Exploration of (deep) neural network architectures in systematic reviews with ASReview

Bart-Jan Boverhof*, Ayoub Bagheri (supervision) , and Rens van de Schoot (supervision)

*bjboverhof@gmail.com

1 Introduction

Systematic Reviews are "top of the bill" in research. The number of scientific studies are increasing exponentially in many scholarly fields. Performing a sound systematic review is a time-consuming and sometimes boring task. The ASReview software is designed to accelerate the step of screening abstracts and titles with a minimum of papers to be read by a human with no or very few false negatives[1].

The current models implemented in the ASReview software are mostly relatively simple classifiers, such as for example the Naive Bayes (NB) classifier, oftentimes combined with term frequency-inverse document frequency (TFIDF) feature extraction. Out of a combination of various machine learning models and feature extraction strategies, the latter combination was found to perform the most optimal, and was specified as default in the ASReview software consequently[2]. Recent research by the ASReview team has mostly been focused elsewhere than contrasting different models. As a consequence, the currently implemented models mostly constitute relatively simple machine-learning classifiers, such as the aforementioned NB or for example a Logistic regression classifiers.

Only recently, research has been initiated by the ASReview team in order to explore the implementation of more complex models, such as for example (convolutional) neural networks. The current research builds upon this groundwork, amongst others laid out by Jelle Teijema[3], who explored the feasibility of a convolutional neural network (CNN) architecture, as well as the utility of a different feature extraction strategy for a large dataset of about 50.000 records. Again, the NB TFIDF combination was found to perform very adequately, however, was found to be slightly outperformed by a CNN with Doc2Vec feature extraction[3]. This gain in performance, although still of relatively modest size, imparts that CNN's are at least able to slightly outperform NB on a big dataset, and calls for some further exploration as a consequence.

In the current project I aim to take some further steps, and explore if and how deep-learning models may enhance the performance of a systematic review with ASReview. The central research questions of this project is: *Are (convolutional) neural networks able to improve the performance of a systematic review with ASReview over classical machine-learning models on a dataset of 50.000 records.* It should furthermore be noted that the objective of this project is not to perform an all-encompassing simulation study: this lies outside the scope of the context in which this project was carried out (i.e. an 7.5 ECTS internship). Hence, this report should not be interpreted as an exhaustive investigation, but rather as an initial exploration. My focus is on proposing an exemplary use-case of (various) neural network architecture applications and concepts, and in addition explore some other potentially interesting avenues, such as switching models during the screening process and hyperparameter optimisation. In addition, with the current report I don't aim to provide a publishable scientific research paper, but rather a concise and relatively informal overview of my explorations.

2 Methodology

This section provides a description of the utilised dataset, software and the explored model architectures.

2.1 Dataset

The utilised dataset for utilised for the assessment of model performances is systematic review on depressive relapse and recurrence of including a total of 50.936 paper abstracts, out of which 63 were selected as relevant[4]. The dataset is characterised by the relatively big size of about 50.000, which may be considered quite substantial for a systematic review. Since neural networks constitute a relatively large number of parameters, performance generally depends on the size of the datasets, for which the utilised dataset is an eligible choice in this project.

2.2 Simulation Study Setup

An overview of all investigated models is shown in Table 1. The subsequent sections will elaborate further on these included models and their concepts.

Table 1. Overview of contrasted models

	Model	Feature Extraction
Baseline Models	Naive Bayes	TFIDF (Default)
	Logistic Regression	TFIDF (Default)
	Naive Bayes	TFIDF 5000
	Logistic Regression	TFIDF 5000
Neural Network	Convolutional Neural Network	Doc2Vec
	Naive Bayes + Convolutional Neural Network	TFIDF 5000
	Naive Bayes + Convolutional Neural Network	TFIDF 5000 + Doc2Vec
	Naive Bayes + Cnn Literature: Convolutional Neural Network[5]	TFIDF 5000 + Doc2Vec

2.2.1 Baseline models

In this simulation study, various newly constructed models are proposed in addition to the currently implemented ASReview default models. Both the default Naive Bayes and Logistic Regression (LR) classifiers with the default TFIDF feature extraction are included. The performance of these models will function as baseline to compare the performances of the newly constructed neural networks against. In addition to these two combinations, a NB and LR classifier with a modified version of the TFIDF feature extraction algorithm (only including the top 5000 occurring words) are implemented as well (see also Section 2.2.5).

2.2.2 (Convolutional) Neural Networks

The explored neural networks are all CNN's. The CNN architecture consists of four to six convolutional blocks. The network depth (i.e. amount of convolutional blocks) is determined dynamically within the simulation run, again every 300 iterations by means of hyperparameter optimisation (see also Section 2.2.3). A schematic overview of the possible network architectures is depicted as Figure 1. Each block consists of a convolutional layer with ReLU activation, in somce cases followed by a max pooling, and closed with a dropout layer. (Max) Pooling layers are commonly employed in CNN's, usually succeeding a convolutional layer with the purpose of reducing dimensionality. The objective of such layers are to down-sample features into a more compact space, hereby only retaining essential information, and thus omitting redundant information[6]. Dropout layers drop a randomly-selected number of neurons and their connections throughout the training process. More specifically, within each full training run through the data, i.e. each epoch a different randomly selected set of neurons are fixed to be inactive. Doing so helps to reduce overfitting, particularly for complex network architectures[7].

2.2.3 Hyperparameter optimisation

Substantial improvements in neural network performance have been attained by utilizing hyperparameter optimisation (HPO), especially in the case of CNN's[8]. The amount of layers, the amount of neurons and the amount of filters are arbitrarily choices that are required to be specified when designing a neural network architecture. The optimal network architecture may be strongly dependent upon a specific situation at hand, and is therefore likely to differ across circumstances. Even so, within a single systematic review the optimal network architecture may also differ: at different stages of the systematic review, the amount of data being processed differs. I.e., at the start of a systematic review the amount of records being processed is quite low, whereas nearing the end it is naturally much bigger. Theoretically it could be that, as only few records are being processed at the start of the review, a relatively simple network architecture is most appropriate. On the other hand, near the end of a systematic review as a relatively large amount of data is being processed, a more complex and sophisticated network architecture might be beneficial.

It is due to the aforementioned that after each at the start of the systematic review, and subsequently after each 300 iterations a new HPO run is conducted, allowing for different CNN architectures within a single systematic review. Optimisation is done by minimising training loss. The obtained set of hyperparameters are utilised in the subsequent 300 iterations, after which a new HPO run is initialised. Each run consists of a total of 80 trials of a maximum of 75 epochs. HPO is conducted with the Optuna library for Python[9].

2.2.4 Model switching

In the aforementioned research by Teijema (2021) it was shown that a the performance within a systematic review can, at least be slightly, improved by starting out with a relatively simple model and switching to a more complex model later in the

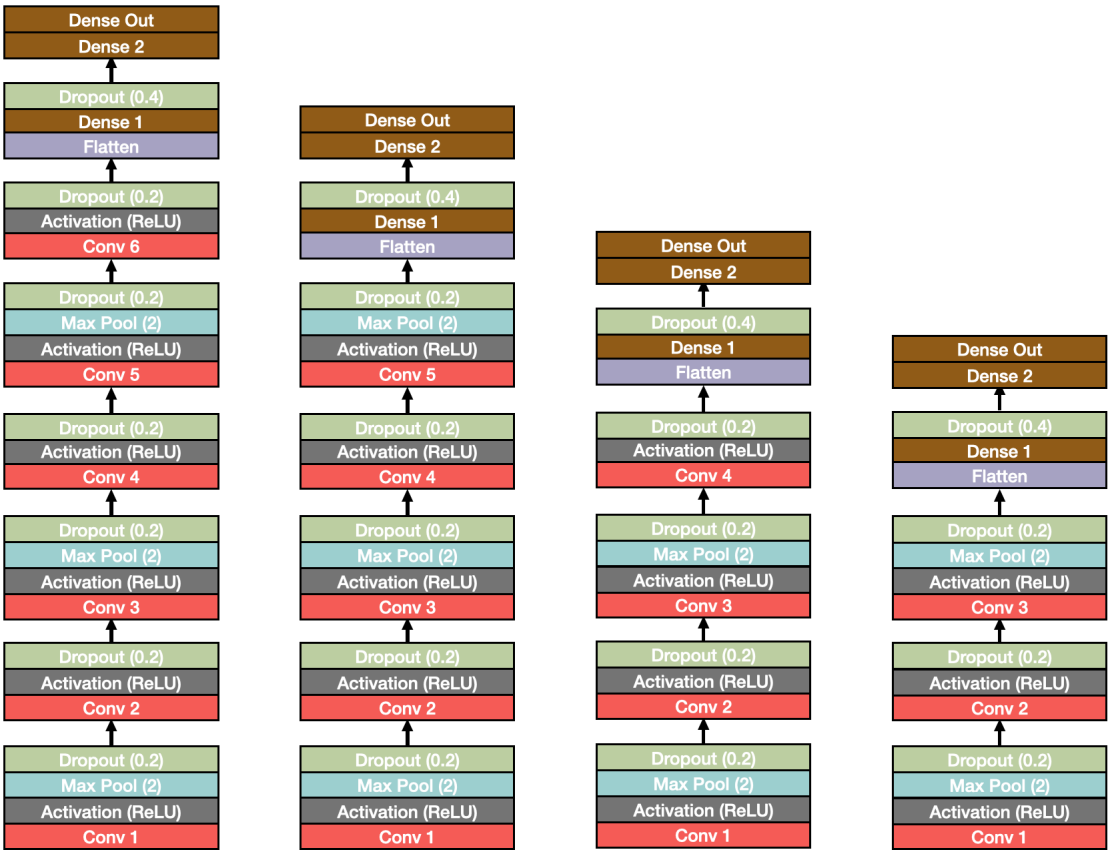

Figure 1. Possible architectures of the newly proposed Convolutional Neural Network

process[5]. I refer to this model as the CNN literature model. Indeed, the combination between a NB - CNN switch was found to perform best. Consequently, a number of four different models that combine NB and a CNN are implemented. The switchpoint is set at 500 iterations, which implies that for all NB + CNN combinations, the first 500 records are being obtained by means of NB, whereas the subsequent records are found with the respective CNN. The underlying rationale is that a CNN may be more able to detect those records as compared with a more simplistic model, however it firstly requires a substantial amount of data before being able to.

2.2.5 Feature Extraction

In the previously mentioned simulation study by Ferdinands (2020) it was shown that the Logistic and Naive Bayes classifier both perform most optimal in combination with TFIDF feature extraction, for which the specified baseline models all make use of TFIDF. The vocabulary size of the default TFIDF applications equals 81234 for this specific dataset. In addition to the default TFIDF as implemented in ASReview, an alternative version is implemented only including the top 5000 most frequently appearing words, referred to as TFIDF 5000. The reason for doing so is to decrease running time and memory issues for the CNN implementations that make use of TFIDF.

In addition, several of the implemented neural networks use Doc2Vec feature extraction. The Gensim Doc2Vec[10] feature extraction was implemented for ASReview by Teijema (2021)[11]. It should be noted that it is currently impossible to specify different feature extraction strategies within a single systematic review with ASReview. Consequently, the switchers that initiated with a Naive Bayes TFIDF strategy were ran as a separate model first, after which all proposed records until that point were specified as priors to the CNN. Though this is not an elegant and satisfying end solution for making use of different models and feature extraction strategies within a single systematic review, it does enable us to explore the merit of such an approach.

3 Results

3.1 Baseline models

Results of the four baseline models are depicted as Figure 2.

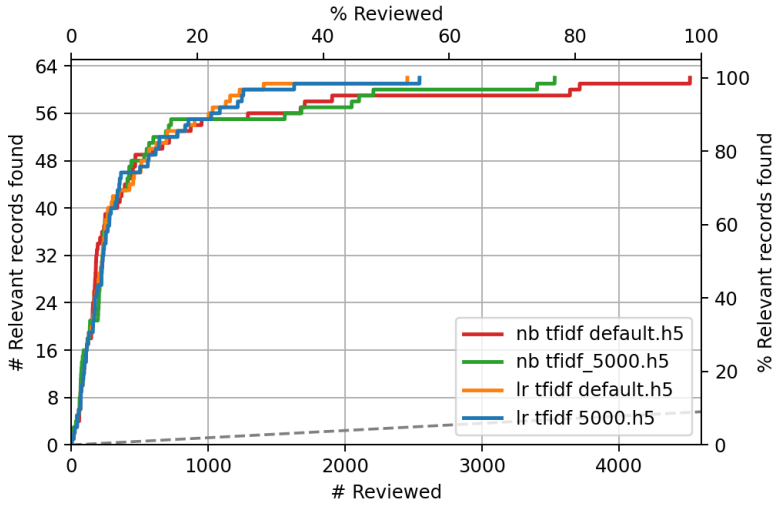

Figure 2. Baseline model results

It can be discerned that until roughly 1000 iterations all baseline models perform relatively equal. After this point performance diverges. Both LR models outperform the NB models, however there is no noticeable difference in between both LR models. Both models are observed to have found all relevant records after around 2500 reviewed records.

As noted, both NB models show worse performance when contrasted against the LR models. The NB model with the default TFIDF is observed to have found all relevant records after having reviewed roughly 4500. The NB model with the newly proposed TFIDF of maximum of 5000 words is observed to performs better, and is shown to have found all relevant records in roughly 1000 records less as compared with the NB with default TFIDF.

3.2 CNN model switchers

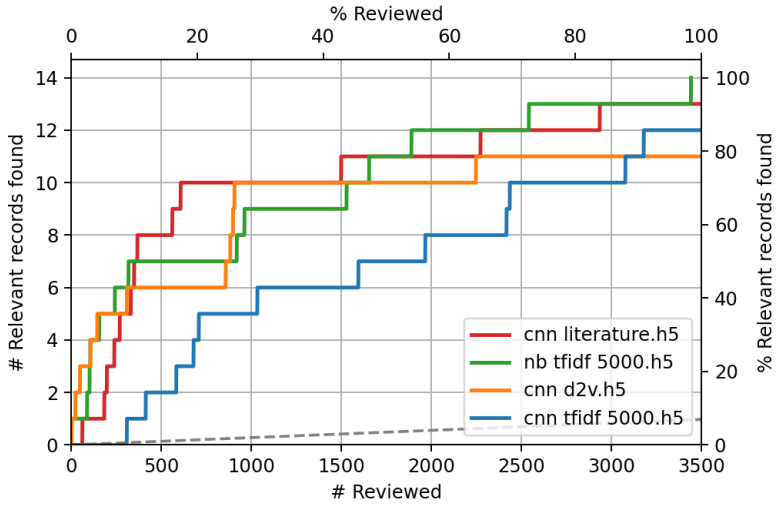

Figure 3. Convolutional Neural Network results

Results of the four CNN's are depicted in Figure 3. It should be noted that the first 500 iterations, which have been ran with the NB TFIDF 5000, are omitted from the plot, and thus the bottom x-axis refers to the amount of reviewed records after the switchpoint. This implies that 49 out of 63 records are found within the first model, whereas the latter 14 remain to be found with the second model, the latter which is solely visualised. The full part of the NB model with TFIDF 5000 after 500 iterations is also included for reference. Due to hardware and time limitations, only a constrained number of iterations have been ran.

It becomes apparent that the NB TFIDF 5000 only model outperforms all specified NB + CNN switch models. The latter is the only model able to find all relevant records in under 4000 iterations. Out of the aforementioned, the literature based model as developed by Teijema (2021) performs slightly better as compared with the other CNN's, however, all are unable to find all relevant records within the specified number of iterations.

4 Conclusion & Discussion

Out of the baseline models, the specified Logistic Regression models were observed to outperform the Naive Bayes models by a substantial margin. No noticeable difference in performance was found when contrasting the the default TFIDF LR model with the TFIDF maximum 5000 words LR model. Quite a substantial difference in terms of performance was however found between both NB models, in which the one with TFIDF 5000 was found to outperform the default TFIDF.

None of the models switching from NB TFIDF 5000 to a CNN were found to outperform the classical machine learning models. Subsequently we conclude that no evidence is found in favor of using one of the specified hybrid models in the light of the current dataset.

4.1 Discussion & Future Research

The inability of the hybrid models included in this study to outperform the classical ML models, however, does not necessarily imply that neural networks should be ruled out alltogether. Several promising avenues avenues still remain unexplored in the current study, most of which may be ascribed toward the lack of time that was inherent to this project. The following sections will propose some of these avenues for future research.

One unexplored avenue is the utility of a pre-trained word embedding. Especially deep neural networks may benefit from utilising pre-trained word embeddings, for example in combination with Word2Vec feature extraction. There are at least two major advantages when using such an approach. First and foremost, using pretrained word embeddings may help increase performance. This gain in performance could be of substantial size, and may help neural networks to challenge conventional machine learning models in terms of performance. Another advantage of using pretrained word embeddings would be a decrease in training time. Training a word embedding from scratch is very computationally demanding, and consequently takes a lot of time. Since time is of the essence in a systematic review, it is important to reduce training times where possible, and using pre-trained word embeddings will contribute to this.

Another potentially interesting avenue of inquiry is into the feature extraction strategy. The current default implemented in ASReview is TFIDF, including every word occurrence throughout the entire dataset. When working with a big dataset such as in the current study, this results in astronomically sized data objects, and memory issues arise when making use of more complex models such as neural networks. These memory issues were successfully solved by implementing TFIDF feature extraction of solely the 5000 most occurring words. In addition to this, it has been shown that this modified feature extraction strategy yielded a substantial increase in performance for the ASReview default model (i.e. Naive Bayes). It would therefore interesting to further enquire into the already implemented feature extraction strategies, and hereby learn more about why and when such differences in performance appear.

Lastly, in order for using a switcher switcher effectively, it is beneficial to explore the possibility of feature extraction switching as well. The reason for this is that different models work best with different feature extraction strategies: for example, it was found in this study classical machine models such as the Naive Bayes performs more optimal with TFIDF feature extraction, whereas neural networks perform more optimal with Doc2Vec. Having the option for switching feature extraction on the fly will open up many possibilities with regards to utilising different models in one systematic review. It would become possible to use several models in one review, each most suitable to handle a specific point within the systematic review. The ability for each of these models to make use of the most optimal feature extraction may be a precursor for the aforementioned approach to work, and may therefore be an interesting avenue for future research.

References

1. van de Schoot, R. *et al.* An open source machine learning framework for efficient and transparent systematic reviews. *Nat. Mach. Intell.* **3**, 125–133 (2021).
2. Ferdinands, G. Active learning for efficient systematic reviews. (2020).
3. Teijema, J. Asreview-study-model-switching. (2021).
4. Brouwer, M. E. *et al.* Psychological theories of depressive relapse and recurrence: A systematic review and meta-analysis of prospective studies. *Clin. psychology review* **74**, 101773 (2019).
5. Teijema, J. Asreview cnn 17 layer model plugin, DOI: 10.5281/zenodo.5084887 (2021).
6. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521**, 436–444, DOI: 10.1038/nature14539 (2015).
7. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal machine learning research* **15**, 1929–1958 (2014).
8. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *The J. Mach. Learn. Res.* **13**, 281–305 (2012).
9. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A next-generation hyperparameter optimization framework. 2623–2631, DOI: 10.1145/3292500.3330701 (2019).
10. Rehurek, R. & Sojka, P. Gensim–python framework for vector space modelling. *NLP Centre, Fac. Informatics, Masaryk Univ. Brno, Czech Repub.* **3** (2011).
11. Teijema, J. Asreview wide doc2vec plugin, DOI: 10.5281/zenodo.5084877 (2021).