Supporting Systematic Literature Reviews Using Deep-Learning-Based Language Models

Rand Alchokr
Otto-von-Guericke University
Magdeburg, Germany
rand.alchokr@ovgu.de

Manoj Borkar
Otto-von-Guericke University
Magdeburg, Germany
manoj.borkar@st.ovgu.de

Sharanya Thotadarya
Otto-von-Guericke University
Magdeburg, Germany
sharanya.hunasamaranahalli@ovgu.de

Gunter Saake
Otto-von-Guericke University
Magdeburg, Germany
saake@ovgu.de

Thomas Leich
Harz University & METOP GmbH
Weringerode & Magdeburg, Germany
tleich@hs-harz.de

ABSTRACT

Background: Systematic Literature Reviews are an important research method for gathering and evaluating the available evidence regarding a specific research topic. However, the process of conducting a Systematic Literature Review manually can be difficult and time-consuming. For this reason, researchers aim to semi-automate this process or some of its phases. **Aim**: We aimed at using a deep-learning based contextualized embeddings clustering technique involving transformer-based language models and a weighted scheme to accelerate the conduction phase of Systematic Literature Reviews for efficiently scanning the initial set of retrieved publications. **Method**: We performed an experiment using two manually conducted SLRs to evaluate the performance of two deep-learning-based clustering models. These models build on transformer-based deep language models (i.e., BERT and S-BERT) to extract contextualized embeddings on different text levels along with a weighted scheme to cluster similar publications. **Results**: Our primary results show that clustering based on embedding at paragraph-level using *S-BERT-paragraph* represents the best performing model setting in terms of optimizing the required parameters such as correctly identifying primary studies, number of additional documents identified as part of the relevant cluster and the execution time of the experiments. **Conclusions**: The findings indicate that using natural-language-based deep-learning architectures for semi-automating the selection of primary studies can accelerate the scanning and identification process. While our results represent first insights only, such a technique seems to enhance SLR process, promising to help researchers identify the most relevant publications more quickly and efficiently.

CCS CONCEPTS

• **General and reference**; • **Software and its engineering**;

KEYWORDS

Systematic Literature Review, Language Models, Deep Learning, BERT

ACM Reference Format:
Rand Alchokr, Manoj Borkar, Sharanya Thotadarya, Gunter Saake, and Thomas Leich. 2022. Supporting Systematic Literature Reviews Using Deep-Learning-Based Language Models. In *The 1st Intl. Workshop on Natural Language-based Software Engineering (NLBSE'22), May 21, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3528588.3528658

NLBSE'22, May 21, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9343-0/22/05.
https://doi.org/10.1145/3528588.3528658

1 INTRODUCTION

Systematic Literature Reviews (SLRs) are a well-established research method used within the evidence-based research paradigm. In most scientific fields, such as medicine and software engineering, Systematic Literature Reviews have become a popular method for conducting literature analyses and meta-studies to aggregate evidence on a topic [17, 32]. Such types of studies are helpful to synthesize and assess the available empirical data. However, despite the methodological benefits of an Systematic Literature Review, it can be an extremely tedious and time-consuming process — especially with the number of scientific publications constantly increasing every year. This makes the task of identifying the most relevant publications an exceedingly difficult task for any researcher. Consequently, there has always been a need for automating Systematic Literature Reviews, and various tools provide different types of support [14, 22, 31]. Generally, tools assisting SLRs in retrieving publications based on a query that uses *term frequency and inverse document frequency* (TF-IDF) to identify similar publications, and display these to the user. However, one of the main drawbacks of such existing tools is their inability to capture semantics of the actual texts, which could help to improve the correct selection of publications according to the specified query and publications previously selected by the user.

In this paper, we address the tedious process of identifying relevant primary studies during the conduct phase Figure 1 of an Systematic Literature Review. For this purpose, we employ the state-of-the-art transformer-based BERT model variants BERT [7] and S-BERT [30] at different granularities of the input text, i.e. word, sentence, and paragraph to semi-automate this phase, and thus support the entire Systematic Literature Review method. To evaluate our proposed technique, we compare the models' resulting clusters of publications with the outcomes of two Systematic Literature

Reviews conducted manually by researchers (i.e., we compare the publications selected by the models and the researchers). In detail, our contributions are:

- We propose a semi-automated technique to cluster the initial set of publications of an Systematic Literature Review using contextualized embeddings obtained from variants of the BERT language model.
- We introduce a weighting scheme to add value to publications close to the Systematic Literature Review search string.
- We evaluate our technique based on an empirical comparison with two existing manually conducted Systematic Literature Reviews.
- We publish a persistent open-access repository comprising the source code and datasets we used for the experiments and evaluation[1].

Our results indicate that our proposed technique is a promising way to improve the automation of conducting Systematic Literature Reviews where clustering based on embedding at paragraph-level using *S-BERT-paragraph* represents the best performing model setting in terms of optimizing the required parameters such as correctly identifying primary studies, number of additional documents identified as part of the relevant cluster and the execution time of the experiments. We sketch our further plans of improving, integrating, and evaluating our technique in more detail, which highlights interesting research opportunities.

In Section 2, we discuss Systematic Literature Review methodology with relevant work of automation tools, an overview of language models and neural networks. We describe our contribution to the conducting phase of the Systematic Literature Review approach and the models adopted to generate the embeddings in Section 3. Next in Section 4, we discuss the evaluation of our experiments and its results. To our knowledge, this is one of the first works involving deep learning infrastructures with Systematic Literature Reviews. In Section 5, we list several threats to validity and finally we conclude in Section 6.

2 BACKGROUND AND RELATED WORK

In this section, we discuss the Systematic Literature Review methodology and previous work related to its automation. We also discuss some recently proposed techniques related to neural network and language models.

2.1 Systematic Literature Review

Typically any scientific research work begins with a literature review. However, a literature review is not considered of a significant scientific value if not conducted properly following specific guidelines. An Systematic Literature Review has three main phases, where each of them is divided into several steps, as proposed by Kitchenham and Charters [18], and summarized in Figure 1.

(1) Planning: involves identifying the need for a review, establishing research questions and search queries, and developing review protocols.
(2) Conducting: involves the retrieval of relevant research, selecting the primary studies, and data synthesis.
(3) Reporting: involves formulation of the final report with the findings and analysis of the gathered evidence.

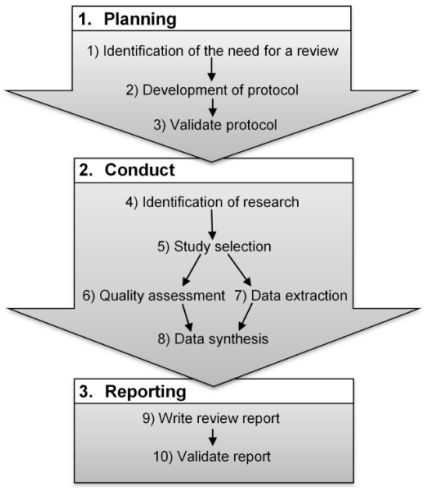

Figure 1: The three phases involved in a Systematic Literature Review process.

The goal of an Systematic Literature Review is to summarise existing evidence and identify gaps. However, the labor-intensive, error-prone, and time-consuming nature of this method has led researchers to develop a range of tools to provide support by automating parts of its process. (SLuRp) [3], (StArt) [15], (SLR-Tool) [11] are some of the very few tools that support all stages of the Systematic Literature Review. Although these tools support the entire Systematic Literature Review process, some of them have difficulties such as, complex installation procedures or lack of support for protocol development or collaboration. Some of the proposed tools majorly contribute to the conducting phase, where the main focus lies on primary studies' selection and data synthesis. While other tools, such as (PEx) [9] and (ReVis) [8, 10] employ text mining and visualization techniques to construct mappings between the research documents, site content analyzer tool uses entity recognition methodologies [5] and SLRONT [6] uses evidence-based ontology to support Systematic Literature Reviews.

Several studies evaluated some of the available tools based on a criteria of features [23, 24]. Their results show a need for more research employing new techniques to improve the effectiveness and efficiency in identifying the relevant primary studies in the conducting phase. A required improvement would be capturing the semantic of the text. This learning of contextualized relationships at word, sentence, and paragraph level is made possible through the integration of deep learning architectures in the form of language models, and this is what we intend to achieve through our work.

2.2 Language Models

Over the last few years, language models ,which are unsupervised multi-task learners, have become a crucial component of Natural

[1] https://doi.org/10.5281/zenodo.6356195

Language Processing. They have been employed at the foundation of several Natural Language Processing tasks, such as, question answering, machine translation, reading comprehension, and summarization. These tasks which were typically approached with supervised learning on task-specific datasets are now tackled unsupervised by using language models. These models are pre-trained on gigantic amount of free-text data to learn predicting the probability of a sequence of words.

The earlier forms of language models used statistical techniques like linguistic rules, N-grams, and Hidden Markov Model (HMM) to learn the probability distribution over a sequence of words. Some count-based models such as Principal Component Analysis (PCA), and topic and neural probabilistic models [12, 27] involve unsupervised learning based on word frequency and global word co-occurrence matrix. The theory under study is that the words that occur together in the same context share similar and related semantic meanings. Context-based models, such as Skip-Gram [26] and Continuous Bag-of-Words (CBOW), model the text data using a sliding window of appropriate size that moves along the sentences and are trained to predict the probabilities of a target word/words given the rest of the context within the sliding window. Additionally, language models such as GloVe (Global Vectors) combine ideas of count and context-based models [13].

2.3 Neural Networks in NLP

Neural networks have shown important capabilities and effectiveness over the recent years in modeling language. Neural machine translation models were initially meant for language translation tasks [25]. They set the trend for using the word embeddings learned by the encoder of a bi-directional (LSTM) autoencoder for downstream tasks. Embeddings from language models (Elmo) picked this up a notch via producing contextualized word-level representations by pre-training the model in an unsupervised way [28]. Cross-View training is a semi-supervised learning approach where representations are improved by training in a supervised and unsupervised manner [4].

The current state-of-the-art language models utilize components of transformer architecture which inculcates multi-head self-attention mechanisms, thus helping the model learn long-range dependencies or relationships between words and sentences over longer sequences. Language models such as OpenAI-GPT [16], Bidirectional Encoder Representations from Transformers (BERT) [7], ALBERT [19], RoBERTa [21], and Text-to-Text Transfer Transformer (T5) [29] are trained on a giant collection of free text corpora and come in multiple variants where contextual relationships and meanings are learned via techniques and tasks, such as masking, sentence order prediction, and translation.

Li et al. [20] employed a pre-trained BERT model to obtain sentence level embeddings and clusters on them to form sets of documents. While BERT has been trained with masked sequence and next sentence prediction tasks, Sentence-BERT (S-BERT) [30] is a modification of the BERT network to derive contextual and meaningful sentence embeddings using siamese and triplet networks. Additionally, Li et al. [20] used named entities as part of the weightage schemes to enhance the embeddings for clustering. We, on the other hand, explore clustering documents based on word, sentence, and paragraph-level embeddings and define our own weighting schemes based on the search string designed during the planning phase of the Systematic Literature Review process.

3 EXPERIMENTAL SETUP

In the following section, we explain our experiments beginning with identifying the dataset we use and the three basic modules we employ. These modules, as illustrated in Figure 2, involve:

1) Contextualized embeddings modules for getting various levels representations.
2) Weighting schemes module.
3) Clustering module.

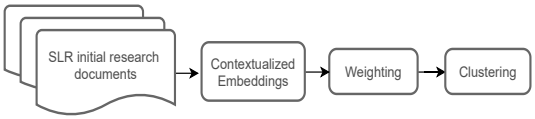

Figure 2: Overview of our Framework.

3.1 Dataset

The dataset used in our experiments comprises of two collections, each belongs to an Systematic Literature Review study:

SLR_1 "Bad Smell Detection Using Machine Learning Techniquesa systematic literature review", 2019 [1]: 153 documents as an initial set of results from Scopus[2] and 12 out of them were manually identified as primary studies.

SLR_2 "Automatic software refactoringa systematic literature review", 2019 [2]: 174 documents as an initial set of results from Scopus, out of which 9 are manually identified as primary studies.

For each Systematic Literature Review, we replicated the search methodology as their respective authors originally performed by applying the search string on Scopus and retrieving the initial set of results. The bibliographic information could be exported in .bib or .csv formats. This resulting corpus is considered as the dataset for our experiments. Each of the retrieved documents have entries, such as title, abstract, author, keywords, year of publication, publisher, venue, ISBN, and journal data.

As part of data gathering and cleaning, we first checked the dataset for duplicates, incomplete entries, and missing information. Then we extracted the text entries only, namely: title, abstract, and keywords. We retain only the ones that have complete data for title and abstract, as they are essential for learning the document-level embeddings. However, incomplete and missing keyword entries are not considered as part of the retention criteria, because often a large number of research articles do not include keywords or index terms, and eliminating such documents may lead to poor identification of relevant research.

Finally, after extracting and cleaning the data, each dataset contains four columns as illustrated in Table 1- **title**, **abstract**, **keywords**, and **cluster labels**. These cluster labels were manually added to indicate whether this document is among the primary

[2] https://scopus.com

studies set based on the decision of the researchers conducting the Systematic Literature Review. All documents are labeled as relevant (1), whilst the rest of the documents are assigned a non-relevant label (0).

Table 1: Overview of the document entries we extracted from Scopus.

Variable name	Description
Title	The title of the document.
Abstract	The abstract of the document.
Keywords	The identified keywords of the document.
	Added manually:
Cluster label	A Boolean indicator–> primary study (1) or not (0).

3.2 Contextualized Word Embeddings - Documents Embedding Module:

Existing Systematic Literature Review tools retrieve documents based on the query string that uses (TF-IDF) where the similarity or dissimilarity of documents depends only on word frequencies. Inability to capture the semantics between text documents is one of the main drawbacks of the existing tools. To overcome this problem, here we propose to use contextualized word embeddings to capture the semantic similarity between documents. Contextual embeddings capture the usage of words across varying contexts and encode this information in a vector. This vector representation of a word changes when the context in which this word is used changes. We argue that using contextualized embeddings to represent documents will help end-users gain a better understanding of the underlying document space.

As we aim to understand how well the state-of-the-art deep learning architectures in the field of Natural Language Processing can help in capturing the intrinsic relationship between the documents. The documents embedding module forms the basis for learning clusters and grouping documents using the learned contextualized representations. In our experiment, we use BERT [7] and S-BERT [30] as our pre-trained language models to generate document-level contextualized embeddings. As mentioned before, BERT has been trained with masked sequence and next sentence prediction tasks. However, S-BERT is a modification of the BERT network to derive contextual and meaningful sentence embeddings using siamese and triplet networks. For implementation we used Python and Scikit-learn[3] which is a free software machine learning library.

3.2.1 Data pre-processing. To explore what level of embeddings best captures the contextualized relationships between documents, we designed our experiments to accommodate learning of contextualized meanings at word, sentence, and document levels. In the case of BERT model, we design the input pipeline to obtain word and sentence level embeddings, while for S-BERT we modify the pipeline to obtain sentence and document level embeddings.

As shown in Figure 3, for word and sentence level embeddings, we break the document text into groups of individual words and sentences, respectively. However, at the document level, we treat the entire chunk of title and abstract of a document as one big sentence. Additionally, only in the case of BERT model, we tokenize the words and add special tokens such as [CLS], [SEP], and [PAD] to establish the start and end of sentences for the model to learn sentence level embeddings.

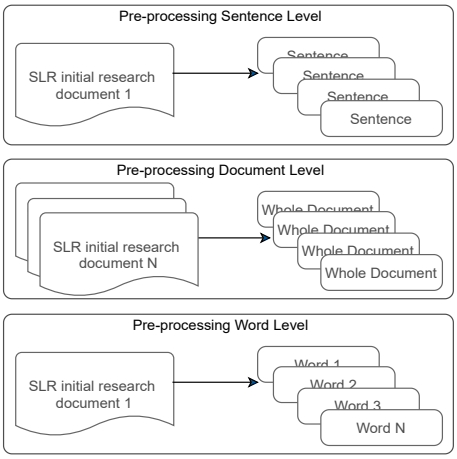

Figure 3: Pre-processing data for BERT, S-BERT models.

3.2.2 BERT - Transformer based language model. The architecture of BERT makes use of the transformer encoders to derive a language model trained on a corpus of billions of words. The usage of bi-directional multi-layer transformer encoders to capture the left-to-right context, as well as the right-to-left context, gives it the potential to capture long-term dependencies over past and future. To encourage the bi-directional nature, instead of training to predict the target given the context, BERT is trained on two tasks: *Masked Language Model* and *Next Sentence Prediction* - The former involves randomly masking a certain percentage of tokens in the sequence allowing the model to learn by predicting masked words without having prior information about the replaced words in the sequence. The second is inspired by the logical understanding that the downstream tasks rely more on sentence relationship than at word level and involve an auxiliary task of training a binary classification model to predict the next sentence given the previous sentence. BERT makes use of the same base model for all kinds of end-tasks, eliminating the idea of a task-specific model and improves upon the uni-directional nature of contextual understanding. It can be used for a large variety of Natural Language Processing tasks, often requiring a minimal amount of changes in the architecture, and have very few parameters to learn for each of the sub-tasks.

In our experiments with BERT for obtaining word-level embeddings, we encode the title and abstract of the documents and concatenate over the last four hidden layers' outputs to derive a fixed-sized embedding of 768 dimensions. For each document, we average over all the word embeddings for words occurring in the documents. Similarly, for sentence-level embeddings, we break the text content(title and abstract) of documents into sentences and for each sentence, we derive a fixed-size embedding of 768 dimensions

[3] https://scikit-learn.org/stable/

and average over all the sentence embeddings for each document to obtain document-level embeddings from them.

3.2.3 Sentence-BERT (S-BERT).. S-BERT was fine-tuned on the Natural Language Inference (NLI) dataset using siamese and triplet network structures [30]. It was basically designed to perform Natural Language Processing tasks in low computational time. In S-BERT, a pooling operation is added to the output of BERT to generating a fixed-sized sentence embedding. S-BERT has three pooling strategies (i.e *[CLS] Token strategy, Mean strategy, and MAX strategy*), evaluated on the STSbenchmark datasets[4]. The results showed that the *Mean pooling strategy* outperformed the other two strategies. Hence, we have used it in our experiments. The generation of document embeddings for a given document is carried out by a three-steps process. Figure 4 gives an overview of the implementation technique.

1) The text inside the document is split into sentences using the *Spacy sentence tokenizer* as sentences with a length greater than a certain threshold are removed. Based on the results of our experiments we set the threshold value to 20.
2) Each sentence is passed to S-BERT to derive a fixed-sized embedding of 768 dimensions.
3) Sentence embeddings obtained for the sentences belonging to a document are then averaged to extract a document level embedding.

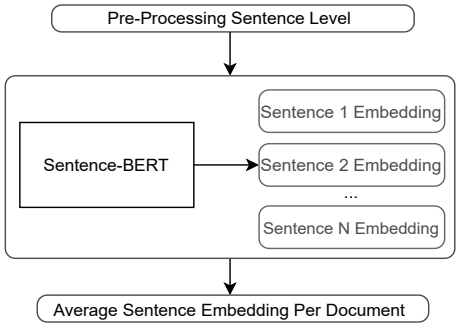

Figure 4: S-BERT implementation technique.

3.3 Weighting Schemes Module:

Systematic Literature Review mainly involves finding relevant research documents, given a search string. The primary studies are the relevant research documents among the obtained initial set of papers. These are manually identified and labeled as relevant(1), meaning relevant for the study. Our weightage scheme is designed with the idea of recognizing relevant documents closer to the search string, unique words recognized by the highest (TF-IDF) scores and the search document.

The search string for finding primary studies in the conducting phase of the Systematic Literature Review contains keywords. We make use of this search string to identify the main keywords and create a list of study-relevant keywords. We also construct another keyword list, consisting of words that are identified by the highest (TF-IDF) scores among the entire set of titles and abstracts relevant for the study. For every sentence in the document (D), a count N_D keeps track of the number of words from the search string and/or the number of words from the (TF-IDF) keyword list, that matches with the words in the sentences of (D). The Sentence weight is:

$$W_s = \frac{N_k + 1}{N_s} \tag{1}$$

N_k denotes the number of matching words in a sentence with the search string keyword list and/or (TF-IDF) keyword list.
N_s denotes the total number of words in the sentence.
W_s denotes the weighted contribution of each sentence to the document.

The inclusion of length of words for each sentence, while calculating the W_s, ensures that the weights are normalized and therefore can be simply averaged over for calculating the document level embeddings from these sentences.

The document embedding using the weightage scheme is:

$$W_d = \frac{\Sigma_{s \in D}\ W_s \times E_s}{\Sigma_{s \in D}\ |s|} \tag{2}$$

$s \in D$ indicates each sentence that belongs to the document.
W_s denotes the weighted contribution of each sentence to the document, calculated in the previous step.
E_s denotes the sentence embedding obtained from embeddings module.
W_d denotes the weighted document embedding.

Figure 5 illustrates BERT, S-BERT weightage module.

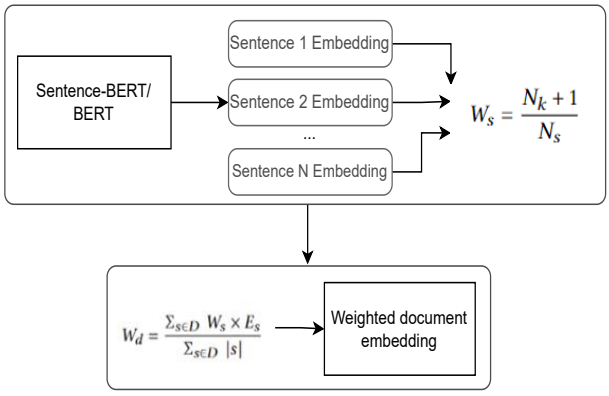

Figure 5: BERT, S-BERT Weightage module.

3.4 Document Clustering Module:

To recognize relevant and non-relevant documents during the conducting phase of the Systematic Literature Review, we use the clustering method. Clustering is a technique used in machine learning to group similar objects. The objects which are in the same cluster are more similar to each other as compared to the objects present in other different clusters. There are different kinds of clustering algorithms such as K-means clustering, Density-Based Spatial clustering, HDBSCAN, and Hierarchical clustering. In our experiments, we use *K-means* algorithm to group similar documents. We chose

[4] http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark

K means for below two reasons i.e. since the number of documents are expected to grow exponentially K means method can easily adapt to new instances and with respect to time complexity they are scalable for large data sets. The basic algorithm for K-means is described below in algorithm 1.

As an input to the clustering module, we use one of the weighted or unweighted document level embeddings - obtained directly in case of paragraph-level or averaged over words and sentences in the document for word and sentence-level structure respectively). We first identify the appropriate number of clusters for the embeddings using knee locator of *Kneedle algorithm* made available via Python kneed library[5]. This elbow determines the k value of the k-means algorithm to divide our embeddings into k clusters. Once k clusters are formed, we identify the cluster which consists majority of the documents labeled as primary study as per the true labels. This forms our relevant cluster.

Algorithm 1 Basic Algorithm for K-means

Input: Document Embedding vectors and number of clusters i.e. K

Output: K clustered sets

0: **procedure** K-MEANS
1: *Randomly select K centroids*
2: *Calculate the similarity of each Document Embedding with K centroids*
3: *Assign the Document Embedding vector to the cluster centroid whose distance from the cluster centroid is minimum of all the cluster centroids*
4: *Recompute the centroids of the newly formed clusters*
5: **repeat**
steps 3 and 4
6: **until** *Centroids of newly formed clusters do not change*

4 RESULTS AND DISCUSSION

The designed experiments focus on exploring the possibilities over multiple configurations in different modules. Firstly, we experiment with the input data provided to determine how language models perform on different lengths and sorts of text. There are three ways in which we associate every document with a sequence of text.

1) Only the title is considered, we noticed that even though it has the shortest sequence length, it contains important keywords that define the document.
2) Only the abstract is used to represents the document as input data. Abstracts are slightly longer sequences of text in comparison to the title and often contain multiple sentences to form sentence-level relationships within the document set.
3) Both title and abstract are combined with subsequent word separators to form the biggest sequences of input data to be passed to the module.

In the document embeddings module, we use variants of BERT and S-BERT to compare how well they perform against each other in providing document-level embeddings from words, sentences and paragraphs. Through the weighting schemes, we wanted to explore if the identification of relevant documents became more prominent. We compare the clusters obtained using unweighted and weighted scheme approaches.

For evaluating the clusters on document embeddings, we use *Fowlkes-Mallows Index* (FMI), which is generally used when the ground truth labels are known. In our dataset, the labels are identified and annotated manually as relevant (1) and non-relevant (0) as the primary studies to the respective SLR study. FMI is defined as the geometric mean of the pairwise precision and recall:

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \tag{3}$$

- TP : True Positives (i.e. number of pair of points that belongs to the same clusters in both true and predicted labels)
- FP : False Positives (i.e. the number of pair of points that belongs to the same clusters in true labels and not in the predicted ones)
- FN : False Negatives (i.e the number of pair of points that belongs in the same clusters in the predicted labels and not in the true labels)

We calculate the FMI scores for the entire document set and the score ranges from 0 to 1. A high value indicates a good similarity between the two clusters.

In our **first dataset** SLR_1, 12 documents which are manually identified as primary studies form our relevant cluster. As noticed, BERT-UW and our TF-IDF(baseline) achieve good results in terms of identifying 11 out of 12 documents in the relevant cluster with a FMI score of 0.661 and 0.741 for BERT-UW and TF-IDF(baseline) respectively for the whole studies set. However, it is important to consider the number of additional documents as part of the relevant cluster, as it relieves the tedious process of going through extra documents which are irrelevant for the given SLR. Therefore, there needs to be a good balance between identifying correct documents from relevant cluster and not identifying large number of additional documents as part of that cluster. Considering this, S-BERT-W outperforms BERT-UW, TF-IDF(baseline) and all other models with only 10 additional documents and by identifying 8 out of 12 primary documents. This is also in accordance with the highest FMI score of 0.899.

For the **second dataset** SLR_2, TF-IDF(baseline) achieves better results than the rest of the models in terms of identifying 8 out of 9 primary study documents, but when we consider additional documents, the S-BERT-paragraph model performs better as it identifies 6 out of 9 primary studies and in addition, just has 22 other documents as part of the relevant cluster in comparison to TF-IDF(baseline) which clusters 104 other documents in addition to 8 primary study ones. It should also be noted that, SBERT-paragraph which gives out embeddings at the paragraph level, outperforms the rest at word or sentence level. This is also in accordance with the highest FMI score of 0.851.

Based on the performance, we can safely say that, although TF-IDF model fetches better results in terms of identifying more primary studies, it comes with the baggage of carrying a large number of additional documents along with it. On considering the ratio of primary studies to non-primary studies in the relevant cluster (i.e the number of additional documents), we argue that for each dataset, one of the settings with deep language models

[5] https://pypi.org/project/kneed/

Table 2: Results for SLR_1 - Total Documents = 153, Primary studies = 12.

Model	Data	# Level	Correct PS	FMI Score	# additional docs	# Clusters
TF-IDF(baseline)	Title+Abstract	Word	11/12	0.741	39	3
BERT_UW	Title+Abstract	Word	11/12	0.661	63	3
BERT_W	Title+Abstract	Sentence	8/12	0.774	30	3
SBERT_W	Title+Abstract	**Sentence**	**8/12**	**0.899**	**10**	4
SBERT_UW	Title+Abstract	Sentence	10/12	0.728	41	4
SBERT_Paragraph	Title+Abstract	Paragraph	8/12	0.833	20	3

UW = UNWEIGHTED, W = WEIGHTED, Correct PS = Correctly identified Primary Studies

Table 3: Results for SLR_2 - Total Documents = 174, Primary studies = 9.

Model	Data	# Level	Correct PS	FMI Score	# additional docs	# Clusters
TF-IDF(baseline)	Title+Abstract	Word	8/9	0.690	104	2
BERT_UW	Title+Abstract	Word	6/9	0.752	43	4
BERT_W	Title+Abstract	Sentence	6/9	0.699	59	3
SBERT_W	Title+Abstract	Sentence	5/9	0.745	44	3
SBERT_UW	Title+Abstract	Sentence	6/9	0.796	33	4
SBERT_Paragraph	Title+Abstract	**Paragraph**	**6/9**	**0.851**	**22**	5

UW = UNWEIGHTED, W = WEIGHTED, Correct PS = Correctly identified Primary Studies

(BERT and S-BERT when combined with Weighted or Unweighted schemes), always achieves good results by identifying very few additional documents in the primary study set and outperforms the TF-IDF(baseline) in finding a good balance. However, the robustness and consistency of the models with different settings, can only be tested as the experiments are extended to more and varied datasets within the extent of SLR study.

Another interesting notion that comes into the picture, when we have to deploy the model into production is the execution time. This becomes even more important when the number of initial search documents extracted or the text corpus is significantly larger. In terms of execution time, S-BERT-paragraph setting has lower values in comparison to other settings of the models. Therefore, we consider the proposed method can help in accelerating the SLR process. Despite the acceptable performance of the proposed method, there is a potential risk of not extracting all the available primary studies on the target topic of the SLR and to guarantee the benefits of this method, more evaluation with a larger dataset is needed.

5 THREATS TO VALIDITY

We replicated two Systematic Literature Reviews and compared our modules' selection and clustering of primary studies with the manual results achieved by the researchers. One potential threat could relate to deploying the search strings on digital libraries. The obtained results may change over a period of time resulting in differences in the datasets' results. Additionally, in our experiments, we made use of pre-trained models which have several advantages in terms of execution time and ease of deployment and usage, however, when the number of initial documents is significantly larger, it makes more sense to fine-tune the model to the dataset and its vocabulary. Therefore, the evaluation so far needs more development, as it is infeasible to manually evaluate the method at large-scale. Adoption of analysis tools that can support the evaluation process, would considerably help research in this area and generalize our approach to other domains.

6 CONCLUSION

In this paper, we present a novel approach to semi-automate the conducting phase of the Systematic Literature Review where identification and synthesis of relevant research called primary studies are of key interest. We simplify this process by integrating deep learning-based language models, specifically BERT and S-BERT in the setup. These models are utilized to extract contextualized embeddings from the title and abstract of documents at word, sentence and paragraph levels. We also utilize a weightage scheme to give preference to documents that are in close relation with the Systematic Literature Review search string, and finally cluster the document level embeddings using the k-means algorithm to find clusters of similar documents.

To accomplish our tasks, we used two datasets each related to an Systematic Literature Review study. Firstly, Our results demonstrate that clustering on contexualized embeddings obtained via language models(BERT and S-BERT) outperforms our baseline model(TF-IDF) having a good trade-off between identifying the relevant set of primary studies and number of additional documents retrieved as part of the relevant cluster. In terms of specific model settings, S-BERT-paragraph and S-BERT weighted generally have a good trade-off when it comes to correctly identifying most of the primary study documents as relevant research, identifying less number of additional documents and outperform other settings in terms of execution time. The results also outline that the weightage schemes are inconsistent in comparison to unweighted ones in cases of both BERT and S-BERT models. While for the first dataset, the weighted approach identifies lesser documents from the set of primary study. For the second dataset, the unweighted approches for both BERT and SBERT models are better. However, this cannot be concluded with experiments performed over just two datasets.

To conclude, few of the experiments performed on our entire document set revealed promising results and could be used as a good starting point for identifying the primary studies for Systematic Literature Review. Learning contextualized relationships between

documents using deep learning based architectures such as language models helped identifying most of the primary studies and also helped in dealing with the drawbacks of the manual process of the Systematic Literature Review conducting phase generally.

7 FUTURE WORK

Engaging contextualized embeddings using deep learning based architectures such as language models in Systematic Literature Review is a promising aspect. More work is needed to explore the potential benefits of deep learning text analysis capabilities. This work paves way for ongoing research in this direction. An essential future experiment would be utilizing the full document text, instead of just the title and abstract. Another interesting area of exploration pertains to topic modeling experiments that we currently start investigating. Additionally, improving the evaluation part of the method by adopting analysis tools is an essential requirement. A bigger SLR corpus covering more research problem domains are needed to see consistent insights from the results of the BERT models. And as the research in the area of Natural language processing shows promising rise, more and more language based models such as XLNet, GPT-2, RoBERTa etc., which are pre-trained on humongous corpus are made available and can be explored as part of the future work in this domain. Furthermore, language models such as SciBert trained specifically on scientific text can further improve our work. Therefore, to accelerate the process we invite interested researchers to join us in this research to provide a better level of reliability in enhancing the conducting phase of Systematic Literature Review methodology.

REFERENCES

[1] Ahmed Al-Shaaby, Hamoud Aljamaan, and Mohammad Alshayeb. 2020. Bad Smell Detection Using Machine Learning Techniques: A Systematic Literature Review. *Arabian Journal for Science and Engineering* 45 (01 2020).

[2] Abdulrahman Baqais and Mohammad Alshayeb. 2020. Automatic software refactoring: a systematic literature review. *Software Quality Journal* 28 (06 2020).

[3] David Bowes, Tracy Hall, and Sarah Beecham. 2012. SLuRp - A tool to help large complex systematic literature reviews deliver calid and rigorous results. In *EAST'12 - Proc. 2nd Int. Work. Evidential Assess. Softw. Technol.* 33–36.

[4] Kevin Clark, Minh Thang Luong, Christopher D. Manning, and Quoc V. Le. 2020. Semi-supervised sequence modeling with cross-view training. In *EMNLP 2018*. 1914–1925.

[5] Daniela Cruzes, Manoel Mendonça, Victor Basili, Forrest Shull, and Mario Jino. 2007. Automated Information Extraction from Empirical Software Engineering Literature: Is that possible?. In *ESEM 2007*. 491–493.

[6] Daniela Cruzes, Manoel Mendonça, Victor Basili, Forrest Shull, and Mario Jino. 2007. Using context distance measurement to analyze results across studies. In *ESEM 2007*. 235–244.

[7] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019*. 4171–4186.

[8] Katia R. Felizardo, Gabriel F. Andery, Fernando V. Paulovich, Rosane Minghim, and José C. Maldonado. 2012. A visual analysis approach to validate the selection review of primary studies in systematic reviews. *Information Software Technology*. 54, 10 (2012), 1079–1091.

[9] Katia Romero Felizardo, Elisa Yumi Nakagawa, Daniel R.C. Feitosa, Rosane Minghim, and José Carlos Maldonado. 2010. An Approach Based on Visual Text Mining to Support Categorization and Classification in the Systematic Mapping. In *EASE 2010*.

[10] Katia R. Felizardo, N. Salleh, Rafael M. Martins, Emilia Mendes, Stephen G. Macdonell, and José C. Maldonado. 2011. Using visual text mining to support the study selection activity in systematic literature reviews. In *ESEM 2011*. 77–86.

[11] Ana M. Fernández-Sáez, Marcela Genero Bocco, and Francisco P. Romero. 2010. SLR-Tool a tool for performing systematic literature reviews. In *ICSOFT 2010*, Vol. 2. 157–166.

[12] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Journal of Machine Learning Research* 9 (2010), 297–304.

[13] Hanson, Edward R. 1971. Musicassette Interchangeability. the Facts Behind the Facts. *Journal of the audio engineering society* 19, 5 (1971), 417–425.

[14] Edgar Hassler, Jeffrey C. Carver, David Hale, and Ahmed Al-Zubidy. 2016. Identification of SLR Tool Needs - Results of a Community Workshop. *Information and Software Technology* 70 (2016), 122–129. https://doi.org/10.1016/j.infsof.2015.10.011

[15] Elis Hernandes, Augusto Zamboni, Sandra Fabbri, and André Di Thommazo. 2012. Using GQM and TAM to evaluate StArt – a tool that supports Systematic Review. *CLEI Electron. J.* 15, 1 (2012).

[16] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *ACL 2018* 1 (2018), 328–339.

[17] Barbara A. Kitchenham, David Budgen, and O. Pearl Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press. https://doi.org/10.1201/b19467

[18] Barbara A. Kitchenham and Stuart Charters. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE-2007-01. Keele University and University of Durham.

[19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR 2020*. 1–17.

[20] Yutong Li, Juanjuan Cai, and Jingling Wang. 2020. A Text Document Clustering Method Based on Weighted BERT Model. In *ITNEC 2020*. 1426–1430.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. In *ICLR 2020*.

[22] Christopher Marshall and O. Pearl Brereton. 2013. Tools to Support Systematic Literature Reviews in Software Engineering: A Mapping Study. In *International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 296–299. https://doi.org/10.1109/esem.2013.32

[23] Christopher Marshall and Pearl Brereton. 2013. Tools to support systematic literature reviews in software engineering: A mapping study. In *ESEM 2013*. IEEE, 296–299.

[24] Christopher Marshall, Pearl Brereton, and Barbara Kitchenham. 2014. Tools to support systematic reviews in software engineering: A feature analysis. In *EASE 2014*.

[25] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *NIPS 2017*, Vol. 2017-December. 6295–6306.

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR 2013*. 1–12.

[27] Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *AISTATS 2005*.

[28] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT 2018*. 1–10.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li Peter, and J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* (2019), 1–67.

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP 2019*.

[31] Yusra Shakeel, Jacob Krüger, Ivonne von Nostitz-Wallwitz, Gunter Saake, and Thomas Leich. 2020. Automated Selection and Quality Assessment of Primary Studies. *Journal of Data and Information Quality* 12, 1 (2020), 4:1–26. https://doi.org/10.1145/3356901

[32] He Zhang and Muhammad A. Babar. 2013. Systematic Reviews in Software Engineering: An Empirical Investigation. *Information and Software Technology* 55, 7 (2013), 1341–1354.