Open Source, Web-based Machine-Learning Assisted Classification System

Mireia Roser Consarnau

Universitat Oberta de Catalunya (UOC)

Abstract

The aim of this article is to provide a design overview of the web based machine learning assisted multi-user classification system. The design is based on open source standards both for multi-user environment written in PHP using the Laravel framework and a Python based machine learning toolkit, Scikit-Learn. The advantage of the proposed system is that it does not require the domain specific knowledge or programming skills. Machine learning classification tasks are done on the background automatically. The paper commences with a review of literature on applications of text mining the most common type of data available, discusses the main system components and outlines the process flow with examples. System effectiveness is also evaluated using a dataset comprised of music lyrics divided into two classes of 245 songs each, using a support vector machine (SVM) classifier, function words feature set, three fold cross-validation and 10:90 train test split.

Index Terms

Open Source, classification system, text analysis.

I. Introduction

Text mining is an umbrella term that covers techniques for text classification, summarization and topic detection among others. Information storage and retrieval as well as the natural language processing lay in the core of the text mining techniques making the human produced textual artifacts machine readable. Statistical modeling and machine learning techniques are used to perform the text mining tasks. There are many applications where text mining techniques may be found useful and even essential. For example sentiment analysis in market research, topic discovery and classification in patent analysis and summarization in literature search are the areas where text mining simplifies processing of the textual content and automates analysis

for the user. The following paper presents a novel approach for web-based classification of user-uploaded content. Traditionally, the machine learning tools require programming skills and knowledge of sophisticated machine learning algorithms. This approach eliminates the domain specific knowledge and turn machine learning text classification into a user friendly activity that can be performed with a few clicks.

II. Background

In the following sections the literature review is presented, which examines some of the recent trends in text mining, with an aim to incorporate these findings in the design of the text classification prototype.

A. Classification

A study by Aggarwal, Zhao, Yu [1] proposed a method for classification of textual data by extracting and combining features from both the content and meta-data either embedded in the content or related to the content. The contribution of their approach is such that much of the text classification methods does not take meta-data into account, whereas [1] developed two algorithms COntent and Auxiliary attribute based Text clustering (COATES) and COntent and auxiLiary attribute-based Text classification (COLT) that are aimed at enhancing the feature sets by incorporating the meta-data to improve classification accuracy. To demonstrate the superiority of the proposed method, a series of experiments were conducted. The performance of the COLT-based method was compared to that of the Nave Bayes and the Support Vector Machine classifiers.

Meta-data, or the side-information [1] comes in various forms. For example texts may be in-text embedded web links or text file properties such as usage statistics, name of the creator and owner and access information such as file access logs. All these extraneous attributes by themselves can be used as discriminators in clustering or classification, however they, or be combined with content data. The meta-data usage should be carefully considered as they may become the noise source. ”While such side-information can sometimes be useful in improving the quality of the clustering process, it can be a risky approach when the side-information is noisy. In such cases, it can actually worsen the quality of the mining process. Therefore, we will use an approach which carefully ascertains the coherence of the clustering characteristics of the side information with that of the text content” [1].

And hence arose a challenge of developing a method of using the meta-data and the content data features in such a way that they enhance classification and minimize interference. The proposed approach is a combination of partitioning and probabilistic estimation processes that estimate coherence of the meta-data features with textual features. The process involves finding meta-data features that behave similarly when clustering using the textual features.

The results of the experiments suggest that the COLT algorithm which combines features extracted from text as well as the meta-data-based features, does increase the accuracy rates and outperform Nave Bays and Support Vector Machine classifiers using only the text-based features [1].

Tseng, Lin and Lin [2] applied text mining to patent analysis. Patent documentation contains a substantial amount of information to be processed. In addition to technical details, patent documents contain relations to other inventions, development and business trends, industrial applications and use. This information may be used for development of investment policies and planning. Current strategies are aimed at extracting structured information including the filing dates, names of the assignees and citations and analyze these data using bibliometric and data mining techniques. There are several text mining approaches at work that mimic how patent analysts deal with the analytical patent review. These techniques include ”text segmentation, summary extraction, feature selection, term association, cluster generation, topic identification, and information mapping” [2].

Authors have proposed a method of automation of patent analysis to simplify the review and decision making based on clustering. The method also generates patent maps for topic analyses and enhances the patent analysis tasks including patent classification, organization and prior art search. Term summaries allow users to conduct the prior-art search. The output is visualized depicting the trends and links between the documents. The traditional methods often do not go farther than extracting the title and the abstract information, while the proposed method extracts textual content from the entire document. The method was tested using a set of 612 patents, using 6 segment s per patent. Term frequency was used to generate the topic maps. The results suggest that it is more effective to classify documents based on partial matching of its segments than that by the whole document.

B. Text analysis

The issue of performance is critical to the qualitative assessment of the text mining systems. There is a magnitude of conditions that may affect the outcome [3]. The real world data is noisy and may not be readily processed. Pre-processing of texts may affect the quality of classification, thus having a baseline and methods of controlling it are essential to the quality of the text processing task. Hashimi, Hafez and Mathkour (2015) propose a set of criteria for evaluation of effectiveness of the text mining techniques. Their survey of more than 130 research articles has identified 25 different criterions in 4 categories. The proposed criteria has been selected by weighing their occurrence in the literature and includes: Usability, Comprehensiveness, Flexibility, Complexity, GUI, Business goals satisfaction, Specific Goals Achievement, Support KPI, Support MLI, Support Compliance, Support to Individual Contents, Extraction, Aggregation, Clustering, Indexing, Accumulation, Identification, Effectiveness of Content, Text Interface, Passive, Summarization, Contraction, Abbreviation, Achievement of Text Learning.

A study by Nassirtoussi, Aghabozorgi, Wha, Ngo (2014) examines the relationship between the text mining of sentiments and financial markets. The authors argue that this area of research is lacking technical framework due to its multidisciplinary nature. Their survey of the related literature aims to compare the text mining systems and identify common elements in the applications of text mining for the market analysis. The application of market prediction using textual data from social media sources is comprised of three components which include: linguistics, behavioral economics and machine learning. The former is the natural language that needs to be translated into the machine readable code, the second is the underlying psychological factors influencing the decision making as we the expression of sentiments that may influence the business decision and the last is the set of techniques for computing the prediction of the decision based on the language used to express the relationship towards a product or service. The notion of price in behavioral economics is considered a perceived value. The price is in the mind of the beholder and not perceived as the cost of the product production. This presupposes a variance in interpretation among individuals. The relationship between behavioral manifestations of confidence and pessimism towards a product or service serve as a predictor of the market activity. This is useful because, the textual features are considered behavioral predictors of another type of activity. The market prediction systems take two types of data, the textual data and the

market data. The textual data may be acquired from several sources such as news feeds and social media. As well academic journals and scientific literature are the suitable candidates for in-depth analysis [4].

Once the data is acquired, the analysis follows the standard protocol which include data pre-processing, feature-selection, dimensionality-reduction, feature-representation, machine learning algorithm selection, model training in case of supervised learning and prediction. The survey results suggest that studies present the findings by using a confusion matrix to depict the results. Accuracy is the most common metric used to evaluate the model performance and often include recall, precision and the F1 score. The accuracy ranges between the 50%-70% where the ”results above 55% have been considered report-worthy [2]. Market prediction from textual data bears similarity with other types of textual analyses such as authorship identification, profiling and attribution. For example imbalanced datasets may pose a challenge because classifiers are often biased towards larger training sets. And similarly to stylometric research, the market prediction research varies in the types of experiments, techniques and datasets employed making it difficult to conduct an objective comparison of the effectiveness of the proposed techniques.

A study by Scandariato, Walden, Hovsepyan, and Joosen [5] has applied text mining to identification of security vulnerabilities in software source code. The machine learning techniques are used to predict vulnerable software components by analyzing term frequencies. To assess the feasibility of the proposed method, Scandariato, et al. [5] conducted an experiment using 182 releases of 20 Android OS applications.

The results suggest that it is possible to apply text mining techniques for prediction of security vulnerabilities. The measures of precision and recall are used to support the findings. Two classifiers have been used in the experiments and included Nave Bayes and Random Forest. Weka machine learning tool has been used to run the experiments using default classifier parameters with the exception of random forest algorithm where the number of trees have been increased to one hundred. A bag of words approach has been adopted where term frequencies are the features and used in predicting the outcome. Similar to spam detection, classification of the source code on basis of presence of security vulnerabilities has two classes: vulnerable and not vulnerable. A piece of code was considered vulnerable if at least one vulnerability warning were reported by the static code analyzer. All source code was passed through the code analyzer to label the set. 10-fold cross-validation was employed. Furthermore, discretization the process of transforimg continuous data into discrete categories appeared to have a positive effect on performance. The

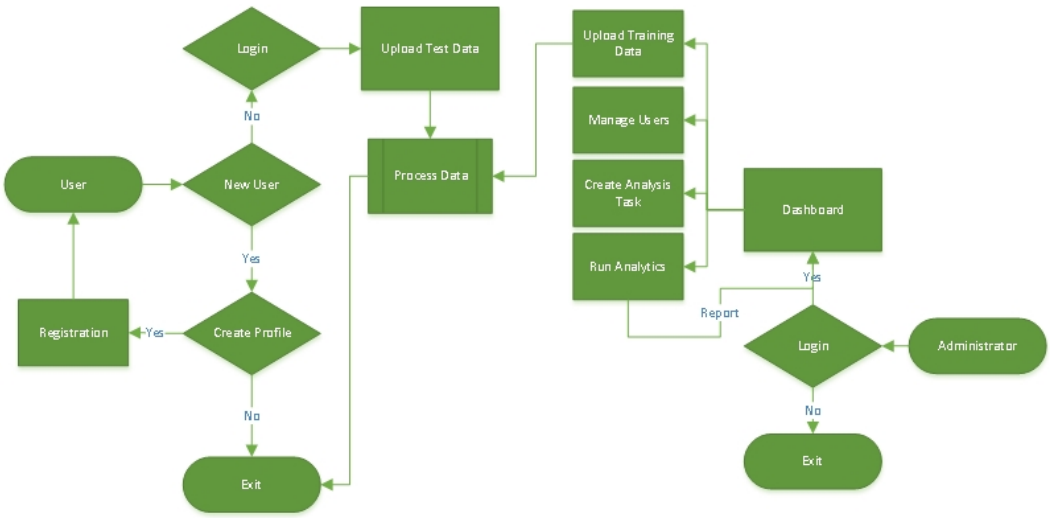

Fig. 1. System design

results ranged from 0.59-0. 99 prediction scores and 0.43-0. 92 for recall scores, which are comparable to the earlier studies.

III. Framework Design

The framework is comprised of the front end that provides file and user management functionally and analytical engine at the back end responsible for data processing, modeling, analysis and reporting. The front end takes advantage of the open source PHP framework (Laravel)[6], whereas the analytics engine is driven by the Python machine learning library(Scikit-Learn). The main system components could be organized into three main functional groups which include: user management, file management, and analysis [7]. In the following sections, each component is described in greater detail. Figure 1 depicts the overall system design and its individual components.

A. User Management task

One of the advantages of using the open source development framework Laravel is that it comes with tools necessary for maintaining a multi-user environment right out of the box. Its user management module enables self-registration of users and easy integration with a mail server for confirmation and retrieval of the account information. It also comes with user authentication capability. The proposed classification system utilizes two distinct user groups: the standard users

and the admin users, where the former provide information for the analysis and the latter group may analyze the uploaded information. Furthermore, user accounts can be managed from within the admin panel and include such functionality as password resets, account information updates in addition to deletion and creation of new user accounts.

B. File management task

Considering that the system is capable of running both supervised and unsupervised machine learning tasks, the file management requirements are such that both the administrative and the standard users should be able to upload information in a form of documents. When the machine learning task is supervised a training set needs to be provided. This task is managed by the administrator who uploads classification dataset. In a scenario where each individual user provides unique content, username is treated as label and the training data is linked to each users profile. The users then upload the test files and administrator run the analysis task. Each and every file uploaded is listed either under the training set category or the test set and can be then deleted. The front end acts as a graphical interface to managing the uploaded text files. It is responsible for data collection and management. In unsupervised learning task, only the administrative user manages the files, which are uploaded to the system. These files can be managed I a similar fashion to that of the supervised method.

C. Analysis task

The analysis task is performed through the link to external module written in Python. The analysis code takes advantage of the Scikit-learn machine learning library. The front end passes on the parameters to the python analysis script to execute the classification task. The output is then channeled back to the front end to display the outcome of classification. Unfortunately, the PHP language does not have a powerful machine learning capabilities and thus an external processing is required. The choice of the algorithm and the way data is passed on the classifier would depend on the type of classification task. In unsupervised classification, only one set of data is passed on to an algorithm, where the supervised classification required training set to build a model and fit the new, unseen data to that model to obtain a prediction. Thus, the algorithm and pre-processing functions need to be tailored specific to each task. The output can be displayed as a prediction of a class label or visual representation of different data clusters, as well as the probabilities of each class.

1) Classifier: Classification utilizes the SVM [8] algorithm which constructs optimal hyperplane for separable patterns both either linear or non-linear. Support vectors are the data points that lie closest to the hyperplane or the decision surface. The decision function is defined by training samples.

- Training data $\{\mathbf{x}_i, y_i\}$ $i = 1, \ldots, l$, $\mathbf{x}_i \in R^n$, and $y_i \in \{-1, 1\}$
- On a separating hyperplane: $\mathbf{x}\mathbf{w} + b = 0$, where
  - w normal to the hyperplane
  - $\frac{|b|}{\|\mathbf{w}\|}$ is the distance to origin
  - $\|\mathbf{w}\|$ Euclidean norm of $\mathbf{w}$
- d_+, d_- shortest distances from labeled points to hyperplane
- Define margin $m = d_+ + d_-$
- Task: find the separating hyperplane that maximizes m
- For the separating plane:

$$\mathbf{x}_i\mathbf{w} + b \geq +1, \quad y_i = +1 \tag{1}$$

$$\mathbf{x}_i\mathbf{w} + b \leq -1, \quad y_i = -1 \tag{2}$$

$$\equiv \tag{3}$$

$$y_i(\mathbf{x}_i\mathbf{w} + b) - 1 \geq 0, \qquad \forall i \tag{4}$$

- For the closest points the equalities are satisfied, so:

$$d_+ + d_- = \frac{|1 - b|}{\|w\|} + \frac{|-1 - b|}{\|w\|} = \frac{2}{\|w\|} \tag{5}$$

The analysis task can be organized into five distinct processes each responsible for data manipulation. The general system process flow is presented in Figure 2. First data undergoes pre-processing task, where the noise items such as direct quatations and names are removed. Features may be autmatically by an algorithm where only the most relevant features are used or pre-defined based. In this case, the function words [9] are used as a feature set and are built-in. The function words are then extracted and the rest of the data is omitted. Each function word is vectorized and counted. Vocabulary matrix is constructed based on the function word counts in the entire training dataset. New data undergoes same pre-processing and vectorization steps. The new data is passed on to the classifier which predicts a label for every sample in the test set based on the model constructed. The predictions are than passed on the fornt-end which produces are readable report.

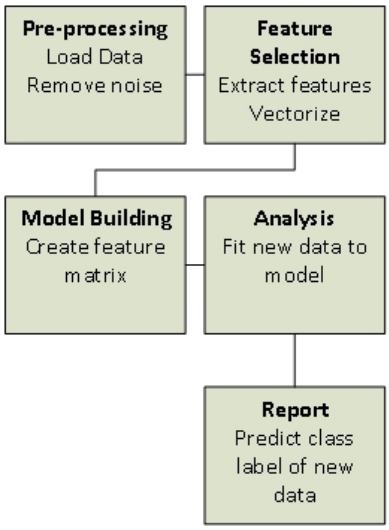

Fig. 2. Data process flow

IV. Framework evaluation

The previous sections have provided background information into the nature of classification task, as well outlined the system components. In this section, I am going to present findings of conducting supervised classification of two classes of music lyrics using the support vector machine (SVM) classifier, and English function words as features. The data set is comprised of lyrics to songs of two genres, country and rap, 245 items in each genre. Because I am dealing with two classes and the input can only be classified into one of two non overlapping classes (C_1, C_2), this task can be defined as a binary classification problem [10]. It wold be safe to assume that the trained model may occasionally misclassify the music genre and attach a label of a different class. This would affect the accuracy and by extension performance of the classifier. Although this framework is not aimed at developing the best classifier, but rather a system that allows to employ a variety of classification schemas, this preliminary work is important to evaluate the overall look and feel of the classification system through the lens of the user and administrator. To this end, performance is evaluated and reported using the four measures of accuracy, precision, recall and F1 score. Three-fold cross validation methodology was employed, where 10% of the data set was used for training and 90% for testing. Accuracy score is overall effectiveness of a classifier. When converted into percentage, it shows the ratio of correctly classified samples. Precision shows class agreement of the data labels with the positive labels given by the classifier. The measure of recall shows effectiveness of a classifier to identify

positive labels. And the F measure also known as the F1 score shows the relationship between true matches and predicted ones.

For this task 3 user accounts have been created. The administrative user is the one conducting the analysis. Class 1 user uploads the country songs, and the Class 2 user uploads the Rap songs. The file management screen is depicted in Figure 3. Upon uploading the entire dataset to the framework, the administrator runs the analysis report which provides a confusion matrix with predictions for each of the sample in the test set and evaluation the performance.

A. Results

The results show a very promising trend with an average F1 score of 0.82 and accuracy of 84% using a set of 490 documents and a feature set comprised of 429 features. The results are as follows.

Genre	**Precision**	**Recall**	**F1**	**Documents**
Class1 (Country)	0.88	0.73	0.80	245
Class 2 (Rap)	0.79	0.91	0.84	245
Avg/Total	0.83	0.82	0.82	490

V. Conclusion

The proposed classification system takes advantage of the open source technologies including the Laravel framework for PHP front end development and Scikit-learn python library for conducting machine learning tasks. The results of the experiment suggest the ease of use and feasibility of using such approach to enable non-programmers or individuals unfamiliar with complexity of machine learning to conduct machine learning tasks through an intuitive web based interface. Laravel’s user management module allows to add users and create workgroups and be used in scenarios where user input is required such as research projects, marketing surveys, stock market analytics and data mining. The classification algorithm provides a high degree of accuracy and is robust to account for other type of data such as weather report and real-estate market prediction.. The future research should on expansion of the algorithm base, to allow users to select or manipulate algorithm-level variables to further tweak the model. The modular nature of this system allows to employ any feature set and any algorithm making it a universal tool for the data analysis.

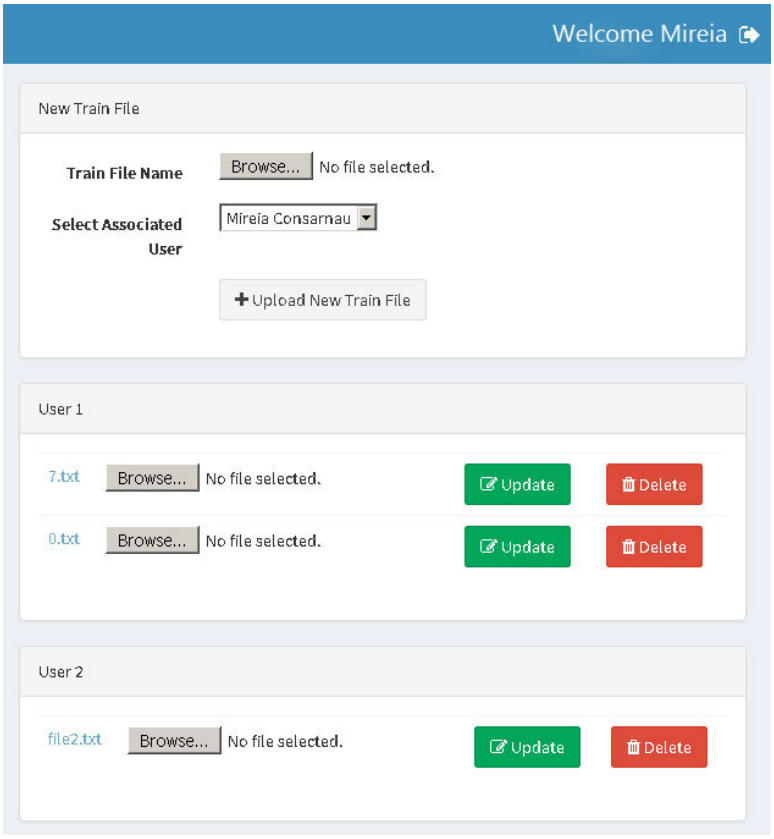

Fig. 3. File management screen

REFERENCES

[1] C. C. Aggarwal, Y. Zhao, and P. S. Yu, "On text clustering with side information," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 894–904.

[2] Y.-H. Tseng, C.-J. Lin, and Y.-I. Lin, "Text mining techniques for patent analysis," *Information Processing & Management*, vol. 43, no. 5, pp. 1216–1247, 2007.

[3] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[4] S. Agarwal and H. Yu, "Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion," *Bioinformatics*, vol. 25, no. 23, pp. 3174–3180, 2009.

[5] R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, "Predicting vulnerable software components via text mining," *Software Engineering, IEEE Transactions on*, vol. 40, no. 10, pp. 993–1006, 2014.

[6] R. Saunier, *Getting Started with Laravel 4*. Packt Publishing Ltd, 2014.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "A multi-class svm classifier utilizing binary decision tree," *Informatica*, vol. 33, no. 2, 2009.

[9] R. Arun, V. Suresh, and C. V. Madhavan, “Stopword graphs and authorship attribution in text corpora,” in *2009 International Conference on Semantic Computing (ICSC)*. IEEE, 2009, pp. 192–196.

[10] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.