Accelerating literature screening for systematic literature reviews with Large Language Models – development, application, and first evaluation of a solution

Paul Herbst and Henning Baars

University of Stuttgart, Chair of Information Systems I, Germany

Abstract

A systematic literature review (SLR) is a cornerstone of any academic endeavor. Nevertheless, literature reviews are time-consuming, arduous and a regular point of contention. A selection of adequate keywords for a database search that casts a net, that is not too wide and not too narrow, and the selection of filtering criteria in particular cause difficulties. The application of Machine Learning (ML) and Natural Language Processing (NLP) to support these tasks has been proposed before. But the emergence of Large Language Models (LLMs) and Generative Pretrained Transformer models (GPT) bring new options for automation that might capture semantic details that elude former approaches. We discuss application options for the different steps of a literature review and propose, implement, and test a solution for screening large amounts of abstracts in a short amount of time. Our initial results suggest a vast automation potential, despite some risks and limitations that have to be further navigated.

Keywords

machine learning, large language model, abstract screening, systematic literature review [1]

1. Introduction

Literature reviews provide the foundation for any research project. In some cases, they are used to contribute the related work or the conceptual foundations for a specific research project, in others, the literature review stands on its own [1–5] – a standalone literature review [1, 3]. The latter approach is particularly suited for broader topics with hundreds or thousands of relevant papers that warrant a separate quantitative analysis. Unsupported, the related tasks can cost several person-years, a large portion of which is the identification of relevant papers alone. The literature gives examples of costs that go up as high as 100.000 USD and beyond [6].

Usually, literature reviews are done with keyword searches in literature databases, although in some cases, all publications of a defined outlet are scanned manually [5]. It has been suggested before to apply Natural Language Processing (NLP) methods and/or Machine Learning (ML) techniques to partly automate this step. The literature already demonstrates some promising results. However, "classical" NLP approaches come with built-in limitations esp. because of the equivocality, vagueness, ambiguity, and context-dependencies of human language [7, 8]. A promising approach to handle these challenges are transformer-based large language models (LLMs). First introduced in 2019 they have recently shown unprecedented results in a slew of NLP tasks [9–11], and it is therefore plausible to apply them to literature reviews as well. A direct application of state-of-the-art LLMs, however (e. g. ChatGPT on top of a GPT-4 foundational model), is currently still delivering sobering results: The models make up authors, years, and publications or present papers that do not fit the actual subject [12]. However, there are alternative ways to tap into the potential of LLMs that deliver better results. In this paper we show a multistep approach that uses *text-embeddings* or contextual embeddings to create an initial classification of the paper abstracts using established machine learning approaches. Text-embeddings are vectors, generated by an LLM [13] for capturing the semantics of a word, sentence or paragraph. Following this initial classification is the usage of the natural language understanding capability of an LLM for the final selection using few-shot learning.

[1] *LWDA: Learing, Knowledge, Data, Analysis 2023, October 09–11, 2023, Marburg, Germany*

Therefore, our research question is: How can LLMs be exploited for an automatic screening of abstracts in a SLR?

2. Conceptual Foundations

There is a plethora of literature on how to conduct a systematic literature review (e.g. [1–5]). In the following, we go by the structure suggested by vom Brocke et al. [5]: They distinguish between 5 phases that are applied cyclically [5]: 1. definition of review scope 2. conceptualization of the topic, 3. literature search (keyword-based or by screening all papers), 4. literature analysis and review, 5. research agenda. While we experimented with all five phases with various LLMs, we so far only got reasonable results for phase 3 which will be our focus here.

In phase 3, the choice of pertinent keywords is a central problem [3]. The alternative is scanning all publications from all relevant outlets individually, which is often not feasible due to the required time. Here, the possibility to apply NLP methods as an automation option comes into play.

Traditionally, NLP usually starts with the preprocessing of the text that among others can include steps for the removal of "irrelevant" stop-words, syntactical corrections, lemmatization and stemming, i.e. reducing words to their grammar-independent core, a part-of-speech tagging that marks the grammatical role of words and phrases, and the use of thesauri or ontologies to deal with homonyms or synonyms [14, 15]. After that, each unit of text is characterized by a selection of words that sets it apart from the rest, usually with the so called "TF-IDF" metric [16]. The sequence or deeper meaning of the words is discarded; hence this is called a "bag of words" (BoW) approach.

More recent NLP approaches use so called embeddings which are vectors of numbers that are attributed to a unit of text (a string of characters or words – a "token", a sentence or a document) and thereby position the text in a "semantic space", i.e. the vector represents and places the texts meaning [13, 17]. Besides calculating vectors based on the above-mentioned TF-IDF approach, there are two other techniques to create word embeddings [17]:

1. Static Embeddings can be generated using pre-trained models. While there are some large pretrained Static Word embedding models like Googles Word2Vec for domain-specific texts, it is also possible to train own models. The vectors learned can then be used to measure syntactic and semantic word similarity [18].
2. Contextualized Embeddings like ELMo, BERT and GPT-3 are pre-trained models that compute embeddings for a sentence dynamically, taking the context of a word into account [19].

Among others, the embeddings can be applied in similarity searches, for document retrieval and entity extraction, as well as for classification or clustering applications.

Contextualized embeddings can be produced with a "transformer" architecture, a type of artificial neural network that was originally designed for a transformation of sequences into new sequences (seq2seq), e. g. for language translation. A specialty of transformers is that they take a large sequence (a "context window") of text into account at once, calculate the relative importance of all tokens to all other tokens ("attention"/"self-attention") from multiple angles ("multiple attention heads"), and are trained by predicting omitted or subsequent tokens [20]. Recent pre-trained models ("generative pretrained-models", GPTs) have several billion to a few trillion weight/parameters and are trained with enormous text-corpuses. The GPTs are meant to represent foundational models (e. g. OpenAI GPT-3.5, OpenAI GPT-4, Google Bard, Meta Llama/Llama 2 etc.) that can be "fine-tuned" and applied for various "down-stream tasks", like ChatGPT for chat. Pretrained models can be accessed directly or via Application Programming Interfaces (APIs).

3. Related Work

To fathom the state of the art in applying NLP for the automation of literature review tasks in general and literature scanning in particular, we conducted a "traditional" systematic literature review (according to the recommendations of vom Brocke et al. [5]). We used the four databases AIS library (1.287 hits, 5 relevant), IEEE-Xplore (216 hits, 8 relevant), ACM digital library (205 hits, 8 relevant) and Web of Science (937 hits, 29 relevant) with the search string "systematic literature review" AND (automated OR automation OR "large language model" OR llm OR "natural language processing" OR nlp). The resulting pre-selection was deduplicated. After removing papers without full text access and following a more detailed screening, this selection was narrowed down to a total of 21 relevant papers. The following is an overview of the topics studied in those papers.

While we are focusing on the abstract screening step of the SLR, there are more steps that have the potential to be supported by machine learning. Torre-Lopez et al. [21] provided a detailed report over those possibilities for the different phases of the SLR. One of them being the possibility of supporting the generation of the search string [22, 23]. Additionally, there have been multiple studies conducted about the applied NLP techniques, trends, and challenges [6, 24–28]. Furthermore, there have been studies of the automation potential in certain domains e. g. the clinical domain [29, 30].

The possibility to use Machine Learning and NLP algorithms to decrease the effort needed for conducting a SLR has been studied for some time. Initial work was conducted by Cohen et al. [31] using BoW representations of abstracts in combination with Machine Learning. They also introduced a measurement scale that allows to rank models against each other, namely "work saved over sampling" (WSS). It is a weighted variant of the F-measure (2*precision*recall/(precision+recall)) with a threshold of 0.95 for the recall (WSS@95) [31]. WSS@95 is still widely used to benchmark abstract screener models against each other [32, 33], despite of criticism that the ratio of relevant papers in the test-set influences the maximum score that can be achieved using this measure [34].

Later studies also mainly focused on traditional NLP techniques, esp. based on BoW and TF-IDF approaches [6, 35–37]. Due to the mentioned shortcomings of these approaches, they still lack the contextual awareness that Transformer based neural networks provide [19]. Transformers deliver results that are more nuanced by being able to understand the semantics of a text [38]. The only study using contextual embeddings that was found by the SLR was conducted by Alchokr et al. [38] who claimed to have achieved relatively high precision. Yet they were not able to meet the recall score of 0.95 for relevant abstracts proposed by Cohen et al. [31].

In summary, the research into the application of transformer-generated embeddings is still in its infancy with only one relevant source. We build up on these ideas by combining the embeddings with a classification model, namely a Balanced Random Forest, a GPT-based Vector Embedding Model as well as augmenting the initial results with a final classification using a direct application of a GPT-based LLM.

To allow for accessibility and reproducibility [21, 33] we publish our source code on GitHub.[2]

4. Methodology and Solution Design

For answering our research question, we utilized the design science approach [39–41]. We followed the recommendations of Österle at al. [41] who distinguish between the phases of analysis, drafting, and evaluation, although our evaluation is so far still of a preliminary nature.

4.1. Design phase – requirements and approach

As for the design, we identified four core requirements that a tool for supporting or automating the abstract screening process of an SLR should fulfill. These requirements are:

[2] https://github.com/paul-herbst/llm-for-slr

Table 1
Requirements

ID	Requirement
R1	**Reproducibility** a core aim of a SLR is to produce transparent, reliable, and valid results [5]
R2	**Very low percentage of false negatives** not omitting relevant papers is a core concern of a SLR [31]
R3	**Low percentage of false positive**s the aim of the automation needs to be to reduce the manual work as much as possible, therefore avoid false positives
R4	**Efficiency** the solution should apply computing resources in a frugal manner, esp. avoiding the application of LLMs in a large-scale manner if that can be avoided

While standard literature database queries somewhat fulfill R1 and R2, they often generate a high percentage of false positives [42]. With regards to our own literature search for this paper, out of a total of 2.465 hits, only 50 (2%) were at least partially relevant to our SLR topic (21 after further refinement).

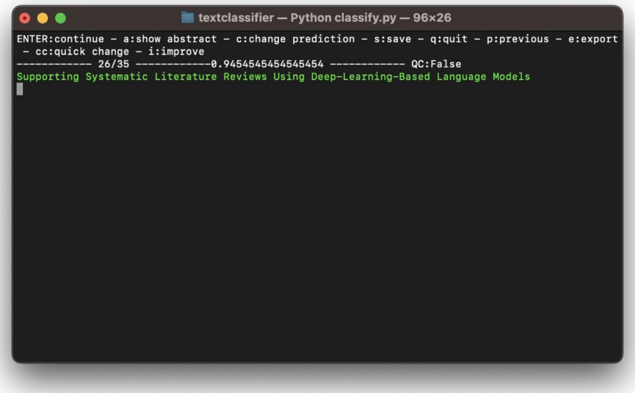
Our search highlights the shortcomings of a keyword-based database query: The database is not capable of processing the difference between our intention to find papers about the *automation of the SLR*, so it returns all the papers that include *a SLR about the topic of automating something*. Similar issues arise whenever the *intended subject of the query* is not the *actual subject of the paper* but rather appears in its context, in an example, or in the framing of a different subject or when a certain degree of abstraction is intended. Other examples include queries for "analytics and artificial intelligence as a research subject rather than a research method", for the "efficiency of method x in general but not only in a specified domain" etc. We deem this as being a typical problem, that results from the inability to address semantic context – an issue that also arises with a traditional NLP BoW approach.

We chose to counter this with an LLM-based approach that incorporates both the language context in general as well as our subject namely relevant abstracts. Since a direct query of an LLM is prone to hallucinations (see section 1), we developed a prototype that melds the natural language understanding capabilities of LLMs with established ML models. More concretely, we chose to apply the OpenAI API for the generation of embeddings of the abstracts which we further processed with a classifier. Note that the results are not vendor or product specific, as similar features can be used in other LLMs, esp. open-source ones like S-BERT and Llama 2. The field is also developing dynamically with new alternatives that are introduced on an almost monthly basis.

The embeddings of the abstracts are further classified for relevance using an established ML approach that – unlike a direct classification with a current LLM – can be applied with reasonable costs and thereby supports RQ4. We decided for a balanced random forest (BRF) [43], as it mitigates the imbalance between irrelevant and relevant papers, which is inherent to most literature reviews. A BRF works like a Random Forest, apart from its tree-building step in which it under-samples the majority class and ensures an equal representation of classes.

This approach is further refined by feeding the classification results to a LLM for a final evaluation. We found that this helped with spotting smaller semantic differences or eliminating errors from the training data. In our case, we prompted OpenAI's GPT-4 model with the instruction to classify all abstracts that got unclear classification results from the random forests. This prompt is formulated in natural language and enriched with 2-5 examples of abstracts labeled as relevant or irrelevant ("few-shot learning"). With this step we are leveraging the capability of the LLM to understand natural language and use it to classify a given text, based on a defined and specific context.

4.2. Drafting phase – prototype



```
ENTER:continue - a:show abstract - c:change prediction - s:save - q:quit - p:previous - e:export
 - cc:quick change - i:improve
------------ 26/35 ------------0.9454545454545454 ------------ QC:False
Supporting Systematic Literature Reviews Using Deep-Learning-Based Language Models
```

Figure 1: Screenshot of the command line interface of the prototype

In the following section, we explain our prototype, beginning with a high-level overview (cf. Figure 2). This is followed by a more detailed look at each of the components.

We implemented a Python-based tool with a command line interface as our prototype (cf. Figure 1). In a first step we extract the abstracts from several literature databases and feed them into a reference management software that is capable of being accessed with an APIs (Zotero). For the next step, we compiled a training dataset of abstracts with known results that we used for training the balanced random forest classifier (training component). The classifier operates with the embeddings (vectors) of the abstracts that are generated by the OpenAI-embedding endpoint. The model is then employed as a preliminary classification tool for narrowing down an extensive corpus of papers retrieved by a full scan of an initial broad search. It filters down the results to only those relevant to the subject at hand. Unclear results then undergo the refinement step that leverages GPT-4 with a few-shot text classification (refinement component).

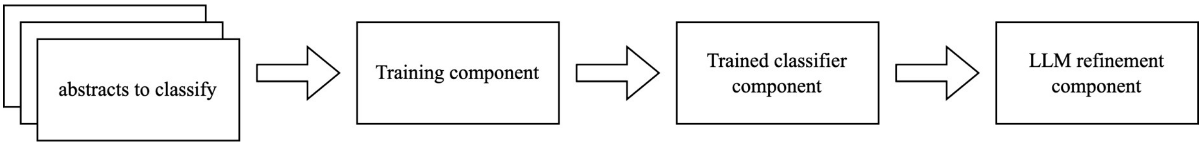


Figure 2: Classification process overview

4.3. Training component

The prototype needs several positive and negative examples for training. Since the classification conducted later is grounded in this foundational data, careful consideration should be given to the selection of these papers. They can for example come from an initial exploratory search or from a previous SLR. Our results indicate that about ten relevant and ten irrelevant papers are sufficient to achieve a satisfactory result. It should be noted that the irrelevant papers should vary in topic and the training examples should include edge-cases.

For each abstract in the training set, the prototype creates an object consisting of the paper's title, its abstract, and its authors. This object is sent to the OpenAI API to create the text embedding which is persisted together with a corresponding relevancy tag. This action can be performed for batches of abstracts at once. The prototype also provides the possibility to update the training set and to revert changes to the training set to a previous point. After that, a Balanced Random Forest is trained using the training vector embeddings. Fig 2 shows an overview of the data preparation steps.

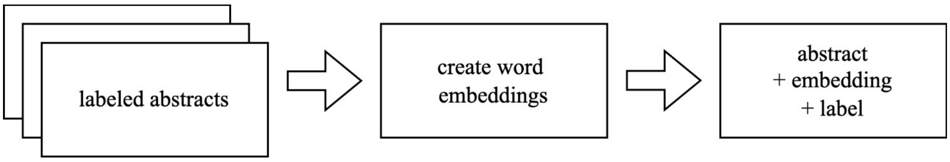


Figure 3: Overview of training data preparation

4.4. (Trained) classifier component

After training the RBF, it can be applied to pre-classify all abstracts stored in the reference management software. This is achieved by creating vector embeddings for each abstract in the same way as for the training data. The BRF is then used to predict the relevancy of the abstracts (it calculates a probability that the abstract is in the "relevant paper class"). In our tests, the following thresholds delivered reliable results: A relevancy score below 0.6 is classified as 'irrelevant' and above 0.75 as "relevant". These thresholds are based on our use of the prototype to provide a balanced tradeoff between the avoidance of false negatives and false positives (RQ2 and RQ3). They can be changed depending on the specific setting and the required rigor of the SLR.

The initial classification already narrows down the number of hits substantially, but it sometimes fails to classify a paper with sufficient certainty, in our case meaning an assigned relevancy value between 0.6 and 0.75.

4.5. LLM refinement component

To further refine the search within this uncertain category, we employed GPT-4. The LLM was fed with a detailed explanation of the SLR topic, along with four examples of abstracts marked as relevant and irrelevant. This additional step helped to further narrow down the search.

The following is an example of the structure of the data provided to the LLM. As we used GPT-4, which is trained as a conversational LLM, we were required to provide the information in a chat-like structure. The prompt we used was *"You are a classifier that predicts whether a paper is relevant based on a prompt. Only ever answer with 'relevant' or 'not relevant'."*

This system-prompt is followed by a "simulated", precomposed conversation between the user and the assistant. The user messages are in the format:

> PROMPT: Is this a paper about automating or semi-automating the process of a systematic literature review? Think carefully and read thoroughly. If the paper is not about the automation potential of SLRs, answer 'not relevant'.
> TITLE: *the title of the paper*
> ABSTRACT: *the abstract of the paper*

For each of these user messages the response of the assistant is either 'relevant' or 'irrelevant', depending on the abstract provided. This is done for two relevant and two irrelevant papers. Figure 4 depicts the structure of the prompt that is provided to the LLM.

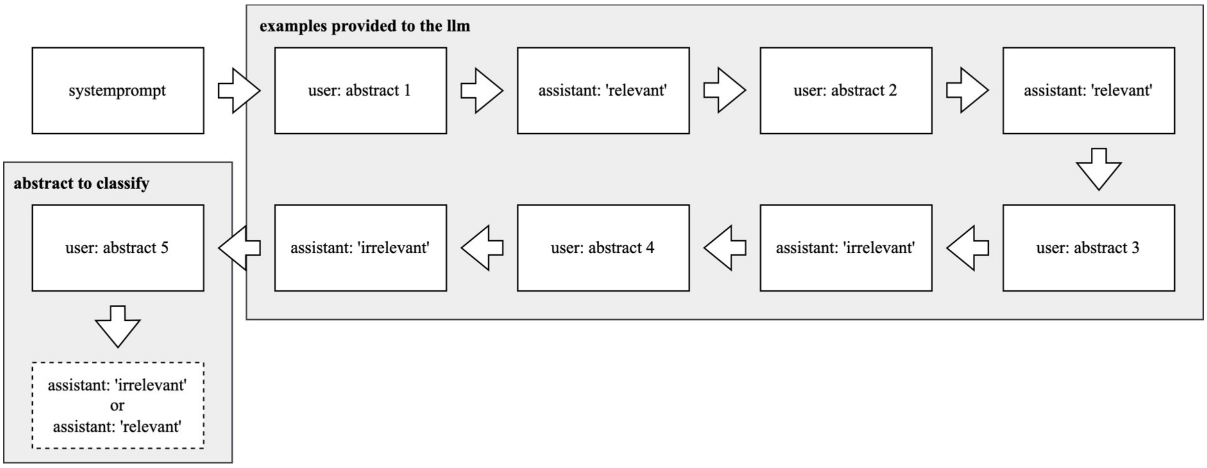


Figure 4: Structure of the few-shot classification prompt

Afterwards the LLM is given the abstracts with uncertain classification results. The outputs of the LLM are parsed and converted. Should the output be anything else than 'relevant' or 'irrelevant', the prototype messages the user and asks for a manual classification. Notably, this did not happen once during our testing. Although it might sound appealing to use this GPT-4 based approach for all papers, it's important to mention that using GPT-4 to classify every abstract in a multi-shot manner would be financially prohibitive in many cases (not matching RQ4). Hence, the rationale behind merging the two approaches, random forest classification and GPT-4 review, for an efficient and cost-effective classification process. Note that these restrictions might become obsolete with future, cost efficient LLMs.

After classifying the abstracts, the user has the possibility to manually make changes to the relevancy classification of the models. After that, the classification can be exported, and the user can decide if the classification should be added to the training set of the Random Forest Classifier.

4.6. Evaluation phase

In our preliminary evaluation, we scrutinized the solution with respect to our four requirements.

RQ1: Reproducibility. A cornerstone of the SLR is the ability of other researchers to validate the findings (RQ1). While LLMs themselves might be a black box and are not inherently explainable, our prototype was developed with reproducibility in mind and allows the exact reproduction of the conducted SLR given the following prerequisites are met:

1. The papers that are analyzed by the prototype must be the same.
2. The papers to train the random forest must be the same.
3. The object that is transformed to a vector needs to be in the same format.
4. The vector produced for each object must be the same as in the original SLR.
5. The "seed" (initialization of the randomization) for the Random Forest must be the same.
6. The system-prompt for the LLM must be the same.
7. The few-shot examples for the LLM must be the same.
8. The hyperparameter "temperature" of the LLM must be set to zero to produce deterministic responses.

This incurs that in a publication of the results of a SLR conducted with this solution, the inputs for the points above are ideally published alongside the results. To facilitate this, our prototype creates a 'receipt' file that documents these parameters and can be used to reproduce the classification. This document could be added to the paper or uploaded to some repository to decrease friction for researchers wanting to reproduce the SLR.

RQ2-RQ4: False positives and false negatives and efficiency. Three researchers applied the prototype in two literature research projects. The average got false positive rates of around 0.5 with minimal false negative rates. It needs to be noted that we were able to reduce the person hours necessary for the literature screening by a factor between 6 and 10.

5. Discussion

In this paper we present a novel approach of leveraging Transformer-Based LLMs to substantially decrease the manual workload of abstract screening during SLRs, while avoiding missing relevant literature (false negatives). This is accomplished by chaining two different classification steps, in form of a Random Forest Classifier to classify contextualized embeddings of the abstracts and GPT-4 in a multi-shot classification form.

Despite all benefits, our approach also comes with some drawbacks. Mainly, the researcher must give away control to a black box. This can also make the SLR less transparent for other researchers. Still, it must be considered, that the existing approach for SLR is not without flaws either.

Our testing showed that LLMs are not yet capable to automate SLR in a “zero-shot” fashion, i. e. just asking for relevant papers for some research topic. However, in our case a 4-example few-shot learning approach already led to satisfactory results when providing an abstract to classify.

As following research steps, we particularly want to address a more rigorous evaluation. We conceive an experiment-based setting in which this LLM-based solution is systematically compared with a manual screening, a keyword-based approach, and a BoW approach.

It needs to be highlighted that the LLM field is continually developing, and it is prudent to assume that within the foreseeable future, they can support or even automate more steps of an SLR, from the support of the conceptualization of the literature review to the proposition of a research agenda.

References

[1] A. Fink, Conducting research literature reviews: from the internet to paper, Sage Publications, Los Angeles, 2014.

[2] J. Webster, R. Watson, Analyzing the Past to Prepare for the Future: Writing a Literature Review, MIS Quarterly 26 (2002) 13–23.

[3] Y. Levy, T. J. Ellis, A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research, InformingSciJ. 9 (2006) 181–212. https://doi.org/10.28945/479.

[4] C. Okoli, K. Schabram, A Guide to Conducting a Systematic Literature Review of Information Systems Research, SSRN Journal (2010). https://doi.org/10.2139/ssrn.1954824.

[5] J. vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, A. Cleven, Reconstructing The Giant On The Importance Of Rigor In Documenting The Literature Search Process, in: Proceedings of the 17th European Conference on Information Systems. , Verona, 2009.

[6] R. van Dinter, B. Tekinerdogan, C. Catal, Automation of Systematic Literature Reviews: A Systematic Literature Review, Information and Software Technology 136 (2021) 1–16. https://doi.org/10.1016/j.infsof.2021.106589.

[7] S. Jusoh, A study on nlp applications and ambiguity problems. Journal of Theoretical and Applied Information Technology 96 (2018) 1486–1499.

[8] L. Galke, A. Diera, B. X. Lin, B. Khera, T. Meuser, T. Singhal, F. Karl, A. Scherp, Are We Really Making Much Progress in Text Classification? A Comparative Review, 2023. URL: http://arxiv.org/abs/2204.03954.

[9] X. Chen, J. Ye, C. Zu, N. Xu, R. Zheng, M. Peng, J. Zhou, T. Gui, Q. Zhang, X. Huang, How Robust is GPT-3.5 to Predecessors? A Comprehensive Study on Language Understanding Tasks, 2023. https://doi.org/10.48550/arXiv.2303.00293.

[10] A. Koubaa, GPT-4 vs. GPT-3.5: A Concise Showdown, 2023. URL: https://www.techrxiv.org /articles/preprint/GPT-4_vs_GPT-3_5_A_Concise_Showdown/22312330/2. https://doi.org/10.36227/techrxiv.22312330.v2.

[11] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, B. Ge, Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models, 2023. URL: http://arxiv.org/abs/2304.01852. https://doi.org/10.48550/arXiv.2304.01852.

[12] R. Qureshi, D. Shaughnessy, K.A.R. Gill, K.A. Robinson, T. Li, E. Agai, Are ChatGPT and large language models "the answer" to bringing us closer to systematic review automation? Systematic Reviews 12 (2023). https://doi.org/10.1186/s13643-023-02243-z.

[13] J. Camacho-Collados, M.T. Pilehvar, Embeddings in Natural Language Processing, in: Proceedings of the 28th International Conference on Computational Linguistics: Tutorial Abstracts, International Committee for Computational Linguistics, Barcelona, Spain (Online) 2020, pp. 10–15. https://doi.org/10.18653/v1/2020.coling-tutorials.2.

[14] D. Sullivan, Document Warehousing and Text Mining: Techniques for Improving Business Operations, Marketing, and Sales, New York, 2001.

[15] N. Milić-Frayling, Text processing and information retrieval, in: A. Zanasi (Ed.), WIT Transactions on State of the Art in Science and Engineering, WIT Press, 2005, pp. 1–45. https://doi.org/10.2495/978-1-85312-995-7/01.

[16] G. Sidorov, Vector space model for texts and the tf-idf measure. in: SpringerBriefs in Computer Science. Springer, 2019, pp. 11–15. https://doi.org/10.1007/978-3-030-14771-6_3.

[17] S. Selva Birunda, R. Kanniga Devi, A Review on Word Embedding Techniques for Text Classification, in: J.S. Raj, A.M. Iliyasu, R. Bestak and Z.A. Baig (Eds.), Innovative Data Communication Technologies and Application, Springer, Singapore, 2021, pp. 267–281. https://doi.org/10.1007/978-981-15-9651-3_23.

[18] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, 2013. URL: http://arxiv.org/abs/1301.3781.

[19] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237. https://doi.org/10.18653/v1/N18-1202.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is All you Need. in: Advances in Neural Information Processing Systems. Curran Associates Inc., 2017.

[21] J. de la Torre-Lopez, A. Ramirez, J.R. Romero, Artificial intelligence to automate the systematic review of scientific literature, COMPUTING, 2023. https://doi.org/10.1007/s00607-023-01181-x.

[22] L. Cairo, G.F. de Carneiro, M.P. Monteiro, F.B. e Abreu, Towards the Use of Machine Learning Algorithms to Enhance the Effectiveness of Search Strings in Secondary Studies, in: Proceedings of the XXXIII Brazilian Symposium on Software Engineering. Association for Computing Machinery, New York, NY, USA, 2019, pp 22–26. https://doi.org/10.1145/3350768.3350772.

[23] A.E. Kwabena, O.-B. Wiafe, B.-D. John, A. Bernard, F.A.F. Boateng, An automated method for developing search strategies for systematic review using Natural Language Processing (NLP), METHODSX 10 (2023). https://doi.org/10.1016/j.mex.2022.101935.

[24] L. Feng, Y.K. Chiam, S.K. Lo, Text-Mining Techniques and Tools for Systematic Literature Reviews: A Systematic Literature Review, in: Proceedings of the 24th Asia-Pacific Software Engineering Conference (APSEC), 2017, pp. 41–50. https://doi.org/10.1109/APSEC.2017.10.

[25] H. Muller, S. Pachnanda, F. Pahl, C. Rosenqvist, The application of artificial intelligence on different types of literature reviews - A comparative study, in: Proceedings of the 2022 International Conference on Applied Artificial Intelligence (ICAPAI). IEEE Norway Sect, CIS Chapter, 2022, pp. 38–44. https://doi.org/10.1109/ICAPAI55158.2022.9801564.

[26] Y. Shakeel, J. Krüger, I. von Nostitz-Wallwitz, C. Lausberger, G.C. Durand, G. Saake, T. Leich, (Automated) literature analysis: threats and experiences, in: Proceedings of the International Workshop on Software Engineering for Science, Association for Computing Machinery, New York, NY, USA, 2018, pp. 20–27. https://doi.org/10.1145/3194747.3194748.

[27] Y. Shakeel, J. Krüger, I.V. Nostitz-Wallwitz, G. Saake, T. Leich, Automated Selection and Quality Assessment of Primary Studies: A Systematic Literature Review, J. Data and Information Quality 12 (2019). https://doi.org/10.1145/3356901.

[28] R. Ros, E. Bjarnason, P. Runeson, 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, Association for Computing Machinery, New York, NY, USA, 2017, pp. 118–127. https://doi.org/10.1145/3084226.3084243.

[29] T. Tsubota, D. Bollegala, Y. Zhao, Y. Jin, T. Kozu, Improvement of intervention information detection for automated clinical literature screening during systematic review, JOURNAL OF BIOMEDICAL INFORMATICS 134 (2022). https://doi.org/10.1016/j.jbi.2022.104185.

[30] M. Michelson, K. Reuter, 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 (2019). https://doi.org/10.1016/j.conctc.2019.100443.

[31] A.M. Cohen, W.R. Hersh, K. Peterson, P.-Y. Yen, Reducing Workload in Systematic Review Preparation Using Automated Citation Classification, J Am Med Inform Assoc 13 (2006) 206–219. https://doi.org/10.1197/jamia.M1929.

[32] A. O'Mara-Eves, J. Thomas, J. McNaught, M. Miwa, S. Ananiadou, Using text mining for study identification in systematic reviews: a systematic review of current approaches, Syst Rev 4 (2015). https://doi.org/10.1186/2046-4053-4-5.

[33] W. Kusa, A. Hanbury, P. Knoth, Automation of Citation Screening for Systematic Literature Reviews Using Neural Networks: A Replicability Study, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg and V. Setty (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham, 2022, pp. 584–598.

[34] W. Kusa, A. Lipani, P. Knoth, A. Hanbury, An analysis of work saved over sampling in the evaluation of automated citation screening in systematic literature reviews, Intelligent Systems with Applications 18 (2023). https://doi.org/10.1016/j.iswa.2023.200193.

[35] A. Bravo, L. Bennetts, P. Atanasov, Accelerating the Early Identification of Relevant Studies in Title and Abstract Screening, in: Prodeedings of the 2021 INTERNATIONAL SYMPOSIUM ON COMPUTER SCIENCE AND INTELLIGENT CONTROLS (ISCSIC 2021), 2021, pp. 132–140. https://doi.org/10.1109/ISCSIC54682.2021.00034.

[36] T. Georgieva-Trifonova, Continued Supporting a Systematic Literature Review by Applying Text Mining Methods, in: 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH), 2022, pp 1–5. https://doi.org/10.1109/INFOTEH53737.2022.9751318.

[37] G. Rizzo, F. Tomassetti, A. Vetro, L. Ardito, M. Torchiano, M. Morisio, R. Troncy, Semantic enrichment for recommendation of primary studies in a systematic literature review, DIGITAL SCHOLARSHIP IN THE HUMANITIES 32 (2017) 195–208. https://doi.org/10.1093/llc/fqv031.

[38] R. Alchokr, M. Borkar, S. Thotadarya, G. Saake, T. Leich, Supporting Systematic Literature Reviews Using Deep-Learning-Based Language Models. in: 2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering (NLBSE), 2022, pp. 67–74. https://doi.org/10.1145/3528588.3528658.

[39] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems 24 (2007) 45–77. https://doi.org/10.2753/MIS0742-1222240302.

[40] A.R. Hevner, S.T. March, J. Park, S. Ram, Design Science in Information Systems Research. MIS Quarterly 28 (2004) 75–105. https://doi.org/10.2307/25148625.

[41] H. Österle, R. Winter, W. Brenner (Eds.), Gestaltungsorientierte Wirtschaftsinformatik: ein Plädoyer für Rigor und Relevanz, Infowerk, Nürnberg, 2010.

[42] H. Scells, G. Zuccon, B. Koopman, Automatic Boolean Query Refinement for Systematic Review Literature Search, in: The World Wide Web Conference, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1646–1656. https://doi.org/10.1145/3308558.3313544.

[43] L. Kobyliński, A. Przepiórkowski, Definition Extraction with Balanced Random Forests, in: B. Nordström and A. Ranta (Eds.), Advances in Natural Language Processing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp.237–247. https://doi.org/10.1007/978-3-540-85287-2_23.