Mining Scientific Articles Powered by Machine Learning Techniques

Carlos A. S. J. Gulo[1,2], Thiago R. P. M. Rúbio[1,3],
Shazia Tabassum[1,4], and Simone G. D. Prado[5]

1 Departamento de Engenharia Informática, Faculdade of Engenharia,
Universidade do Porto, Porto, Portugal

2 PIXEL Research Group, UNEMAT, Brazil
sander@unemat.br

3 LIACC – Artificial Intelligence and Computing Science Laboratory,
Universidade do Porto, Porto, Portugal
reis.thiago@fe.up.pt

4 LIAAD, Inesctec, Porto, Portugal
shazia.tabassum@inesctec.pt

5 Departamento de Computação , Faculdade de Ciências, Universidade Estadual
Paulista, Bauru, Brazil
simonedp@fc.unesp.br

Abstract

Literature review is one of the most important phases of research. Scientists must identify the gaps and challenges about certain area and the scientific literature, as a result of the accumulation of knowledge, should provide enough information. The problem is where to find the best and most important articles that guarantees to ascertain the state of the art on that specific domain. A feasible literature review consists on locating, appraising, and synthesising the best empirical evidences in the pool of available publications, guided by one or more research questions. Nevertheless, it is not assured that searching interesting articles in electronic databases will retrieve the most relevant content. Indeed, the existent search engines try to recommend articles by only looking for the occurrences of given keywords. In fact, the relevance of a paper should depend on many other factors as adequacy to the theme, specific tools used or even the test strategy, making automatic recommendation of articles a challenging problem. Our approach allows researchers to browse huge article collections and quickly find the appropriate publications of particular interest by using machine learning techniques. The proposed solution automatically classifies and prioritises the relevance of scientific papers. Using previous samples manually classified by domain experts, we apply a Naive Bayes Classifier to get predicted articles from real world journal repositories such as IEEE Xplore or ACM Digital. Results suggest that our model can substantially recommend, classify and rank the most relevant articles of a particular scientific field of interest. In our experiments, we achieved 98.22% of accuracy in recommending articles that are present in an expert classification list, indicating a good prediction of relevance. The recommended papers worth, at least, the reading. We envisage to expand our model in order to accept user's filters and other inputs to improve predictions.

1998 ACM Subject Classification H.3.3 Information Search and Retrieval

Keywords and phrases Machine Learning, Text Categorisation, Text Classification, Ranking, Systematic Literature Review

Digital Object Identifier 10.4230/OASIcs.ICCSW.2015.21

2015 Imperial College Computing Student Workshop (ICCSW 2015).
Editors: Claudia Schulz and Daniel Liew; pp. 21–28
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ICCSW 15

1 Introduction

Literature review is one of the most important phase of research to ascertain the state of the art of any specific domain area [4, 17, 6]. However, is possible to perform a literature review guided by different methodologies [17], for instance using traditional literature review or systematic literature review. In summary, a traditional literature review aims to perform a collection of information about researchers, theories and hypothesis, as such as how to solve a research problem using novel methodologies [4]. The outcome of this type of literature review, in general, is in reports or in thesis' chapter. Otherwise, the Systematic Literature Review (SLR) can be defined as a set of procedures which allows analysing systematically an interesting literature locating, appraising, and synthesising the most relevant researches in a domain area. The description of procedures should allow a reproducible literature review. Besides it, a literature review phase also helps to identify the gaps and challenges in that area. However, searching for interested articles in electronic databases do not retrieve the most relevant content indeed, although, the search engines recommend articles in which the specific keywords occur.

Automatic recommendation of articles is a challenging problem [20], mainly because the most scientific article contents are represented by text [15]. Text representation is critical in some text processing applications such as text categorisation [1], information retrieval[10], and topic modelling[3, 9]. Moreover, it's not a trivial process generating useful features from text representation to be used in many machine learning algorithms to support natural language processing [15].

The motivation for the development of the proposed model is providing automatically an efficient way of recommend, classify and rank important scientific literature. The manual process of finding and reviewing the most relevant literature that supports a research hypothesis is time consuming and error-prone, although researches [20, 3, 9, 1] for recommending scientific articles to users based on other users' ratings have showed good results. Our contribution in this new stage of our work is an automatic recommender system design focused on systematic literature review methods. The proposed solution is based on machine learning techniques and the process automatically classifies and prioritise the relevance of scientific papers.

In this paper we combine text mining and machine learning techniques as support to identify the most relevant literature using a data set collection searched in many journal repositories: ACM Portal [1], Engineering Village [2], IEEE Xplore [3], ScienceDirect [4], Web of Science [5]. Data set is analysed quantitatively in order to reduce the time used to review papers to write the literature review about our research domain: *high performance computing as support to computer aid diagnostic systems using medical images.*

Text Mining is a common process of extracting relevant information using a set of documents [8]. It provides basic preprocessing methods, such as identification, extraction of representative characteristics, and advanced operations as identifying complex patterns [9, 8]. Document classification is a task that consists of assigning a text to one or more categories: the name of its class of subject, and main topics.

The rest of the paper is organised as follows. In section 2 reviews the related works. Section 3.1 presents articles recommendation module, in Section 3 the experiments performed

[1] `http://dl.acm.org/`
[2] `http://www.engineeringvillage.com/`
[3] `http://ieeexplore.ieee.org`
[4] `http://www.sciencedirect.com`
[5] `http://apps.webofknowledge.com`

using Naive Bayes and the results obtained with the sets of scientific articles considered in the automatic text recommendation, are discussed in Section 3.2, which is followed by the concluding remarks in Section 4.

2 Background

Automatic recommendation of scientific articles consists on many sub-tasks, namely: data collection, text processing, data division, features extraction, feature selection, data representation, classifier training, applying the classification model, and performance evaluation [9].

Starting with data collection, we have to manage gathering the relevant references from known databases, such as literature repositories or other specific way to get documents. With this data, text preprocessing should remove undesirable information that represent noise. Stop words are removed (prepositions, pronouns, articles, adverbs and other auxiliary words) and the resulting words are steamed[21, 9].

Feature extraction reflects the terms we want to extract from the text. It may be related to the content (keywords) or not (author name, publication date, etc.), depending on data mining goals. At this step, the data is stored as a matrix that match the selected features with their weighting in the text. The calculation of the weighting can be obtained using statistical methods, such as the frequency on the documents (absolute or relative) [22, 7].

At this point, the data is divided into two main sets: training and test. We apply classifier algorithms to the training set in order to obtain a model that can predict a class or label to unseen data (test). These models usually recur to statistical approaches or machine learning paradigms. There is no ideal ratio of training data to testing data. The classification performance is the average performance of implemented classification models[24, 12]

Machine learning algorithms consist on recognising patterns from a data set and we aim to evaluate the extrapolation with unknown data. Many statistical algorithms can be used to create a model for classifying or labelling, such as [2, 5, 16, 18, 19, 23]: Latent Semantic Analysis (LSA) Language Model, Gaussian Model, Bayesian Model, among others. Various techniques are used: Support Vector Machine (SVM), Naive Bayes classifier, K-Nearest Neighbour (K-NN), Rocchio Algorithm, Decision Trees, Ensemble Classifiers, Inductive Logic Programming (ILP).

The last steps are the performance evaluation and the classification itself. As the training data have already a target value (previous) classification, if we present this same data to the trained model, the resulting performance is then obtained. The most common metrics for evaluating performance are accuracy, recall and precision. Recall correspond to the ability of the algorithm in retrieving the most relevant documents, meanwhile precision shows the capacity of the model in excluding not interesting documents. Once the model's predictive performance is adjusted, the final step consists on presenting new and unseen data (the test set from data division) and get the final result of the classification model [21].

3 Experiments And Model Design

The infrastructure used to perform the experiments and also illustrates the obtained results was composed of the Rapidminer Predictive Analytics Platform available to download through the Rapidminer website [6]. We have used the Rapidminer to construct the models and analyse

[6] `https://rapidminer.com/` – Rapidminer is a visual environment for predictive analytics, and it's considered easy-to-use just following the simple and intuitive instructions, and it's not required programming any code line to build models and make predictions

Table 1 Total of articles searched in journal repositories.

Repositories	Publication	
	Searched Queries	Papers
ACM Portal	("medical image") and ("high performance computing" or "parallel computing" or "parallel programming") and (PublishedAs:journal) and (FtFlag:yes) and (AbstractFlag:yes)	28
Engineering Village	(((((medical imag*) WN KY) AND ((high NEAR/0 performance NEAR/0 comput*) WN KY)) AND ((parallel NEAR/0 comput*) WN KY)) AND ((parallel NEAR/0 programm*) WN KY)), Journal article only, English only	22
IEEE Xplore	(((medical imag*) AND (("high performance comput*" OR "parallel programm*") OR "parallel comput*")))	68
ScienceDirect	"medical image" AND ("high performance computing" OR "parallel computing" OR "parallel programming")[Journals(Computer Science,Engineering)]	409
Web of Science	((("high performance comput*") OR ("parallel comput*") OR ("parallel programm*")) AND ("medical imag*"))	72
Total		599

the application of the algorithm. A portable computer equipped with an Intel(R) Core(TM) i7-2630QM 2.0 GHz, 8GB of RAM (DDR3 1333 MHz), Linux Debian Stretch (64 bits) operating system.

Data set used in experiments was built using the searching queries in each repository included in the previous Table 1, and composed by 575 observations (after removing 24 duplicated references), and 4 variables (*id, Title, Abstract, and Priority).* The analysed variable is text data, the *Abstract*, and its unstructured data. Unstructured data has variable length, one observation contains a scientific text, it has variable spelling using singular and plural forms of words, punctuation and other non alphanumeric characters, and the contents are not predefined to adhere to a set of values, it requires converting it to structured data for further processing. The preprocessing steps, provided by Text Mining methods, are responsible to make everything lowercase, remove punctuation and spaces, extract words from the data, replace synonyms, plural and other variants of words with a single term, reduce words to their stem, and remove common English stop-words, finally, create the structured data in table format where each word becomes a variable with a numeric value for each record [8].

3.1 Ranked-recommendation Based On Machine Learning Classification Model

This research work is new in terms of the methodology used to rank and prioritize papers. As a general classification model, we classify the scientific papers using a Naive Bayes classification algorithm. As a novel method we improve over it by extending the model to build a ranking model over the classification model as shown in Figure 1. This model uses the word list from the trained model and the already classified model using Naive Bayes Classifier. Then generates a ranking model which can be used as a recommendation system for future searches.

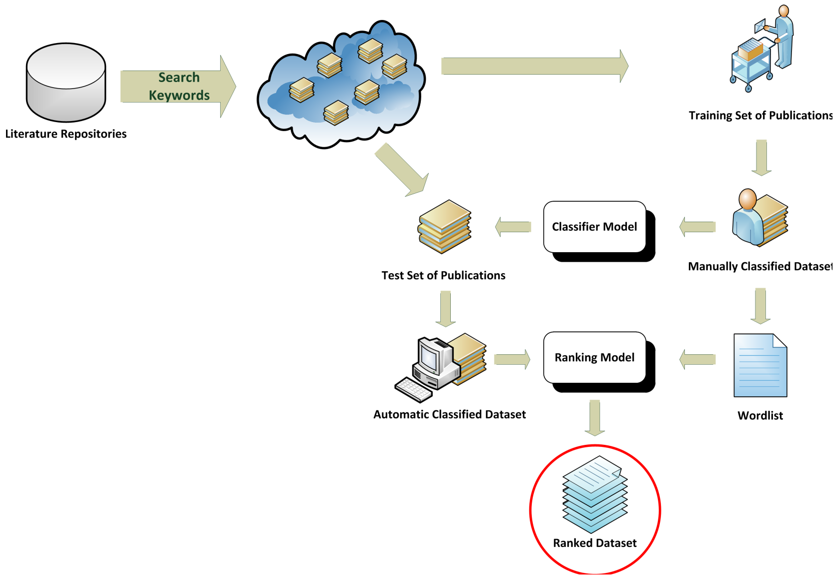

Figure 1 Architecture of the model process.

We are interested in establishing an automatic process able to classify and rank publications from a personal literature collection. Our main goal is to achieve the same level of relevance as performed by a human expert. We have seen that this process consists in various parts, following the steps described in Section 2 and selected the Naive Bayes algorithm as our first attempt to classify scientific papers. Although, Naive Bayes algorithm is not considered the most precise, otherwise is very simple to work with and to configure [23]. Using a previously retrieved data set, a human expert in a specific domain has analysed each one of the observations and classified the priority of the references regarding two main criteria: relevance of the reference and adequacy to the interested scientific domain. Its analysis consisted in classifying the reference into three priority classes:

- *Prio1*: References that are very relevant and adequate to the expected search;
- *Prio2*: References that are not so relevant but still adequate;
- *Prio3*: References that somehow interesting to the new research, but not the main source of knowledge.

A Naive Bayes model could then, be trained using the classification given by the expert. The classifier Naive Bayes is a supervised learning algorithm based on the Bayes theorem, which has strong independence features. Naive Bayes can be used with other models and play the role of vectorizer [11], obtaining hybrid models that best fit in certain classifications.

Figure 2 shows the Rapidminer model with the selected blocks responsible for training and performance evaluation of the Naive Bayes. The process of automatic classifying publications starts then, with a selected set of keywords that represent the context and the area of interest. We make a search in literature databases looking for the references that matches our filtering criteria (defined by our systematic review protocol). This set of references is the main data set we want to analyse, then it's divided into two parts: training and test. The training set corresponds to a smaller fraction of references that will be submitted to the user (expert in that domain) so that it should be classified manually. Simultaneously, based on the most important concepts presented on the analysed set, is created a *dictionary* of terms.

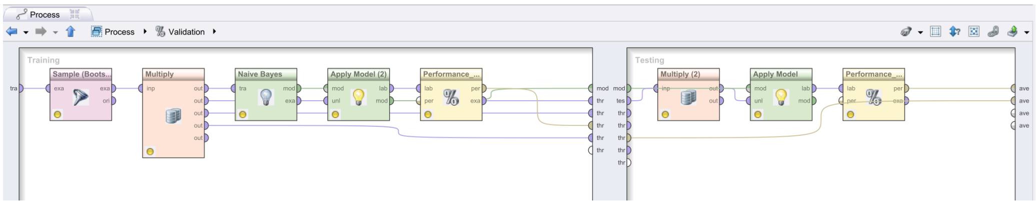

Figure 2 Naive Bayes Training process.

Table 2 Naive Bayes Results.

	true prio1	true prio2	true prio3	class precision
pred. prio1	833	0	0	100.00%
pred. prio2	0	211	63	77.01%
pred. prio3	0	0	2423	100.00%
class recall	100.00%	100.00%	97.47%	

3.2 Results

A Naive Bayes model is created and learns the classification patterns used by the domain expert. When this classifier model is applied to test the reference data set, the result is an automatic classified set of references. As seen in Table 2, after tuning sets, the final model obtained: 98.22% of accuracy, 92.84% of precision and 99.15% of recall. The importance of the Naive Bayes model created here is to guide the configuration of new models for different study fields.

We have used K Fold Cross Validation for estimating the performance of the classifier. In k fold cross validation sometimes called rotation estimation the data set D is randomly split into k mutually exclusive subsets the folds $D_1, D_2 .....D_k$ of approximately equal size. The inducer is trained and tested k times; each time t $\in \{1, 2...k\}$, it is trained on $D \setminus D_t$ and tested on D_t. The cross validation estimate of accuracy is the overall number of correct classifications divided by the number of instances in the data set. Formally let $D_{(i)}$ be the test set that includes instance $x_i = (v_i, y_i)$ then the cross validation estimate of accuracy [13].

$$acc_{cv} = \frac{1}{n} \sum_{(v_i,y_i)\in D} \delta(I(D \setminus D(i), v_i), y_i) \qquad (1)$$

For many methods of text analysis, specifically the so called “bag-of-word” approaches, we created a common data structure for the text (Document Term Matrix – DTM) [14, 21, 9]. This is a matrix in which the rows represent references and columns represent terms. The values represent how often each word occurred in each reference. Not all terms are equally informative of the underlying semantic structures of texts, and some terms are rather useless for this purpose. In order to produce text statistics, for instance, the most common terms in the text, we used the Term Frequency Inverse Document Frequency (TFIDF) [9].

TFIDF, is a numerical statistic which indicates how important a term is to a reference in our collection. It is often used as a weighting factor in text mining. The TFIDF value increases proportionally to the number of times a term appears in the reference, but is offset by the frequency of the term in the collection, which helps control the fact that some terms are generally more common than others. Variations of the TFIDF weighting scheme are often used by search engines as a central tool in scoring and ranking a reference's relevance given a user query [9]. TFIDF was successfully used for stop-words filtering and classification. One of the simplest ranking functions is computed by summing the TFIDF for each query term; many more sophisticated ranking functions are variants of this simple model[10, 9].

$$TFIDF(i) = \frac{Frequency(i) * N}{df(i)}, \tag{2}$$

$$\mathrm{R} = (\alpha * \frac{1}{prio}) * (\frac{wordsinwordlist}{totalwords}), \tag{3}$$

where *wordinwordlist* is the frequency of words in all documents, and *totalwords* is the number of words in the collection. Here, we apply the *dictionary-based approach* and create a ranking mechanism to obtain a relevance score (R) for each paper. The relevance score R is calculated in the Equation (3), where a paper is considered more relevant depending on its priority (1, 2 or 3) and the percentage of the most relevant terms are present in its abstract. Finally, we prune the ranked publications set recommending the top 10 most relevant references for a specific search.

4 Conclusion And Future Work

We proposed a model for recommending scientific articles to users based on abstract content using a personal collection of references. In general, building a large amount of labelled training data for text classification is a labour-intensive and time-consuming task. Our study showed that this approach works well considering our initial purpose and make good predictions on recommending scientific articles based on references collection. We believe that our approach have promising results, mainly because it's suitable to be applied in all domains. The results demonstrated the effectiveness and applicability of automated reference classification methods for management and updating a systematic literature review, required in all research project. In future work we will compare our model with Support Vector Machines and Boosting, besides integrating the first model developed previously in [9].

Acknowledgements. The first author thanks Universidade do Estado de Mato Grosso (UNEMAT – Brazil) and Faculdade de Engenharia da Universidade do Porto (FEUP – Portugal), for the support given. The work of Thiago, has been funded through a IBRASIL Grant. IBRASIL is a Full Doctorate programme selected under Erasmus Mundus, Action 2 STRAND 1, Lot 16 and coordinated by University of Lille. Simone developed her postdoctoral research funded by CAPES (Brazil) – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Proc no. BEX 1338/14-5).

References

1 Yindalon Aphinyanaphongs and Constantin F. Aliferis. Text categorization models for retrieval of high quality articles in internal medicine. In *AMIA Annual Symposium Proceedings*, pages 31–5, 2003.
2 David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.
3 David M. Blei and John D. Lafferty. Topic models. In *Text Mining: Classification, Clustering, and Applications*. Chapman & Hall/CRC, 2009.
4 Harris M. Cooper. *The structure of knowledge synthesis*. Knowledge in Society. 1988.
5 Ludovic Denoyer and Patrick Gallinari. Bayesian network model for semi-structured document classification. *Information Processing & Management*, 40(5):807–827, 2004.

6 Tracy Edinger and Aaron M. Cohen. A large-scale analysis of the reasons given for excluding articles that are retrieved by literature search during systematic review. In *AMIA Annual Symposium Proceedings*, pages 379–387, 2013.

7 Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in R. *Journal of Statistical Software*, 25(5):1–54, 2008.

8 Louise Francis and Matthew Flynn. Text mining handbook. In *Casualty Actuarial Society E-Forum*. Casualty Actuarial Society E-Forum, 2010.

9 Carlos A.S.J. Gulo and Thiago R.P.M. Rúbio. Text mining and scientific articles and using the R language. In *Proceedings of the 10th Doctoral Symposium in Informatics Engineering – DSIE*, pages 60–69, Porto, 2015.

10 Andreas Hotho, Andreas Nürnberger, and Gerhard Paab. A brief survey of text mining. *LDV Forum – GLDV Journal for Computational Linguistics and Language Technology*, 2005.

11 D. Isa, L.H. Lee, V. Kallimani, and R. RajKumar. Text document preprocessing with the bayes formula for classification using the support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, 20(9):1264–1272, 2008.

12 Mohammad S. Khorsheed and Abdulmohsen O. Al-Thubaity. Comparative evaluation of text classification techniques using a large diverse arabic dataset. *Language Resources and Evaluation*, 47(2):513–538, 2013.

13 Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI – International Joint Conference on Artificial Intelligence*, pages 1137–1145. Morgan Kaufmann, 1995.

14 Guy Lebanon, Yi Mao, and Joshua Dillon. The locally weighted bag of words framework for document representation. *J. Mach. Learn. Res.*, 8:2405–2441, 2007.

15 S. Massung, ChengXiang Zhai, and J. Hockenmaier. Structural parse tree features for text representation. In *ICSC – International Conference on Semantic Computing*, pages 9–16, 2013.

16 Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 Workshop on Text Learning*, 1999.

17 Chitu Okoli and Kira Schabram. A guide to conducting a systematic literature review of information systems research. *Sprouts: Working Papers on Information Systems*, 10(26), 2010.

18 Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, 2012.

19 M. Mahdi Shafiei and Evangelos E. Milios. A statistical model for topic segmentation and clustering. In Sabine Bergler, editor, *Advances in Artificial Intelligence*, volume 5032, pages 283–295, 2008.

20 Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Knowledge Discovery and Data Mining*, 2011.

21 Sholom M. Weiss, Nitin Indurkhya, and Tong Zhang. *Fundamentals of Predictive Text Mining*. Springer, 2010.

22 Sholom M. Weiss, Nitin Indurkhya, Tong Zhang, and Fred J. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, 2005.

23 Hwanjo Yu, ChengXiang Zhai, and Jiawei Han. Text classification from positive and unlabeled documents. In *Proceedings of the twelfth International Conference on Information and Knowledge Management*, pages 232–239, 2003.

24 Yangchang Zhao. R and data mining: Examples and case studies. In Yangchang Zhao, editor, *R and Data Mining*, pages 1–4. Academic Press, 2013.