Maximizing Text Mining Performance: the Impact of Pre-Processing

Lanera C[1], Berchialla P[2], Baldi I[1], Gregori D[1]
[1]Unit of Biostatistics, Epidemiology and Public Health, Department of Cardiac, Thoracic and Vascular Sciences, University of Padova, V. Loredan 18, 35131 Padova, Italy
[2]Department of Clinical and Biological Sciences, University of Torino, Via Santena 5bis, 10126 Torino, Italy

Abstract
In recent years, with the rise of Electronic Medical Records (EMRs), there has been a dramatic increase of text to analyze. While the selection and tuning of the outperforming algorithm, which are intertwined to the scalability and robustness of the algorithm itself, can be implemented in ready-to-use systems, the pre-processing is still not an automated step of the analysis. The importance of the pre-processing relies on the fact it serves as the basis of any further analysis and a poor pre-processing can hamper the performance even of the best tuned algorithm. In this work, we studied the impact of the most common text pre-processing steps, such as stripping white space, removing stop-words, stemming or building n-Grams, on classification. The motivating example is the classification of EMRs. The pre-processing is assessed in conjunction with neural networks, support vector machines and boosting to highlight their synergistic impact and the importance of the order in which the single steps are carried out.

Key Words: Pre-processing, Machine learning, Electronic Medical Records

1. Introduction

In recent years, with the rise of Electronic Medical Records (EMRs), there has been a dramatic increase of text to analyze. Pre-processing the data is the process of cleaning and preparing the text for classification. The classification task is typically accomplished by Machine Learning Techniques (MLTs) which are tools used for analyzing large and complex data sets called Data Mining aimed at discovering knowledge in an automatic or semi-automatic process[1]. While the selection and tuning of the outperforming MLT, which are intertwined to the scalability and robustness of the algorithm itself, can be implemented in ready-to-use systems, the pre-processing is still not an automated step of the analysis[2 3].

Narrative EMRs fields usually contain lots of uninformative words which represent noise. Keeping those words makes the dimensionality of the problem high and the classification more difficult. Having the data properly pre-processed stems from the need to reduce the noise in the text to help improve the performance of the classifier and speed up the classification process.

This study aims to assess several different pre-processing tasks in text mining, and to what extent the order they are carried out can affect the performance of machine learning algorithms for text classification. The study is conducted on a large corpus of EMRs on infectious diseases. Text Mining (TM) narrative EMRs data to investigate disease burden

is a relatively new and promising approach in the field of infectious diseases, where manual review of narrative fields is the current form of text exploration [4].

Our objective is to assess and compare the overall performance of different pre-processing strategies given a MLT, chosen according to the benchmark suggested by the literature[5-7]. Secondarily, we aim at disentangling the pre-processing impact from the specific MLT impact on the classification/prediction accuracy[8].

2. Materials and Methods

2.1 Data source

The source of data is primary care records maintained by general practitioners on visits, diagnosis, prescriptions, hospitalizations and specialist visits over a 10-year span. The aim was to identify incident infectious diseases out of 1,230,355 entries. At the time of writing, authors are not entitled for confidentiality reasons to disclose all the data information.

2.2 Pre-processing procedures

At first, a simple regular expression match and a fast manual review were implemented to identify in a semi-supervised way a subset to be used for training the machine on out-of-doubt positives/negatives. More in detail, pattern matching was used for each infectious disease of interest only into the short-text fields and an independent researcher was asked to select only the most obvious positive cases out of them. Negatives were defined as patients not matching any of the above patterns in any field of any of their entries. The excluded records were marked as "doubt" and are the base on which the final performance statistics were computed.

The pre-processing procedures considered fall in three main groups: reducing, modificating and enriching procedures. The first group is about "removing something", like extra non wanted white spaces or symbols, numbers or even words we do not need the MLT take care about. Mainly, this procedure removes noise. We selected "remove non-word text", "remove stop-words" and "strip whitespace" procedures.

The second group concerns "merging something", like different inflection or counts (i.e. singular or plural) of the same words, different kinds of typing nouns or words at the beginning of a sentence with capital letters but even words with the same field of semantic action. Mainly, this procedure merges concepts we think could be lost otherwise with the risk to allow important information to become noise. We selected "lowering" and "stemming" procedures.

The last group is about "producing something", like "new words". To be more precise, in the context of text analyses we distinguish from human words called simply "words" and computer words called "tokens"[9]. Normally each word is considered as a token as well as the opposite. Using this procedure, we are able to consider (and so they are passed to the algorithms as they are a single indivisible piece of information) a group of words, in the case of n-Gram a consecutive sequence of exactly n words. Mainly, this procedure produces new tokens useful (hopefully) to bring information which could be lost otherwise, e.g. negations. We selected "producing n-Grams[9]", limiting $n \in \{1, 2, 3\}$ and considering only increasing subsets of them, i.e. only 1-gram (i.e. no generation of tokens, other than the words just considered), 1-gram $\cup$ 2-gram (i.e. all the words plus all the consecutive couple of words) and 1-gram $\cup$ 2-gram $\cup$ 3-gram.

All strategies are listed in table 1 with the following criteria: procedures executed on the same strategy are identified, in each row, by the same letter. Moreover, procedures belonging to the same strategy are executed in order from the leftmost one to the right when there were the same or no number near the letter, otherwise they are executed according to the numbering. For examples on the third line "two groups" f1-f3 and f2 will be executed on the same pre-processing strategy (which executes no procedure in the merging groups), ordered as follows: removing non-words, stop-words and next the extra white space + producing 2-Gram + removing non-words, stop-words and finally the extra white space.

Table 1: Strategies implemented: strategies are identified by letters. Numbers gives the ordering of the procedures applied to each strategies. If none or same number, the order of execution is from left to right.

Executing	*Removing group*			*Merging group*		*Producing group*	
	non-word	*white space*	*stop-words*	*lower*	*stem*	*2-Gram*	*3-Gram*
one procedure	a	b	c	d	e	f	g
one group	a			b		c	d
two groups	a b c d2 e2 f1-f3 g1-g3			a h i		b d1 f2 h	c e1 g2 i
all groups	a b c3 d3 e1-e4 f1-f4			a b c2 d2 e2 f2		a c1 e3	b d1 f3

Next the various types/sequences of functions and procedures identified for the pre-processing are applied on the whole database and the selected metrics are registered at each step.

The final analysis is carried out transforming the resulting output of each pre-processing step into a Document-Term Matrix (DTM) filled using the term frequencies - inverse document frequencies weights and passing it to each of the chosen algorithms. For each pre-processing strategy we record and compute the following metrics: i) computational time to asses each single pre-processing step, to assess the overall pre-processing procedure and to perform the training each MLT; ii) maximum ram needed to asses each one of the overall procedures from the pre-processing to the training of the MLT; iii) confusion matrix of the predicted results with computation of sensibility, precision, specificity, accuracy and F1 score.

2.3 Machine Learning techniques

2.3.1 Artificial Neural Network with one hidden layer

Artificial Neural Networks[10] [11] are models inspired by the structure and/or function of biological neural networks. They are a class of pattern matching that are commonly used for regression and classification problems but are really an enormous subfield comprising hundreds of algorithms and variations. The one we have selected is the perceptron with one hidden layer. This model may be considered as a logistic regression where the covariates are pre-processed using a non-linear transformation projecting the input data in a space (intermediate hidden layer) where it becomes linearly separable.

2.3.2 Support-Vector Machine (SVM)

SVM[12] is a supervised learning method useful for classification and regression analysis. The method is based on the construction of one or more hyperplanes in high or infinite dimension leading to separate the data according to output variable. For this purpose, the

better hyperplane has the largest distance to the nearest data belonging to class, they are the functional margin, for greater margin the classification error is lower.

2.3.3 Boosting

The Boosting is a general approach for improving the accuracy of any given learning algorithm. Here we considered an adaptation by Tuszynski[13] of the logitboost algorithm[14] aimed at speeding-up the whole procedure when applied on very large data sets. The algorithm implements the standard boosting technique[15], i.e. sequential use of a classification algorithm that is represented in this case by a decision tree.

2.4 Software

Analyses were performed using R software version 3.2.4 (now updated at the last version 3.3.0) and the following packages were used:

- RTextTools 1.4.2: Automatic Text Classification via Supervised Learning
- tm 0.6-2: Text Mining Package
- stringr 1.0.0: Simple, Consistent Wrappers for Common String Operations
- RWeka 0.4-26: R/Weka Interface
- SnowballC 0.5.1: Snowball stemmers based on the C libstemmer UTF-8 library
- slam 0.1-32: Sparse Lightweight Arrays and Matrices

3. Results

At present we have carried out the strategy “e” as referred in the last rows of table 1, (i.e. the simpler one which includes all procedures). Preliminary results are reported below. After the first “globally-reached-the-end” execution, we are now working on fixing bugs and adopting patches as well as a *defensive programming* approach in order to be able to run a single loop-function which will automatically collect all the interested metrics for each of the selected sets of pre-processing procedures. For this reason, at the moment no metrics on computational time nor space complexity are rigorously collected. Approximately, on our 2 x 3.0 GHz quad-core server equipped with 128 GB of ram the whole performed procedure takes a couple of days to run.

As preliminary results for Boosting algorithm, the procedure has identified in a couple of days 3,472 true positives out of 3,481 and no false positives, 4,150 true negatives and 9 false negatives (table 2).

Table 2: Preliminary results for Boosting algoritmhs and strategy "e".

Pre-processing group	*sensitivity*	*specificity*	*accuracy*	*F-measure*
stopwords g1	0.656	0.566	0.582	0.173
spaces g1	0.656	0.566	0.582	0.173
non words g1	0.656	0.566	0.582	0.173
stemming g2	0.656	0.565	0.58	0.173
lowering g2	0.996	0.429	0.524	0.207
2-gram	0.812	0.577	0.617	0.208
g1 + g2 + 2-gram	0.996	0.426	0.522	0.206

g2 + g1 + 2-gram	0.996	0.426	0.522	0.206
2-gram + g2 + g1	0.987	0.453	0.542	0.210
2-gram + g1 + g2	0.987	0.453	0.542	0.210
g2 + g1 + 2-gram + g1	0.996	0.426	0.522	0.206

4. Discussion

Pre-processing text is a time consuming task when using MLTs for classifying EMRs and it is important to establish to what extent it is necessary.
In this work we applied different pre-processing tasks and different strategies involving multiple pre-processing tasks to EMRs text with the aim to classifying records according to the presence of infectious diseases using MLTs.
In a previous work, Goncalves et al[16] applied stop word removal, stemming, replacing words with synonyms and pruning (i.e. removing too frequent words) and combination of them on MEDLINE documents. On a variety of MLTs they showed the effect of pruning in boosting accuracy of algorithms. However, they did not consider bi-grams among their set of pre-processing steps.
Munkova and colleagues[17] showed that when using algorithms for sequence rules extraction, removing stop-words has an impact on the quality of extracted rules.
In our data, the pre-processing tasks that affected most the MLT sensitivity are lowering and bigrams. In general, while not performing lowering, using two or more pre-processing tasks along with bigrams gives a boost to MLT sensitivity. On the other hand, building bigrams is the pre-processing task that better improves the MLT accuracy with respect to not performing any kind of pre-processing task.
Further analyses are required first to assess if the same results are observed also on different MLTs (like SVM, NN, decision trees, Bayes Net, K-nearest neighbours and ensemble algorithm), secondly if the value of lowering and bigrams pre-processing is not due to the peculiarity of the EMRs focused on infection disease classification.

References

1. Witten IH, Frank E, Hall MA. Data Mining: Practical Machine Learning Tools and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc 2011.
2. Afzal Z, Schuemie MJ, van Blijderveen JC, et al. Improving sensitivity of machine learning methods for automated case identification from free-text electronic medical records. *BMC Med Inform Decis Mak* 2013;13:30. doi: 10.1186/1472-6947-13-30
3. Jensen PB, Jensen LJ, Brunak S. Mining electronic health records: towards better research applications and clinical care. *Nat Rev Genet* 2012;13(6):395-405. doi: 10.1038/nrg3208
4. Wang Z, Shah AD, Tate AR, et al. Extracting diagnoses and investigation results from unstructured text in electronic health records by semi-supervised machine learning. *PLoS One* 2012;7(1):e30412. doi: 10.1371/journal.pone.0030412
5. Riemenschneider M, Heider D. Current Approaches in Computational Drug Resistance Prediction in HIV. *Curr HIV Res* 2016;14(4):307-15.
6. Koo CL, Liew MJ, Mohamad MS, et al. A review for detecting gene-gene interactions using machine learning methods in genetic epidemiology. *Biomed Res Int* 2013;2013:432375. doi: 10.1155/2013/432375

7. Kapetanovic IM, Rosenfeld S, Izmirlian G. Overview of commonly used bioinformatics methods and their applications. *Ann N Y Acad Sci* 2004;1020:10-21. doi: 10.1196/annals.1310.003
8. Haddi E, Liu X, Shi Y. The role of text pre-processing in sentiment analysis. *Procedia Computer Science* 2013;17:26-32.
9. Marafino BJ, Davies JM, Bardach NS, et al. N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *J Am Med Inform Assoc* 2014;21(5):871-5. doi: 10.1136/amiajnl-2014-002694
10. Venables WN, Ripley BD. Modern Applied Statistics with S. New York: Springer-Verlag 2002.
11. Ripley BD. Pattern Recognition and Neural Networks: Cambridge University Press 1996.
12. Cortes C, Vapnik V. Support-vector networks. *Machine Learning* 1995;20(3):273-97.
13. Tuszynski J. caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc., 2014 [Available from: https://CRAN.R-project.org/package=caTools.
14. Dettling M, Buhlmann P. Boosting for tumor classification with gene expression data. *Bioinformatics* 2002;19(9):1061-69.
15. Freund Y, Schapire RE. Experiments with a new boosting algorithm. . *Proceedings of the thirteenth international conference on machine learning, Morgan Kaufmann* 1996:148-56.
16. Goncalves CA, Goncalves CT, Camacho R, et al. The impact of pre-processing on the classification of MEDLINE documents. Proceedings of the 10th International Workshop on Pattern Recognition in Information Systems, PRIS 2010, in Conjunction with ICEIS 2010, 2010:53-61.
17. Munková D, Munk M, Vozár M. Data Pre-processing Evaluation for Text Mining: Transaction/Sequence Model. *Procedia Computer Science* 2013;18:1198-207. doi: http://dx.doi.org/10.1016/j.procs.2013.05.286