Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 219 (2023) 1193–1199

Procedia
Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022

A machine learning approach for mapping and accelerating multiple sclerosis research

António Lopes[a]*, Bruno Amaral[b]

[a]Iscte – Instituto Universitário de Lisboa, Lisboa, Portugal
[b]Lisbon Collective, Lisboa, Portugal

Abstract

The medical field, as many others, is overwhelmed with the amount of research-related information available, such as journal papers, conference proceedings and clinical trials. The task of parsing through all this information to keep up to date with the most recent research findings on their area of expertise is especially difficult for practitioners who must also focus on their clinical duties. Recommender systems can help make decisions and provide relevant information on specific matters, such as for these clinical practitioners looking into which research to prioritize. In this paper, we describe the early work on a machine learning approach, which through an intelligent reinforcement learning approach, maps and recommends research information (papers and clinical trials) specifically for multiple sclerosis research. We tested and evaluated several different machine learning algorithms and present which one is the most promising in developing a complete and efficient model for recommending relevant multiple sclerosis research.

Keywords: machine learning; recommender systems; multiple-sclerosis; artificial intelligence; research information;

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: alsl@iscte-iul.pt

10.1016/j.procs.2023.01.401

1. Introduction

The amount of research in the health sciences is staggering and overwhelming for researchers trying to keep up with the most recent and relevant papers and studies for their respective areas. As depicted in Fig. 1, the number of indexed papers in Scopus for the research areas of medicine, neuroscience and pharmacology alone are now surpassing the 1 million mark per year. The uptick in 2020 and 2021 can be naturally explained by the important reaction to the COVID-19 pandemic, but the trend before that period was already indicative of the overwhelming amount of research done in these areas.

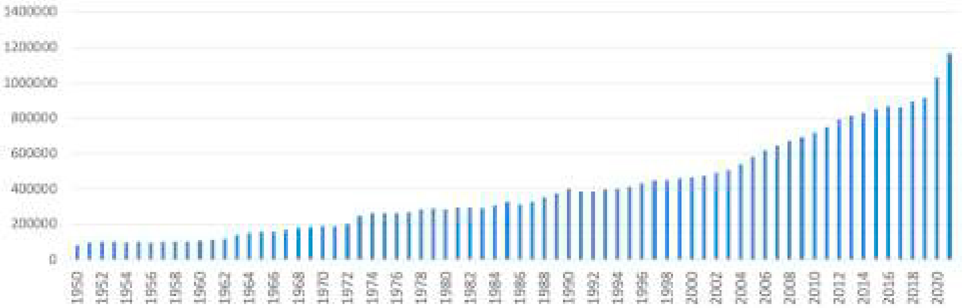

Fig. 1. Number of indexed papers in Scopus in the areas of Medicine/Neuroscience/Pharmacology

The inability to accompany the research evolution in these areas is especially difficult for practitioners that also must deal with their clinical duties, such as overseeing patient care. This raises the importance of having an intelligent system that can help researchers parse through enormous amounts of research material to find and prioritize relevant papers and studies. Artificial intelligence-based (AI) recommender systems (RS) can play this role by leveraging the power of machine-learning (ML) algorithms to process very large sets of papers (and other research-related information) and rank them in order of relevancy to a particular research area.

In this paper, we present a case study of such an approach applied to the field of multiple sclerosis (MS). Using a ML algorithm, we developed a supervised learning process to build a model that can determine the relevancy of research items (papers and clinical trials) in the field of MS. This model was then integrated into a recommender system that is responsible for collecting the data from multiple sources, aggregate this data into an analysis pipeline, and classify the research items to feed a website that researchers, physicians, practitioners and even patients can use to access the relevant information for their work on MS.

The recommender system, called Gregory-MS (https://gregory-ms.com/), is also continually improving through a reinforcement learning process. Every few days, the admin team reviews the recommendations from the system to determine if they are adequate, and then tags the articles accordingly. The goal is to continuously improve the ML model to focus on improving and accelerating MS research.

The rest of the paper is structured as follows: Section 2 provides the scope for this work and reviews some related work; Section 3 describes the details of the development of the ML model and the Gregory-MS system; Section 4 provides the details on experimental results; and finally we conclude the paper on Section 5, where we discuss the limitations of this work and establish some guidelines for future work.

2. Related Work

The goal of RS is to help end-users determine what is relevant information in the context of a large pool of unknown data. This kind of systems has been present from the beginning of the World Wide Web [9] and is now widely used in e-commerce [3], social networks [5], and several other domains and big companies like Netflix [1].

The basis of a recommender system is the filtering process that can weed out irrelevant data and leave only the relevant data. The way this filtering process can be done to extract or generate knowledge varies depending on what data and resources are available. From that perspective, recommender systems can have the following categories [10]:

- **Collaborative filtering** - when personal user data (previous ratings, logs, preferences, or other actions performed by the user) is used to develop a user profile that represents the basis for the recommendation model.
- **Content-based filtering** - when the RS uses the data itself to extract knowledge that is globally relevant for all (or a subgroup of) end-users.
- **Hybrid filtering** - some combination of the previous two approaches or with other information-based approach (like using graphs or feature extraction [6][11]).

A survey of research-paper RS [2] shows that Content-based filtering was used in 55% of the reviewed articles, and only 18% of the reviewed literature uses collaborative filtering, leaving the rest to a variety of different hybrid approaches. Such a difference in the use of these filtering techniques is easily explained by the fact that using collaborative filtering approaches is dependent on the existence of explicit user ratings or some other kind of user-provided data that are needed to make personalized recommendations. Since many users are reluctant to provide explicit ratings or perform other actions in these systems, collaborative filtering-based recommender systems tend to be difficult to implement. In fact, content-based filtering has been widely used for recommender systems for quite a while now, including in this academic and research recommendation setting.

The research described in [7] and [12] are quite similar and are based on the concept of key-phrase extraction, where the documents tagged by the user (and new documents harvested from different sources, like the web) are both parsed through this key-phrase extraction module to a common representation (n-grams). They then use a matching module to find documents that are related and, thus, can be recommended to the user. The matching module is based on the cosine similarity, which produces similarity values for each category of n-grams to compute a unique score for each paper. The highest score papers are then recommended to the user. N-grams are a great way to build the similarity approach as described in these papers, however, because of their extreme sparsity and the fact that they can only interpret unseen instances with respect to learned training data, they are only well-suited for extremely large amounts of training data.

The authors in [13] describe an approach for a RS for research papers based on a multiple-facet comparison between papers. The described system starts by collecting the input from the user (whether a person's name or a set of keywords). It then uses that input to determine the set of papers to use: if the input is a person's name, the system will use that person's published papers as a basis for comparison; if the input is a set of keywords, the system will choose the top 5 papers from a keyword-based query. The system will then use the text in the title and abstract of a paper, vectorize it and then uses the cosine similarity to find related papers from a list of harvested papers.

In [14], the authors describe an approach in which a user's access logs are used to perform a collaborative filtering approach for ranking-oriented paper recommendations. They also use cosine similarity to find the similarity between the tagged papers and the harvested papers. Using user's logs has the advantage of not having to explicitly request the user to rank some papers (which some users are unwilling to do), since it only requires to monitor the normal user's activity in a certain environment. However, the context of a user activity or the intention behind it may not be ascertained by the user's logs. For example, the fact that the user accessed a particular document does not represent that the document is in fact of value for that user. It is natural that, for the user to evaluate the relevance of the paper to their work, they must in fact access and read the document. But that shouldn't be assessed as being a relevant interaction for classifying other papers, considering that the consulted paper may not be indeed relevant.

As described by the previous approaches, the research on RS for research papers has been evolving, but has not yet leveraged the potential and important step that is the adoption of ML algorithms (as assessed in [10]), which allows to further extend the personalization and recommendation capabilities of such systems.

For the RS to be able to recommend some data to its users, it must have some sort of matching, classification, or prediction algorithm. This algorithm can be created in one (or a mix) of the following ways [10]:

- **Supervised learning** - when algorithms are provided with (human-generated) training data that represents positive and negative examples, and they must generate a model that represents that knowledge.

- **Unsupervised learning** - when algorithms are just given a dataset and must develop some sense of that data on their own without any human input or feedback.
- **Semi-supervised learning** - when algorithms are provided with training data that has some correct and incorrect answers but there's also some missing information, and they must learn the representing model.
- **Reinforcement learning** - when algorithms learn based on external feedback (positive or negative) given either by a thinking entity or the environment over time.

In a review of ML-based RS [10] it is stated that 76% of the reviewed articles use a supervised learning approach and only 22% use an unsupervised learning approach. Our recommender system, Gregory-MS, was built using a supervised content-based filtering machine learning algorithm that uses past classification data (composed of a large dataset of annotated research papers and clinical trials relevant for MS) to generate a model that can perform predictions on future data. Next section details the technical approach used to build the system.

3. Technical Approach

The Gregory-MS system was built to aid researchers, practitioners, physicians, and patients that are interested in keeping up with the most recent and relevant work on MS. The system includes a set of components that enable the complete harvesting, training, and recommending process. This architecture is described in Fig. 2.

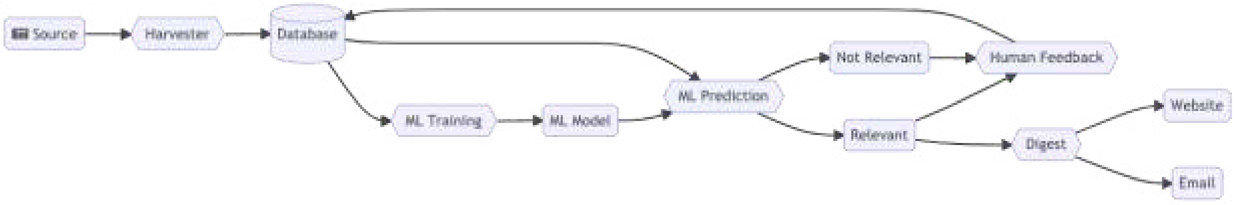

Fig. 2. Gregory-MS system's architecture

The system is composed of five main components:

- **Harvester** - This is the process that is responsible for gathering data from multiple sources like PubMed, BioMedCentral, Sage, Scielo, and many others. It collects the metadata on papers and clinical trials from these sources.
- **ML Training** - This is the machine learning training process, which takes the raw data from the database (and the annotated data from the human feedback process, when available) and generates the ML model that will be used to perform the predictions.
- **ML Prediction** - This is the machine learning prediction process, which takes the newly harvested data and the ML Model and performs predictions on whether this new data is relevant or not.
- **Human Feedback** - This is the basis for the reinforcement learning process. A human user reviews the recommendations done by the ML Prediction phase and provides feedback on whether those recommendations are accurate or not. This in turn feeds the database with newly annotated data that will be used in the next iteration of the ML Training process.
- **Digest** - This is a batching process that generates relevant information for Gregory-MS stakeholders. It generates a newsletter email with the most recent and relevant information and feeds the necessary data that is always available in the main public website.

The main component that it is important to detail is the ML Training process, which is depicted in Fig. 3.

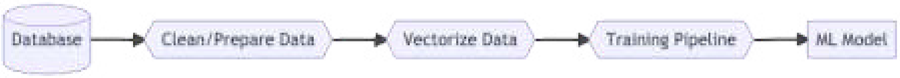

Fig. 3. Detail of the ML Training process

We use a supervised learning algorithm that processes the annotated data (by a human expert) and trains a ML model. See Section 4 for more details on which model we use.

As depicted in Fig. 3, the process begins by cleaning and preparing the raw data before training the model. Besides cleaning unusable characters (like special symbols and data like HTML tags) and *stopwords* (common words that are void of context for this kind of training, such as “the”, “is”, “at”), we also perform *stemming* for all the words in the input text. *Stemming* is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. For example, the words “program”, “programmer” and “programming” are all converted to their base form “program”. This helps reduce the number of different words that the model must process.

The second step of the ML Training process is the tokenization of the input data. Tokenizers are necessary because this kind of ML algorithms only work with numbers, so we must turn the words from the source data into numbers (or tokens). This is done automatically by the tokenizer. During this process, when the terms in the dataset are being converted into the corresponding tokens, the TF-IDF [8] algorithm is used to give more weight to certain terms. TF-IDF is short for Term Frequency - Inverse Document Frequency, and it is a numerical statistic that is intended to reflect how important a word is to a document in the context of a collection of documents. The TF-IDF value increases proportionally to the number of times a word appears in the document, but it is offset by the number of documents that contain the word, which helps to adjust for the fact that some words are just more common than others (and thus do not contribute to determine the relevancy of a single document in relation to the complete list of documents). In a survey of RS [2], it was shown that TF-IDF was the most applied weighting scheme, accounting for 70% of the reviewed approaches.

The final step in this ML Training process is the actual training pipeline. At this stage, the process divides the entire dataset into two different datasets: the first one (holding 80% of the records in the original dataset) is the dataset used for training the ML Model; the second one (holding the remaining 20%) is the dataset used for testing the accuracy of the trained ML Model.

Next, given the fact that input data has a very imbalanced set of records (around 22% of the records are annotated as being relevant and 78% not relevant), the training process adjusts this disparity with a randomly *undersampling* of the majority class (”not relevant”), to provide a more equally balanced dataset. Afterwards, the training process applies the actual classifier that is responsible for performing the training and generation of the ML Model.

Next section details the steps that we took to choose the actual algorithm for training the ML Model.

4. Evaluation

To determine which algorithm should be used in the Gregory-MS system, we tested and evaluated several different algorithms using a dataset of more than 8700 research papers related to MS. These papers were harvested by using a simple keyword search for MS and had the following distribution of publication date: 2020 (156), 2021 (6042), 2022 (2573). The following algorithms were tested: Gaussian Naive Bayes, Multinomial Naive Bayes, Linear Support Vector Classification, and Logistic Regression.

Table 1 depicts the average results obtained in a multi-iteration test run of the ML Training process (the process described in Section 3).

Table 1. Evaluation of ML Training algorithms

Algorithm	Accuracy	Recall	Precision	F-Measure
Gaussian Naive Bayes (GNB)	0.688	0.507	0.342	0.408
Multinomial Naive Bayes (MNB)	0.816	0.137	**0.981**	0.240
Linear Support Vector Classification (LSVC)	**0.925**	**0.815**	0.831	**0.823**
Logistic Regression (LR)	0.919	0.745	0.853	0.795

To understand the results, some concepts need to be explained:

- **Accuracy** - This represents the simple ratio of correctly classified samples (between the test dataset and the prediction results).
- **Recall** - This represents the ratio TP / (TP + FN) where TP is the number of true positives and FN the number of false negatives. The *Recall* is intuitively the ability of the classifier to find all the positive samples.
- **Precision** - This metric indicates whether the model can correctly identify all the positive samples without accidentally marking too many negative samples as positive.
- **F-Measure** - This metric can be interpreted as a harmonic mean of the *Precision* and *Recall*, which is calculated with following formula: F = 2 * (precision * recall) / (precision + recall).

For all these metrics, the values vary between 0 (meaning the worst) to 1 (meaning the best), which means a higher score represents a better performance.

When evaluating a set of algorithms in their ability to generate classification models, it is important to use this other group of metrics (especially in unbalanced datasets). In a review of RS [10], it is stated that Precision, Recall and F-measure are among the most popular performance metrics used in the reviewed studies, totalling almost 50% of all occurrences. This is because using only the accuracy metric may not be enough to understand the efficacy of a certain algorithm. This is evident in this case as well.

If we consider the Accuracy metric, we see that the LSVC and LR algorithms are very close in performance. However, accuracy does not consider the disparity between true positives/negatives and false positives/negatives. It is important to consider the remaining metrics to infer which algorithm delivers the best results. From these results, we can conclude that, even though LSVC and LR have similar accuracies, and both generate a similar (but low) number of False Positives, LSVC is better because it is able to generate fewer False Negatives than LR. We have, thus, decided to use LSVC in the Gregory-MS system.

5. Future Work

In this paper, we described the machine learning-based approach to map and predict the relevancy of research data for MS. The described approach applies to the specific scenario of MS, but it can, in fact, be applied to other areas, depending solely on the existence of annotated data for each different research area.

Although these preliminary results are promising, there are some limitations with the chosen approach. The fact that the process can only be bootstrapped by an annotated dataset, which represents quite a bit of manual labour for a human with expertise of a particular research area, makes this approach less desirable, especially when it is hard to find clinical practitioners or researchers that are available to provide feedback on the system's recommendations.

With that in mind, we intend to test and evaluate more advanced approaches that do not require initiating this processed with previously annotated data, such as using pre-trained transformers models [4] that allow extracting more semantic and contextual meaning from unlabelled data.

Acknowledgements

The authors would like to acknowledge the important support of Project Galileo (Cloudflare), Lisbon Collective (https://lisboncollective.com/) and OneOverZero (https://github.com/oneoverzero). The authors would also like to thank the reviewers of the paper for helping to create this improved version of the paper.

References

[1] Amatriain X, Basilico J. Recommender systems in industry: A Netflix case study. In Recommender systems handbook 2015 (pp. 385-419). Springer, Boston, MA.

[2] Beel J, Gipp B, Langer S, Breitinger C. Paper recommender systems: a literature survey. International Journal on Digital Libraries. 2016 Nov;17(4):305-38.

[3] Buettner R. Predicting user behavior in electronic markets based on personality-mining in large online social networks. Electronic Markets. 2017 Aug;27(3):247-65.

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[5] Elmongui HG, Mansour R, Morsy H, Khater S, El-Sharkasy A, Ibrahim R. TRUPI: Twitter recommendation based on users' personal interests. InInternational Conference on Intelligent Text Processing and Computational Linguistics 2015 Apr 14 (pp. 272-284). Springer, Cham.

[6] Felfernig A, Le VM, Popescu A, Uta M, Tran TN, Atas M. An Overview of Recommender Systems and Machine Learning in Feature Modeling and Configuration. In15th International Working Conference on Variability Modelling of Software-Intensive Systems 2021 Feb 9 (pp. 1-8).

[7] Ferrara F, Pudota N, Tasso C. A keyphrase-based paper recommender system. In Italian research conference on digital libraries 2011 Jan 20 (pp. 14-25). Springer, Berlin, Heidelberg.

[8] Jones KS. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation. 1972.

[9] Pazzani M, Billsus D. Learning and revising user profiles: The identification of interesting web sites. Machine learning. 1997 Jun;27(3):313-31.

[10] Portugal I, Alencar P, Cowan D. The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications. 2018 May 1;97:205-27.

[11] Ramzan B, Bajwa IS, Jamil N, Amin RU, Ramzan S, Mirza F, Sarwar N. An intelligent data analysis for recommendation systems using machine learning. Scientific Programming. 2019 Oct 31;2019.

[12] Sugiyama K, Kan MY. Scholarly paper recommendation via user's recent research interests. In Proceedings of the 10th annual joint conference on Digital libraries 2010 Jun 21 (pp. 29-38).

[13] Uchiyama K, Nanba H, Aizawa A, Sagara T. OSUSUME: cross-lingual recommender system for research papers. In Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation 2011 Feb 13 (pp. 39-42).

[14] Yang C, Wei B, Wu J, Zhang Y, Zhang L. CARES: a ranking-oriented CADAL recommender system. In Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries 2009 Jun 15 (pp. 203-212).