CLASSIFYING NONPROFITS USING MACHINE-LEARNING: BENCHMARK AND GUIDE FOR SOCIAL SCIENTISTS SOLVING REAL-WORLD PROBLEMS

JI MA AND ISHA KANANI *

University of Texas at Austin

May 7, 2019

Abstract

This research note reports the use of supervised machine-learning algorithms in classifying the nonprofit organizations in the United States. Mission statements and project descriptions are collected from the 990 forms as text data, and classifications using National Taxonomy of Exempt Entities are collected from the National Center for Charitable Statistics at the Urban Institute. Three text classification algorithms are experimented: Naïve Bayes, Random Forest, and Neural Network. The Neural Network classification achieves the best results with an average accuracy of 9*.9% (standard deviation **), recall *** (standard deviation **), and precision *** (SD **). An open-source Python package *npocat* is developed and shared using the trained algorithms. Future projects are discussed.

*J.M.: maji@austin.utexas.edu, LBJ School of Public Affairs and RGK Center for Philanthropy and Community Service; I.K.: ishakanani@utexas.edu, School of Information.

Contents

List of Tables

List of Figures

1 Introduction

Although the voluntary and philanthropic organizations have long been existent for numerous centuries, the so-called “nonprofit sector” was only coined in the 1970s by scholars, policy makers, and nonprofit practitioners (Hall, 2006). A major reason for assembling the diverse organizations as a conceptual whole is to legitimize the existence of these organizations and the benefits these organizations receive (Hall, 2006, pp. 54–55; Barman, 2013). From Durkheim’s ([1912] 2012) perspective, the order and structure of a society can be reflected by a classification system. The National Taxonomy of Exempt Entities (NTEE) developed by the National Center for Charitable Statistics (NCCS), the most widely used classification system, is one of the efforts legitimizing the existence of nonprofit sector (Hodgkinson, 1990; Hodgkinson & Toppe, 1991). As Barman (2013, p. 105) cite Clarke and Casper (1996, p. 601): “The ways in which different entities (people, animals, plants, diseases, etc.) are organized into classificatory groups reveal something of the social, cultural, symbolic, and political contexts within which classifications occur.”

The development of NTEE classifications can date back to the 1980s (Hodgkinson, 1990, pp. 8–9, 11). In 1982, NCCS assembled a team of experts working on creating a taxonomy for nonprofit organizations. The first draft of the taxonomy, entitled “National Taxonomy of Exempt Entities” (NTEE), came out in 1986 and published in 1987. In the early 1990s, NCCS had classified nearly one million nonprofits using NTEE. In 1995, the Internal Revenue Service (IRS) adopted the NTEE coding system, took over the tasks of assigning and maintaining the classifications, and started to release the Business Master File with NTEE codes (US Internal Revenue Service, 2013, 2014).

Two agencies took the task of assigning NTEE codes: NCCS and IRS. Before 1995, NCCS coded nonprofits according to the program descriptions in Part III and VIII of Form 990, supplemented with information from Form 1023 (“Application for Recognition of Exemption”) and additional research (National Center for Charitable Statistics, 2006, p. 16). After 1995, IRS began to issue “new exempt organizations an NTEE code as part of the determination process,” and “the determination specialist assigns an NTEE code to each organization exempt under I.R.C. §501(a) as part of the process of closing a case when the organization is recognized as tax-exempt” (US Internal Revenue Service, 2013, p. 1).

The NTEE classification system supported many applied or academic studies of nonprofit organizations which have critical economic and political roles in society. For example, NTEE provides a framework on which the social and economic activities of nonprofits can be mapped and compared with other types of organizations in a society (e.g., Roeger, Blackwood, & Pettijohn, 2015). It can also serve as an analytical tool for measuring the organizational capacity in different service domains and inform the practitioners and policymakers in decision-making (Hodgkinson & Toppe, 1991). Scholars also use NTEE codes for sampling purposes (e.g., Carman & Fredericks, 2010; Okten & Weisbrod, 2000) or as independent variables (Sloan, 2009). The invention of NTEE also provides a fundamental necessity for comparative international research, facilitating the study of “global civil society” (Hodgkinson, 1990; Lester M. Salamon & Anheier, 1992; Lester M Salamon, Anheir, & coaut, 1996; Vakil, 1997).

The NTEE classification system, although one of the best we have so far, still has several drawbacks. First, because it only assign one major category code to an organization, it cannot accurately describe a nonprofit's programs which are usually diverse and across several service domains (Grønbjerg, 1994, p. 303). Although a program classification system was developed later (Lampkin, Romeo, & Finnin, 2001), it is not widely used probably because it is impractical to assign codes to massive amount of programs. Second, the assignment of NTEE codes is not complete because it is "based on an assessment of program descriptions contained in Parts 3 and 8 of the Form 990" and "program descriptions were only available for some organizations" (National Center for Charitable Statistics, 2006, p. 16). A recent study found the number of organizations in Washington State with a specific NTEE code could be significantly increased if the mission statements were used for coding (Fyall, Moore, & Gugerty, 2018). Third, NTEE codes are static but nonprofit organizations' activities may change over time. Recoding existent NTEE assignments is extremely onerous, and this may be one of the reason that IRS does not have a procedure by which the nonprofits can request the change of their NTEE codes (US Internal Revenue Service, 2013). In general, the tremendous human labor needed for classification is a prominent challenge, and such challenge is also an evident barrier for improving any classification system.

Numerous social scientists experimented with applying computational research methods in classifying objects for solving real-world problems (e.g., Baćak & Kennedy, 2018; Fyall et al., 2018; Grimmer & Stewart, 2013; Nelson, Burk, Knudsen, & McCall, 2018). We respond to the challenge by applying the advances in computational linguistics and made the following contributions to the growing literature: 1) we established a standardized workflow and benchmarks which future studies of nonprofit or classification in other social science disciplines can build on and compare to; 2) we achieved 83.47% overall accuracy for classifying the nonprofits into 10 broad categories, and 77% for classifying into 25 major groups; 3) we developed a Python software package for scholars to classify text statements using NTEE codes; 4) we released all source codes and data for replication purposes and future studies, this is particularly important since there are many caveats in tuning algorithms which cannot be detailed by this paper.[1]

2 Method

2.1 Working with Texts and Research Workflow

Classifying texts is a typical task of automatic content analysis and usually employs three types of methods: dictionary, supervised, and unsupervised methods (Grimmer & Stewart, 2013, pp. 268–269). The dictionary methods use a predefined dictionary of words to classifying the texts. Although accurate, this approach is not capable to deal with the variations and contexts of language. The supervised method is an improved solution which uses computer algorithms to "learn" the linguistic patters in a dataset classified by human coders. Unlike the dictionary and supervised methods which require predefined categories of interest, unsupervised methods can discover linguistic patters in texts without inputting any knowledge of classification. However, unsupervised method's validity can be problematic because the returned classifications may not be theoretically meaningful. To take the advantage of existing human-coded NTEE classifications and the experience of existing studies (Fyall et al., 2018; Nelson et al., 2018), this study employs a supervised approach as Figure 1 illustrates.

[1] Follow this link for the complete working directory with detail instructions: https://github.com/***

Figure 1: RESEARCH WORKFLOW

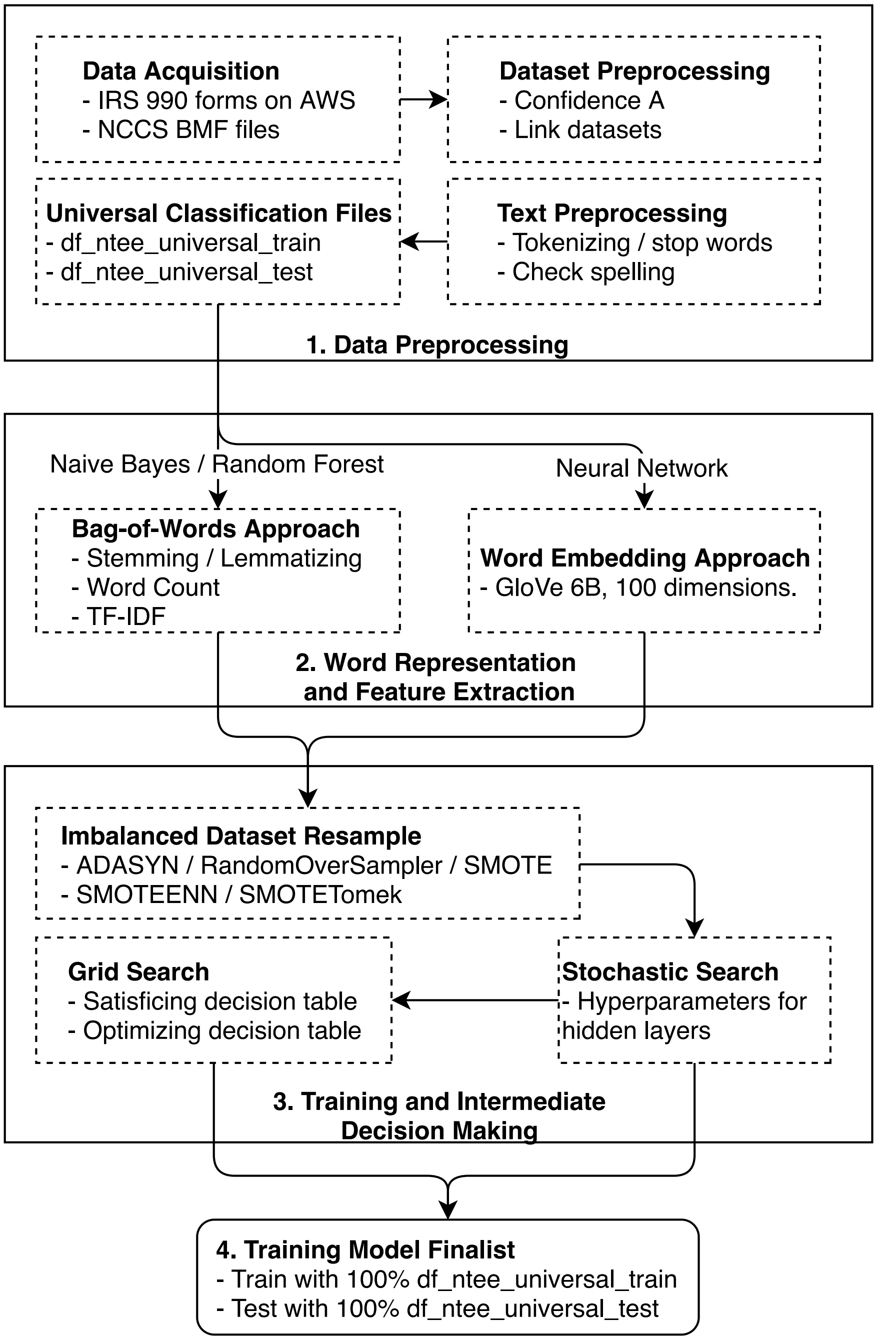

Table 1: LOCATIONS OF TEXT FIELDS IN DIFFERENT FORMS

	Mission Statement	Program Description
990	Part I, Line 1; Part III, Line 1	Part III, Line 4; Part VIII, Line 2a-e, Line 11a-c; Schedule O
990-EZ	Part III	Part III, Line 28-30; Schedule O
990-PF	–	Part IX-A; Part XVI-B

Figure 1 shows this paper's complete workflow. We implement four stages of analysis: 1) *preprocessing stage* includes data acquisition and the preprocessing of datasets and texts; 2) *feature extraction* includes bag-of-words (used by Naive Bayes and Random Forest algorithms) and word embedding (used by neural network algorithms); 3) at *training and intermediate decision making* phase, we use stochastic and grid search to train, search, and optimize the machine learning algorithms; 4) we *train the model finalist* with the complete dataset and prepare the trained model for public use. The following part introduces the four phases in detail.

2.2 Data Preprocessing

Data acquisition and dataset preprocessing. We collected text records from form 990, 990-EZ, and 990-PF, and supplemented these records with program descriptions from Schedule O. Form 990 (Return of Organization Exempt From Income Tax) is submitted by most of the nonprofit organizations. For smaller organizations with "gross receipts of less than $200,000 and total assets of less than $500,000 at the end of their tax year" (US Internal Revenue Service, 2018, p. 1), they can file Form 990-EZ (Short Form Return of Organization Exempt From Income Tax), a shorter version of Form 990. Private foundations use Form 990-PF (Return of Private Foundation). The texts describes organizational activities in two forms: overall mission statement and specific program description. Table 1 summarizes these text fields' specific locations in different forms.

Classification records (i.e., NTEE codes) are collected from the 2014-2016 Business Master Files on NCCS website.[2] This study deals with two types of NTEE classifications: 10 broad category and 26 major groups. Table 2 shows the relationship between broad categories and major groups. A detail list of the 26 major groups can be found in US Internal Revenue Service (2014). The accuracy of classification is indicated by a letter of A, B, or C, and "confidence level of A ... indicates that there is at least a 90 percent probability that the major group classification is correct" (National Center for Charitable Statistics, 2006, p. 16). The intercoder reliability of records at confidence A level should approximate 100% (Stengel, Lampkin, & Stevenson, 1998, p. 147) – this measure is particularly important because the NTEE codes were assigned by human coders from different organizations (i.e., IRS and NCCS) over different periods of time.

From 2014 to 2016, 56.12% records are classified at A level, 37.32% at B level, and 6.56% at C level. For training purposes, we only use records at confidence level A and drop all X/Z category (i.e., unknown or unclassified). About 1.76% organizations changed their NTEE codes between 2014 and 2016. We dropped

[2] https://nccs-data.urban.org

Table 2: NTEE-CC CLASSIFICATION SYSTEM

Broad Category Code	Explanation	Major Group Code
I	Arts, Culture, and Humanities	A
II	Education	B
III	Environment and Animals	C, D
IV	Health	E, F, G, H
V	Human Services	I, J, K, L, M, N, O, P
VI	International, Foreign Affairs	Q
VII	Public, Societal Benefit	R, S, T, U, V, W
VIII	Religion Related	X
IX	Mutual/Membership Benefit	Y
X	Unknown, Unclassified	Z

the records of these organizations since these records are less credible.

Text Preprocessing. Texts in sentences need to be "tokenized" into words before analysis, which is called "tokenization" in natural language processing. We also removed stop words (e.g., "the", "a" and punctuation marks) and checked spelling errors using algorithms based on "minimum edit distance" (ie., the minimum number of editing operations needed to change one word into another; Jurafsky and Martin (2017, p. 26)).

Universal Classification Files (UCFs). The final step at data preprocessing stage is to divide data records into training and testing datasets (i.e., files in `/dataset/df_ntee_universal/`) that are mutually exclusive and can be used for benchmarking future models. The *Universal Classification File Training* (UCF-Training) (`df_ntee_universal_train.pkl.gz`) is used for developing models and consists of 80% total records. The *Universal Classification File Testing* (UCF-Testing) (`df_ntee_universal_test.pkl.gz`) is used for testing trained models' performance and consists of 20% total records. Table 3 presents the two datasets' composition by major groups. The UCFs approximate the composition of organizations reported to the IRS except group A ("arts, culture, and humanities") and T ("philanthropy, voluntarism, and grantmaking foundations"). The consequence is that, comparing to human coders, the trained models are less likely to categorize organizations as T but more likely to categorize organizations as A.

2.3 Word representation and feature extraction

The machine learning algorithms can only work on numeric vectors that are transformed from the tokenized sentences. A variety of transformation methods can "represent" words as vectors, and good methods should be able to easy the process of extracting "features" from texts. In general, there are two approaches to word representation: bag-of-words and word embedding.

2.3.1 Bag-of-words approach

Bag-of-words approach considers words in texts as mutually independent, as a result, disregards the order of words in text. For example, "we are health service organization" and "health organization service are we" are the same bag-of-words. This approach serves as the basis for developing many simple lan-

Table 3: Composition of Universal Classification Files

Major Group	Training (#)	Training (%)	Testing (#)	Testing (%)	Reported (#)	Reported (%)
A	17,010	11.02%	4,291	11.11%	35,813	6.77%
B	25,827	16.72%	6,419	16.63%	67,879	12.83%
C	3,323	2.15%	827	2.14%	9,054	1.71%
D	4,239	2.75%	1,034	2.68%	8,740	1.65%
E	9,015	5.84%	2,307	5.98%	25,643	4.85%
F	2,301	1.49%	543	1.41%	8,481	1.60%
G	5,053	3.27%	1,353	3.50%	10,697	2.02%
H	467	0.30%	126	0.33%	2,203	0.42%
I	2,947	1.91%	740	1.92%	8,687	1.64%
J	4,772	3.09%	1,132	2.93%	15,841	2.99%
K	2,009	1.30%	522	1.35%	7,444	1.41%
L	5,942	3.85%	1,537	3.98%	20,428	3.86%
M	4,693	3.04%	1,140	2.95%	10,857	2.05%
N	15,460	10.01%	3,925	10.17%	43,987	8.31%
O	1,731	1.12%	409	1.06%	7,878	1.49%
P	9,180	5.94%	2,318	6.00%	40,880	7.73%
Q	1,987	1.29%	436	1.13%	7,288	1.38%
R	1,064	0.69%	257	0.67%	2,830	0.53%
S	14,459	9.36%	3,603	9.33%	48,387	9.14%
T	2,032	1.32%	541	1.40%	84,338	15.94%
U	1,000	0.65%	225	0.58%	3,039	0.57%
V	350	0.23%	85	0.22%	940	0.18%
W	8,357	5.41%	2,038	5.28%	20,862	3.94%
X	4,566	2.96%	1,098	2.84%	20,699	3.91%
Y	6,640	4.30%	1,701	4.41%	15,712	2.97%
Z	–	–	–	–	547	0.10%
Total	154,424	100.00%	38,607	100.00%	529,154	100.00%

Source: Numbers and percentages reported to IRS (i.e., last two columns) are from McKeever, Dietz, and Fyffe (2016). Dashed lines separate the ten broad categories.

guage models because it can efficiently represent the possibility of word's occurrence in texts (Bengfort, Bilbro, & Ojeda, 2018). We adopt two methods in this study to represent the texts: count vector and Term Frequency-Inverse Document Frequency.

Count vector counts the number of occurrences of all the words in a given text. Given a set of statements, the algorithm first builds an index of all unique words from the collection which is called vocabulary index. The algorithm then represent the texts using words' frequencies and vocabulary index. Table 4 presents a simple example of count vectors, in which "we focus on education" is represented as vector $[1,1,1,0,0,0,0]$

Term Frequency-Inverse Document Frequency (TF-IDF) normalizes raw word frequencies using the number of documents in which the word appears. As Eq. 1 presents, tf_{ij} is the frequency of word i in mission statement j, weighted by the inverse document frequency (i.e., idf_i; Eq. 2), where N^{total} is the number of total mission statements and N^i is the number of mission statements that word i appears. The underly-

Table 4: EXAMPLE OF COUNT VECTORS

statements X vocabulary	we	focus	on	education	health	care	about
we focus on education	1	1	1	0	0	0	0
health care care	0	0	0	0	1	2	0
we care about	1	0	0	0	0	1	1

ing assumption of TF-IDF is that the words appear in all statements are not as important as those occur in a limited number of statements (Jurafsky & Martin, 2017, p. 278).

$$w_{ij} = tf_{ij} \cdot idf_i \tag{1}$$

$$idf_i = log(\frac{N^{total}}{N^i}) \tag{2}$$

We need to "normalize" the texts to reduce the vocabulary size before transforming using either count vector or TF-IDF, because the same word can have numerous spelling variations. For example, "environments," "environmental," and "environment" represent the same root word (i.e., *stem*) "environ." Otherwise, the ML models will suffer from "the curse of dimensionality": as the feature increases, the data becomes more discrete and less informative to decision making (Bellman, [1961] 2015, p. 94).

The process of finding stems is called "morphological parsing" which includes two primary methods: *stemming* and *lemmatizing* (Jurafsky & Martin, 2017, p. 25). Stemming slices longer strings to smaller ones according to a series of predefined rules. For example, "ational" is transformed to "ate" in all words ending with the former string. Therefore, stemming tend to have errors of both over- and under-parsing. Lemmatizing is a more advanced method which reduces a word to its stem by analyzing its meaning.

2.3.2 Word embedding approach

Disregarding the contexts in which the words appear is an evident drawback of bag-of-words approach. The word embedding approach represents words in a multi-dimensional space (i.e., each word has a vector value), in which words that often appear together in texts have closer distance with each other (Jurafsky & Martin, 2017, p. 290; Bengfort et al., 2018, p. 65). We can either train our own word vectors which require a large corpus and is time-consuming, or use pre-trained word vectors. We use the 100-dimension word vectors pre-trained from a corpus of 6 billion word tokens (Pennington, Socher, & Manning, 2014).

2.4 Training and intermediate decision making

2.4.1 Imbalanced dataset resampling

Training using imbalanced dataset like UCF-Training can bias our estimation because machine learning algorithms cannot extract enough information from minority classes (e.g., group H and V). Therefore, resampling the imbalanced dataset to build a more balanced one is crucial for predicting minority classes. We experimented with four strategies of over-sampling (i.e., ADASYN, RandomOverSampler, and

SMOTE) and two strategies of over-sampling followed by under-sampling to reduce the noise (i.e., SMOTEENN and SMOTETomek; Lemaître, Nogueira, and Aridas, 2017). The influence of resampling is substantial: the F1 score of predicting minority class broad category VI (i.e., major group Q) were improved from 15% to over 30% in our pilot experiments.

2.4.2 Classifiers for training

Naïve Bayes (NB) classifier is built on Bayeś theorem. It is one of the simplest classifiers to learn and implement among all machine learning algorithms and built on simple conditional probability principles. The classifier assumes all features extracted from the texts are conditionally independent, which is wrong in most cases. But the classifier is efficient and has proven to be useful for a variety of tasks even on a small dataset (Jurafsky & Martin, 2017, p. 76; Grimmer & Stewart, 2013, p. 277). We tested two types of NB classifiers: multinomial and Complement NB classifiers (Rennie, Shih, Teevan, & Karger, 2003).

Random Forest classifier is implemented by developing multiple prediction models. Each model in this algorithm is trained by different data, and then all of these models are asked to predict for the same record. A prediction class that is elected by most of these small algorithms is given as the prediction result by the random forest algorithm. It uses the word "forest" because each small algorithm trained is a decision tree (Quinlan, 1986, p. 83). A decision tree represents a set of questions that usually have Yes/No answers. The process starts from the top of the tree with one question, and based on the answer, we further run down on either one side of the tree, and answer another question and repeat till we reach the end of the tree. Each decision tree is trained on a different training set (Breiman, 1996, p. 124).

Take our study for example, if we provide 5,000 statements with their NTEE codes to a random forest with five trees, each tree will randomly select a thousand records to train. Each tree includes new words at different levels of the tree. For example, the tree starts with word "emergency," it will have two branches for "yes" or "no," leading to another word and so on, till the bottom of the tree where the NTEE code is. Since each tree is trained on a different set, when a record is given for prediction, each tree predicts the class independent of other tree. In total of five class predictions will be collected in this case, and the class which has the highest occurrence in the prediction results is given as the final predicted class by the random forest algorithm.

Since each decision tree in a Random Forest classifier is provided with a unique set of records for the training purpose, it strengthens the performance of the overall forest. The classifier however is difficult to visually interpret. It takes a little effort to visualize how decision trees work and understand the algorithm, unlike the Naïve Bayes approach.

Neural Network (NN) classification is built on the concepts of a neuron structure of the human mind. Each neuron in the network is connected to a few other neurons of the network by a numerical value called "weight." The neurons process records each one in turn, and learn by looking at their classification (i.e., NTEE code in this case) with the known previous NTEE codes of records. With every new record the neurons learn, they update the connection value "weight" to update the model (Collobert & Weston, 2008, p. 163). After the network is done processing each record of the training set, it has final weights for each connection between two neurons. When a testing set is provided, the neurons use the final weights to predict the NTEE code. Depending on the architecture of the neurons, we can design a variety of NNs (e.g., the basic fully connected, Recurrent, and Long Short-Term Memory). This study uses Convolutional NN (CNN) following other scholars' recommendation (Zhang & Wallace, 2015).

2.4.3 Measuring algorithm performance

An algorithm's performance can be measured by numerous metrics, but social scientists particularly concern two questions while solving real-word problems: 1) how many predicted observations are correct (i.e., *precision* calculated by Eq. 3)? 2) how many observations are correctly predicted (i.e., *recall* calculated by Eq. 4). Answering the two questions is critical for social scientists to apply ML research methods, and we will analyze the methodological implications and recommended practices in discussion section.

In Eq. 3, k is one of the NTEE codes, $\#Org_k^{corr}$ is the number of organizations correctly classified as k, and $\#Org_k^{pred}$ is the number of organizations predicted as k. $\#Org_k^{corr}$ will always be smaller than or equal to $\#Org_k^{pred}$ because ML algorithms can hardly predict every observation right. For example, $Precision_B = 0.75$ indicates that 75% of all the organizations classified as "education" are correct.

$$Precision_k = \frac{Org_k^{corr}}{Org_k^{pred}} \tag{3}$$

Assuming robust human-coders' coding of nonprofits as true value, in Eq. 4, Org_k^{true} is the number of organizations that belong to k category. For example, $Recall_B = 0.80$ denotes that 80% of the organizations classified as "education" by robust human-coding are correctly identified by the algorithm.

$$Recall_k = \frac{Org_k^{corr}}{Org_k^{true}} \tag{4}$$

2.4.4 Intermediate decision making

The goal of this study is to find the best ML algorithm with appropriate parameters. We can either try some of the configurations randomly (i.e., *stochastic search*), or iterate all possible configurations (i.e., *grid search*). For NB and RF algorithms, we used the latter approach. For NN algorithms, we first used stochastic search to narrow down the configurations of hidden layers, and then conducted a grid search for the input and output layers' parameters using CNN. The grid search for all possible parameters (over 2 million combinations) is impossible even by using one of the most advanced super computing clusters in the world.

We conduced two rounds of grid search. The first found is for *satisficing decision making* in which we only considered the configurations that can perform at the top 5 percent. Then we ran the second found grid search for *optimizing decision making* in which we increased the values of some parameters to allow the algorithms to reach their performance ceilings. We then choose the best algorithm and parameters for final training.

Table 5: COMPARING CONVOLUTIONAL NEURAL NETWORK CLASSIFIERS

Code	Precision-*N*	Precision-*R*	Recall-*N*	Recall-*R*	F1-*N*	F1-*R*	%Obs.
I	87%	83%	85%	87%	86%	85%	11%
II	85%	91%	88%	78%	86%	84%	17%
III	76%	83%	90%	82%	82%	82%	5%
IV	76%	88%	87%	70%	81%	78%	11%
V	85%	77%	86%	90%	85%	83%	30%
VI	59%	44%	8%	21%	14%	29%	1%
VII	88%	83%	76%	79%	81%	81%	17%
VIII	65%	71%	77%	70%	71%	71%	3%
IX	90%	80%	85%	92%	88%	85%	4%

Notes: *N* = No resampling; *R* = Resampling.

3 Results

3.1 Selecting the best classifier

For multi-class classification task (i.e., more than two classes to predict), it is difficult to measure the overall performance because for each category the performance differs. Table 5 presents the performance of CNN classifiers with and without resampling. Because the dataset is imbalanced, the classifier has a poor performance on category *VI International, Foreign Affairs* without resampling. Training the classifier with resampled dataset substantially improves the F1 score from 14% to 29%, but slightly sacrifices the performance on other categories. So which one should we choose?

We choose the classifier trained without resampling as the best model because even the F1 score of *VI* is substantially improved, we still cannot use the predicted results of this category (21% identified among which only 44% are correct). We recommend not sacrificing the performance on other categories as researchers need to manually check or completely drop this category in their analysis anyway. This rationale applies to selecting other classifiers.

3.2 Performance of best model: Convolutional Neural Network

In general, CNN classifier achieves the best performance. For classifying the 10 broad categories, 83.47% records in the UCF-Testing dataset are correctly recognized; for the 25 major groups task, 77% are correctly classified. Detail satisficing decision table can be found in source code repository published online (`/output` folder). The precision and recall for each category or group varies as Table 6 and Table 7 present.

Our CNN classier approximates or even outperforms human coders on many broad categories (i.e., I, V, VI, VIII) and major groups (i.e.,). For example, the classifier outperforms human coders on broad category *VII Public, Societal Benefit*: 76% *VII* organizations are identified, and among these identified organizations, 88% are correct – nearly 12% higher than human coders' performance. For major group *W Public, Society Benefit*: 89% *W* organizations are identified, and among these identified organizations, 83% are correct – 29% higher than human coders' performance.

Table 6: PERFORMANCE OF MODEL: BROAD CATEGORY

Code	H-Precision	Precision	Recall	F1
I	88.49%	86.85%	84.81%	85.82%
II	92.72%	84.58%	88.13%	86.32%
III	87.00%	76.08%	89.58%	82.28%
IV	92.00%	75.90%	87.29%	81.20%
V	86.00%	90.07%	85.30%	87.62%
VI	76.64%	85.22%	85.55%	85.38%
VII	76.00%	59.32%	8.03%	14.14%
VIII	86.57%	87.91%	75.92%	81.48%
IX	89.52%	65.38%	76.87%	70.66%

Notes: H-Precision = Human Coder Precision, compiled from Stengel et al. (1998, p. 153).

Table 7: PERFORMANCE OF MODEL: MAJOR GROUP

Code	H-Precision	Precision	Recall

Notes: H-Precision = Human Coder Precision, compiled from Stengel et al. (1998, p. 153).

3.3 Functions of software package for classifying texts using NTEE codes

4 Discussion

Complementing or replacing?

Practical suggestions to social scientists solving real-world problems. The results of performance in this paper indicates that, for social scientists who want to apply computational methods in their research should be “cautiously confident.” The key here supporting our confidence is a robust validation (Grimmer & Stewart, 2013, p. 271). Otherwise, it will be “garbage in, garbage out.” Many factors can influence the validity of the algorithm. For example, the algorithm may have a poor performance on a datset that is structurally different from the training dataset. We strongly suggest that readers should check the annotations in our scripts posted online to understand the caveats and make necessary optimizations according to their own research question.

Social scientists should also take the advantage of high performance computing (HPC) research infrastructures (e.g., Keahey et al., 2018). The ML algorithms can achieve best performance only when trained with a large amount of data, and such training process consumes a huge amount of computing resources which is far beyond the capacity of the most advanced personal computers. At the grid search phase of this study, we used two most advanced GPU accelerators (NVIDIA Tesla P100) for NN training and six 48-CPU computing servers for NB and RF training. HPC infrastructures are widely used in natural sciences but it is still new to social scientists. We encourage methodology workshops to incorporate the introduction of HPC infrastructures as part of their syllabus.

Future studies.

Acknowledgments

Earlier version of this paper was presented at 2019 West Coast Nonprofit Data Conference. The authors acknowledge the Texas Advanced Computing Center at The University of Texas at Austin for providing cloud computing resources (Chameleon Cloud) that have contributed to the research results reported within this paper.

References

Baćak, V., & Kennedy, E. H. (2018). Principled Machine Learning Using the Super Learner: An Application to Predicting Prison Violence. *Sociological Methods & Research*, 0049124117747301. doi:10.1177/0049124117747301

Barman, E. (2013). Classificatory Struggles in the Nonprofit Sector: The Formation of the National Taxonomy of Exempt Entities, 1969—1987. *Social Science History*, *37*(1), 103–141.

Bellman, R. E. ([1961] 2015). *Adaptive Control Processes, A Guided Tour*. doi:10.1515/9781400874668

Bengfort, B., Bilbro, R., & Ojeda, T. (2018). *Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning* (1 edition). Beijing Boston Farnham Sebastopol Tokyo: O'Reilly Media.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. doi:10.1007/BF00058655

Carman, J. G., & Fredericks, K. A. (2010). Evaluation Capacity and Nonprofit Organizations: Is the Glass Half-Empty or Half-Full? *American Journal of Evaluation*, *31*(1), 84–104. doi:10.1177/1098214009352361

Clarke, A. E., & Casper, M. J. (1996). From Simple Technology to Complex Arena: Classification of Pap Smears, 1917-90. *Medical Anthropology Quarterly*, *10*(4), 601–623.

Collobert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 160–167). ICML '08. doi:10.1145/1390156.1390177

Durkheim, É. ([1912] 2012). *The Elementary Forms of the Religious Life*. Courier Corporation.

Fyall, R., Moore, M. K., & Gugerty, M. K. (2018). Beyond NTEE Codes: Opportunities to Understand Nonprofit Activity Through Mission Statement Content Coding. *Nonprofit and Voluntary Sector Quarterly*, *47*(4), 677–701. doi:10.1177/0899764018768019

Grimmer, J., & Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, *21*(3), 267–297. doi:10.1093/pan/mps028

Grønbjerg, K. A. (1994). Using NTEE to classify non-profit organisations: An assessment of human service and regional applications. *Voluntas: International Journal of Voluntary and Nonprofit Organizations*, *5*(3), 301–328. doi:10.1007/BF02354038

Hall, P. D. (2006). A Historical Overview of Philanthropy, Voluntary Associations, and Nonprofit Organizations in the United States, 1600–2000. In W. W. Powell & R. Steinberg (Eds.), *The nonprofit sector: A research handbook* (pp. 32–65). Yale University Press.

Hodgkinson, V. A. (1990). Mapping the non-profit sector in the United States: Implications for research. *Voluntas: International Journal of Voluntary and Nonprofit Organizations*, *1*(2), 6–32. doi:10.1007/BF01397436

Hodgkinson, V. A., & Toppe, C. (1991). A new research and planning tool for managers: The national taxonomy of exempt entities. *Nonprofit Management and Leadership*, *1*(4), 403–414. doi:10.1002/nml.4130010410

Jurafsky, D., & Martin, J. H. (2017). *Speech and Language Processing* (3rd draft).

Keahey, K., Riteau, P., Stanzione, D., Cockerill, T., Mambretti, J., Rad, P., & Ruth, P. (2018). Chameleon: A Scalable Production Testbed for Computer Science Research. In J. Vetter (Ed.), *Contemporary High Performance Computing: From Petascale toward Exascale* (1st ed., Vol. 3). Chapman & Hall/CRC Computational Science. Boca Raton, FL: CRC Press.

Lampkin, L., Romeo, S., & Finnin, E. (2001). Introducing the Nonprofit Program Classification System: The Taxonomy We've Been Waiting for , Introducing the Nonprofit Program Classification System: The Taxonomy We've Been Waiting for. *Nonprofit and Voluntary Sector Quarterly*, *30*(4), 781–793. doi:10.1177/0899764001304009

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res. 18*(1), 559–563.

McKeever, B. S., Dietz, N. E., & Fyffe, S. D. (2016). *The Nonprofit Almanac: The Essential Facts and Figures for Managers, Researchers, and Volunteers*. Rowman & Littlefield.

National Center for Charitable Statistics. (2006). *Guide to Using NCCS Data*. Urban Institute. Washington, DC.

Nelson, L. K., Burk, D., Knudsen, M., & McCall, L. (2018). The Future of Coding: A Comparison of Hand-Coding and Three Types of Computer-Assisted Text Analysis Methods. *Sociological Methods & Research*, 0049124118769114. doi:10.1177/0049124118769114

Okten, C., & Weisbrod, B. A. (2000). Determinants of donations in private nonprofit markets. *Journal of Public Economics*, *75*(2), 255–272. doi:10.1016/S0047-2727(99)00066-3

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). doi:10.3115/v1/D14-1162

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. doi:10 . 1007 / BF00116251

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (pp. 616–623). ICML'03. AAAI Press.

Roeger, K. L., Blackwood, A. S., & Pettijohn, S. L. (2015). The Nonprofit Sector and Its Place in the National Economy. In J. S. Ott & L. A. Dicke (Eds.), *The Nature of the Nonprofit Sector* (Third edition, pp. 22–37). Boulder, CO: Westview Press.

Salamon, L. M. [Lester M.], & Anheier, H. K. (1992). In search of the non-profit sector II: The problem of classification. *Voluntas: International Journal of Voluntary and Nonprofit Organizations*, *3*(3), 267–309. doi:10.1007/BF01397460

Salamon, L. M. [Lester M], Anheir, H. K., & coaut. (1996). *The international classification of nonprofit organizations ICNPO-Revision 1, 1996*. Baltimore, Md: The Johns Hopkins University Institute for Policy Studies.

Sloan, M. F. (2009). The Effects of Nonprofit Accountability Ratings on Donor Behavior. *Nonprofit and Voluntary Sector Quarterly*, *38*(2), 220–236. doi:10.1177/0899764008316470

Stengel, N. A. J., Lampkin, L. M., & Stevenson, D. R. (1998). Getting It Right: Verifying the Classification of Public Charities in the 1994 Statistics of Income Study Sample. In Statistics of Income Division & Internal Revenue Service (Eds.), *Turning Administrative Systems Into Information Systems* (Vol. 6, pp. 145–167). Statistics of Income Division, Internal Revenue Service.

US Internal Revenue Service. (2013). IRS Static Files No. 2013-0005. https://www.irs.gov/pub/irs-wd/13-0005.pdf.

US Internal Revenue Service. (2014). Exempt Organizations Business Master File Information Sheet. https://www.irs.gov/pub/irs-soi/eo_info.pdf.

US Internal Revenue Service. (2018). 2017 Instructions for Form 990-EZ. https://www.irs.gov/pub/irs-pdf/i990ez.pdf.

Vakil, A. C. (1997). Confronting the classification problem: Toward a taxonomy of NGOs. *World Development*, *25*(12), 2057–2070. doi:10.1016/S0305-750X(97)00098-3

Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *arXiv:1510.03820 [cs]*. arXiv: 1510.03820 `[cs]`