Classifying nonprofits using supervised machine-learning: Benchmark and guide for social scientists solving real-world problems

Ji Ma and Isha Kanani [*]

University of Texas at Austin

May 12, 2019

Abstract

This research classified nonprofit organizations in the United States using supervised machine-learning algorithms according to text descriptions, built a workflow and benchmark to apply computational linguistics for future sociological studies, and can help researchers more accurately profile a civil society. We achieved 83% overall accuracy for classifying the nonprofits into 10 broad categories, and 78% for classifying them into 25 major groups. Our study suggests that machine-learning algorithms, and neural networks in particular, can substantially and reliably improve researchers' productivity because they can approximate or even outperform human coders on many categories. This leaves human researchers free to focus on the categories on which the algorithms have poor performance. We provided a clear strategy and operationalization method on how machine-learning classifiers can "complement" human researchers. A Python software package has been developed for scholars. Practical suggestions and future directions are also 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 voluntary and philanthropic organizations have long existed in numerous centuries, the term "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 was to legitimize their existence and the benefits they 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) is the most widely used classification system and represents one of the efforts put forth to legitimize the existence of the nonprofit sector (Hodgkinson, 1990; Hodgkinson & Toppe, 1991). Barman (2013, p. 105) cites the following observation from 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 be dated back to the 1980s (Hodgkinson, 1990, pp. 8–9, 11). In 1982, the NCCS assembled a team of experts who were working on creating a taxonomy for nonprofit organizations. The first draft of the NTEE came out in 1986 and was published in 1987. In the early 1990s, the NCCS had classified nearly one million nonprofits using the NTEE. In 1995, the Internal Revenue Service (IRS) adopted the NTEE coding system, took over the task 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 were responsible for assigning the NTEE codes: the NCCS and the IRS. Before 1995, the NCCS coded nonprofits according to the program descriptions in Parts III and VIII of Form 990, which were supplemented with information from Form 1023 ("Application for Recognition of Exemption") and additional research (National Center for Charitable Statistics, 2006, p. 16). After 1995, the IRS began to issue "new exempt organizations an NTEE code as part of the determination process," and "the determination specialist [assigned] 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 [was] recognized as tax-exempt" (US Internal Revenue Service, 2013, p. 1).

The NTEE classification system has supported many applied and academic studies on nonprofit organizations that have critical economic and political roles in society. For example, the NTEE provides a framework through which the social and economic activities of civil society can be mapped and compared with other sectors in the society (e.g., Roeger, Blackwood, & Pettijohn, 2015). Scholars can use NTEE codes to sample nonprofits of interests (e.g., McVeigh, 2006; Okten & Weisbrod, 2000; Sharkey, Torrats-Espinosa, & Takyar, 2017; Vasi, Walker, Johnson, & Tan, 2015) or as independent variables (Sloan, 2009). The NTEE can also serve as an analytical tool to measure organizational capacity in different service domains and inform practitioners and policy makers' decision making (Hodgkinson & Toppe, 1991). Moreover, the NTEE is a fundamental necessity for comparative international research and facilitates the study of "global civil society" (Hodgkinson, 1990; Salamon & Anheier, 1992; Salamon & Anheir, 1996; Vakil, 1997).

But the NTEE classification system, although one of the best we have so far, still has several drawbacks. First, because the NTEE only assigns one major category code to an organization, it cannot accurately describe a nonprofit's programs which are usually diverse and spread across several service domains (Grønbjerg, 1994, p. 303). Even though a program classification system was later developed (Lampkin, Romeo, & Finnin, 2001), it is not widely used, probably because it is impractical to assign codes to a massive number 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 mission statements were used for coding (Fyall, Moore, & Gugerty, 2018).

Third, NTEE codes are static while nonprofit organizations' activities may change over time. Recoding existent NTEE assignments is extremely onerous, and this may be one of the reasons that the IRS does not have a procedure through which nonprofits can request a change to their NTEE codes (US Internal Revenue Service, 2013).

Fourth, a vast amount of grassroots organizations are not classified and remain missing in existing datasets because an organization "that normally has annual gross receipts of $50,000 or less" is not required to report to the IRS (US Internal Revenue Service, 2019). As Smith (1997) estimates, the IRS listings ignore about 90% of nonprofits, most of which are grassroots associations. Organizational activities at the grassroots level are particularly important for sociological and political studies, and most studies fail to consider these grassroots organizations because of the dataset limitation (e.g., McVeigh, 2006; Sharkey et al., 2017; Vasi et al., 2015).

The tremendous amount of human labor needed for classification is a prominent challenge and is also an evident barrier to improving any classification system. Numerous social scientists have experimented with applying computational methods to classify objects and solve real-world problems (e.g., Baćak & Kennedy, 2018; Fyall et al., 2018; Grimmer & Stewart, 2013; Nelson, Burk, Knudsen, & McCall, 2018). We respond to this challenge by applying the advances made in computational linguistics and contribute to the growing literature from four aspects: 1) we established a standardized workflow and benchmarks that future studies of nonprofits or typologies in other social science disciplines can build on and make comparisons to, 2) we achieved 83% overall accuracy for classifying the nonprofits into 10 broad categories and 78% for classifying them into 25 major groups, 3) we developed a Python software package for scholars to classify text descriptions using NTEE codes, and 4) we released all source codes, data, and work history for replication purposes and future studies. This last point is particularly important since there are many caveats in tuning algorithms that cannot be detailed in this paper.[1]

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

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, Lines 2a-e, Lines 11a-c; Schedule O
990-EZ	Part III	Part III, Lines 28-30; Schedule O
990-PF	–	Part IX-A; Part XVI-B

2 Method

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

Figure 1 presents this paper's complete workflow. We implement four stages of analysis: 1) the *preprocessing stage* includes data acquisition and the preprocessing of datasets and texts; 2) *feature extraction* includes a bag-of-words representation (used by naïve Bayes and random forest algorithms) and word embedding (used by neural network algorithms); 3) the *training and intermediate decision-making* phase, is where we use stochastic and grid search to train, search, and optimize the machine-learning algorithms; and 4) the last phase involves *training the model finalist* with the complete dataset and preparing the trained model for public use. The rest of this section introduces the four phases in detail.

2.1 Data preprocessing

Data acquisition and dataset preprocessing. We collected text records from Forms 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 nonprofit organizations. 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) 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 describe organizational activities in two forms: the overall mission statement and specific program descriptions. Table 1 summarizes these text fields' specific locations on the different forms.

Classification records (i.e., NTEE codes) were collected from the 2014–2016 Business Master Files on

Figure 1: RESEARCH WORKFLOW

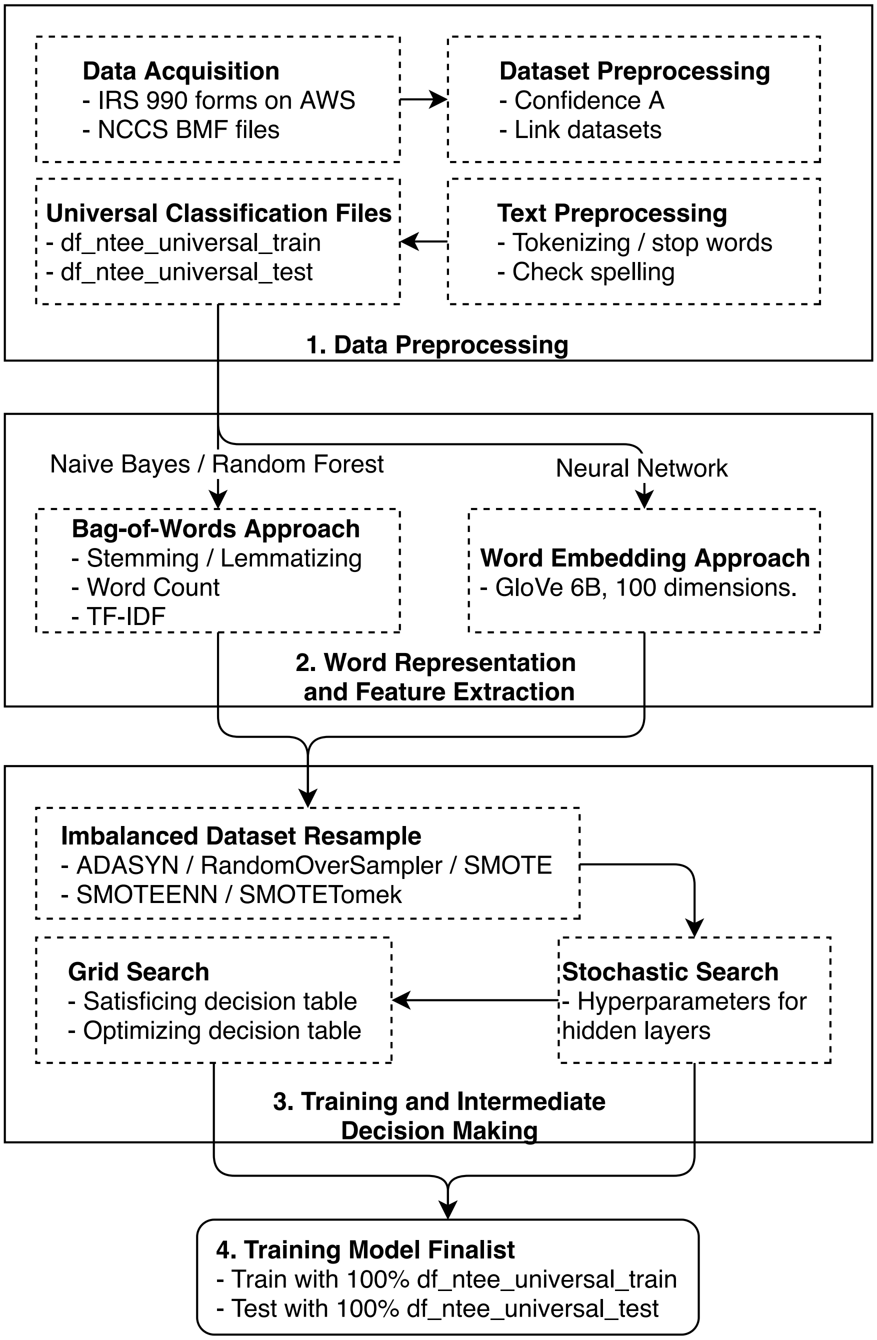

Table 2: NTEE-CC CLASSIFICATION SYSTEM

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 NCCS website.[2] This study deals with two types of NTEE classifications: 10 broad categories and 26 major groups. Table 2 shows the relationship between the broad categories and major groups. A detailed list of the 26 major groups can be found through the US Internal Revenue Service (2014). The accuracy of a classification is indicated by the letters of A, B, and C, where a "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 level A 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., the IRS and NCCS) over different periods of time.

From 2014 to 2016, 56.12% of records were classified at level A, 37.32% at level B, and 6.56% at level C. For training purposes, we only used records at confidence level A and dropped all records in the *X/Z* category (i.e., unknown or unclassified). About 1.76% of organizations changed their NTEE codes between 2014 and 2016. We dropped the records of these organizations as well 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" (i.e., the minimum number of editing operations needed to change one word into another; Jurafsky & Martin, 2017, p. 26).

Universal Classification Files (UCFs). The final step in the data preprocessing stage is to divide data records into training and testing datasets (i.e., files in `/dataset/UCF/`) that are mutually exclusive and can be used to benchmark future models. The *Universal Classification File Training* (UCF-Training; `df_ucf_train.pkl.gz`) is used to develop models and comprises 80% of the total records. The *Universal Classification File Testing* (UCF-Testing; `df_ucf_test.pkl.gz`) is used to test a trained model's performance and comprises 20% of the total records. Table 3 presents the two datasets' composition by major groups. The UCFs approximate the composition of organizations reported to the IRS except for groups *A* ("arts, culture, and

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

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%

Note: Numbers and percentages reported to the Internal Revenue Service (i.e., the last two columns) are from McKeever, Dietz, and Fyffe (2016). Dashed lines separate the 10 broad categories.

humanities") and *T* ("philanthropy, voluntarism, and grantmaking foundations"). The consequence is that, compared to human coders, the trained models are less likely to categorize organizations as *T* and more likely to categorize organizations as *A*.

2.2 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 ease the process of extracting "features" from texts. In general, there are two approaches to word representation: bag-of-words and word embedding.

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

2.2.1 Bag-of-words approach

The bag-of-words approach considers words in texts as being mutually independent and thus disregards the order of the words. For example, "we are health service organization" and "health organization service are we" are the same from a bag-of-words perspective. This method serves as the basis for developing many simple language models because it can efficiently represent the possibility of a word's occurrence in texts (Bengfort, Bilbro, & Ojeda, 2018). We adopted 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 that is called the vocabulary index. The algorithm then represent the texts using word frequencies and the vocabulary index. Table 4 presents a simple example of count vectors in which the statement "we focus on education" is represented as the 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 a given 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 in which word i appears. The underlying assumption of TF-IDF is that any words appearing in all the statements are not as important as those occurring 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 machine-learning 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 into 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 tends to have both over- and under-parsing errors. Lemmatizing is a more advanced method that reduces a word to its stem by analyzing its meaning.

2.2.2 Word embedding approach

Disregarding the contexts in which the words appear is an evident drawback of the bag-of-words approach. The word embedding approach is a new advancement (Mikolov, Chen, Corrado, & Dean, 2013) and was suggested by Nelson et al. (2018, p. 28) as a future direction for sociological studies. It represents words in a multidimensional space (i.e., each word has a vector value), and in this space words that often appear together in texts are closer in distance to each other (Jurafsky & Martin, 2017, p. 290; Bengfort et al., 2018, p. 65). We can either train our own word vectors, which would require a large corpus and is time-consuming, or use pretrained word vectors. In this study, we use 100-dimension word vectors pretrained from a corpus of 6 billion word tokens (Pennington, Socher, & Manning, 2014).

2.3 Training and intermediate decision making

2.3.1 Imbalanced dataset resampling

Training using an imbalanced dataset such as UCF-Training can bias our prediction of minor classes because machine-learning algorithms cannot extract enough information from these classes (e.g., groups H and V). Therefore, resampling the imbalanced dataset to build a more balanced one is crucial for predicting minority classes. We experimented with three 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, & Aridas, 2017). The influence of resampling is substantial: the F_1 score of predicting minority class major group Q was improved from 15% to over 30% in our pilot experiments.[3]

2.3.2 Classifiers for training

The *naïve Bayes (NB) classifier* is built on Bayes' theorem. It is one of the simplest classifiers to learn and implement among all machine-learning algorithms and is 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: the multinomial and complement NB classifiers (Rennie, Shih, Teevan, & Karger, 2003).

The *random forest (RF) 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 make a prediction for the same record. A prediction class that is elected by most of these small algorithms is given as the prediction result by the RF 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 run down either side of the tree and answer another question. We repeat this process until reaching the end of the tree. Each decision tree is trained on a different training set (Breiman, 1996, p. 124).

[3] Although major group Q and broad category VI represent the same group of organizations, for computer algorithms, the classification contexts are different; therefore, performance on this category varies.

Take our study for example. If we provide 5,000 statements with their NTEE codes to an RF with 9 trees (i.e., each tree corresponds to a broad category code), each tree will randomly select a thousand records to train. Each tree includes new words at different levels of the tree. For example, a tree starting with the word "emergency" will have a branch for "yes" and "no" that leads to another word and so on– all the way to 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 the other trees. A total of 9 class predictions were collected in this case, and the class with the highest occurrence in the prediction results was given as the final predicted class by the RF algorithm.

Since each decision tree in an RF classifier is supplied a unique set of records for training purposes, the performance of the overall forest is stronger. The classifier however is difficult to visually interpret. Unlike the Naïve Bayes approach, it takes some efforts to visualize how decision trees work and understand the algorithm.

Neural network (NN) classification is built on the structure of a neuron in the human mind. Each neuron in the network is connected to a few other neurons of the network by a numerical value called "weight." Each neuron processes 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, or long short-term memory). This study uses convolutional NN (CNN) following other scholars' recommendation (Zhang & Wallace, 2015).

2.3.3 Measuring algorithm performance

An algorithm's performance can be measured by many 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.

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

F_1 score (Eq. 5), the harmonic mean of precision and recall, was introduced to balance the two measures.

$$F_{1k} = \frac{2 \cdot Precision_k \cdot Recall_k}{Precision_k + Recall_k} \tag{5}$$

2.3.4 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 (240 parameter combinations for NB and RF, 7,200 for NN, detail history files are in folder `output`). 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.

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 F_1 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 F_1 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. For social scientists, mathematical improvements may not make substantial and practical meaning. This rationale

Table 5: Comparing Convolutional Neural Network classifiers

Code	Precision-*N*	Precision-*R*	Recall-*N*	Recall-*R*	F_1-*N*	F_1-*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%

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

Table 6: Performance of best model on broad category

Code	H-Precision	Precision	Recall	F_1
I	88%	87%	85%	86%
II	93%	85%	88%	86%
III	87%	76%	90%	82%
IV	92%	76%	87%	81%
V	86%	85%	86%	85%
VI	77%	59%	8%	14%
VII	76%	88%	76%	81%
VIII	87%	65%	77%	71%
IX	90%	90%	85%	88%

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

applies to selecting other classifiers.

3.2 Performance of best model

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., *V*, *VI*, *VIII*) and major groups (i.e., *A*, *G*, *J*, *N*, *S*, and *W*). 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*: 86% *W* organizations are identified, and among these identified organizations, 87% are correct – 29% higher than human coders' performance.

Table 7: PERFORMANCE OF BEST MODEL ON MAJOR GROUP

Code	H-Precision	Precision	Recall	F_1
A	88%	80%	87%	83%
B	93%	85%	85%	85%
C	86%	65%	74%	69%
D	90%	80%	90%	85%
E	92%	77%	78%	78%
F	86%	51%	60%	55%
G	65%	68%	68%	68%
H	73%	55%	19%	28%
I	84%	71%	71%	71%
J	72%	86%	67%	75%
K	82%	63%	68%	66%
L	83%	70%	76%	73%
M	88%	87%	90%	88%
N	88%	83%	93%	88%
O	91%	65%	61%	63%
P	88%	64%	57%	60%
Q	77%	43%	36%	39%
R	67%	46%	21%	28%
S	75%	84%	79%	81%
T	78%	66%	32%	43%
U	76%	52%	22%	31%
V	24%	0%	0%	0%
W	58%	87%	86%	86%
X	87%	68%	71%	70%
Y	90%	84%	91%	88%
Z	10%	–	–	–

Note: H-Precision = Human Coder Precision, compiled from Stengel et al. (1998, p. 153). Dashed lines separate the ten broad categories.

3.3 Python package for classifying texts

We developed a Python package (`npoclass`) for classifying texts using NTEE codes, and scholars can use it free of charge.[4] Although the package can work on any texts, we expect it should perform best on nonprofit organization-related narratives. `npoclass` has the following features, and detail instructions are in the package's documentation:

1. Take a single string text or a list of of text descriptions as input;
2. Return predicted codes of both broad category and major group;
3. Return probability on each code of broad category and major group;
4. Parallelize predicting process on multi-core computers.

4 Discussion

We achieved 83% overall accuracy for classifying the nonprofits into 10 broad categories according to their text descriptions, and 78% for classifying them into 25 major groups. We detailed the caveats and built a workflow and benchmark of applying computational linguistics for future sociological studies.

An encouraging conclusion of this study is, the ML algorithms, neural networks in particular, can substantially and reliably improve researchers' productivity as they can approximate or even outperform human coders on many categories. Therefore, human researchers can focusing on the categories on which the algorithms have poor performance. We provided a clear strategy and operationalization method on how machine-learning classifiers can "complement" human researchers (Nelson et al., 2018, p. 25).

This study also enables social scientists to examine social, political, and economic activities at grassroots level. As discussed in the introduction section, NTEE as a typology at organizational level is not accurate because one organization can have multiple programs, and the classification of programs has never been done before. This study can help researchers to code nonprofits' activities at program level, profiling a more accurate civil society.

Some 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

[4] https://github.com/****

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 can make numerous improvements based on the workflow and benchmark introduced in this paper. First, students on this topic can experiment with more classifiers and parameters. For example, using a more accurate nonprofit-specific glossary and stemmer (Paxton, Velasco, & Ressler, 2019), and a large-scale competition is also in preparation.[5] We also deposited our working directory with all datasets, source codes, and historical versions on GitHub, enabling future large-scale collaborations on this project possible. Second, scholars are invited to use the Python software package for their own empirical studies and provide any feedback. Third, computational social scientists can apply the workflow in this paper to other domains of inquiry. Last but not least, we are advancing a multi-lingual version of this project to assist the study of/with nonprofits/NGOs in non-English speaking countries. This step is essential to develop the global civil society studies further.

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

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

[5] https://***.

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.

McVeigh, R. (2006). Structural Influences on Activism and Crime: Identifying the Social Structure of Discontent. *American Journal of Sociology*, *112*(2), 510–566. doi:10.1086/506414

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*. arXiv: 1301.3781 [cs]

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

Paxton, P., Velasco, K., & Ressler, R. (2019). Nonprofit-Specific Glossary and Stemmer. https://web.archive.org/web/20190509160945/https://www.pamelapaxton.com/990missionstatements.

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

Sharkey, P., Torrats-Espinosa, G., & Takyar, D. (2017). Community and the Crime Decline: The Causal Effect of Local Nonprofits on Violent Crime. *American Sociological Review*, *82*(6), 1214–1240. doi:10.1177/0003122417736289

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

Smith, D. H. (1997). The Rest of the Nonprofit Sector: Grassroots Associations as the Dark Matter Ignored in Prevailing "Flat Earth" Maps of the Sector. *Nonprofit and Voluntary Sector Quarterly*, *26*(2), 114–131. doi:10.1177/0899764097262002

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.

US Internal Revenue Service. (2019). Annual Exempt Organization Return: Who Must File. https://www.irs.gov/charities-non-profits/annual-exempt-organization-return-who-must-file.

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

Vasi, I. B., Walker, E. T., Johnson, J. S., & Tan, H. F. (2015). ‘‘No Fracking Way!” Documentary Film, Discursive Opportunity, and Local Opposition against Hydraulic Fracturing in the United States, 2010 to 2013. *American Sociological Review*, *80*(5), 934–959. doi:10.1177/0003122415598534

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]`