Paper Selection and Information Extraction in Healthcare
3-cfu Project Work

Riccardo Marvasi, Edoardo Saturno, Lucia Gasperini, Arianna Albertazzi
Master's Degree in Artificial Intelligence, University of Bologna
{ riccardo.marvasi, edoardo.saturno, lucia.gasperini, arianna.albertazzi3 }@studio.unibo.it

Abstract

Systematic reviews are fundamental in synthesizing vast amounts of research literature on specific topics. With the increasing volume of available data, the role of Artificial Intelligence (AI) in conducting systematic reviews has become crucial. This report investigates the application of machine learning techniques for automating the paper screening process in systematic reviews (Girish Sundaram, 2022). The objective is to enhance efficiency and accuracy in identifying relevant papers for inclusion. Traditional approaches relying on manual screening are labor-intensive and prone to human error, prompting the exploration of automated methods. We evaluate the performance of various machine learning models, alongside a *BERT-based mode*l (Jacob Devlin, 2019). Our approach involves pre-processing the dataset, splitting it into training, validation and test sets, and training the models on multiple splits. Findings reveal that while traditional models demonstrate promising performance, the BERT-based model offers a competitive alternative with potential for further improvement.

1 Introduction

The systematic review process plays a pivotal role in synthesizing vast amounts of research literature to inform evidence-based decision-making in various fields (Iain J Marshall, 2018). However, the manual screening of papers for inclusion in systematic reviews can be labor-intensive and time-consuming. Thus, automating or semi-automating this process through machine learning techniques has garnered significant attention due to its potential to enhance efficiency and reduce bias (James Thomas, 2017).

Traditionally, researchers have relied on manual screening by human reviewers to identify relevant papers for inclusion in systematic reviews. While effective, this approach is resource-intensive and prone to human error (Annette M. O'Connor, 2019) (Siddhartha R. Jonnalagadda, 2015). Furthermore, as the volume of scientific literature continues to grow exponentially, the need for more scalable and efficient screening methods becomes increasingly apparent. Recent advancements in natural language processing and machine learning have paved the way for automated screening methods, offering a promising solution to this challenge (Alison O'Mara-Eves, 2015). These methods typically involve training machine learning models on labeled datasets of papers to predict their inclusion status based on features extracted from titles, abstracts, and keywords (Iain J Marshall, 2015).

Traditional machine learning models offer simplicity and interpretability but may struggle with capturing complex patterns in text data. On the other hand, deep learning models like *BERT* can automatically learn intricate patterns from raw text data, potentially improving performance but requiring large amounts of data and computational resources (Jacob Devlin, 2019).

In our study, we conducted a series of experiments to evaluate the effectiveness of both traditional machine learning models and a BERT-based approach for automated paper screening in systematic reviews. We utilized the SYNERGY dataset (De Bruin et al., 2023), to train and evaluate our models. Specifically, we trained some traditional machine learning models, such as *RandomForestClassifier*, *LogisticRegression*, *SVM*, *NB*, and *XGB* on the training set and fine-tuned the BERT model using transfer learning techniques and comparing their performance against the established benchmarks that are between *78%* and *98%* (Gerald Gartlehner, 2019).

For evaluation, we measured the performance of each model on the validation set using standard metrics such as accuracy, precision, recall and F1-score.

2 System description

In our study, we developed and evaluated two distinct systems for automating paper screening in systematic reviews: *traditional machine learning* models and a *BERT-based approach*. In this section, we provide a detailed description of each system, including their architectures and components.

For the traditional machine learning models, we employed several commonly used algorithms, including *RandomForestClassifier*, *LogisticRegression*, *SVM*, *NB* and *XGB*. These models are well-established in the literature for various classification tasks (Breiman, 2001) (Daniel Jurafsky, 2024) (Bernhard Scholkopf) (Vijaykumar B., 2014). In terms of architecture, we followed standard implementations widely used in machine learning literature and we utilized existing libraries such as *scikit-learn* (Fabian Pedregosa, 2011) in Python to implement these models. The core algorithms were not original; however, we tailored the preprocessing steps and evaluation procedures to suit the specific task of automated paper screening in systematic reviews.

While, for the BERT-based approach (Jacob Devlin, 2019) we utilized an architecture which comprises multiple Transformer layers for bidirectional encoding. Each Transformer layer consists of self-attention mechanisms and feed-forward neural networks, allowing the model to effectively capture contextual information from input sequences. BERT's architecture facilitated the extraction of rich semantic representations from textual data, essential for discerning the relevance of papers in systematic reviews.

More specific implementation details are explained in the *Experimental setup and results* section.

3 Data

In this report, we present an analysis of a systematic review dataset made by *Christian Appenzeller-Herzog* in 2020. This dataset is part of a collection of datasets, the SYNERGY dataset, which includes the study selection of 26 systematic reviews.

SYNERGY represents a comprehensive and openly accessible dataset focusing on study selection within systematic reviews, encompassing 169,288 scholarly works across 26 systematic reviews. A mere 2,834 (1.67%) of these academic works are included in the categorized binary dataset of systematic reviews. This distinctive attribute renders the SYNERGY dataset invaluable for the advancement of information retrieval algorithms, particularly for dealing with sparse labels. With an array of variables available per record, such as titles, abstracts, authors, references, and topics, this dataset holds significant utility for researchers in fields spanning NLP, machine learning, network analysis, and beyond. In aggregate, the dataset presents a vast pool of 82,668,134 trainable data points (De Bruin et al., 2023).

Specifically, we conducted our study using a dataset comprising details from 3454 medical papers. This dataset includes information such as the title, abstract, year of publication, authors of the articles, and a binary label indicating whether the article is included (labeled as 1) or not included (labeled as 0) in a systematic review for common therapies related to Wilson's disease. The dataset exhibits a significant imbalance, with only 5.2% of the papers passing the screening process and being considered relevant for the study (*Figure 1*).

This imbalance indicates a scarcity of positive instances (included papers) compared to negative instances (non-included papers) in the dataset. For this reason, the traditional splitting into *training*, *validation* and *test* sets must be handled with caution, as there is a risk of ending up with a validation set that contains very few positive instances, leading to biased evaluation results. Another risk involves encountering an unfortunate split where the training set contains too few included papers, which can reduce models' ability to accurately recognize them. This scenario can lead to models not

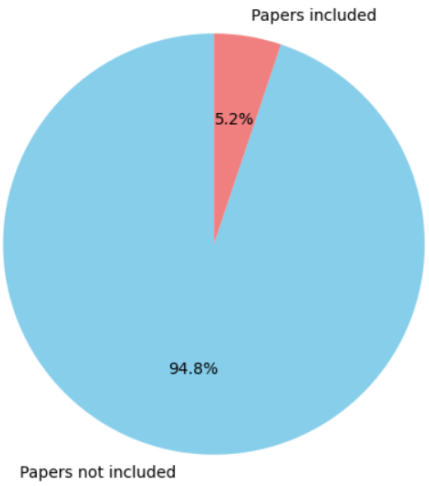

Figure 1: Distribution of the label column

adequately learning the patterns associated with relevant papers. To address this challenge, our study utilized five different splits of the dataset. We trained and evaluated all models on each split individually, and then calculated the average results across all splits. This approach ensures a more robust evaluation and provides a balanced representation of model performance across various dataset configurations.

Another significant issue in the dataset is the presence of a considerable amount of missing data, as illustrated in *Table 1*.

Column	Number of missing data
Title	1
Abstract	1094
Keywords	67
Authors	1
Year	0
Date	1130
Label	0

Table 1: Distribution of the NaNs in the dataset

While some of the missing data points are irrelevant for our task, there are others that significantly impact our work. This is particularly evident in the case of the *1094* missing abstracts, which account for *31.7%* of the dataset. We tackled this issue using two parallel approaches:

1. In the first approach, we replaced the NaN values with a label " *Missing*" and continued with the study. This allowed us to include all data points in the analysis, even those with missing information.

2. In the second approach, we removed all papers where the abstract was missing. This approach aimed to create models whose evaluation performances were not influenced by a significant portion of missing important data.

In both cases we considered as relevant columns just the ones containing the title, the abstract and the keywords of the articles, concatenating them in a new column. Apart from handling NaN values and removing non-relevant features, we did not make any other modifications to the original dataset. All other pre-processing steps were "model-specific" and are detailed in the next section.

4 Experimental setup and results

In order to fully assess the efficacy of different techniques and ensure a complete comparison between various architectures, we have defined and evaluated multiple models. In particular, our study was structured into two distinct parts. The first part focused on analyzing the performances achieved by models utilizing traditional sparse representations for documents. In contrast, the second part delved into an analysis of a model leveraging dense data representation, specifically a BERT model.

4.1 Models exploiting sparse vectors

As the primary methodology in our study, we investigated five classic machine learning models. Each of these models exploits a sparse vector representation created through *TF-IDF* encoding. The models used were

- Random Forest Classifier (RFC)
- Support Vector Machine (SVM)
- Logistic Regressor (LR)
- XGBoost (XGB)
- Naive Bayes (NB)

In the pre-processing pipeline, the column containing the concatenation of abstract, title and keywords of the selected paper undergoes several key steps in order to simplify the classification task of the defined models. Firstly, each token is converted to lowercase to standardize the text and reduce vocabulary size, then, as it is common when using sparse vector representations, common English stopwords (like "*the*," "*is*," etc.) and punctuation marks are removed from the tokenized list, to enhance the focus on meaningful words. After stopwords and punctuation removal, *lemmatization* (reducing words to their base or root form) is performed, again with the aim of normalizing variations of words and reducing feature sparsity.

Once the preprocessing steps are completed, the text data is transformed into a numerical representation using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. At this step, we decided to keep in the TF-IDF matrix just the top 5000 most frequent terms. This choice optimizes the matrix for both efficiency and relevance to the specific classification task at hand, ensuring that the resulting numerical representation captures

meaningful information while minimizing computational overhead and unnecessary sparsity. The presented pipeline transforms raw text data into a format that can be effectively used for training machine learning models, facilitating accurate and efficient text analysis tasks.

The five models previously mentioned are individually trained on the five distinct versions of the training set (as explained in Section 3) and then utilized to generate predictions for the validation set. To ensure a comprehensive and fair comparison, we have employed multiple evaluation metrics, specifically accuracy, precision, recall and F1 score. It's worth noting that due to the limited distribution of included papers relative to the total dataset, the recall metric is particularly informative, as it provides insight into the proportion of correctly identified included papers by the model, making it a crucial measure of model performance in this context. The results presented in *table 2* are the average of the models trained on the five different sets.

	Acc	Pre	Rec	F1
RFC	0.944	0.733	0.093	0.165
LR	0.947	0.843	0.166	0.277
SVM	0.952	0.960	0.201	0.332
NB	0.939	0.000	0.000	0.000
XGB	0.957	0.748	0.423	0.540

Table 2: Results of the models based on tf-idf vectorization

As briefly introduced in *Section 3*, a parallel study was conducted on an alternate version of the dataset where papers without abstracts were excluded. The aim of this second experiment is understanding how much the absence of abstracts impacts on the performance of the models, providing some insights of how much the classification of the model is based on the abstract and how much it is based on the other data (keywords and title).

	Acc	Pre	Rec	F1
RFC	0.955	1.000	0.161	0.277
LR	0.941	0.750	0.097	0.172
SVM	0.950	0.970	0.264	0.415
NB	0.939	0.000	0.000	0.000
XGB	0.957	0.801	0.444	0.571

Table 3: Results of the models on the dataset where papers without abstracts were excluded

4.2 Models exploiting dense vectors

In the second part of our study on the "information extraction" task we developed a BERT model, an architecture able to capture contextual relationships between words in the text, and with the ability of understanding the meaning of words based on their surrounding context. This time the preprocessing pipeline comprehends a tokenization of the input text data which includes padding and truncation after 150 tokens.

The 150 value was determined following a study on the average length of the abstracts, together with balancing resource consumption and performance considerations. Increasing the maximum sequence length (MSL) for the BERT model, in fact, leads to poor generalization. This is probably due to the fact that the model becomes too focused on specific details within the data, especially the abstracts, which are a key component. Furthermore, the increase in MSL also results in longer training times per epoch and requires more epochs to achieve comparable results to previous settings of MSL.

Additionally, class weights are computed based on label distribution for model training, in order to partially compensate for its imbalanced nature. The core of the architecture are the BERT (*BERT-based uncased*, in particular) embeddings obtained transforming the tokenization of the input data, followed by a linear classifier for binary classification. The model is optimized using the "*AdamW*" optimizer (with a starting learning rate of $1e-5$) and the weighted binary cross-entropy loss, considering the over-mentioned computed class weights. The dense representations are obtained after three epochs of training, each with 32 as batch size. *Figure 2* represents graphically the architecture of the model.

Given that the initial parameters of the BERT model significantly impact the final outcome, we utilized five different seeds, in order to consider multiple initial configurations. This approach resulted in the creation of five distinct models, all sharing the same architecture but differing in their initialization parameters. Again, in order to guarantee a robust evaluation, the model is trained and tested on five different versions of the training set. The results reported in *table 4* are the average, for each model, on the five variants.

Following the same reasoning made in *section 4.1*, a parallel study was conducted on an alternate version of the dataset where papers without ab-

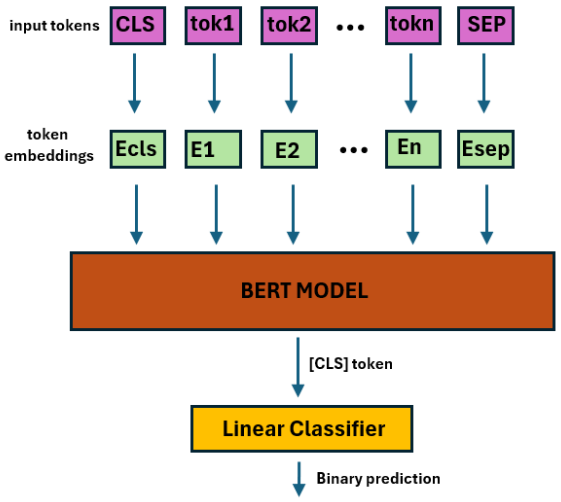

Figure 2: Graphical representation of the architecture of the model

	Acc	Pre	Rec	F1
BERT1	0.951	0.608	0.434	0.506
BERT2	0.947	0.540	0.434	0.481
BERT3	0.944	0.626	0.528	0.573
BERT4	0.945	0.573	0.374	0.453
BERT5	0.949	0.475	0.470	0.472

Table 4: Results of the BERT model

stracts were excluded. The new results are reported in *table 5*.

	Acc	Pre	Rec	F1
BERT1	0.937	0.514	0.529	0.521
BERT2	0.953	0.768	0.440	0.559
BERT3	0.947	0.639	0.513	0.569
BERT4	0.942	0.574	0.553	0.575
BERT5	0.931	0.452	0.409	0.429

Table 5: Results of the BERT model on the dataset where papers without abstracts were excluded

5 Discussion

As we can see from the presented result, the only baseline that exhibits stable and coherent performance, providing a meaningful contrast with the BERT model, is the XGB. Notably, the primary discrepancies between the XGB model and other baselines lie in the F1 score and recall value. This baseline model demonstrates a commendable ability to maintain a substantial number of true positives while minimizing false positives. In the other cases the inner working of the baseline is not complex enough to learn how to properly classify the data: this can be observed by looking at the recall column which has values that reach zero.

Moving on the analysis of BERT models we can see that, on one hand, they are able to reach a discrete performance but, on the other, they cannot clearly beat the XGB model. Specifically, only specific initialization of the BERT models applied on particular datasets manage to outperform XGB. This is not compatible with a straightforward usage of BERT with respect to the baselines in this particular task. In fact BERT requires quite some time to be trained and the time is not always an infinite resource while developing application in NLP. One of the main reasons why BERT fails to consistently outperform XGB can be attributed to the relatively small amount of data present in the dataset. BERT architectures excel in distinguishing patterns within vast amounts of data, and due to the limited dataset size, they are unable to fully leverage their capabilities. In fact, by increasing the MSL value it was noted that the model needed more and more epochs to reach the old performance at a lower MSL. It was concluded that these were due to the fact that by considering larger chunks of the abstract the data becomes too specific to individual articles and the model is increasingly less able to generalize to unseen data as we increase the MSL. Another important reason behind BERT behaviour can be reconducted to the distribution of labels within the dataset splits. This was initially suspected when certain models exhibited strong performance on specific dataset splits but faltered on others. An analysis of the distribution of labels revealed that only 5% of labels were considered positive: a split can generate a chain reaction were many true labels ends up in the validation and test set, thus setting the training set as not representative on the real distribution of the data. Obviously this can happen only in some dataset split and explain incoherent behaviour of BERT.

Regarding these results an important question arises: how much does the labels distribution affects the performance of the models? As it has been already reported in previous sections many rows of the dataset lack the abstract value and two different training were performed for each model: first with a full dataset and then with a filtered one where the NaN abstract rows were removed. Regarding the baseline models the previously reported tables shows that there is no significant difference

between the two strategies, thus enforcing the idea that baseline models do not rely heavily on label distribution within a dataset. On the other hand regarding the BERT model results, we can see that they are slightly better in the filtered dataset training. This difference hindered the model's learning process since the labels are based on human analysis of the abstract. Consequently, the model struggled to make accurate predictions for these NaN values. With this approach BERT were also able to surpass XGB.

6 Conclusion

In conclusion, the analysis presented in this paper sheds light on the performance of baseline models, particularly the *XGB* model, in comparison to BERT models in a specific task within natural language processing. The *XGB* model emerges as a stable and coherent baseline, showcasing commendable performance, particularly in terms of F1 score and recall value. Its capacity to preserve a significant number of true positives while concurrently minimizing false positives underscores its robustness.

However, the BERT models, while capable of discrete performance, do not consistently surpass the XGB model. This lack of consistent superiority can be attributed to several factors. Firstly, the relatively small dataset size limits the BERT models' ability to fully leverage their capabilities in distinguishing patterns within vast amounts of data. Increasing the MSL parameter exacerbates this issue, leading to decreased generalization ability as the model becomes increasingly specific to individual articles.

Moreover, the distribution of labels within the dataset splits also plays a crucial role in the performance of BERT models. Certain dataset splits may lead to an imbalance in label distribution, affecting the model's ability to learn accurately across all classes. This discrepancy is particularly evident when comparing performance on filtered datasets where NaN abstract rows are removed.

The findings underscore the importance of considering label distribution and dataset size when evaluating model performance in NLP tasks. While baseline models exhibit consistent performance regardless of label distribution, BERT models are more sensitive to such variations, indicating the need for careful dataset pre-processing and model tuning to achieve optimal performance.

Moving forward, there are several promising directions for future research based on the findings and limitations of this study.

One avenue is the exploration of techniques for augmenting datasets to increase their size and diversity. This could involve methods such as data synthesis, domain adaptation, or the integration of additional information sources to enhance model performance.

Another important avenue is increase the size of GPUs used for training BERT. Not only the use of better GPUs can reduce training time but it also make it possible to increase batch size. This will in turn improve the generalization skill of bert models thus leading to better results.

Additionally, researchers could explore ensemble learning approaches that combine predictions from multiple models, including baseline models and BERT variants, to improve overall performance and robustness. Ensemble techniques have the potential to mitigate individual model limitations and provide more reliable predictions across diverse datasets and scenarios.

7 Links to external resources

Appenzeller Herzog Dataset by ASReview team https://raw.githubusercontent.com/asreview/systematic-review-datasets/metadata-v1-final/datasets/Appenzeller-Herzog_2020/output/Appenzeller-Herzog_2020.csv

Multinomial Naive Bayes' For Documents Classification and Natural Language Processing by Arthur V. Ratz https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-

References

John McNaught Makoto Miwa Sophia Ananiadou Alison O'Mara-Eves, James Thomas. 2015. Using text mining for study identification in systematic reviews: a systematic review of current approaches.

James Thomas Paul Glasziou Stephen B. Gilbert-Brian Hutton Annette M. O'Connor, Guy Tsafnat. 2019. A question of trust: can we build an evidence-based future? bmc medical research methodology.

Alex Smola John Shawe-Taylort John Platt Bernhard Scholkopf, Robert Williamson§. Support vector method for novelty detection.

Leo Breiman. 2001. Random forest. *Machine Learning*, 45:5–32.

James H. Martin Daniel Jurafsky. 2024. Logistic regression. *Speech and Language Processing*.

Jonathan De Bruin, Yongchao Ma, Gerbrich Ferdinands, Jelle Teijema, and Rens Van de Schoot. 2023. SYNERGY - Open machine learning dataset on study selection in systematic reviews.

Alexandre Gramfort Vincent Michel-Bertrand Thirion Olivier Grisel-Mathieu Blondel Peter Prettenhofer Ron Weiss Vincent Dubourg Jake Vanderplas Alexandre Passos David Cournapeau Matthieu Brucher Matthieu Perrot Edouard Duchesnay Fabian Pedregosa, Gael Varoquaux. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12.

Linda Lux Lisa Affengruber-Andreea Dobrescu Angela Kaminski-Hartenthaler Meera Viswanathan Gerald Gartlehner, Gernot Wagner. 2019. Assessing the accuracy of machine-assisted abstract screening with distillerai: a user study.

Daniel Berleant Girish Sundaram. 2022. Automating systematic literature reviews with natural language processing and text mining: a systematic literature review.

Byron C Wallace Iain J Marshall, Joël Kuiper. 2015. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials.

Byron C Wallace-Jon Brassey James Thomas Iain J Marshall, Rachel Marshall. 2018. Rapid reviews may produce different results to systematic reviews: a meta-epidemiological study.

Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Iain Marshall Byron Wallace Steven McDonald Chris Mavergames Paul Glasziou Ian Shemilt Anneliese Synnot Tari Turner Julian Elliott; Living Systematic Review Network James Thomas, Anna Noel-Storr. 2017. Living systematic reviews: 2. combining human and machine effort.

Mark D. Huffman Siddhartha R. Jonnalagadda, Pawan Goyal. 2015. Automating data extraction in systematic reviews: a systematic review.

Trilochan Vijaykumar B., Vikramkumar. 2014. Bayes and naive-bayes classifier.