DESCRIPTION AND PARAMETERS OF EACH ALGORITHM USED IN THE ENSEMBLE CLASSIFIER

To make the ML predictions, first, we deconstructed the grants in the training sets (e.g., FY2013 1R01s) and test sets (e.g., FY2014 1R01s) by extracting features from the text data. The grant titles and abstract texts from each set were converted to lowercase, tokenized, and stemmed prior to removing stopwords and punctuation. We used these features as input to train the algorithms. Next, we used each algorithm (described below) to predict whether each grant in the test set was prevention or non-prevention research. We classified each grant as prevention research or not based on the majority predictions of the ensemble classifier.

LIBLINEAR

This algorithm is an open source library for large-scale linear classification.[11] It supports logistic regression and linear support vector machines. Words appearing in the title were distinguished from words appearing in the abstract and we used the frequency of each word in a grant as its feature weight. The parameters for our LIBLINEAR algorithm were L2-regularized logistic regression with default values for all other parameters.

OpenNLP MaxEnt

This algorithm is an open source categorizer for the classification of text in pre-defined categories. It utilizes a maximum entropy modeling.[12] Words appearing in the title were distinguished from words appearing in the abstract and we used the frequency of each word in a grant as its feature weight. Parameters for our MaxEnt algorithm were Generalized Iterative Scaling (GIS) using 100 iterations with no smoothing and default values for all other parameters.

NEURAL NETWORK

This is a deep learning algorithm that can classify complicated data to extract patterns and detect trends that are too complex to be noticed by other machine learning techniques.[13] It utilizes a non-linear approach. The neural network was trained on word embeddings that were initialized using word2vec[25] and then trained using a corpus of 2,438,876 NIH grant applications from FY2000–2016. To construct a word2vec-based representation of each grant, we filtered words

that occurred in fewer than 20 grants. Then we obtained the 300-dimensional vector for each remaining word, noun phrase,[26] and Medical Subject Headings (MeSH) term[27] in the grant, and averaged these vectors to produce a vector representation for the entire grant. We used deeplearning4j to implement our Neural Network algorithm with the following default parameter settings: one iteration of each batch of data; Xavier weight initialization scheme; Tanh hidden layer activation; softmax output layer activation; negative likelihood loss function; and stochastic gradient descent optimization algorithm. We trained one network for 500 epochs and the other network for 750 epochs; both had a learning rate of 0.02 and batch size of 1000.

SCALED RELATIVE FREQUENCY RATIO

This algorithm was developed by the NIH Office of Portfolio Analysis. It uses ratios of relative frequencies between two or more categories to identify highly category-specific items.[28] Thus, in the test set, each feature in a grant's abstract was weighted more than those in the title, and then summed for each category. A grant was predicted to belong to the category with the highest weighted sum of that grant's features (each feature was weighted as described below). Words appearing in the title were distinguished from words appearing in the abstract and we used the frequency of each word in a grant as its feature weight. We further removed all features occurring in fewer than 5% of the grants in the training set and used mutual information[26] to limit the number of features to 750 per category (i.e., prevention or non-prevention).

C = number of categories

F = number of features

K_c = number of training examples for category c

V_{cef}= value of feature f for example e for category c

GT = global total feature sum

$$= \sum_{C=1}^{C} \sum_{e=1}^{Kc} \sum_{f=1}^{F} V_{cef}$$

CT_c = category c total feature sum

$$= \sum_{e=1}^{Kc} \sum_{f=1}^{F} V_{cef}$$

CF_{cf} = category feature sum for category c, feature f

$= \sum_{e=1}^{Kc} V_{cef}$

GF_f = global feature sum for feature f

$= \sum_{C=1}^{C} \sum_{e=1}^{Kc} V_{cef}$

CD_{cf} = category feature dominance for category c, feature f

$= CF_{cf} / CT_c$

OD_{cf} = outside category feature dominance for category c, feature f

$= \max \{ \min \{ (GF_f - CF_{cff}) / (GT - CT_c) , 10^8 \} , 10^{-8} \}$

CS_{ce} = category score for category c, testing example e

$= \sum_{f=1}^{F} V_{cef} (CD_{cf} / OD_{cf})$