic modeling technique is Non-Negative Matrix Factorization which is further outlined in [17] is Non-Negative Matrix Factorization. As for feature selection, [24] analyzed the cost-effective nature of using the hybrid feature selection techniques.

C. Regression Models

Related learning models offer a variety of other regression approaches for our project [25]–[28]. For example, Support Vector Regression [29] which attempted to minimize the generalization error bound so as to achieve generalized performance. By shortening this margin of error, the model estimated within these bounds to achieve high levels of accuracy. Another regression model is Gradient Descent, further outlined in [30], which allows further optimization of linear regressors or neural networks. It employs an iterative algorithm to find the local minimum of a differentiable cost function which allows

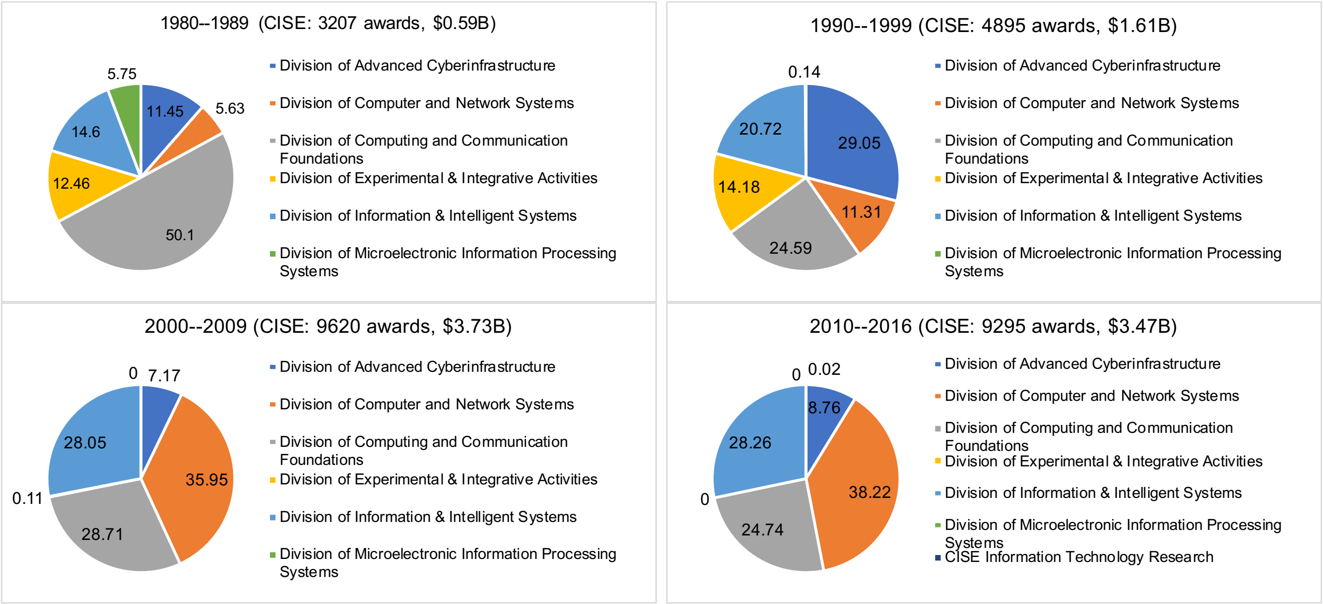

Fig. 8. The trend of fund allocation by NSF CISE over decades (from 1980 to 2016).

the model to maximize the effectiveness of the weights on the function. Other notable regression techniques include Polynomial, Lasso, Ridge, StepWise, and ElasticNet Regression [31]. They focus on multidimensional data and functions of varying degree. Finally, other related feed-forward neural networks include the Adaptive Linear Neuron (ADALINE) and Backpropagation models [25]. These models offer another path we could pursue to improve our results in the near future. The Multi-Layer Perceptron we used in this project proved to be most effective with its use of a non-linear activation function allowing for complex, non-linear decision boundaries [26].

VI. Conclusions and Future Work

Our research has shown the significance and impact of combining various natural language processing techniques with a multi-layer neural network to improve the objective estimation of success in NSF-funded projects. By applying these methods, we significantly improved the coefficient of determination and proved that the textual data from the project descriptions were critical in predicting outcomes. We believe that our model can be improved to achieve higher accuracy and intend to employ deeper NLP techniques (e.g., BERT language model). We also plan to tune the topic modeling process to achieve highly explainable topics that give clearer distributions of the features scores. Furthermore, we plan to train a meta learner on learning the feature-outcome relationships in hundreds of NSF directorates, divisions, and programs.

Acknowledgement

This work is supported by NSF grant IIS-1849816.

References

[1] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[2] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.

[3] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1825–1837, 2018.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[5] H. M. Wallach, "Topic modeling: beyond bag-of-words," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 977–984.

[6] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012, vol. 329.

[7] D. W. Ruck, S. K. Rogers, and M. Kabrisky, "Feature selection using a multilayer perceptron," *Journal of Neural Network Computing*, vol. 2, no. 2, pp. 40–48, 1990.

[8] M. Blanc, T. Esmail, C. Mascarell, and R. Rodriguez, *Predicting project outcomes: a simple methodology for predictions based on project ratings*. The World Bank, 2016.

[9] D. K. Becker, "Predicting outcomes for big data projects: Big data project dynamics (bdpd): Research in progress," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 2320–2330.

[10] V. Bremer, D. Becker, S. Kolovos, B. Funk, W. Van Breda, M. Hoogendoorn, and H. Riper, "Predicting therapy success and costs for personalized treatment recommendations using baseline characteristics: data-driven analysis," *Journal of medical Internet research*, vol. 20, no. 8, p. e10275, 2018.

[11] V. C. Smith, A. Lange, and D. R. Huston, "Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses." *Journal of Asynchronous Learning Networks*, vol. 16, no. 3, pp. 51–61, 2012.

[12] G. Rossetti, L. Milli, F. Giannotti, and D. Pedreschi, "Forecasting success via early adoptions analysis: A data-driven study," *PloS one*, vol. 12, no. 12, 2017.

[13] R. Weber, M. Waller, J. Verner, and W. Evanco, "Predicting software development project outcomes," in *International Conference on Case-Based Reasoning*. Springer, 2003, pp. 595–609.

[14] D. Wang, M. Jiang, Q. Zeng, Z. Eberhart, and N. V. Chawla, "Multi-type itemset embedding for learning behavior success," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2018.

[15] D. Wang, T. Jiang, N. V. Chawla, and M. Jiang, “Tube: Embedding behavior outcomes for predicting success,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1682–1690.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[17] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*, 2001, pp. 556–562.

[18] T. Jiang, T. Zhao, B. Qin, T. Liu, N. V. Chawla, and M. Jiang, “The role of ”condition”: A novel scientific knowledge graph representation and construction model,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1634–1642.

[19] M. Jiang, Q. Li, X. Zhang, M. Qu, T. Hanratty, J. Gao, and J. Han, “Truepie: Discovering reliable patterns in pattern-based information extraction,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2018.

[20] X. Wang, H. Zhang, Q. Li, Y. Shi, and M. Jiang, “A novel unsupervised approach for precise temporal slot filling from incomplete and noisy temporal contexts,” in *The World Wide Web Conference*, 2019, pp. 3328–3334.

[21] Q. Zeng, M. Yu, W. Yu, J. Xiong, Y. Shi, and M. Jiang, “Faceted hierarchy: A new graph type to organize scientific concepts and a construction method,” in *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 2019, pp. 140–150.

[22] I. Bíró and J. Szabó, “Latent dirichlet allocation for automatic document categorization,” in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2009, pp. 430–441.

[23] X. Wang and E. Grimson, “Spatial latent dirichlet allocation,” in *Advances in neural information processing systems*, 2008, pp. 1577–1584.

[24] J. Pirgazi, M. Alimoradi, T. E. Abharian, and M. H. Olyaee, “An efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets,” *Scientific Reports*, vol. 9, no. 1, pp. 1–15, 2019.

[25] V. Pellakuri, D. R. Rao, and J. Murthy, “Modeling of supervised adaline neural network learning technique,” in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, pp. 17–22.

[26] B. Widrow and M. A. Lehr, “Artificial neural networks of the perceptron, madaline, and backpropagation family,” in *Neurobionics*. Elsevier, 1993, pp. 133–205.

[27] L. Liu, F. Zhu, M. Jiang, J. Han, L. Sun, and S. Yang, “Mining diversity on social media networks,” *Multimedia Tools and Applications*, vol. 56, no. 1, pp. 179–205, 2012.

[28] M. Jiang, C. Faloutsos, and J. Han, “Catchtartan: Representing and summarizing dynamic multicontextual behaviors,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2016, pp. 945–954.

[29] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, “Support vector regression machines,” in *Advances in neural information processing systems*, 1997, pp. 155–161.

[30] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.

[31] H. Theil, “A rank-invariant method of linear and polynomial regression analysis,” in *Henri Theil's contributions to economics and econometrics*. Springer, 1992, pp. 345–381.