Expert Systems With Applications 182 (2021) 115261

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A decision support system for automating document retrieval and citation screening

Raymon van Dinter [a,1], Cagatay Catal [b,*], Bedir Tekinerdogan [a]

[a] Information Technology Group, Wageningen University & Research, Wageningen, The Netherlands
[b] Computer Science and Engineering, Qatar University, Doha, Qatar

ARTICLE INFO

Keywords:
Systematic literature review (SLR)
Citation screening
Document retrieval
Decision support
Automation
Deep learning
Convolutional neural network
Natural language processing

ABSTRACT

The systematic literature review (SLR) process includes several steps to collect secondary data and analyze it to answer research questions. In this context, the document retrieval and primary study selection steps are heavily intertwined and known for their repetitiveness, high human workload, and difficulty identifying all relevant literature. This study aims to reduce human workload and error of the document retrieval and primary study selection processes using a decision support system (DSS). An open-source DSS is proposed that supports the document retrieval step, dataset preprocessing, and citation classification. The DSS is domain-independent, as it has proven to carefully select an article's relevance based solely on the title and abstract. These features can be consistently retrieved from scientific database APIs. Additionally, the DSS is designed to run in the cloud without any required programming knowledge for reviewers. A Multi-Channel CNN architecture is implemented to support the citation screening process. With the provided DSS, reviewers can fill in their search strategy and manually label only a subset of the citations. The remaining unlabeled citations are automatically classified and sorted based on probability. It was shown that for four out of five review datasets, the DSS's use achieved significant workload savings of at least 10%. The cross-validation results show that the system provides consistent results up to 88.3% of work saved during citation screening. In two cases, our model yielded a better performance over the benchmark review datasets. As such, the proposed approach can assist the development of systematic literature reviews independent of the domain. The proposed DSS is effective and can substantially decrease the document retrieval and citation screening steps' workload and error rate.

1. Introduction

A systematic literature review (SLR) is a means of identifying, evaluating, and synthesizing all available research relevant to a particular research question, or topic area, or phenomenon of interest (Kitchenham & Charters, 2007). Kitchenham and Charters (2007) proposed a guideline where the SLR process consists of twelve steps to increase rigor and reproducibility. However, as the literature published is proliferating, the manual production of systematic reviews requiring increased human workload. With a median of 8 months after the last search, a systematic review is often outdated before publication as they take so much time to produce (Beller, Chen, Wang, & Glasziou, 2013). Furthermore, Michelson and Reuter (2019) calculated the financial cost of an SLR study. They also provided a total time estimate to complete a systematic review, which was found to take 1.72 years for a single scientific reviewer. A single review would cost $141,194.80. On average, the total cost of all SLRs per year to each of the ten major academic institution amounts to $18,660,304.77, and for each pharmaceutical company is $16,761,234.71. They also called for action to develop automation tools to speed up the SLR process since the high human workload and cost of a systematic review may pose a barrier to their consistent application to carefully assess the promise of studies.

Research identification is an essential step in the SLR process, including two substeps: developing a search string and document retrieval. The development of a search string aims to gather all relevant literature (e.g., high recall) while excluding as much irrelevant literature (e.g., high precision) as possible. Subsequently, document retrieval aims to collect all literature that matches a search string. As every publication venue has developed its custom search engine, reviewers need to input customized search strings into every publication venue they want to

* Corresponding author.
E-mail addresses: raymon.vandinter@wur.nl (R. van Dinter), ccatal@qu.edu.qa (C. Catal), bedir.tekinerdogan@wur.nl (B. Tekinerdogan).
[1] https://orcid.org/0000-0002-1811-8803.

https://doi.org/10.1016/j.eswa.2021.115261
Received 15 March 2021; Received in revised form 13 May 2021; Accepted 19 May 2021
Available online 25 May 2021

include in the review. Additionally, reviewers must manually collect incomplete data fields for data extraction, which is a tedious and time-consuming process.

The primary study selection process attempts to identify the critical relevant literature fairly and with high rigor using study selection criteria. An experienced reviewer is estimated to screen up to two articles per minute (Wallace, Trikalinos, Lau, Brodley, & Schmid, 2010). A survey by Marshall (2016) showed that reviewers experience the primary study selection process as highly time-consuming and error-prone because the process is highly repetitive; a reviewer must read each title, abstract, and full-text of hundreds to thousands of citations. To this end, the reduction of the error rate of the document retrieval is aimed at in our research.

Fig. 1 shows the workflow of these two steps, described by Kitchenham and Charters (2007) and Tsafnat et al. (2014). To illustrate the workflow, we can call the following example. Reviewers often perform a database search, which is part of the document retrieval step. Afterward, they select relevant articles based on the title, which is part of the study selection process. When the relevant articles – selected based on the title – are identified, they are exported to a reference manager. Additional data is collected, which is part of the document retrieval step. At last, the definitive selection of included articles is selected based on abstract, which is part of the primary study selection step. We can see that identifying research and selecting primary studies is a nonchronological process, as the steps blend to achieve the highest efficiency.

To cope with the challenges above, an open-source decision support system (DSS) is proposed that supports the document retrieval step, preprocessing of the dataset, and citation classification.

Our research questions are formulated as follows:

- RQ-1: What is the feasible approach to design a Decision Support System for the automation of SLRs independent of the application domain?
- RQ-2: How applicable is Deep Learning to support the automation of the document retrieval and citation screening process for SLRs?

With the provided DSS, reviewers can fill in their search strategy and manually label only a subset of the citations. In contrast, the remaining unlabeled citations are automatically classified and sorted based on probability. Our approach aims to be accessible to anyone that wants to review scientific literature. Our DSS uses data available in all research domains, and we designed the system to run in the cloud for free and without any required programming knowledge.

Using the DSS, a generic search query and strategy can be used to retrieve documents that match the search query in selected scientific databases. We provided support for PubMed, ScienceDirect, and SpringerLink. We have chosen to automate document retrieval for these three databases, as they are widely used in Medicine and Software Engineering (van Dinter, Tekinerdogan, & Catal, 2021) and include more high-quality articles compared to the other databases. Furthermore, all three databases have been used extensively in the Python community. The document retrieval of these three databases shows that the concept is viable and can be iteratively updated with additional support for other databases.

As the selection of primary studies provides a binary output (i.e., *included* and *excluded*), we use natural language processing and machine learning algorithms to semi-automate the decision-making process. In this process, natural language processing is used to generate features from textual data, where machine learning uses these features to find patterns in the data. This often significantly reduces the time required on reading hundreds to thousands of titles and abstracts, which leaves more time for the synthesis of literature and timely publication. Even though the presented citation screening model uses classification techniques, we export the complete list of citations sorted based on their probability of inclusion. Through this method, reviewers can still view all literature, regardless of their inclusion, which provides them security on finding all relevant literature (Howard et al., 2016; Kontonatsios, Spencer, Matthew, & Korkontzelos, 2020). The sorting of citations reduces the screening workload, considering that reviewers need only to perform a study selection on top-ranked citations. In contrast, the bottom citations are automatically excluded from the review by the reviewer (Kontonatsios et al., 2020).

A recent SLR study on the automation of SLR studies shows that most existing semi-automatic citation screening methods adopt document representation techniques, such as bag-of-words and TF-IDF, that rely on words' frequency (van Dinter et al., 2021). Therefore, the feature representation of documents naturally ignores the readily available information on the context of those words. Furthermore, most studies use domain-dependent document metadata, such as Medical Subject Headings (MeSH), while (Howard et al., 2016) shows MeSH terms contribute to just 1% of the work saved. In contrast, this paper presents a domain-independent Multi-Channel CNN approach. This approach leverages the meaning of essential words and sentences from the title and abstract through word embeddings, which are used to generate informative document features (Colón-Ruiz & Segura-Bedmar, 2020). The proposed method uses parallel CNN architectures with varying kernel sizes followed by a feed-forward neural network to learn these essential words and phrases for the citation screening process. To the best of our knowledge, this paper presents the first decision support system leveraging deep learning techniques (van Dinter et al., 2021). Since the deep learning-based system can generate document features from the context of words instead of static features, our proposed system is more suitable for automation tasks.

To assess the performance of our Multi-Channel CNN approach, we use a dataset containing five toxicology reviews with many citations by Howard et al. (2016). We compare our results against two benchmark studies (Howard et al., 2016; Kontonatsios et al., 2020). Kontonatsios et al. (2020) used Neural Networks for feature extraction and an SVM for classification, while Howard et al. (2016) rank citations based on term frequencies and latent Dirichlet allocation for topic modeling.

Our contributions to reduce time consumption and human error of the document retrieval and primary study selection processes are as follows:

- An improved process through our novel combination of the intertwined document retrieval and primary study selection steps.

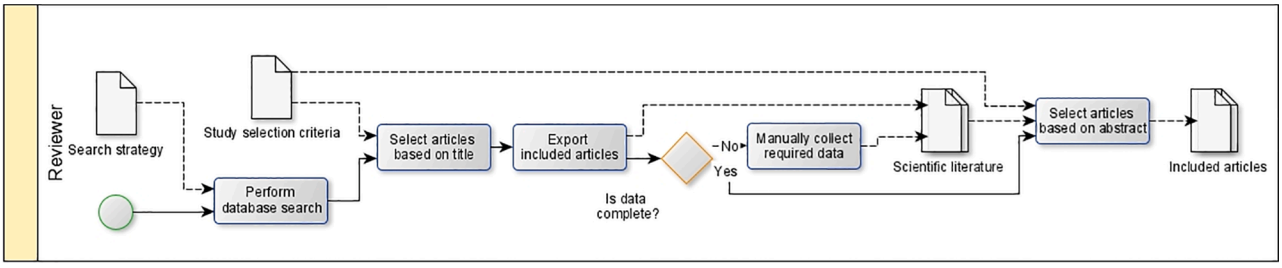

Fig. 1. Gold standard for document retrieval and primary study selection as described by (Kitchenham & Charters, 2007; Tsafnat et al., 2014).

- Reviewers can use the tool in the cloud without preliminary programming knowledge, as we have made our decision support system easy-to-use by developing it as a Markdown Form in Google CoLab.
- The DSS is domain-independent using title and abstract solely.
- Retrieval of documents through three scientific database APIs with the possibility to expand.
- General to specific search term conversion through an automated field and date tagging algorithm for developing database-specific search queries.
- Implementation of a Multi-Channel CNN model into the system.
- A quantitative analysis of the effect of the system.
- Our decision support system is publicly available at https://github.com/rvdinter/decision-support-system

The following sections are organized as follows: Section 2 presents the related work and background. Section 3 presents the adopted research methodology. Section 4 describes the results. Section 5 presents the discussion. Finally, Section 6 presents the conclusion and future work.

2. Related work

The Systematic Review Toolbox (Toolbox, 2014) is a renowned catalog of tools that support the systematic literature review process. The toolbox provides 235 tools that can be used in the SLR process, categorized based on approach, cost, discipline, and the step(s) of automation. However, searching for a free-to-use and multidisciplinary tool for document retrieval and primary study selection provides just one tool: Colandr (Colandr, n.d.). Colandr is a web-based application for conducting evidence reviews in collaboration. Even though it is open access, it is not open source. This way, it is not transparent which algorithms and techniques have been used, and researchers cannot build upon them.

Besides, Beller et al. (2018) list automation tools that can be used to speed up the systematic literature review process and set 8 guidelines for creating a systematic review tool. Beller et al. (2018) mention the importance of the introduction of free decision support systems, as non-profit research groups often do not have funding to pay for ongoing licenses. They also notice that systems are often created in isolation, cannot be integrated into a more extensive system, and are left to deprecate. Therefore, the development of open-source software is critical. It enables developers to incorporate it into another system easily. Similarly, O'Connor et al. (2019) state barriers to why researchers don't use systematic review automation tools to speed up the process. The main causes are that reviewers are reluctant to use automation tools, as most tools are not transparent, and the set-up process is often too complicated. Therefore, our DSS needs to be fully open-source and easily usable by reviewers.

Furthermore, Tsafnat et al. (2014) and Marshall and Wallace (2019) list tools useful for systematic reviews. However, none of the tools listed can perform both study search and selection. van Altena, Spijker, and Olabarriaga (2019) conducted a survey that concludes that not many researchers use a systematic review tool. When tools were used, participants often learn about them from their environments, such as colleagues, peers, or organizations. Tools were often chosen based on user experience, either by experience or from colleagues or peers. To show the speed of automated SLR studies, Clark et al. (2020) performed an SLR study using a suite of tools. Four FTEs made use of a tool for each step in the SLR process. As a result, they completed the research in just 2 weeks with a draft manuscript developed within 61 h.

In (2006), (Cohen et al.) proposed an approach to support the primary study selection process, which has been widely used since then, as it has been adopted by (Bui, Jonnalagadda, & Del Fiol, 2015; Cohen, Ambert, & McDonagh, 2009; Cohen et al., 2006; García Adeva, Pikatza Atxa, Ubeda Carrillo, & Ansuategi Zengotitabengoa, 2014; Kontonatsios et al., 2020; Ouhbi, Kamoune, Frikh, Zemmouri, & Behja, 2016; Rúbio & Gulo, 2016; Sellak, Ouhbi, & Frikh, 2015), regardless of machine learning task (i.e., classification or ranking) used. This gold standard for the automation of the study selection process has been illustrated in Fig. 2. Another methodology to support the primary study selection process is Visual Text Mining, proposed by (Felizardo, Nakagawa, MacDonell, & Maldonado, 2014; Malheiros, Hohn, Pinho, Mendonca, & Maldonado, 2007; Zdravevski et al., 2019). Finally (Hashimoto, Kontonatsios, Miwa, & Ananiadou, 2016; Kontonatsios et al., 2017; Miwa, Thomas, O'Mara-Eves, & Ananiadou, 2014; Wallace, Small, Brodley, & Trikalinos, 2010) provide workflows focusing on Active Learning techniques to support the primary study selection process.

A study by Ros, Bjarnason, and Runeson (2017) argues that the ideal tool for study search and selection would provide paper recommendations. Then, the only manual task required for the researcher is to perform the validations of papers suggested by the tool. This requires a fully automated research identification and a semi-automated selection. Ros et al. (2017) developed a system that automatically refines the search string, inserts them into the Scopus database, applies the snowballing technique, and selects relevant studies using active learning. However, active learning has a major drawback. Cohen et al. (2009) mentions that researchers prefer ranking or classification instead of active learning, as it reverses control from researchers to software, which reduces the study's transparency.

3. Research methodology

This section details the methodology that we follow to automate the document retrieval and primary study selection process of systematic reviews. First, we define the workflow of the automatic document retrieval and classification framework. We then provide more information on the system's algorithm. At last, we show a detailed explanation of the quantitative analysis of our decision support system.

3.1. Document retrieval and primary study selection framework

We propose a framework for a decision support system that incorporates the document retrieval and primary study selection steps because they are tightly connected. We aim to eliminate the repetitive tasks while maintaining the systematic steps as proposed by Kitchenham and Charters (2007). Fig. 3 demonstrates the overall workflow we propose in our study.

A reviewer inserts the search query, publication venues, and timeframe of the search into a Graphical User Interface (GUI). After inserting the required data into the GUI, the DSS collects all articles that match this query. When documents have been retrieved through database API searches, we merge each database's returned documents by alternating rows, as the returned citations have been sorted on relevancy. The system splits the initial citation list into two equivalent-sized sets, namely train, and evaluation. The citations in the train set do not overlap with the citations in the evaluation set. The reviewer manually annotates the train set with include/exclude codes. When reviewers have labeled all articles in the train set and have reached a consensus amongst reviewers, the data is used by the model to learn to generalize which studies are relevant. Afterward, a list of all articles, sorted on probability, is returned as output.

3.2. Data retrieval and citation screening system

In this section, we discuss the development of the proposed DSS. First, we provide a detailed description of the graphical user interface, document retrieval step, preprocessing of the dataset, and citation classification. Subsequently, we show our evaluation approach for quantitative analysis of the method.

3.2.1. The graphical user interface

We have used Google CoLab Markdown Forms to make this DSS

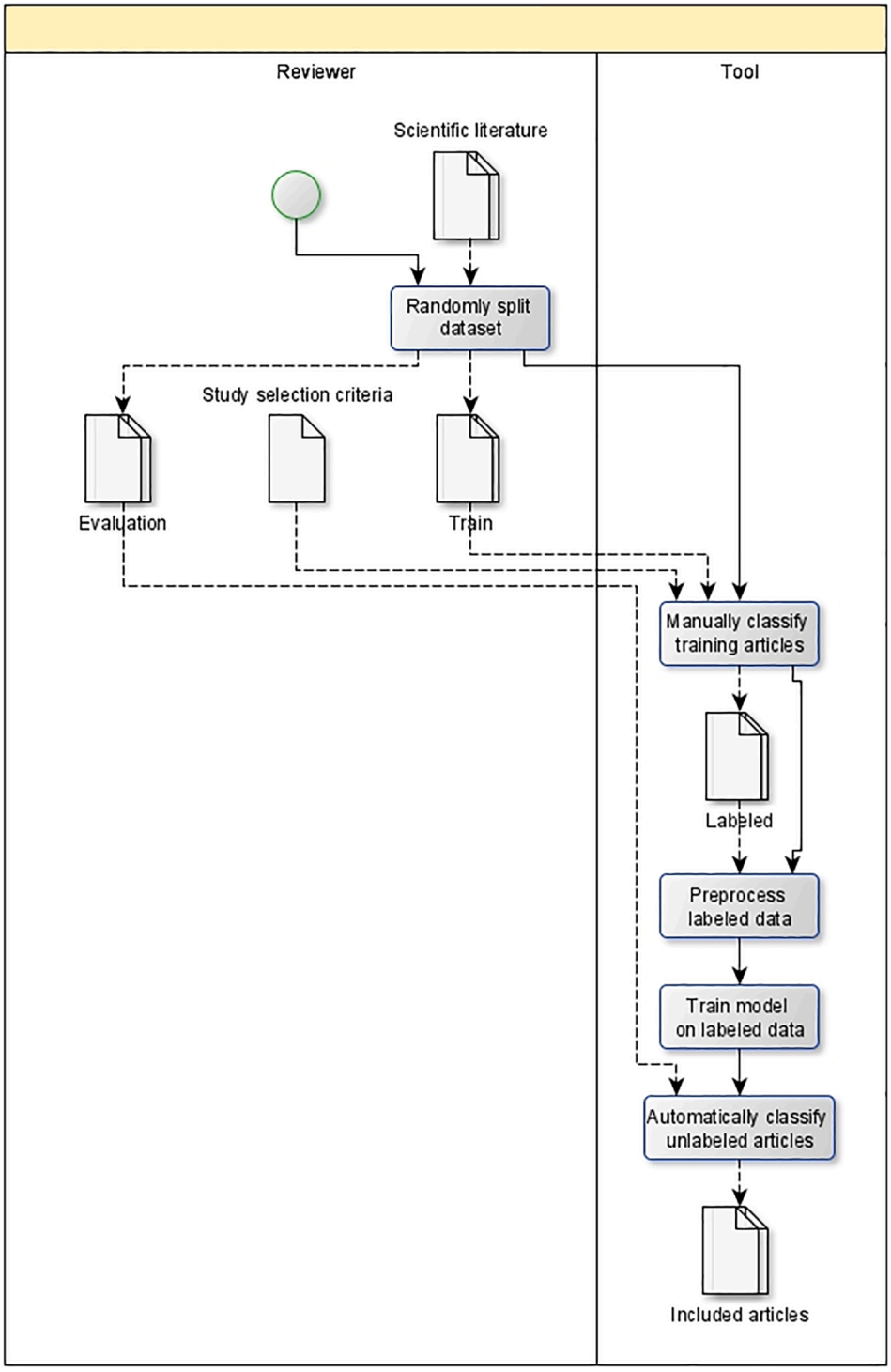

Fig. 2. The gold standard for the automation of the primary study selection process.

visual and accessible to researchers without any programming background while remaining a Notebook containing developers' expansion options. Furthermore, Google CoLab allows researchers to use industry-grade hardware, such as GPUs and TPUs, in the cloud - for free. This eliminates the need for research groups to buy any hardware to run our method.

3.2.2. *Automated document retrieval*

The Markdown Form cell of the document retrieval step is shown in Fig. 4. To execute queries on multiple databases, we require users to write a generic search query. After writing the search query, the user can choose which fields the query should execute (e.g., title and abstract). Users can choose which databases to execute the queries on and the timeframe of the documents. We have developed code to execute queries on APIs from ScienceDirect, PubMed, and Springer.

The PubMed API has built-in query field addition. However, Springer does not. Therefore, we have developed a query refinement algorithm that can add the field, such as title, to the query. For instance, if we want to search the following in the title:

"Automation"AND"SystematicLiteratureReview"

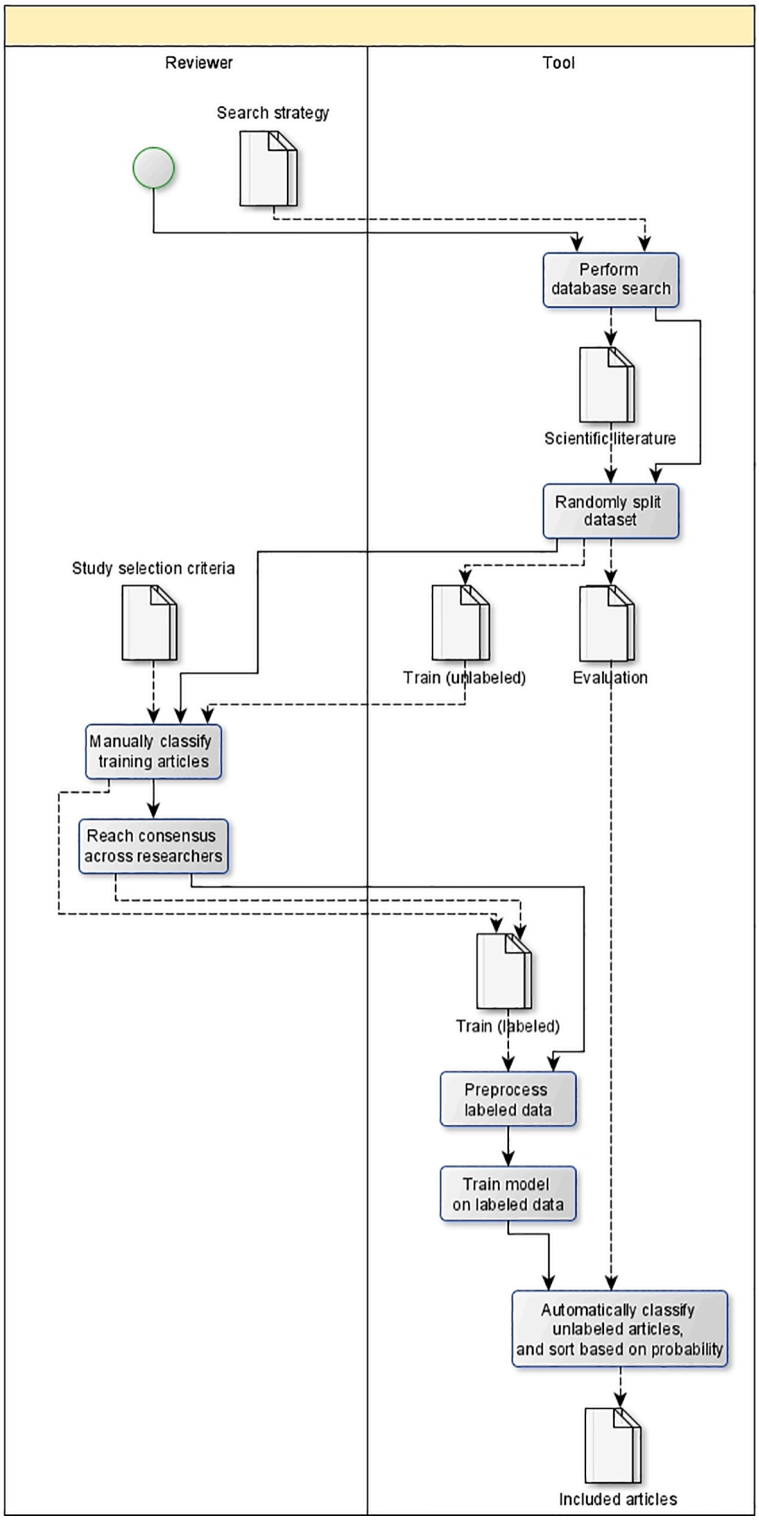

Fig. 3. Workflow of the automatic document retrieval and classification.

We obtain the following refined search query:

title : *"Automation"ANDtitle* : *"SystematicLiteratureReview"*

API keys can be obtained from each databases' developer websites. We must note that you need to be accessing through an IP address from your academic institute to use ScienceDirect's API. Each API returns its values, but the title, year of publication, article identifier, and abstract columns are always included. These are also the features that are used most often by reviewers during the study selection process. After the query execution in selected databases, users can find an MS Excel file containing the features in their working directory, which can be immediately used to select primary studies.

3.2.3. *Preprocessing*

Once all data is collected, the algorithm splits the dataset into two stratified sets; *train* and *evaluate*. The *train* set must be screened and

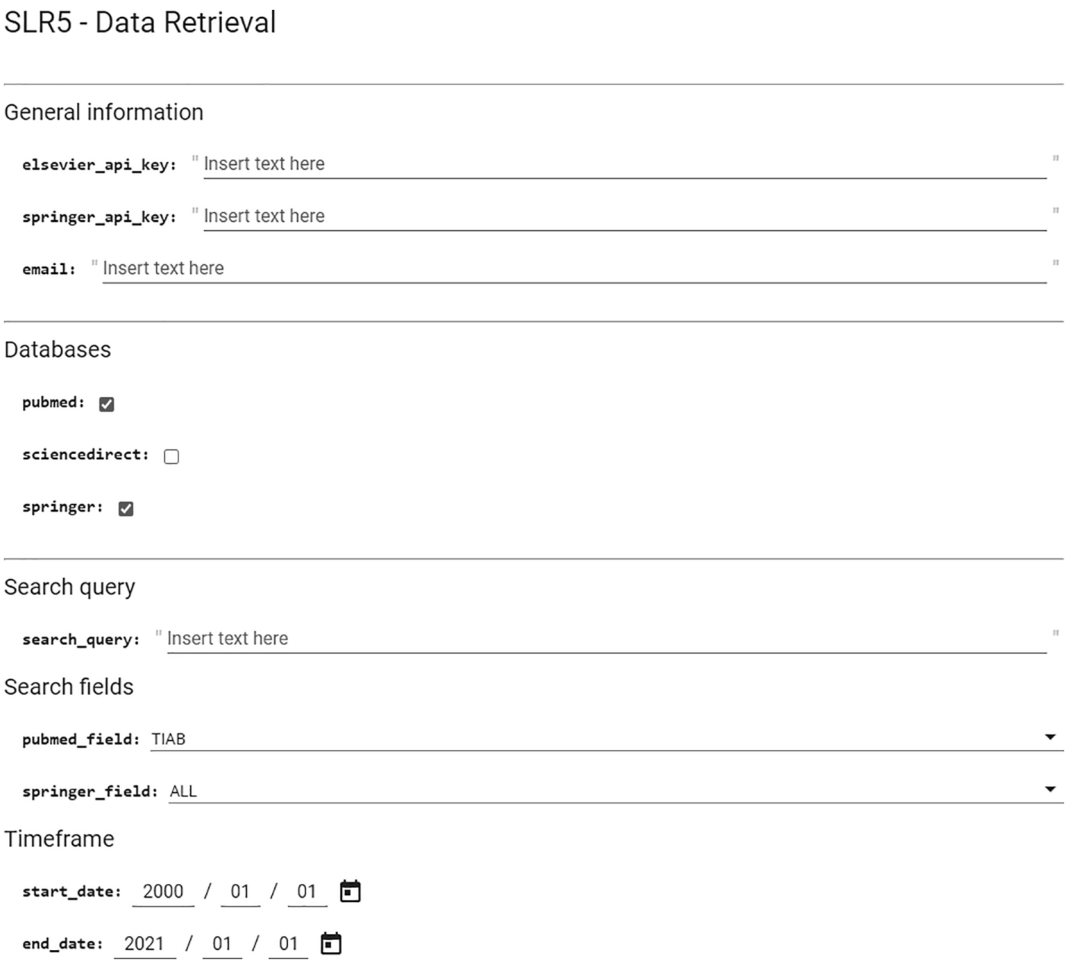

Fig. 4. Data retrieval Graphical User Interface.

classified by a reviewer. Fig. 5 shows the GUI of the study selection UI.

After the reviewer has classified all articles from the *train* set, we preprocess the *train* and *evaluation* sets for the deep learning model. We first concatenate the title and abstract into one feature column to eliminate empty features, as many citations (e.g., interviews or book chapters). Then, we clean the text by splitting the text into tokens, removing its punctuation, converting to lower case, removing non-alphabetic and stop words, removing short tokens of just one character, and applied a minimal token occurrence of 10 times in the entire dataset. We are using the title and abstract as the main sources of our features because they are independent of the database or research domain. These features are also most often used to automate the selection of primary studies (van Dinter et al., 2021). This is partly because the title and abstract can usually be retrieved through database APIs (Langlois et al., 2018; Rúbio & Gulo, 2016), while full-text is often not included. Furthermore, Dieste and Padua (2007) also suggest the use of the title and abstract instead of full-text, as full-text has many challenges regarding cleaning and accessibility. Domain-specific features such as MeSH terms have been excluded, as the Decision Support System must be available to all reviewers.

Once we have cleaned the sets, we split the labeled *train* into a stratified *train* and *validation* set to enable developers to monitor the model during training. The *train*, *test,* and *validation* sets have been split into a 45/50/5 distribution, respectively. Further, we used the Tokenizer API to create numeric word vectors from the feature column. We zip the feature and target columns into a tf.data.Dataset object. Using this object, we can oversample the *training* set using tf.data.experimental.sample_from_datasets() to an even class distribution to avoid class imbalance issues. Once the datasets are complete, we pad the datasets. Feature columns are padded to a size of 600, as most citations have a lower token length.

3.2.4. *Deep learning-based citation screening*

We use a Multi-Channel CNN architecture to support the citation screening process, as they are faster and computationally less expensive than the alternative LSTM architectures that are used for different purposes (Altan et al., 2021)To this day, using large LSTM architectures is not available to all researchers without access to GPU hardware with high memory. However, CNN models allow for impressive results with fewer hardware requirements and lower training times. Furthermore, CNNs also focus on finding keywords and -phrases in text classification, which researchers often do in the SLR process while skim-reading many articles.

The model uses the text feature column as input and provides a confidence score as output. An Embedding layer follows the input to create word embeddings in an end-to-end fashion. We have used 100-dimensional GloVe embeddings trained on the Wikipedia dataset to input our embedding layer (Pennington, Socher, & Manning, 2014). The embedding layer does not need to train its parameters, as we have inserted the embedding matrix, which significantly reduces the training time for the model. The features are fed to the dropout layer, which drops out 60% of the features. Dropout only applies to the model when training to generalize on new data so the full potential can be used. Then, the remaining data is fed to two parallel convolutional channels. For each of the CNN channels, we use a single CNN layer followed by global max pooling. We chose the global max-pooling layer over regular

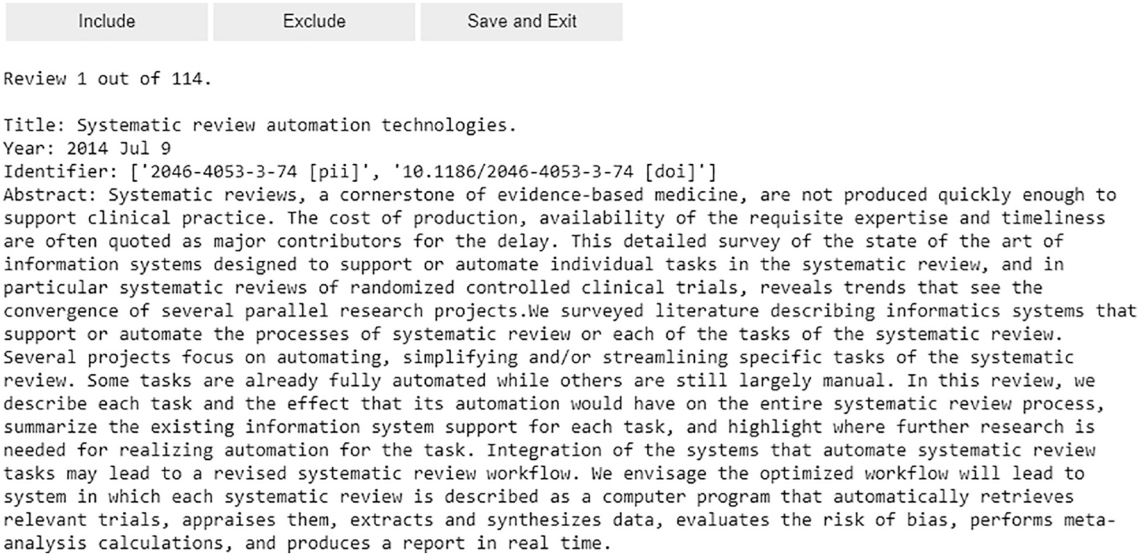

Fig. 5. Widget of the manual study selection UI.

max-pooling, as (Jacovi, Shalom, & Goldberg, 2018) describes: *"Global max-pooling induces a functionality of separating important and not important activation signals using a latent (presumably soft) threshold"* (Jacovi et al., 2018). After the global pooling layer, we concatenate the outputs and put them into a feed-forward network. We apply a 40% dropout to the concatenated vector and send it to the dense layer. The hidden dense and convolutional layers use the ReLu activation function to avoid the vanishing gradient problem, and the dens layers use bias and kernel weight constraints following the unit norm to prevent overfitting. The last dense layer uses a Sigmoid activation function to account for the confidence score, ranging from 0 for *excluded* to 1 for *included*. Table 2 provides the model parameter settings. To find the most compelling citations, the citations are exported and ranked based on the confidence score. As the citations are being ranked as an output, the reviewers remain in control of the screening process (Fig. 6).

Table 1
Major related studies on SLR tools.

No.	Title	SLR steps	Reference
1	A full systematic review was completed in 2 weeks using automation tools: a case study	Completed an SLR in 2 weeks using multiple tools	Clark et al. (2020)
2	Making progress with the automation of systematic reviews: principles of the International Collaboration for the Automation of Systematic Reviews (ICASR)	List tools that can be used, and set 8 guidelines for automating SLRs	Beller et al. (2018)
3	Toward systematic review automation: a practical guide to using machine learning tools in research synthesis	Lists tools that are useful for systematic reviews	Marshall and Wallace (2019)
4	A question of trust: can we build an evidence base to gain trust in systematic review automation technologies?	States barriers why people don't use systematic review automation tools	O'Connor et al. (2019)
5	Systematic review automation technologies	Describe each step in the SLR process, its automation potential, and current tools	Tsafnat et al. (2014)
6	Usage of automation tools in systematic reviews	Concludes that not many researchers are using an SLR tool	van Altena et al. (2019)
7	Software tools to support title and abstract screening for systematic reviews in healthcare: an evaluation	An evaluation of tools that screen citations based on title and abstract	Harrison, Griffin, Kuhn, and Usher-Smith (2020)

3.3. Quantitative analysis framework

This subsection describes how we evaluated our decision support system in a quantitative matter. First, we explain how we obtained our datasets to evaluate our method against a benchmark. Then, we discuss the evaluation settings that we have used for the citation screening step.

3.3.1. Datasets

We have collected five publicly available datasets from (Howard et al., 2016) to evaluate our model. These datasets have been regularly used to evaluate models in the medical domain. We have also collected the WSS@95% results from (Howard et al., 2016; Kontonatsios et al., 2020) as benchmarks for evaluating our results. The dataset by (Howard et al., 2016) contains a list of PMIDs and their corresponding binary label (i.e., 0 for *excluded,* 1 for *included*). We edited our document retrieval module to import the PMID list and search its title, abstract, and metadata through the PubMed API. Table 3 shows the metadata for each of the datasets. Each sample (i.e., citation) contains at least the title, abstract, and label. The 5 datasets from (Howard et al., 2016) are categorized as toxicology reviews. The toxicology reviews are rather extensive, as the researchers used broad search strategies. This is important, as neural networks tend to thrive on large datasets.

From the datasets, an average of approximately 3.87% of abstracts is missing. However, this differs significantly between datasets. For instance, the Neuropathic Pain dataset has 0 abstracts missing, but the Fluoride dataset has 13.60% of its abstracts missing.

3.3.2. Evaluation settings

The primary metric in the automation of the SLR field is Work Saved over Sampling (WSS) (Cohen et al., 2006), which is defined as *" the percentage of papers that meet the original search criteria that the reviewers do not have to read (because they have been screened out by the classifier)."* (Cohen et al., 2006). The formula for WSS is defined as follows:

$$\mathrm{WSS} = \frac{(\mathrm{TN} + \mathrm{FN})}{\mathrm{N}}$$

where *TP* is the number of true positives, *TN* is the number of true

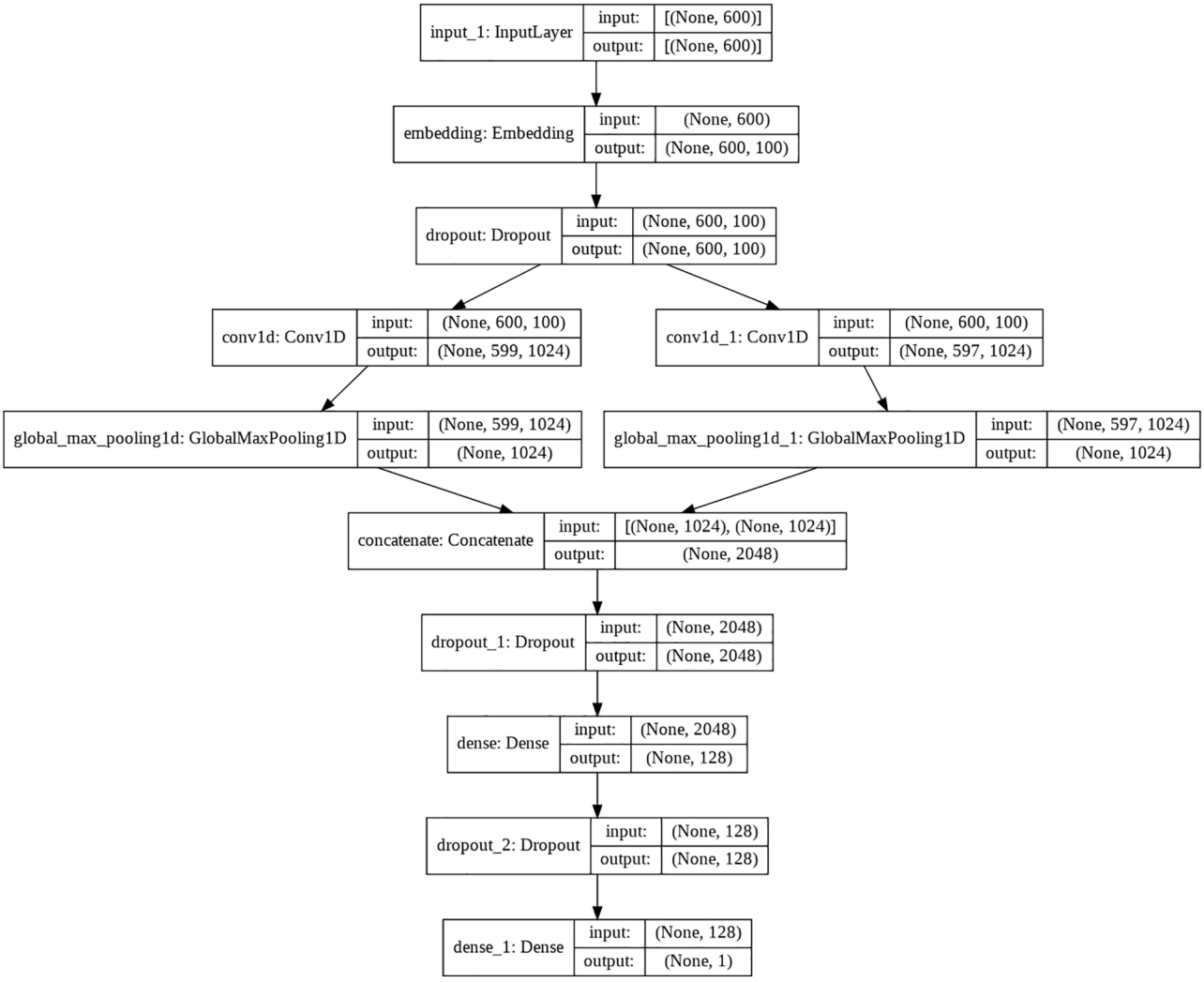

Fig. 6. Decision support system visualized. Other models are similar, except for the number of channels and Conv1D output shape due to kernel size differences.

Table 2
Model parameter settings.

Epochs	Batch size	Dropout input layer	Dropout hidden layers	Filters	No. channels	Kernel sizes	Dense units	Learning rate
15	100	0.6	0.4	1024	2	2/4	128	1E−4

negatives, *FN* is the number of false negatives, *N* is the total number of abstracts in the set. Cohen et al. (2006) stated that one should interpolate the WSS metric at a 95% recall, as work saved must be greater than work saved by plain random sampling (Cohen et al., 2006). When incorporating recall *R* in the formula, we get:

$$\mathrm{WSS@R} = \frac{(\mathrm{TN}+\mathrm{FN})}{\mathrm{N}} - (1-\mathrm{R}) = \frac{(\mathrm{TN}+\mathrm{FN})}{\mathrm{N}} - \left(1 - \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\right)$$

For the model evaluation, we show the WSS@95% results. In more specific, we show the mean WSS@95% after 10 × 2-fold cross-validation. 2-fold cross-validation splits the dataset into two equally sized subsets, with an even distribution of label classes, just as with our proposed framework. We performed this 2-fold cross-validation over 10 fixed seeds to achieve a final estimated mean.

As described by (Ng, 2017), adding more than one metric also makes it more complex to compare algorithms. As (Ng, 2017) explains: *"Having a single-number evaluation metric such as accuracy allows you to sort all your models according to their performance on this metric, and quickly decide what is working best."* (Ng, 2017). Therefore, we keep WSS@95% as our single metric, as it measures the human workload saved in a real-world scenario.

4. Results

After retrieving the title and abstract features through the PubMed API, we preprocessed the features and performed 10x2-fold cross-validation for the Multi-Channel CNN. The 10 × 2-fold cross-validation returns 10 WSS@95% results, of which Table 4 shows the mean.

According to (Cohen et al., 2006), a significant and meaningful workload saving should be at least 10% for the WSS@95% metric. This stems from the fact that the citation screening process of a systematic review, when conducted manually, requires on average ~8.7 FTE to be completed, based on a 38-hour workweek. Therefore, a WSS@95% score of 10%, i.e., 10% of correctly excluded citations +5% of incorrectly excluded citations, results in a workload reduction of ~1.3 FTE. According to expert reviewers, this is a significant reduction of their

Table 3
Datasets adopted by (Howard et al., 2016).

Author	Dataset	# Citations	Eligible citations (%)	Missing abstracts (%)
Howard et al. (2016)	Bisphenol-A (BPA) and obesity	7700	1.44	7.88
	PFOA/PFOS and immunotoxicity	6328	1.50	5.97
	Transgenerational inheritance of health effects	48,638	1.57	4.38
	Fluoride and neurotoxicity in animal models	4479	1.14	13.60
	Neuropathic pain	29,202	17.2	0.00

Table 4
WSS@95% results of 2 benchmark studies versus the Decision Support System. We have printed the highest results of each dataset in bold.

Datasets	Howard et al. (2016)	Kontonatsios et al. (2020)	Decision Support System
BPA	0.752	0.758	**0.792**
PFOA/PFOS	0.805	**0.848**	0.071
Transgenerational	**0.714**	0.707	0.708
Fluoride	0.870	0.799	**0.883**
Neuropain	**0.691**	0.608	0.620

citation screening labor.

We take the DSS scores and compare them to the two benchmark studies by Howard et al. (2016) and Kontonatsios et al. (2020). The experiments that we conducted showed that our proposed DSS yields significant workload savings of at least 10% in 4 out of 5 review datasets. Additionally, we can see that our DSS outperforms the benchmark studies on two datasets, Bisphenol-A and Fluoride. We achieved a 3.6% and 1.3% improvement over the BPA and Fluoride datasets, respectively. The DSS performed poorly on the PFOA/PFOS dataset, as it seemed to overfit.

Furthermore, in Fig. 7, we have also plotted the DSS results in boxplots against the benchmark's means. We can see that the benchmarks' results are often inside the interquartile range (IQR). Also, the five large toxicology review datasets have a narrow IQR and minimum–maximum range. This means that the model has low variance, as the Multi-Channel CNN provides consistent results across the whole dataset.

We want to note that training a single model takes approximately 1–3 min. After that, the model can be used to analyze the test set. In the DSS, we use the Multi-Channel CNN to sort the most relevant articles based on the confidence score. As the model trains rapidly, there is no drawback on time cost for the user-end.

5. Discussion

5.1. Discussion on the results

Due to powerful NLP and deep learning techniques, we can outperform some of the benchmarks in the citation screening process. The use of contextualized word embeddings for citation screening shifts the perspective from *"do the most used words in a title and abstract correspond?"* to *"are there words or sentences with a similar meaning?"*. Furthermore, Multi-Channel CNNs are fast and show a low variance for large datasets, such as the toxicology reviews. Deep learning in this field is shown to be essential to achieve even higher results than the benchmarks.

We developed and evaluated a DSS to retrieve relevant and high-quality citations from PubMed, ScienceDirect, and Springer. The approach can be used to assist the development of systematic literature reviews independent of the domain. The results showed that our proposed DSS outperformed two benchmark datasets in terms of WSS@95%. The obtained results demonstrate that our Multi-Channel CNN-based DSS substantially reduced the screening workload of four systematic review studies by approximately 57%. The results reflect the goal of systematic search: to maximize recall to identify all relevant studies while controlling precision to keep the results manageable. The workload savings varied across the five reviews, from a low WSS@95% score of ~7% on the PFOS-PFOA review to a higher WSS @95% score of ~88% on the Fluoride review. The DSS performed poorly on the PFOS-PFOA review dataset, as it seemed to overfit. We have checked the dataset input, but it has no differences from the other datasets. We can also conclude from looking at the IQR from the boxplots in Fig. 7; the WSS@95% scores remain remarkably consistent within each dataset.

According to (Cohen et al., 2006), a significant and meaningful workload saving should be at least 10% for the WSS@95% metric. This stems from the fact that the citation screening process of a systematic review, when conducted manually, requires on average ~8.7 FTE to be completed, based on a 38-hour workweek. Therefore, a WSS@95% score of 10%, i.e., 10% of correctly excluded citations +5% of incorrectly

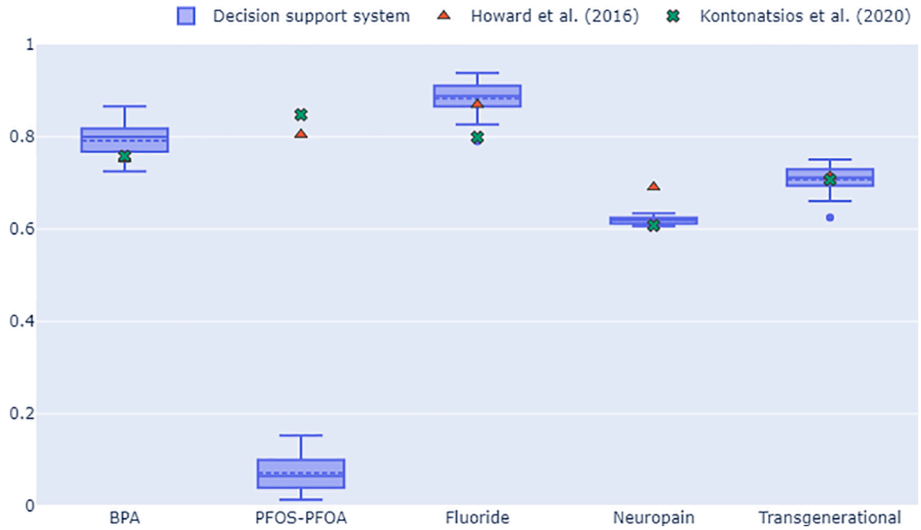

Fig. 7. WSS@95% values of the decision support system and benchmark papers. Benchmark papers WSS@95% values are shown as markers, as they are means. Decision support system WSS@95% values are shown as boxplots. The dotted line in the box stands for the mean; the continuous line represents the median.

excluded citations, results in a workload reduction of ~1.3 FTE. According to expert reviewers, this is a significant reduction of their citation screening labor. The experiments that we conducted showed that our proposed DSS yields significant workload savings of at least 10% in 4 out of 5 review datasets. Thus, it could be potentially used in practical application scenarios for accelerating the citation screening task of systematic reviews.

In practice, our method's workload reduction (i.e., WSS@95% score) achieved by our DSS is relative to the underlying review dataset's size. For example, our Multi-Channel CNN obtained approximately the same WSS@95% performance of 0.7 on both the BPA and the Transgenerational dataset. However, the Transgenerational dataset's validation sample consists of 24,319 citations, and it is substantially larger than the validation sample of the BPA dataset, which consists of 3,850 citations. In practice, this means that a WSS@95% score of 0.708 is equivalent to a workload reduction of 18,433 citations, which are automatically excluded from the Transgenerational review. In comparison, a WSS@95% score of 0.792 translates to a workload reduction of only 3,241 automatically excluded citations for the BPA dataset.

Last, as similarly mentioned by (Kontonatsios et al., 2020), our DSS's limitation is that we need to train our neural network independently for each SLR dataset. This means that we have trained 5 Multi-Channel CNNs corresponding to each dataset. As (Kontonatsios et al., 2020) explains: *"Different systematic reviews may share one or more eligibility criteria (*e.g.*, if included studies are randomized control trials) and thus learned document features could be applied to different reviews."* This could be included in future work.

5.2. General discussion

Our study's main difference with the related work is that we have explicitly developed a DSS, which is domain-independent, open-source, and does not require programming knowledge. Furthermore, our study is the first paper that leverages deep learning to select primary studies—opposed to the tools listed by Toolbox (2014), Tsafnat et al. (2014), and Marshall and Wallace (2019), our study supports the automation of two steps in the SLR process.

Our study shows that shallow machine learning architectures for study selection used domain-dependent finetuning of hand-crafted features in the related work. The need for finetuning is overcome by using a practical and interchangeable NLP preprocessing pipeline using only the title and abstract. The use of just these two input features is in line with the evaluation of tools by Harrison et al. (2020). Furthermore, neural networks are used to find hidden features that eliminate the need for hand-crafted features and finetuning for each domain.

Additionally, other primary studies have not developed a DSS that allows input from multiple databases. Moreover, we follow (Cohen et al., 2009) by classifying citations rather than the Active Learning approach. We did not develop a snowballing algorithm, nor did we build query refinement except for adding the search field.

We found the key papers on the automation of the citation screening process by (Howard et al., 2016; Kontonatsios et al., 2020) and evaluated our results on their benchmark scores. Our study selection, preprocessing, and evaluation methodology steps were corresponding to their studies. Therefore, we have developed a DSS that is reliable and significantly different from shallow machine learning applications, with new and relevant insights.

5.3. Threats to validity

Construct Validity: Construct validity assesses whether the SLR represents the degree to which it measures what it asserts. First, we tried to replicate the model by (Kontonatsios et al., 2020), as it has been recently published and open-source via GitHub. However, we could not achieve similar scores using our dataset. Unfortunately, the corresponding datasets seem not to be accessible anymore. Hence, we validated our code based on the code provided by (Colón-Ruiz & Segura-Bedmar, 2020; Kontonatsios et al., 2020), and others on Kaggle.

Criterion Validity: To assess model WSS@95% results during cross-validation, we have developed a TensorFlow custom metric class. As it needed to measure WSS at a specific recall rate, we used the SensitivityAtSpecificity base class. This base class allows us to calculate a metric at another metric. We have validated this metric using our hand-written calculations. Furthermore, when validating our scores against the benchmark papers, our models seem to score in line with the other papers.

Internal Validity: Internal validity shows the incomplete relationship between results, which may lead to structural errors. We used cross-validation, set 10 seeds to have the same dataset splits consistently, and used fixed model hyperparameters. As these techniques were well-defined in other papers and their open-source code, the model evaluation against benchmark papers was described adequately.

External Validity: This primary study only used published studies as benchmarks that applied machine learning techniques to automate the citation screening process. The scores were required to be mentioned using the WSS@95% score, retrieved by *N*x2-fold cross-validation. Here *N* must be between 5 and 10 rounds. Furthermore, a new machine/deep learning or natural language processing algorithm has not been applied yet to automate systematic literature reviews, like novel transformer algorithms, such as BERT and GPT-3. As these studies have not been published, they have not been discussed regardless of their potential.

Conclusion Validity: The conclusion validity measures the reproducibility of this study. Our study used datasets provided by (Kontonatsios et al., 2020). Furthermore, we made our code open-source, available on this GitHub page (https://github.com/rvdinter/decision-support-system). Our DSS was also discussed among the authors to minimize individual errors. We derived all conclusions based on the tables and figures to avoid subjective interpretation of the researchers' results.

6. Conclusion

This paper has presented a DSS to support the automation of the document retrieval and citation screening process for SLRs. The system aims to improve workflow during retrieving and screening documents and minimize the human workload and error involved in citation screening. Using the proposed DSS, reviewers can follow the procedure by (Kitchenham & Charters, 2007) where they need to fill in their search strategy and manually label only a subset of the citations, while the remaining unlabeled citations are automatically classified and sorted based on their probability. We have demonstrated that by developing a deep learning-based pipeline, we can use the power of deep learning promises to overcome domain-specific challenges in shallow machine learning.

We have further experimented with assessing our Multi-Channel CNN architecture's performance across five publicly available SLR datasets. It was shown that for four out of five review datasets, the proposed method achieved significant workload savings of at least 10%. At the same time, in several cases, our model yielded a better performance over two benchmark review datasets. The approach can be used to assist the development of SLRs independent of the domain. In our future work, we will apply the DSS for various SLRs and focus on the automation of the other steps of the SLR process.

CRediT authorship contribution statement

Raymon Dinter: Conceptualization, Data curation, Software, Writing - review & editing. **Cagatay Catal:** Methodology, Validation, Writing - review & editing. **Bedir Tekinerdogan:** Methodology, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Open Access funding provided by the Qatar National Library.

References

Altan, A., Karasu, S., & Zio, E. (2021). A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Applied Soft Computing, 100*, 106996.

Beller, E., Clark, J., Tsafnat, G., Adams, C., Diehl, H., Lund, H., … Glasziou, P. (2018). Making progress with the automation of systematic reviews: Principles of the International Collaboration for the Automation of Systematic Reviews (ICASR). *Systematic Reviews, 7*, 77.

Beller, E. M., Chen, J. K.-H., Wang, U.-L.-H., & Glasziou, P. P. (2013). Are systematic reviews up-to-date at the time of publication? *Systematic Reviews, 2*, 36.

Bui, D. D. A., Jonnalagadda, S., & Del Fiol, G. (2015). Automatically finding relevant citations for clinical guideline development. *Journal of Biomedical Informatics, 57*, 436–445.

Clark, J., Glasziou, P., Del Mar, C., Bannach-Brown, A., Stehlik, P., & Scott, A. M. (2020). A full systematic review was completed in 2 weeks using automation tools: A case study. *Journal of Clinical Epidemiology, 121*, 81–90.

Cohen, A. M., Ambert, K., & McDonagh, M. (2009). Cross-topic learning for work prioritization in systematic review creation and update. *Journal of the American Medical Informatics Association, 16*(5), 690–704.

Cohen, A. M., Hersh, W. R., Peterson, K., & Yen, P.-Y. (2006). Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association, 13*(2), 206–219.

Colandr. (n.d.). Sign In. In (Vol. 2021).

Colón-Ruiz, C., & Segura-Bedmar, I. (2020). Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics, 110*, 103539.

Dieste, O., & Padua, A. G. (2007). Developing search strategies for detecting relevant experiments for systematic reviews. In First international symposium on empirical software engineering and measurement (ESEM 2007) (pp. 215-224): IEEE.

Felizardo, K. R., Nakagawa, E. Y., MacDonell, S. G., & Maldonado, J. C. (2014). A visual analysis approach to update systematic reviews. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (pp. Article 4). London, England, United Kingdom: Association for Computing Machinery.

García Adeva, J. J., Pikatza Atxa, J. M., Ubeda Carrillo, M., & Ansuategi Zengotitabengoa, E. (2014). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications, 41*(4), 1498–1508.

Harrison, H., Griffin, S. J., Kuhn, I., & Usher-Smith, J. A. (2020). Software tools to support title and abstract screening for systematic reviews in healthcare: An evaluation. *BMC Medical Research Methodology, 20*, 1–12.

Hashimoto, K., Kontonatsios, G., Miwa, M., & Ananiadou, S. (2016). Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of Biomedical Informatics, 62*, 59–65.

Howard, B. E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M. R., … Thayer, K. (2016). SWIFT-Review: A text-mining workbench for systematic review. *Systematic Reviews, 5*, 87.

Jacovi, A., Shalom, O. S., & Goldberg, Y. (2018). Understanding convolutional neural networks for text classification. arXiv preprint arXiv:1809.08037.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In (Vol. 2.3): Keele University.

Kontonatsios, G., Brockmeier, A. J., Przybyła, P., McNaught, J., Mu, T., Goulermas, J. Y., & Ananiadou, S. (2017). A semi-supervised approach using label propagation to support citation screening. *Journal of Biomedical Informatics, 72*, 67–76.

Kontonatsios, G., Spencer, S., Matthew, P., & Korkontzelos, I. (2020). Using a neural network-based feature extraction method to facilitate citation screening for systematic reviews. *Expert Systems with Applications, X*, Article 100030.

Langlois, A., Nie, J.-Y., Thomas, J., Hong, Q. N., & Pluye, P. (2018). Discriminating between empirical studies and nonempirical works using automated text classification. *Research Synthesis Methods, 9*, 587–601.

Malheiros, V., Hohn, E., Pinho, R., Mendonca, M., & Maldonado, J. C. (2007). A visual text mining approach for systematic reviews. In First international symposium on empirical software engineering and measurement (ESEM 2007) (pp. 245-254).

Marshall, C. (2016). *Tool support for systematic reviews in software engineering.* Keele University.

Marshall, I. J., & Wallace, B. C. (2019). Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. *Systematic Reviews, 8*, 163.

Michelson, M., & Reuter, K. (2019). The significant cost of systematic reviews and meta-analyses: A call for greater involvement of machine learning to assess the promise of clinical trials. *Contemporary Clinical Trials Communications, 16*, 100443.

Miwa, M., Thomas, J., O'Mara-Eves, A., & Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *Journal of Biomedical Informatics, 51*, 242–253.

Ng, A. (2017). Machine learning yearning. In Deeplearning.ai (Ed.).

O'Connor, A. M., Tsafnat, G., Thomas, J., Glasziou, P., Gilbert, S. B., & Hutton, B. (2019). A question of trust: Can we build an evidence base to gain trust in systematic review automation technologies? *Systematic Reviews, 8*, 143.

Ouhbi, B., Kamoune, M., Frikh, B., Zemmouri, E. M., & Behja, H. (2016). A hybrid feature selection rule measure and its application to systematic review. In *Proceedings of the 18th international conference on information integration and web-based applications and services* (pp. 106–114). Singapore, Singapore: Association for Computing Machinery.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Ros, R., Bjarnason, E., & Runeson, P. (2017). A machine learning approach for semi-automated search and selection in literature studies. In *Proceedings of the 21st international conference on evaluation and assessment in software engineering* (pp. 118–127). Karlskrona, Sweden: Association for Computing Machinery.

Rúbio, T. R. P. M., & Gulo, C. A. S. J. (2016). Enhancing academic literature review through relevance recommendation: Using bibliometric and text-based features for classification. In *2016 11th Iberian conference on information systems and technologies (CISTI)* (pp. 1–6).

Sellak, H., Ouhbi, B., & Frikh, B. (2015). Using rule-based classifiers in systematic reviews: a semantic class association rules approach. *Proceedings of the 17th international conference on information integration and web-based applications & services*. Brussels, Belgium: Association for Computing Machinery.

Toolbox, S. R. (2014). Search. In (Vol. 2021).

Tsafnat, G., Glasziou, P., Choong, M. K., Dunn, A., Galgani, F., & Coiera, E. (2014). Systematic review automation technologies. *Systematic Reviews, 3*, 74.

van Altena, A. J., Spijker, R., & Olabarriaga, S. D. (2019). Usage of automation tools in systematic reviews. *Research Synthesis Methods, 10*, 72–82.

van Dinter, R., Tekinerdogan, B., & Catal, C. (2021). *Automation of systematic literature reviews: A systematic literature review* (p. 106589). Information and Software Technology.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2010). Active learning for biomedical citation screening. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 173–182). Washington, DC, USA: Association for Computing Machinery.

Wallace, B. C., Trikalinos, T. A., Lau, J., Brodley, C., & Schmid, C. H. (2010). Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics, 11*, 1–11.

Zdravevski, E., Lameski, P., Trajkovik, V., Chorbev, I., Goleva, R., Pombo, N., & Garcia, N. M. (2019). Automation in systematic, scoping and rapid reviews by an NLP toolkit: A case study in enhanced living environments. In *Enhanced living environments* (pp. 1–18). Springer.