Andrew Dragunas AndrewDragunas2021@u.northwestern.edu
Paul Hammonds II PaulHammond2016@u.northwestern.edu
Pravin Kumarappan Pravin.acer@gmail.com
Marsalis Smith Marsalis@u.northwestern.edu
Northwestern University
EECS 349: Introduction to Machine Learning

Predicting Journal Retractions/Corrections

Motivation

The body of scientific literature has greatly increased over the past decades. These research publications are the major avenue for advancing the state of art for a field of study. Through research, we are better able to understand the world around us. However, scientific misconduct, whether it be distortion or fabrication of data, has also increased with time. Major retractions or corrections to published work can have a negative impact on the opinion and confidence in the role of scientists and researchers by the general public. These retracted and corrected articles are a serious concern to the scientific community.

Recently, there have been many examples of major retractions from several scientific publications. The publisher BioMed Central has so far issued over 43 retractions in 2015 amidst an increasing scandal of fabricated peer-reviews. But these retractions are not limited to minor journals or fields of study. In 2010, Harvard economists Carmen Reinhardt and Kenneth Rogoff published a highly influential paper on the effect of debt-to-GDP ratios to a country's economic growth. At the time, this paper was considered the "most influential article cited in public and policy debates about the importance of debt stabilization," and likely lead many countries to impose steep austerity measures soon after the global recession. However, in 2013, it was found that an excel error completely undermined their central hypothesis. To predict whether an article will be retracted or corrected would be of great benefit to the scientific community at large.

Therefore, it is our opinion that to predict whether an article will be retracted or corrected would be of great benefit to the scientific community at large.

Solution

Our first approach to solving our problem started with establishing a comprehensive data set that we found relevant to predicting retracted/corrected publications. We created a MATLAB® function based of one of MATLAB's internal functions, *getpubmed()*. GetPubMed is a function that allows users to access the NCBI database and retrieve information based on the user's search query. We edited the function which allowed us to compile thousands of article data such as Abstract, Authors, Journal Name, Citations, Abstract, *etc*. From the original information

given by PubMed, we extracted other information from several related databases to identify other attributes such as Eigenvalue, Article Importance, University, and Country of Origin, among others. The final features are in *Table 1*.

Table 1: Features and Type

Features	**Type**
First Author	*Discrete*
University	*Discrete*
Country	*Discrete*
Journal	*Discrete*
Impact Factor	*Continuous*
Eigenvalue	*Continuous*
Article Influence	*Continuous*
Total Citations	*Continuous*
Bag of Words (Abstract)	*Continuous*
Publication Type	*Binary*

Data Collection

First, we started collecting our data based on date range was from Jan 1st, 2000 to Dec 31st 2014 without keyword searches. We were limited to 200 papers per day for a total of one million papers due to the PubMed website structure. This method didn't work out very well because we were able to pull out only the journals which were published on daily basis (i.e our pubmed search by dates excluded all the monthly, weekly and yearly journals). Each paper published in PubMed is assigned a unique ID called PubMed ID, so we then collected our data based on these PubMed IDs. We collected all the articles give a PubMed ID between 1,500,000 to 2,500,200. Since our search was based on PubMed ID, it included all journals (monthly, yearly, weekly and daily). When we calculated the number of retractions/corrections in the entire database (i.e. in the entire 1,000,200) we were able to find only 63 retractions/corrections. Since these IDs were not generated based on publication date, we did not know why the number of retractions/corrections were low.

We were able to pull out the entire retraction and correction history in PubMed, and using that we decided to create a random artificial database. We used the highest and lowest PubMed ID to randomly generate a non-repeating list between those IDs. We established our list of

retractions/corrections by selecting a predetermined number of articles from our retractions/corrections database, and dispersed these attributed randomly throughout our data set. This turned out to be our final method for data collection.

Features

The features First Author name and Journal name were obtained from the PubMed database. We created several lookup tables to add additional attributes. We used the Journal Database to cross-reference and add Journal, Impact Factor, Eigenvalue, Total Citations and Article Influence. The University database was used to cross reference the author affiliations and add a University attribute. In a similar manner we obtained the Country attribute. Finally, we implemented bag of words on the abstract field, adding ~10,000 additional attributes. To generate the bag of words, we removed all the stop words from the abstracts and created a unique vocabulary list based of all the remaining words in the abstract. Hence the bag of words consisted of frequency of each word in the unique vocabulary list for each paper.

Testing

Table 2: Test Sets Performance. We found Decision Trees (J48) to be overly aggressive and classify everything as a Non-Retraction, K-Nearest Neighbor (K-NN) to behave well only on small feature sets, and Naive Bayes to behave okay on both small and large training sets alike.

Algorithm	Total Instances	Number of Non-retractions	Number of Retractions	Number of Features	Cross-Validation	Total Accuracy	Mean Error	Correctly Identified Non-Retractions	Correctly Identified Retraction
J48	498	449	49	9	10	90.2%	0.18	100%	0%
K-NN	498	449	49	9	10	86.4%	0.14	92.6%	28.6%
Naive Bayes	498	449	49	9	10	84.5%	0.17	92.4%	12.2%
J48	743	670	73	9	10	90.2%	0.18	100%	0%
K-NN	743	670	73	9	10	85.9%	0.15	93.4%	16.4%
Naive Bayes	743	601	73	9	10	86%	0.19	84.8%	12.3%
J48	998	899	99	9	10	90.2%	0.18	100%	0%
K-NN	998	899	99	9	10	85.4%	0.15	82.5%	26.3%
Naive Bayes	998	899	99	9	10	84.6%	0.17	82.6%	16.2%

The performance of our algorithms on training sets of 100%, 75%, and 50% of 1000 random examples can be seen above in Table 2. Since our goal is to predict whether an article is correctly classified as a retraction or correction, we will use the last column of the above table as our measure of success.

We found that Decision Trees had a high total accuracy (90.2% for all three sizes of training sets), but also extremely aggressive in labeling all examples as false, and 0% as true.

K-Nearest Neighbor and Naive Bayes have similar total accuracy; however, K-Nearest Neighbor is twice as accurate at correctly identifying retractions than Naive Bayes (26.3% vs. 16.2% for 1000 examples).

In addition, we found that the most important features for the task were Author and Journal.

Table 3: Test Set Performance with and without Bag of Words.

Algorithm	Total Instances	Number of Non-Retractions	Number of Retractions	Number of Features	Cross-Validation	Total Accuracy	Mean Error	Correctly Identified Non-Retractions	Correctly Identified Retraction
J48	700	602	98	9	10	86.0%	0.24	100%	0%
K-NN	700	602	98	9	10	77.9%	0.23	88.7%	11.2%
Naive Bayes	700	602	98	9	10	81.4%	0.24	92.2%	15.3%
J48	700	602	98	15924	10	86.0%	0.24	99.8%	0%
K-NN	700	602	98	15924	10	28.0%	0.72	19.2%	81.6%
Naive Bayes	700	602	98	15924	10	69.5%	0.30	77.6%	19.4%

For Decision Trees, including Bag of Words had no change on the classification; for K-NN, the number of correctly identified retractions increased, but there was a subsequent increase in the number of false positives which resulted in a lower total accuracy (likely rooted in the curse-of-dimensionality); for Naive Bayes, including Bag of Words increased the true-positive percentage, but decreased the true-negative percentage.

Result

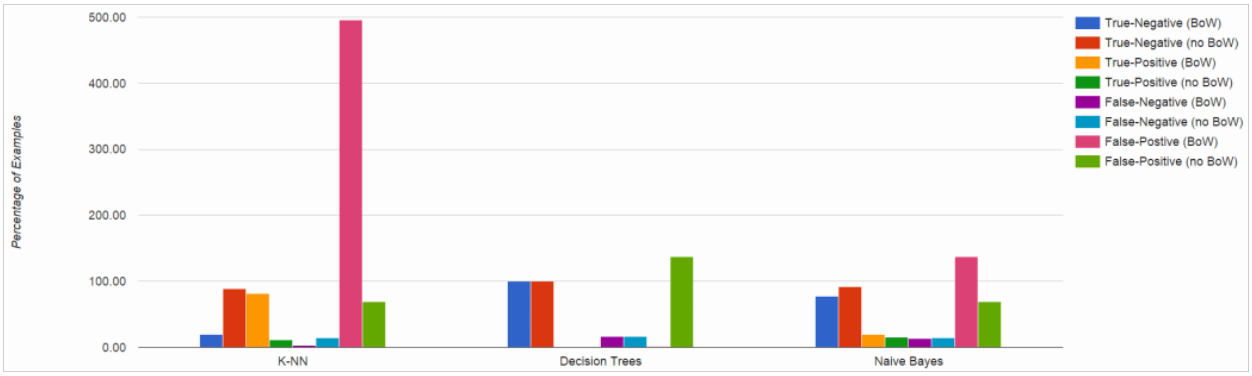

Figure 1: Cross-validated results on 700 example test set, with and without Bag of Words (BoW).

After looking at the retraction and non retraction accuracy values for the three different algorithms, we came to a conclusion that Bag of Words helps in improving the retraction predictions but will result in a much larger number of false-positives. We think this is mainly due to the size of the vocabulary list created by Bag of Words. Even though these attributes may be meaningful, the large dimensions of the feature set resulted in a lot of noise. As an example,

K-Nearest Neighbors with Bag of Words resulted in a 500% false-positive rate. If for example, this tool was used to make decisions on which articles would be published, a disproportion of articles would be flagged as a likely retraction or correction. Decision Trees create a similar problem, with articles that should be rejected being considered for publication.

The use of Naive Bayes provides a balance between overall accuracy and selection of retracted/corrected papers. Both with and without Bag of Words, the true-positive accuracy was 19% and 15%. This represent a large number of publications incorrectly being classified as non-retraction/correction types, but the model was not impacted by increased noise from Bag of Words. In conclusion, the Naive Bayes model was arguably the best algorithm, but still gives a low accuracy rate for detecting retraction / corrections. This is a reasonable outcome when taken into account that the current ratio of retracted/corrected articles to total published articles is less than 0.0001%.

Future Work

Due to the combined number of abstracts having tens of thousands of unique words, limiting the number of unique attributes with bag of words would increase efficiency of computing power. As of now, we were not able to use some of the Machine Learning algorithms since it was computationally expensive and even few other algorithms was not at all able to build a training model. So if by limiting the bag of words not only we can reduce the noise but also build model and try predicting with few additional machine learning algorithms.

We limited our vocabulary list to the abstract of the article. Having a vocabulary list for all combined articles would offer greater insight and possibly increase the prediction power of our model.The other possibility would be to add the number of citations and h-index for each specific authors listed on the article.

In addition to the above, we generated a retraction/correction database of 5178 PubMed IDs. As noted, the Journal and Author Name were the most important features for the task. By creating a master database of every retraction/correction ever published (and adding a proportional number of non-retraction/corrections), we could better catch particular Authors or Journals who are particularly susceptible to scientific misconduct. This would more closely represent real-life situations.