Research Paper

Can Automatic Classification Help to Increase Accuracy in Data Collection?

Frederique Lang[1], Diego Chavarro[1] & Yuxian Liu[2†]

[1]Science Policy Research Unit (SPRU), School of Business, Management and Economics, University of Sussex, Falmer, Brighton, BN1 9SL, United Kingdom
[2]Tongji University Library, Tongji University, Shanghai 200092, China

Citation: Frederique Lang, Diego Chavarro & Yuxian Liu (2016). Can Automatic Classification Help to Increase Accuracy in Data Collection?

Received: Jul. 17, 2016
Accepted: Jul. 29, 2016

Abstract

Purpose: The authors aim at testing the performance of a set of machine learning algorithms that could improve the process of data cleaning when building datasets.

Design/methodology/approach: The paper is centered on cleaning datasets gathered from publishers and online resources by the use of specific keywords. In this case, we analyzed data from the Web of Science. The accuracy of various forms of automatic classification was tested here in comparison with manual coding in order to determine their usefulness for data collection and cleaning. We assessed the performance of seven supervised classification algorithms (Support Vector Machine (SVM), Scaled Linear Discriminant Analysis, Lasso and elastic-net regularized generalized linear models, Maximum Entropy, Regression Tree, Boosting, and Random Forest) and analyzed two properties: accuracy and recall. We assessed not only each algorithm individually, but also their combinations through a voting scheme. We also tested the performance of these algorithms with different sizes of training data. When assessing the performance of different combinations, we used an indicator of coverage to account for the agreement and disagreement on classification between algorithms.

Findings: We found that the performance of the algorithms used vary with the size of the sample for training. However, for the classification exercise in this paper the best performing algorithms were SVM and Boosting. The combination of these two algorithms achieved a high agreement on coverage and was highly accurate. This combination performs well with a small training dataset (10%), which may reduce the manual work needed for classification tasks.

Research limitations: The dataset gathered has significantly more records related to the topic of interest compared to unrelated topics. This may affect the performance of some algorithms, especially in their identification of unrelated papers.

Practical implications: Although the classification achieved by this means is not completely accurate, the amount of manual coding needed can be greatly reduced by using classification algorithms. This can be of great help when the dataset is big. With the help of accuracy, recall,

JDIS
Journal of Data and Information Science
Vol. 1 No. 3, 2016
pp 42–58
DOI: 10.20309/jdis.201619

† Corresponding author: Yuxian Liu (E-mail: yxliu@tongji.edu.cn).

and coverage measures, it is possible to have an estimation of the error involved in this classification, which could open the possibility of incorporating the use of these algorithms in software specifically designed for data cleaning and classification.

Originality/value: We analyzed the performance of seven algorithms and whether combinations of these algorithms improve accuracy in data collection. Use of these algorithms could reduce time needed for manual data cleaning.

Keywords Disambiguation; Machine learning; Data cleaning; Classification; Accuracy; Recall; Coverage

1 Introduction

Retrieving data accurately is one of the first and most important steps in data mining (Porter & Cunningham, 2004). Unfortunately, this may turn out to be one of the most time-consuming and demanding activities in an investigation. When constructing a dataset, one has to ensure that the data gathered are actually related to the subject of interest. Because searches are dependent on keywords, and keywords have different meanings, it is likely that the dataset retrieved has some data unrelated to the subject. Moreover, during the development of a concept, terms may be under constant evolution, which makes the search even harder. This has consequences for the conclusions reached using quantitative techniques, which can be misleading if mistakes are not detected in time.

Keyword searches are widely used for the identification of emerging technologies (Daim et al., 2006). For instance, keywords are used to build research and technological portfolios to adjust the management practices of policy instruments. Through these policy instruments, many countries seek to foster the commercial exploitation of science-based research results (Salo, Mild, & Pentikäinen, 2006; Wallace & Rafols, 2014) and new technologies found through the examination of patents and publications (Kim et al., 2014). However, uses of keywords can bring about unrelated results since the researcher is not always able to determine the specific context of use during a search. There is therefore a need to check the records obtained through the use of keywords before the analysis phase.

Let us take an example to illustrate our point. When searching for the term "crane" in the search engine "Google," the records are related to four different objects. The first one refers to a big machine with a long arm that is used by builders to lift and move heavy things. The second is a type of tall bird that has a long neck and long legs and lives near water. The third is a company, and finally the fourth is a fluid system. This implies that if somebody aims to collect data about a "crane," whichever definition the researcher is interested in, they will end up collecting data about four different objects.

Research Paper

In this paper, we emphasize the care that must be taken when collecting bibliometric data, testing the use of a supervised method to help the researcher deal with ambiguity in the data. In our specific case, we have built a dataset which is related to a specific biomarker called "Her 2." Although we gathered our data from the Web of Science, a database specialized in scientific literature, we still gathered unrelated results caused by the ambiguity between "Her 2" as a biomarker and "her" as a pronoun. Some results containing phrases such as "her 2 children" and "her 2 yellow jackets" etc. were retrieved by our keyword search.

"Her 2" is one of the names for the human epidermal growth factor 2. It is an oncogene that controls its own growth in the breast tissue – a biomarker of great importance for cancer diagnosis and therapy. It was found by several research groups, each group naming this gene in a different way. Shih et al. (1981) identified this kind of gene as a result of transfection studies with DNA from chemically induced rat neuroglioblastomas. Schechter et al. (1984) called this gene neu; Coussens et al. (1985) named the gene they isolated Her-2; Semba et al. (1985) called it C-erbB-2. Later, C-erbB-2, Neu, and Her-2 were found to be the same biomarker (Coussens et al., 1985; Fukushige et al., 1986; Schechter et al., 1984). Yet another way to call this gene is found in Slamon et al. (1987) who called it Her-2/Neu, which is nowadays the prevalent term to refer to it. However, sometimes scientists just use the spellings "her 2," "Her 2," "Her-2," or "HER 2." Although some of the keywords are unique to this biomarker, "her 2" could refer to words with different meanings.

When searching for "Her 2" in the Web of Science, the oldest article found was published in 1970. At first sight we could assume that the biomarker was discovered in 1970. However, the human epidermal growth factor 2 was identified in 1981 (Shih et al., 1981). So the judgment/conclusion from the noisy data can be misleading. Uncritical analysis based on rudimentary article identification strategies may lead to misinterpretation of the development of research areas, thus providing incorrect data for decision-making (Lundberg, 2006). However, excluding this particular keyword in the identification of papers on this topic leads us to lose over 2,000 records.

Although the number of records can be seen as insignificant, depending on the purposes of the research, if the aim is to be comprehensive the lack of 2,000 papers is an important barrier for the construction of the dataset. In this sample, some mistakes could be cleaned manually, but in big datasets, the amount of manual coding is impractical. This shows the relevance of data accuracy and disambiguation for bibliometric analyses, as confirmed by the 2015 International Society of Scientometrics and Informetrics (ISSI) Conference that has listed them as a main topic to call for solutions from the informetrics community (ISSI, 2015).

Article disambiguation strategies are normally focused on cleaning data on authors (Chin et al., 2014; Kim & Diesner, 2015; Liu et al., 2014), institutions (Huang et al., 2014; King, Jha, & Radev, 2014), and acknowledgements (Rotolo, Hopkins, & Grassano, 2014). They do not usually use the subject or the content of the papers. Instead, they identify the articles by combining information on authors, institutions, and journals. In this paper, we put an emphasis on the use of a word for a specific topic in titles and abstracts for disambiguation, and test a supervised procedure that could help with the correct identification of relevant records.

2 Data and Methods

2.1 Dataset

Firstly we used TS="her 2" to retrieve the data from the Web of Science, recalling 8,542 items. Among these items, we excluded those definitely related to the biomarker the epidermal growth factor receptor 2. To do so, we constructed a specific search string to make sure all items recalled from this search string are related to the biomarker "her 2." After a comprehensive literature research, we constructed a search **string 1**:

String 1: TS=("CerbB2*" OR "CerbB-2*" OR "Cer-bB2*" OR "C-erbB2*" OR "Cer-bB-2*" OR "C-erbB-2*" OR "C-er-bB2*" OR "C-er-bB-2*" OR "erbB2*" OR "erbB-2*" OR "er-bB2*" OR "er-bB-2*" OR "HER2" OR "Epidermal growth factor receptor 2" OR "EGFR2" OR "CD340" OR "HER-2/neu" OR "neu/her-2" OR "Anti-her-2*" OR "MVA-BN-HER-2*" OR "CHP-HER-2 vaccin*" OR "HER-2 affitoxin*" OR "Her-2 protein*" OR "her-2 peptid*" OR "her-2 affibod*" OR "her-2 gene*").

And then we performed a search process as shown in Table 1.

Table 1. Search process.

	Search string	Number of items
# 1	TS="her 2"	8,542
# 2	String 1	26,972
# 3	#2 AND #1	6,396
# 4	#1 NOT #3	2,146

We then have 2,146 records that we cannot judge whether they are related to the biomarker "her 2." In order to find an efficient method to clean data, we proceeded to inspect the records manually and found among these 2,146 records, only 98 records are not related to the "her 2" biomarker.

In order to make our dataset more balanced, we searched for Her 5, Her 6… Her 60 and replaced all these numbers with 2. We checked each manually to see if these

records were related to oncogene "her 2." Although we did not use "her 2" as a search term, we still found some records related to oncogene "her 2." With all the data combined, we have a dataset with 2,589 records of which 716 records are unrelated to biomarker "her 2."

2.2 Methods

Chavarro and Liu (2014) used recursive Lesk and keywords distance metrics to disambiguate the meaning of words. This method depends on the definition of the dictionaries of a word. In that study, the authors used one classic article related to "her 2" to define a dictionary and identified all records that are similar to this dictionary (see also Li, Sun, & Datta (2013) for a related approach). The algorithm is based on the similarity of topics. However, one research topic has different aspects. For example, the research on the topic of "her 2" at least concerns the biological property of the biomarker "her 2," the methods to test status of "her 2," and its therapeutic and side-effects. Since the research topic is developing, it is hard to pick up all aspects of a research topic.

In this paper, we opted instead for a classification method that does not involve the creation of a dictionary. This is because in our case we would only be able to build a definition or related words to "her 2" the biomarker, but we would not be able to build a dictionary for the other uses of "her 2" as it can be used in various contexts. For this reason, a different technique was used.

Some machine learning algorithms do not require the use of a dictionary. These algorithms are therefore suited to address the classification problem posed by the dataset on "her 2." Two main types of methods are known in machine learning: supervised and unsupervised methods. Supervised learning predicts an outcome based on input characteristics/data. In supervised learning, the algorithms are therefore designed in two steps. The first is concerned with the training of the algorithm through providing already classified data to the algorithm. The algorithm then learns from the features of the data and the classification associated to it. It attempts to classify data by looking at their characteristics in order to give a predictive classification to each data point. The unsupervised training methods aim at clustering data and finding patterns from the data characteristics, which can be classified by the researcher after running the algorithm.

Supervised methods seem most adapted to our problem because even if a paper related to "her 2" the biomarker may have overlapping characteristics, the unrelated papers may not have similar characteristics and therefore will probably not be identified as a unique cluster. Therefore in this paper, we tested the performance of various algorithms using supervised methods and looking at how accurate they are according to the amount of training and the type of algorithm they use for classification.

2.2.1 Strategy for the Assessment of the Algorithms

We used two approaches in order to assess the performance of the different algorithms. The first approach focuses on the individual performance of the algorithms and the second assesses their combined performance. Finally, we chose the two best performing algorithms to see how they compared to the rest. While the first approach indicates how well each of the algorithms classifies the records, the second relies on the degree of agreement between algorithms in order to achieve accurate results. If each algorithm is considered a trained classifier with its own strengths and weaknesses, when different algorithms agree on a classification, a better performance can be expected than when there is disagreement. This approach has been used by Jurka et al. (2012). The results section provides the outcome of individual and combined approaches.

2.2.2 Steps Used in Machine Learning

As explained earlier, a dataset was built by classifying each data point into two classes. The first class consists of papers that relate to the biomarker "her 2" and the second class consists of the unrelated papers. This dataset was used in order to train the algorithm. However, not all the data were used for this purpose. The dataset was divided into two sets, a training set and a test set. The training set was used, as explained earlier, for the algorithm to learn the pattern of data, which was used for predicting the classification of further data points. Before performing the training of the algorithm, the input data were randomized so that the outcome is not defined by any particular structure of the dataset. The test set was used in order to measure the accuracy of the classification given by the algorithm. Thus with the model built from the training, the algorithm was used to classify the data points from the test set. The predictions were then compared to the manual classification in terms of not only the accuracy of each individual model that has been built but also the accuracy of the models combined. This helped to know the confidence under which we classify the data.

In order to train the data, we decided to use the title and the abstracts (when available). As some abstracts were quite long, this created some problems in running many of the algorithms (as it was too much information to process). For this reason we decided to use all titles and only sentences in the abstracts with an occurrence of "her 2" (under its various forms) as these sentences are the most likely to provide relevant words for performing a classification.

Regarding accuracy, we also tested how much training data the algorithms require in order to become accurate. We want to see how each algorithm performs with different training sizes, and how the increase in the training size improves the

overall algorithm accuracy. To do so, we divided our 2,589 records variably into training/test sets of 10%/90%, 20%/80%, 50%/50%, and 80%/20% to compare the algorithms and their overall accuracy according to the size of the training set.

2.2.3 The Algorithms Used

As mentioned earlier, in this paper we used different algorithms in order to test their accuracy when used for a dual classification on our "her 2" example. The algorithms are found in a package specifically designed to process text in R called RTextTools (Jurka et al., 2012). We used only seven out of the nine algorithms available in the package as two of them are particularly demanding on memory (RAM) and therefore result in errors when processing the model. We used the seven algorithms: Maximum Entropy (MaxEnt) (Jurka, 2012), Lasso and elastic-net regularized generalized linear models (GLMNet) (Friedman, Hastie, & Tibshirani, 2010), Scaled Linear Discriminant Analysis (SLDA) (Peters et al., 2012), Support Vector Machine (SVM) (Meyer et al., 2012), Regression Tree (Tree) (Ripley, 2012), Boosting (Tuszynski, 2012), and Random Forest (Forest) (Liaw & Wiener, 2002).

These algorithms are diverse in terms of not only variety of method but also sampling methods. For instance, MaxEnt and GLMNet are both based on regression methods. The first one classifies data following a multinomial logistic regression model. The second one is based on regression models with Lasso and elastic-net penalties. These help to choose important predictors in the regression and discard the other ones, which reduce prediction errors in many cases when the model has high variability (Hastie, Tibshirani, & Friedman, 2009).

SLDA and SVM are based on linear models. SLDA aims at finding linear decision boundaries based on the closest centroid of a class. This classifier does not perform so well when the classes studied have a higher overlap. SVM is also based on a linear classifier, but is more efficient on overlapping groups. SVM maps the data into a higher dimensional space than it was originally mapped to, and finds a hyperplane that separates the two groups with a maximum distance.

Another algorithm used is Tree. In this method, the space of distribution of data points is iteratively divided and the subdivisions of space are attributed to a class. The last two classification methods, Forest and Boosting, are based on building different models with slightly different training sets (a subset of the training set given). The test set will be run on different models, which will vote to determine the classification. Thus Forest is based on the voting of different tree models. The Boosting method is also based on this type of voting scheme. However, in this model the weights are assigned to each model as a function of how much success they have in predicting correct results from a subset of the training set.

2.2.4 Measures Used

Individual assessment includes two measures that are usually applied to verify the performance of algorithms. The first one is *accuracy* and the second one *recall* and they can be calculated by Equations (1) and (2). Accuracy is the overall number of correct classifications as compared to the whole sample, and recall is the number of correct classifications in each category. These numbers are given in percentages where the higher the percentage, the more accurate and precise the algorithm is. These two measures are complementary in that one evaluates the whole classification, whereas the other evaluates in detail the categories in which the algorithm is better at predicting the results correctly. In order to cross-validate the results, we tested the algorithms with different training datasets: 10%, 20%, 50%, and 80%, in order to understand the impact of size of the training dataset on classification performance. Some of the algorithms could be prone to overfitting when the training dataset is large, and others could underperform when the training dataset is small.

$$\text{Accuracy} = C/n, \tag{1}$$

$$\text{Recall} = C/nmi, \tag{2}$$

where C equals correctly classified items, n the number of items in the dataset, and nmi the number of items in the dataset manually classified in category i.

There is another measure usually used to evaluate algorithm performance at the category level, called *precision*. This measure is the number of correctly classified items in a category divided by the total number of items classified by the algorithm as belonging to this category (Equation (3)):

$$\text{Precision} = C/nai, \tag{3}$$

where C refers to correctly classified items, and nai the number of items classified by the algorithm as belonging to category i.

We have only used recall because we are interested in comparing the algorithms to the golden standard of manual classification. Accuracy and recall are two levels of performance assessment for each algorithm. Accuracy is overall performance, while recall allows us to see if the algorithms are skewed towards one of the two categories.

When combining the algorithms we used two measures of interest. The first is *coverage*, which indicates how many records are agreed by the algorithms on their classification and the second is accuracy, as defined above. We started with the minimum number of agreements, and continued until the maximum number was reached. Usually, the greater the number of algorithms agreeing on a classification, the more likely the record is correctly classified. We used this for both the whole

set of algorithms and the two best performing algorithms. In this paper, we only show the exercise based on a 10% training set, as an example of the case in which the human coder would have to make the least effort. The results can tell us about how algorithms perform on the automatic classification of the biggest test dataset based on the smallest training dataset.

3 Results

3.1 Results of the Individual Benchmark

In this section we look at the individual performance of each algorithm as well as the performance variation according to the size of the training set. In order to assess the performance of each algorithm we also included a *default model*. The default model classifies all the items as being related to "her 2," which gives an accuracy of around 70% in our case, since our training data are skewed towards correctly classified data. The comparison between our trained models and the default model was used as an indicator of the performance of individual algorithms.

3.1.1 Accuracy

Table 2 shows the accuracy of each algorithm trained on training sets of different sizes. We can observe that each algorithm performs differently in terms of accuracy according to the size of the training sample. For instance, Boosting, SVM, and Tree algorithms need only a small amount of training records in order to perform well. They all achieve above 94% accuracy with the smallest training set of 10%. Other algorithms such as Forest, GLMNet, MaxEnt, and especially SLDA do not perform well with small training sets. In the case of Forest, when moving to 20% of the full training set, the algorithm improves its performance by more than 15 percentage points. GLMNet increases significantly its performance when moving from 20% to 50% of the full training set. MaxEnt improves its accuracy somewhat gradually when the size of the training set increases, while SLDA needs a larger training set in order to increase its performance. From 10% to 50% of the training set, this

Table 2. Accuracy – Individual benchmark.

Training set	10%	20%	50%	80%
Default model	73.56%	73.60%	73.11%	69.94%
Forest	80.69%	96.57%	96.52%	96.72%
GLMNet	82.88%	82.58%	94.98%	93.26%
Boosting	95.45%	95.22%	96.68%	95.18%
MaxEnt	86.95%	90.44%	92.89%	93.83%
SLDA	73.56%	74.90%	75.66%	88.25%
SVM	94.29%	95.80%	96.75%	98.07%
Tree	94.38%	95.56%	95.13%	93.83%

algorithm performs barely better than the default model. Compared to the other models SLDA is always at least five percentage points below the second worst performing algorithm. All the other algorithms perform significantly better than the default model. Overall, GLMNet and MaxEnt underperform compared to other models. SVM and Boosting are both high-performance algorithms, especially when we use small training sets. Forest is also a high-performance algorithm, but needs a slightly larger training set than the above-mentioned two algorithms in order to start becoming accurate. Finally, one can observe that some algorithms are prone to overfitting when moving to a larger training set since their accuracy decreases (we can mainly observe this between 50% and 80% of the full training set). This is the case for the Tree, Boosting, and GLMNet algorithms.

3.1.2 Recall

In order to improve our understanding of the performance of the algorithms, we look in this section at how they perform when assessing each individual category (related to "her 2" -Yes-, or unrelated -No-). In order to do so, we used the recall measure for each category, which is displayed in Table 3. One of the first striking results is the underperformance of the algorithms to correctly classify the ones from the -No- category compared to the -Yes- category. This could be due to two reasons. The first could be that the ratio of unrelated items in the training set is highly unbalanced compared to the related items, and therefore we give fewer -No- cases to the algorithms, which creates more difficulty in recognizing them. The second reason is the design of a category. The unrelated category is not focused on a specific topic and therefore words found in the text may be unrelated to other items in this category. When looking at the -Yes- category, one can observe that all algorithms perform extremely well in identifying most of the related documents to "her 2." Most of the algorithms correctly identify over 95% of the documents related to "her 2," with any training size. SLDA is the only one that performs under this threshold for 50% of the training set.

Table 3. Recall of individual algorithms.

Training set / Algorithm	10%		20%		50%		80%	
	No	Yes	No	Yes	No	Yes	No	Yes
Forest	26.95%	100.00%	89.95%	98.95%	90.23%	98.84%	89.74%	99.72%
GLMNet	35.55%	99.88%	34.37%	99.87%	88.22%	97.46%	88.46%	95.32%
Boosting	93.34%	96.21%	95.43%	95.15%	96.26%	96.83%	90.38%	97.25%
MaxEnt	51.14%	99.82%	64.35%	99.80%	73.85%	99.89%	79.49%	100.00%
SLDA	0	100.00%	11.88%	97.51%	44.25%	87.21%	60.90%	100.00%
SVM	84.74%	97.72%	93.78%	96.52%	95.69%	97.15%	97.44%	98.35%
Tree	90.10%	95.92%	91.04%	97.18%	88.79%	97.46%	91.03%	95.04%

Thus we want to focus more on not only how the algorithms perform on the -No- side, but also how well they balance the -Yes- and -No- answers, since we want algorithms to have a good performance on both sides. For the algorithms that we identified as being inaccurate in the previous section with small training sets, we can see here that they perform very poorly for identifying correctly the -No- data (SLDA, Forest, and GLMNet). While Forest and GLMNet improve their -No- classification over the increase of the training size (at 20% and 50%, respectively), they still have the highest proportion of training set imbalance compared to other algorithms. For MaxEnt, while the algorithm performs better than others with smaller training sets, it does not seem to correct its imbalance over training size. MaxEnt and SLDA are the two worst performing algorithms in the two larger training samples. SLDA exhibits the worst performance regardless of the training set. Both Boosting and Tree algorithms have good balance and high accuracy for both categories of smaller training sets, and they seem to become more unbalanced with larger training sets (at 80% and 50%, respectively). Finally, SVM seems to be balanced, although it is slightly better at estimating the -Yes- category. However, it increases its performance on the -No- category when the training set is bigger, to the point that it becomes the best algorithm with very high scores for both -Yes- and -No- at 80% of training.

After looking at both the overall accuracy and recall of the algorithms with training sets of different sizes, we can draw general conclusions about the performance of each algorithm. First of all, the SLDA algorithm is clearly underperforming. This algorithm does not improve significantly the default model. GLMNet and MaxEnt are also underperforming compared to the other algorithms over all training sizes. Concerning the Tree algorithm, it performs well as compared to others with small training sets and has a good balance between -Yes- and -No-, but does not improve as much as others with the training size increasing. The Forest algorithm does not perform very well with small training sets but improves its accuracy over training size. However, it remains highly unbalanced. Finally, Boosting and SVM perform very well from the start. Boosting exhibits a better balance with smaller training sets, but SVM has a better performance overall when the training set is above 20%. SVM also becomes more balanced with larger training sets. In the next sections we test whether it is useful to use a combination of algorithms to predict outcomes.

3.2 Results Combined Benchmark

Table 4 shows the agreement between algorithms:

Table 4. Consensus on the classification of records.

Consensus (Number of algorithms)	Coverage	Coverage No	Coverage Yes	Accuracy	Recall for the Yes category	Recall for the No category
≥ 4	2,330	616	1,714	87.51%	99.88%	53.08%
≥ 5	2,045	336	1,709	93.99%	99.94%	63.69%
≥ 6	1,825	162	1,663	98.19%	100.00%	79.63%
≥ 7	1,606	11	1,595	99.32%	100.00%	N/A

As the number of algorithms that reaches a consensus increases, the number of accurate classifications increases, too. However, the number of records classified decreases (column coverage). In the case studied, it means that a researcher could correctly classify around 69% of the dataset with a coding effort of 10%. There would still be, however, 31% of the dataset that would have to be checked by other means.

This observation, however, has to be taken with caution. While accuracy increases with the number of algorithms agreeing on the classification of records, it is important to note that the biases introduced by some of the algorithms can have important consequences for the recall of the ensemble. If we look at the Recall -Yes- column, we can see that regardless of the number of algorithms agreeing on the classification, it is very close to 100%. In the case of Recall -No-, we can see a very important increase in this measure as more algorithms agree. Interestingly, when all seven algorithms agree the only category that can be predicted is -Yes-. This happens because SLDA is completely biased towards the -Yes- category, creating the impossibility of having any indicator of agreement on the -No- category. Excluding SLDA yields a recall for the -No- category of 79.63%. This is better than the default model, but far from acceptable from a researcher's point of view.

The results show that despite being accurate, the recall achieved for the -No- category would make this approach unsuitable for undertaking a real world classification task. The fact that the -No- category is more diverse than the -Yes- category could make it harder for the algorithms to identify it, in the same way that happened with the individual algorithm. It seems, however, that the inaccuracies caused by each algorithm are reduced when using the consensus approach.

The limitations were particularly noticeable when there is a complete bias of one of the algorithms. This indicates that for this classification task it might be more important to choose an approach that balances the number of algorithms and quality. In the next step we assess the combination of only two of the best individually performing algorithms to see if their accuracy and recall are better than the whole ensemble.

3.3 Results of the Two Best Performing Algorithms

After looking at the algorithm consensus and the advantage and shortcoming of this approach, we look now at the results we could get if one combines the two best performing algorithms identified, namely the SVM and Boosting algorithms. One of the shortcomings of this approach compared to the one above lies in the fact that there are only two algorithms and so one cannot classify the items when there is disagreement between the approaches, and therefore we cannot achieve full coverage. Table 5 shows the results of the combination of the two best algorithms.

Table 5. Performance of the two best algorithms combined.

Algorithm	Coverage	Coverage Yes	Coverage No	Accuracy	Recall for the Yes category	Recall for the No category
SVM and Boosting	2,131	1,620	511	99.06%	99.69%	97.06%

The combination between SVM and Boosting seems to achieve excellent results. When used together, they achieve 91% coverage of the sample tested (2,131/2,330), which is better than the coverage of agreement of five or more algorithms with the above approach. Overall the accuracy of this approach is better than the agreement over six or more algorithms, but looking at the balance between recall of categories, this approach is far superior to the one above. The recall for -No- is much better than the agreement and outperforms most individual algorithms with even larger training sets, the only exception being SVM with 80% of training. One could argue that the agreement of seven algorithms outperforms this approach in terms of recall, but as we have seen with individual algorithm before, this comes at the cost of coverage due to the fact that many algorithms at 10% of training are highly unbalanced towards giving positive answers. Thus one can conclude that at minimum training the approach combining the two best performing algorithms is the most efficient on coverage, but with the inconvenience to manually code 9% of the items that the two algorithms disagree on.

4 Discussion and Conclusion

We have examined the performance of different algorithms on a supervised classification task based on a search for scientific papers. Two techniques were used: the first one was based on the individual performance of each algorithm and the second on their consensus. The two techniques proved better than the default model in most cases, as shown by the accuracy rates. However, a variety of issues arose in this classification task.

Firstly, the amount of training needed to have an acceptable performance varies with individual algorithm. Some algorithms perform well with small amounts of

data, while others need a large training set (and therefore more "human help") to perform better. The discussion on training size has also shown that the statement "the more the better" is not always valid. At some point the algorithms are prone to overfitting when given too much data.

Secondly, not all the algorithms perform well. For instance, SLDA performed quite badly. Many algorithms also did not perform well on the unrelated category (the -No-) compared to their good performance with the -Yes-. This could be explained by both the problem linked to the diversity in the -No- category, and the smaller number of items in this category in our training set. Therefore, when using these techniques for other and larger applications, one may want to look into each category in order to understand its diversity through, for example, cluster analysis.

Finally, the patterns produced by the combination of the seven algorithms allow us to understand the influence of classification mistakes on predictions due to the bias or underperformance of individual algorithm. In this case, even if we included more algorithms, the possibility of predicting the -No- category would be stagnated because of SLDA. When this happens, the power of the ensemble would be determined only by one of the algorithms. The fact that excluding SLDA does not produce completely satisfactory results on the -No- category brings up the question about the relationship between number and quality. This applies not only to automatic classification, but also to human-based classification. In cases in which there are a number of human coders, a situation such as a biased coder could imply inaccuracies in the classification, having impacts on the recall of at least one of the categories. Also training seven algorithms for classification purposes can take a much larger amount of time or computer power than picking the best.

In order to improve this, a solution between individual and combined power should be used. In this case, by using the combined power of the two best performing algorithms, we achieved satisfactory results both on accuracy and recall for each category. Coverage, however, cannot be complete by using any of the combined approaches mentioned. Some sort of manual classification is still needed on the researcher's side. In spite of this, the finding of the satisfactory results achieved by the combination of SVM and Boosting is promising.

In conclusion, we found that a supervised approach to data cleaning is possible. However, this still requires the active involvement of the researcher in the process. Although the classification achieved by this means is not completely accurate, the amount of manual coding needed can be greatly reduced. This is of great help when the dataset is big. With the help of accuracy, recall, and coverage measures, it is possible to have an estimation of the error involved in this classification, which could open the possibility of incorporating the use of these algorithms in software specifically designed for data cleaning and classification.

Acknowledgements

The authors are grateful to Peter Bone for his help with the proofreading of this paper. We wish also to thank Jose Christian for helpful discussion and support. Thanks go also to the colleagues who gave us comments during the 4th Global TechMining Conference and in the Science Policy Research Unit (SPRU) Wednesday Seminar. Yuxian Liu's work was supported by National Natural Science Foundation of China (NSFC) (Grant No.: 71173154), The National Social Science Fund of China (NSSFC) (Grant No.: 08BZX076) and the Fundamental Research Funds for the Central Universities.

Author Contributions

F. Lang (f.lang@sussex.ac.uk) and D. Chavarro (dchavarro@gmail.com) provided the ideas to answer the research questions. F. Lang run the programs and wrote the methodology part. D. Chavarro analyzed the result of program and wrote this part. Y.X. Liu (yxliu@tongji.edu.cn, corresponding author) collected the data and proposed the research questions and a framework for data analysis. She wrote the introduction and data collection parts. The authors discussed every detail of this article, and elaborated the conclusion together. The work of every author is significant and they can be cited in any order.

References

Chavarro, D. & Liu, Y. (2014). How can a word be disambiguated in a set of documents: Using recursive Lesk to select relevant records. Presented in 2014 Annual Global Techmining Conference. Retrieved from http://www.gtmconference.org/abstracts/2014/session1METHODS3.pdf.

Chin, W.S., Zhuang, Y., Juan, Y.C., Wu, F., Tung, H.Y., Yu, T., Wang, J.P., Chang, C.X., Yang, C.P. & Chang, W.C. (2014). Effective string processing and matching for author disambiguation. The Journal of Machine Learning Research, 15, 3037–3064.

Coussens, L., Yang-Feng, T.L., Liao, Y, Chen, E., Gray, A., McGrath, J., …& Ullrich, A (1985) Tyrosine kinase receptor with extensive homology to EGF receptor shares chromosomal location with neu oncogene. Science, 230(4730), 1132–1139.

Daim, T.U., Rueda, G., Martin, H., & Gerdsri, P. (2006). Forecasting emerging technologies: Use of bibliometrics and patent analysis. Technological Forecasting and Social Change, 73, 981–1012.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33, 1–22.

Fukushige, S., Matsubara, K., Yoshida, M., Sasaki, M., Suzuki, T., Semba, K., Toyoshima, K. & Yamamoto, T. (1986). Localization of a novel v-erbB-related gene, c-erbB-2, on human chromosome 17 and its amplification in a gastric cancer cell line. Molecular and Cellular Biology, 6, 955–958.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning. Berlin: Springer. Available at: http://link.springer.com/content/pdf/10.1007/978-0-387-84858-7.pdf.

Huang, S., Yang, B., Yan, S. & Rousseau, R. (2014). Institution name disambiguation for research assessment. Scientometrics, 99, 823–838.

International Society of Scientometrics and Informetrics (ISSI) (2015). International Conference on Scientometrics & Informetris Call for Paper. Retrieved from http://issi2015.ulakbim.gov.tr/.

Jurka, T., Collingwood, L., Boydstun, A., Grossman, E., & Atteveldt, W.V. (2012). RTextTools: A supervised learning package for text classification. The R journal, 5, 6–12.

Kim, B., Gazzola, G., Lee, J.M., Kim, D., Kim, K., & Jeong, M.K. (2014). Inter-cluster connectivity analysis for technology opportunity discovery. Scientometrics, 98, 1811–1825.

Kim, J., & Diesner, J. (2015). Distortive effects of initial-based name disambiguation on measurements of large-scale coauthorship networks. Journal of the Association for Information Science and Technology, 67(6): 1446–1461.

King, B., Jha, R., & Radev, D.R. (2014). Heterogeneous networks and their applications: Scientometrics, name disambiguation, and topic modeling. Transactions of the Association for Computational Linguistics, 2, 1–14.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2, 18–22.

Li, C., Sun, A., & Datta, A. (2013). TSDW: Two-stage word sense disambiguation using Wikipedia. Journal of the American Society for Information Science and Technology, 64(6), 1203–1223.

Liu, W., Doğan, R.I., Kim, S., Comeau, D.C., Kim, W., Yeganova, L., & Wilbur, W.J. (2014). Author name disambiguation for PubMed. Journal of the Association for Information Science and Technology, 65(4), 765–781.

Lundberg, J., Fransson, A., Brommels, M., Skar, J., & Lundkvist, I. (2006). Is it better or just the same? Article identification strategies impact bibliometric assessments. Scientometrics, 66, 183–197.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2012). Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. Retrieved from http://cran-r.c3sl.ufpr.br/web/packages/e1071/e1071.pdf.

Peters, A., Hothorn, T., Ripley, B.D., Therneau, T., Atkinson, B., & Hothorn, M.T. (2012). Package 'ipred': Improved predictors. Retrieved from https://cran.r-project.org/web/packages/ipred/index.html.

Porter, A., & Cunningham, S. (2004). Tech mining: Exploiting new technologies for competitive advantage. Hoboken, New Jersey: John Wiley & Sons.

Ripley, B. (2012). Tree: Classification and regression trees. Retrieved from https://cran.r-project.org/web/packages/tree/index.html.

Rotolo, D., Hopkins, M., & Grassano, N. Do funding sources complement or substitute? The case of the UK cancer research. In the 19th International Conference on Science and Technology Indicators (the STI 2014), (pp 473). Leiden, Netherlands.

Salo, A., Mild, P., & Pentikäinen, T. (2006). Exploring causal relationships in an innovation program with robust portfolio modeling. Technological Forecasting and Social Change, 73, 1028–1044.

Schechter, A.L., Stern, D.F., Vaidyanathan, L., Decker, S.J., Drebin, J.A., Greene, M.I., & Weinberg, R.A. (1984). The Neu Oncogene – An Erb-b-related gene encoding A 185,000-Mr Tumor-antigen. Nature, 312(5994): 513–516.

Semba, K., Kamata, N., Toyoshima, K., & Yamamoto, T. (1985). A v-erbB-related protooncogene, c-erbB-2, is distinct from the c-erbB-1/epidermal growth factor-receptor gene and is amplified in a human salivary gland adenocarcinoma. Proceedings of the National Academy of Sciences, 82, 6497–6501.

Shih, C., Padhy, L.C., Murray, M., & Weinberg, R.A. (1981). Transforming genes of carcinomas and neuroblastomas introduced into mouse fibroblasts. Nature, 290, 261–264.

Slamon, D.J., Clark, G.M., Wong, S.G., Levin, W.J., Ulirich, A., & Mcguire, W.L. (1987). Human breast cancer: Correlation of relapse and survival with amplification of the HER-2/neu oncogene. Science, 235, 177–182.

Tuszynski, J. (2012). caTools: Tools: Moving window statistics. Retrieved from https://cran.r-project.org/web/packages/caTools/index.html.

Wallace, M.L., & Rafols, I. (2014). Research portfolios in science policy: Moving from financial returns to societal benefits. Minerva, 2015, 53(2): 89–115.