Hindawi
Scientific Programming
Volume 2021, Article ID 6641832, 15 pages
https://doi.org/10.1155/2021/6641832

Review Article

Bidirectional Language Modeling: A Systematic Literature Review

Muhammad Shah Jahan,[1] **Habib Ullah Khan**,[2] **Shahzad Akbar**,[3] **Muhammad Umar Farooq**,[1] **Sarah Gul**,[4] **and Anam Amjad**[1]

[1]*Department of Computer Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan*
[2]*Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar*
[3]*Riphah College of Computing, Riphah International University, Faisalabad Campus, Faisalabad 3800, Pakistan*
[4]*Department of Biological Sciences, FBAS, International Islamic University, Islamabad, Pakistan*

Correspondence should be addressed to Shahzad Akbar; shahzadakbarbzu@gmail.com

Received 27 December 2020; Revised 21 April 2021; Accepted 26 April 2021; Published 3 May 2021

Academic Editor: Fabrizio Riguzzi

In transfer learning, two major activities, i.e., pretraining and fine-tuning, are carried out to perform downstream tasks. The advent of transformer architecture and bidirectional language models, e.g., bidirectional encoder representation from transformer (BERT), enables the functionality of transfer learning. Besides, BERT bridges the limitations of unidirectional language models by removing the dependency on the recurrent neural network (RNN). BERT also supports the attention mechanism to read input from any side and understand sentence context better. It is analyzed that the performance of downstream tasks in transfer learning depends upon the various factors such as dataset size, step size, and the number of selected parameters. In state-of-the-art, various research studies produced efficient results by contributing to the pretraining phase. However, a comprehensive investigation and analysis of these research studies is not available yet. Therefore, in this article, a systematic literature review (SLR) is presented investigating thirty-one (31) influential research studies published during 2018–2020. Following contributions are made in this paper: (1) thirty-one (31) models inspired by BERT are extracted. (2) Every model in this paper is compared with RoBERTa (replicated BERT model) having large dataset and batch size but with a small step size. It is concluded that seven (7) out of thirty-one (31) models in this SLR outperforms RoBERTa in which three were trained on a larger dataset while the other four models are trained on a smaller dataset. Besides, among these seven models, six models shared both feedforward network (FFN) and attention across the layers. Rest of the twenty-four (24) models are also studied in this SLR with different parameter settings. Furthermore, it has been concluded that a pretrained model with a large dataset, hidden layers, attention heads, and small step size with parameter sharing produces better results. This SLR will help researchers to pick a suitable model based on their requirements.

1. Introduction

Transfer learning encompasses the model training on large text corpus and utilization of obtained knowledge to downstream tasks [1]. Before the emergence of transformer architecture for transfer learning, unidirectional language models were used extensively but these models faced many limitations such as reliance on unidirectional recurrent neural network (RNN) architecture and limited context vector size. To overcome these gaps, bidirectional language models such that bidirectional encoder representation from transformer (BERT) is introduced to improve the performance of downstream tasks. Bidirectional language models can be applied in a wide variety of tasks such as natural language inference (NLI) [2, 3], paraphrasing at sentence-level [4], Question Answering (QA) systems, and entity recognition at token level [5]. In the beginning, pretraining of bidirectional language models was done via supervised learning [6] but human-labeled datasets are limited. To resolve this issue, the use of a large corpus-based unsupervised learning increased.

Language models are one of the most crucial components of natural language processing (NLP). A language model provides context to distinguish between words and phrases that sound alike in English such as "recognize speech" and "wreck a nice beach" but indeed very different. The language model is a probability distribution over sequences of words and used in information retrieval. There are many types of language models including n-gram, exponential neural network (ENN), and bidirectional. These language models are the backbone of Google Assistant, Amazon's Alexa, and Apple's Siri to analyze the data for the prediction of words. BERT is first deep bidirectional language models based on transformer architecture which means it reads the input from both sides left-to-right and right-to-left while existing models were unidirectional and just read the input from one side. BERT outperforms all existing models.

A large amount of data [7] such as text corpus, domain-specific data (e.g., PubMed, PubMed Central (PMC) [8]), and scientific dataset [9] is available for unsupervised learning. Also, different sentence tokens, e.g., span [10], semantic [11–13], lexical [14], and syntactic [15], are used to pretrain the models. In general, large pretraining objective, unlabelled datasets [16, 17], benchmarks [18, 19], and fine-tuning methods [20, 21] are beneficial in unsupervised learning. Pretrained models developed using unsupervised learning have produced state-of-the-art results due to better use of parallel computing. The resultant models are not only applicable to computer domains but also used in other specific domains, e.g., [22] business [23], medical [24, 25], and science [26]. The performance of downstream tasks directly depends on pretraining of the models which subsequently considers many significant factors such as dataset size, batch size, step size, sequence size, parameters, layers, hidden layers, attention heads, and cross-layer sharing for practical implications. These factors are used in different research studies to get better results of pretrained models, but there is no study available to the best of our knowledge which provides comprehensive review to these research studies. This paper attempts to find answers to the following five research questions (RQs) as follows:

RQ1: what are the significant model types and techniques used for sentence embedding learning?

RQ2: what is the effect of dataset size with different batch, step, and sequence size on the performance of the pretrained model?

RQ3: what is the effect of parameters with different input layers, hidden layers, and attention heads on the performance of the pretrained model in downstream tasks?

RQ4: what are the effective techniques for cross-layer parameter sharing in the pretraining of models?

RQ5: what are the leading datasets used in the pretraining of models?

To find answers to these research questions, we performed an exhaustive systematic literature review (SLR) of thirty-one (31) research papers as presented in Table 1. The contributions of this paper are as follows:

(i) Firstly, this research study discovers all bidirectional language models built upon transformer or Transformer-XL architecture during 2018–2020.

(ii) Secondly, all the important settings of the pretrained model such as size of the dataset, batch size, step size, sequence size, parameters, layers, hidden layers, attention heads, and cross-layer sharing are recognized in this paper.

(iii) Every model is compared with RoBERT that is a replicated BERT model with a large dataset and batch size but with a small step size. The analysis of existing models with RoBERTa is also carried out in this SLR.

Rest of the paper is organized as follows: in Section 2, the research methodology is developed which consists of selection and rejection criteria, search process, quality assessment criteria, data extraction, and synthesis. Section 3 and Section 4 present the results and answers of the five developed questions, respectively. Section 5 discusses the analysis of the selected research studies. Section 6 provides recommendations to the existing research studies. Lastly, Section 7 concludes the whole research study and provides future directions.

2. Research Methodology

This research study is performed based on the guidelines of the systematic literature review standard. Following features that distinguish the systematic literature review from conventional literature review are as follows [53]:

(i) To begin, review protocol is developed based on the research questions.

(ii) Selection and rejection criteria are developed to assess each primary study.

(iii) Search strategy is defined to provide the addition of the most relevant literature in the SLR. It is documented to ensure the completeness of research study.

(iv) Information from each research study is evaluated using quality assessment criteria.

(v) To perform quantitative meta-analysis, review protocol turns out to be the prerequisite.

(vi) This review protocol establishes the basis of SLR due to which it becomes possible to identify the research gaps from the selected area so that new research activities can be positioned.

Two sections (Background and Research Questions) are already provided in the introduction section. Therefore, we

Table 1: Overall hyperparameters.

Paper	Batch size	Max sequence	Learning rate	Step size	Parameters	Layers	Hidden	Attention head
[17]	2K	512	1e − 6	125K	360	24	1024	16
[10]	256	128	1e − 4	2.4M	340	24	1024	16
[11]	32	128	2e − 5	1M	340	24	1024	16
[14]	512	256	5e − 5	1M	114	6	768	12
[15]	400K	256	5e − 5	4K	114	24	1024	16
[27]	256	128	1e − 4	1M	110	12	768	12
[28]	2048	512	1e − 5	500K	340	24	1024	16
[29]	330	512	3e − 5	777K	340	24	1024	16
[20]	32	512	1e − 4	1M	330	24	1024	16
[30]	32	512	1e − 4	1M	340	24	1024	16
[31]	256	128	1	1M	14.5	4	312	12
[32]	256	128	1e − 4	1M	340	24	1024	16
[33]	4096	512	0.00176	125K	233	12	4096	128
[34]	1024	128	1.0e − 4	1M	3.9	48	2560	40
[35]	4096	512	0.00176	125K	233	12	4096	64
[36]	2048	128	0.01	2.1M	11	12	768	12
[37]	2K	512	10^{-3}	125K	356	24	1024	16
[38]	8K	512	1e − 6	500K	360	24	1024	16
[39]	32	128	2e − 5	1M	340	12	768	12
[40]	2048	512	2e − 4	1.75M	335	24	1024	16
[41]	1024	512	5e − 4	400k	33	12	768	12
[42]	32	256	2^{-5} to10^{-5}	1M	66	6	768	12
[43]	256	128	1e − 4	1M	108	12	768	12
[44]	128	128	1e − 4	1M	340	24	1024	16
[45]	256	128	1e − 4	1M	110	12	768	12
[46]	256	128	1e − 4	1M	340	24	1024	16
[47]	256	128	5^{-5} to 10^{-5}	1M	110	12	768	12
[48]	128	512	3e − 4	50K	9.5	24	1024	16
[49]	6	512	1.5e − 5	1M	340	24	1024	16
[50]	8000	512	1e − 6	500K	400	12	1024	12
[51]	5120	128	1.8e − 4	25K	340	24	1024	16
[52]	7680	128	6e − 4	0.5M	110	12	768	12

are omitting both sections and will describe the other four elements in subsequent sections.

2.1. Selection and Rejection Criteria. We defined logical rules for the selection and rejection of the research papers to achieve the objectives of SLR. These rules are as follows:

(i) The selected research studies must target the bidirectional language modeling and the BERT model.

(ii) Selected research studies for this SLR must be published between 2018 and 2020.

(iii) All the research studies selected in this SLR must be from one of these four scientific repositories, i.e., arXiv, Elsevier, ACM, IEEE, and two conferences including NIPS (neural information processing systems) and MLR (machine learning research).

(iv) Duplicate research studies are not selected. Similar content shared by more than one research study is discarded.

2.2. Search Process. Four scientific repositories and two conferences mentioned in Section 2.1 initiated the search process. Five defined keywords, i.e., (1) bidirectional language modeling, (2) pretrained language modeling, (3) biLM, (4) BERT, and (5) transformer, are used to search the research studies as shown in Table 2. We only use AND operator while searching because without AND operator, some keywords produced irrelevant searches. We also used some advanced options provided by databases to refine the search result. For example, while searching for research studies on Science Direct with the keyword "transformer," we receive a lot of results because "transformer" belongs to other domains as well. To generate relevant results, an advanced option is used for publication titles such as "Science of Computer Programming."

We used open coding like process which involves three phases (Phase 1, 2, and 3) and three authors (A1, A2, and A3):

(i) Phase 1: A3 selects all papers which are from mentioned databases.

(ii) Phase 2: A2 checks all papers selected in Phase 1 and checks either these papers are published in between 2018 and 2020.

(iii) Phase 3: A1 selects all the papers provided at the end of Phase 2 which targeted the bidirectional language modeling or share its properties. Phases 1 and 2 are

Table 2: Number of results using keywords.

Sr. no.	Keywords	Operator	Scientific repositories					
		IEEE	ACM	arXiv	Elsevier	NIPS	MLR	
1	Bidirectional language modeling	AND	5	1	24	6	0	1
2	Pretrained language modeling	AND	2	2	18	2	0	0
3	biLM	N/A	1	3	1	1	0	0
4	BERT	N/A	16	6	24	1	1	1
5	Transformer	N/A	1	4	4	0	0	0

straight forward, but in Phase 3, if any of other two authors A1 or A2 had any disagreement, then a voting procedure was followed and majority wins.

In Figure 1, overview of the search process is illustrated. By using keywords shown in Table 2, we received total 94,954 results:

(i) We have rejected 92,876 papers based on the title of the research studies.

(ii) Among 92876, we rejected another 1589 papers by applying selection and rejection criteria on abstract.

(iii) Another 364 research studies are discarded by performing the general study of papers.

(iv) Lastly, after detailed study of 127 research studies, 96 research studies are eliminated and only thirty-one (31) relevant research studies are selected to perform SLR.

2.3. Quality Assessment. For the reliable outcome of the SLR, a quality assessment checklist (QA 1 to QA 5) is developed. Every paper included in this study must satisfy the assessment criteria to ensure high-quality of the selected research studies by answering a few questions in Table 3.

All selected research studies target bidirectional language models or BERT by either using or improving these models. The selected repository-based distribution of research studies is shown in Figure 2. All of the papers are from internationally recognized scientific repositories such as arXiv, IEEE, ACM, Elsevier, NIPS, and MLR. We included arXiv with other databases because most of the work on the bidirectional language model is published in arXiv. Almost all top LMs are developed by big technology organizations, and their work is published in arXiv. It can be analyzed from Figure 2 that arXiv is the most cited database. Also, it is ensured that the selected research study must answer at least one question. We have developed this checklist presented in Table 3 to ensure high-quality findings of our research studies.

2.4. Data Extraction and Synthesis. A template for data extraction and data synthesis is developed in Table 4 to answer the research questions. Data extraction is used to extract the specific and most related data based on selection and rejection criteria (Section 2.1). For data extraction and synthesis, we have extracted the bibliography of the paper and then core findings of the paper such as methodology, pretraining, fine-tuning, and the results are synthesized. We have performed the data synthesis to answer our developed research questions for this SLR.

3. Results

After applying the review protocol (see Section 2), thirty-one (31) research studies published during 2018–2020 are selected to conduct this SLR. We compared all models with RoBERTa [17] which is the replication of BERT [27] with a large dataset, batch size, sequence size, parameter, layers, hidden layers, attention head but with small step size, no parameter sharing, and no sentence representation learning. The main advantage of comparison with RoBERTa is that it is a model built on BERT with slightly changed parameters and can generate fair comparison for all other models used in this research. In this section, the dataset with other parameters effecting the pretraining of models, results based on model structure, pretraining objectives, sharing parameters in pretraining, and model selection for testing are discussed in detail.

3.1. Model Type and Technique Used for Sentence Embedding Learning. We identified four (4) types of pretraining objectives for language representation and three (3) types of sentence representation learning presented in Table 5. Pretraining objectives are as follows: (1) autoencoding is a model type in which the model reconstructs the original data from corrupted inputs. (2) Autoregressive uses the probability distribution that remembers previous states while partially autoregressive uses only one previous state. (3) Autoencoding and autoregressive present objective in which corrupting and knowledge of previous values preserve. (4) Autoencoding and partially autoregressive present objective in which corrupting and partially knowledge of previous values preserve.

Three (3) sentence representation learning tasks are discussed in terms of pretraining objectives in Table 5: Next Sentence Prediction (NSP), Sentence Order Prediction (SOP), and None. (1) NSP is a binary classification that predicts whether two segments that appear consecutively are from the same document. (2) SOP focuses on intersentence coherence with positive examples the same as NSP but negative examples are different and achieved by swapping the documents. (3) If a model used None it means neither NSP nor SOP is used.

3.2. Pretraining Setup. We have divided the pretraining dataset into four categories presented in Table 6 with respect

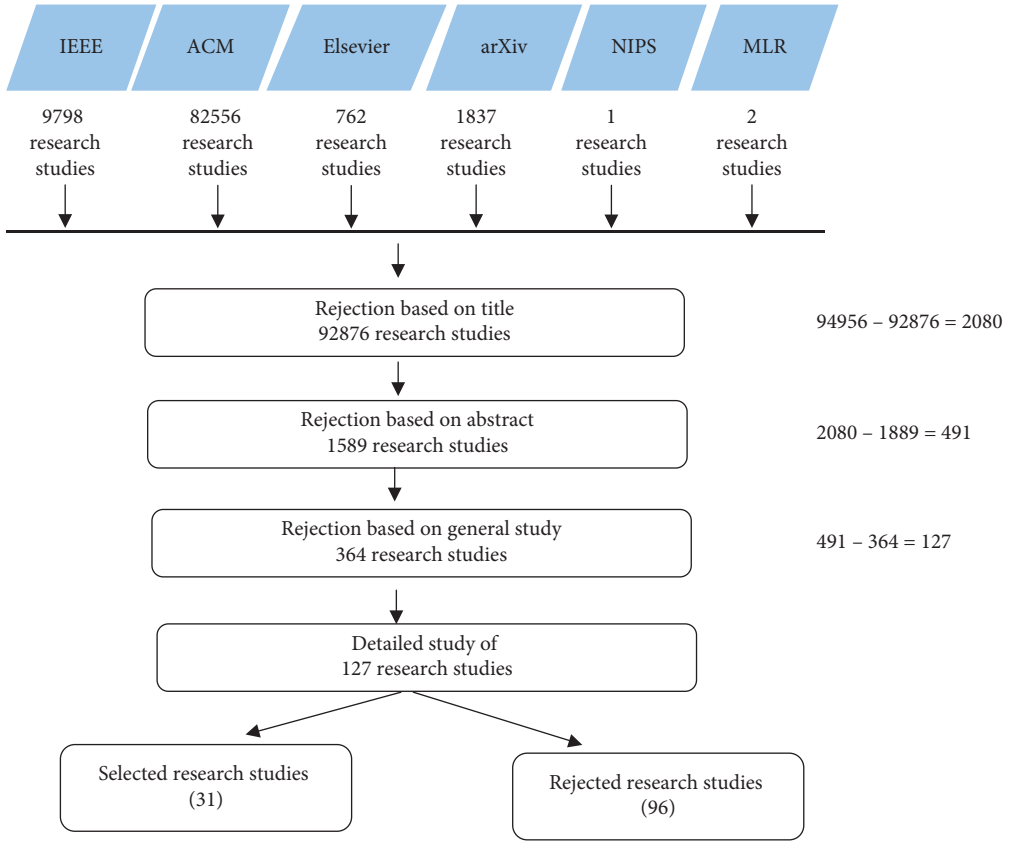

Figure 1: Overview of the search process.

Table 3: Quality assessment checklist.

Sr. no.	Quality assessment checklist
QA 1	Are models using bidirectional language modeling in selected research studies?
QA 2	Do all the papers use either BERT or improve it?
QA 3	Are selected research studies published from 2018 to 2020?
QA 4	Do the selected research papers are from the scientific repositories, NIPS and MLR?
QA 5	Do the selected research papers provide the required answers for developed research questions?

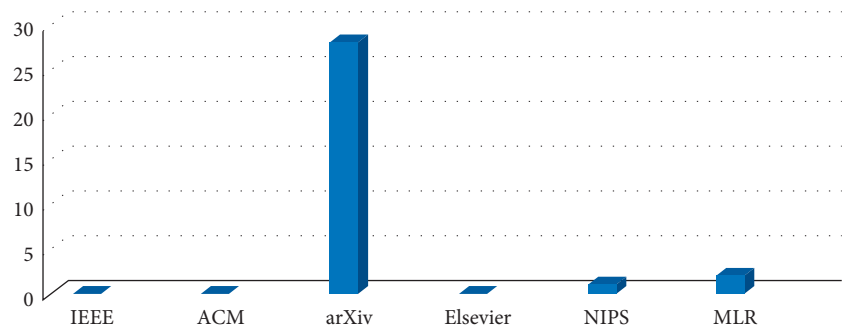

Figure 2: Summary of selected papers.

Table 4: Data extraction and synthesis.

Sr. no.	Description	Details
1	Bibliographic information	Authors, title, research type, publication year, etc.
2	Methodology	The main structure of our study is to extract the methodology of the paper.
3	Pretraining	Pretraining structure of each study is thoroughly analyzed.
4	Fine-tuning	Fine-tuning structure of each study is thoroughly analyzed.
5	Dataset	Datasets used in the selected research studies are identified.

Table 5: Model type and sentence representation learning.

Pretraining objectives	Sentence representation learning		
	NSP	SOP	None
Autoencoding (AE)	[14, 27, 29–32]	[33–35]	[10, 11, 15, 20, 36–49]
Autoregressive (AR)	—	—	[50]
Autoencoding and autoregressive	[51]	—	[28]
Autoencoding and partially autoregressive (PAR)	—	—	[52]

to size such as (1) 1 GB to 99 GB, (2) 100 GB to 149 GB, (3) 150 GB to 199GB, and (4) 200 GB to onwards. The main advantage of this categorization helps in visualization of dataset size used by different language models and the relation of dataset size with step size, batch size, and sequence size. Step size indicates how many steps a program will run and takes data points with respect to time. Batch size is the number of examples that can be utilized in one iteration. Sequence size defines the maximum size of an input. If one increases the sequence size, then it required a lot of computational power and resources to pretrain an LM. We divide the batch and step size into three categories (small, big, and same) while sequence size has two categories (small and same). We make two categories of sequence size because no model has a bigger sequence size than 512. Sequence size of 512 is used by RoBERTa [17] which has 160 GB of training dataset size with 2 K (small) of batch size and 125 K (medium) step size. The comparison of performance of different models is done using GLUE leaderboard. GLUE consists of ten (10) diversified tasks, but we use results of eight (8) of these tasks and leave WNLI and AX due to completely different behavior of these two tasks. Subsequently, SQuAD and then RACE are also used for comparison.

3.3. Effect of Parameters with Different Layers, Hidden Layers, and Attention Heads. As shown in Table 7, we have divided the parameter size of models into seven categories such as (1) 1 M to 99 M, (2) 100 M to 199 M, (3) 200 M to 199 M, (4) 301 M to 349 M, (5) 350 M to 399 M, (6) 400 M to 500 M, and (7) 501M to onwards. Every category contains the parameters used by models in pretraining. We have three categories for parameter size: (1) layers, (2) hidden layers, and (3) attention head, and every category has three subcategories: (1) less, (2) more, and (3) same. All results are compared against RoBERTa [17] which has 360 M parameters, twenty-four (24) layers, 1024 hidden layers, and sixteen (16) attention heads.

3.4. Cross-Layer Parameter Sharing. Cross-layer parameter sharing is a process in which models share some parameters during pretraining with purpose of gaining knowledge. We divide the cross-layer parameter sharing into four different categories as shown in Table 8. (1) In all-shared means both feedforward network (FFN) and attention are shared across layers, (2) in shared-attention, only attention is being shared, (3) in shared-FFN, only FFN is being shared across layers, and (4) not-shared shares nothing during pretraining. For the selected parameters, results are shown in Table 8.

3.5. Dataset Used. In this section, the datasets used are mentioned so that new models could be tested using these datasets. Four datasets are identified: (1) General Language Understanding Evaluation (GLUE): it consists of ten diversified tasks. Some tasks are single-sentence classification and some are sentence-pair classification. GLUE provides split training and testing data to test the performance of pretrained models. It allows us to submit our submissions on the GLUE leaderboard and compare our evaluation results on private held-out test data; (2) Bilingual Evaluation Understudy (BLEU) is used to evaluate the quality of machine translation text from one language to another; (3) the Stanford Question Answering Dataset (SQuAD) has two datasets with SQuAD v1.1 and SQuAD v2.0 with one has answerable questions and the other has unanswerable. SQuADv1.1 consists of 100 K questions while SQuADv2.0 consists of 150 K questions; (4) RACE is a comprehension dataset consists of 28 K passages and 100 K questions.

4. Answers to Research Questions

RQ1: what are the significant model types and techniques used for sentence embedding learning?

Answer: as shown in Table 5, in thirty-one (31) identified models, only six models, ERNIE [14], BERT [27], UniLM [29], StructBERT [30], TinyBERT [31], and MT-DNN [32], use NSP as sentence learning technique while three models, ALBERT [33], Megatron-LM [34], and AlBERT (xxlarge-ensemble) [35], use SOP. It could be seen that twenty-one (21) out of thirty-one (31) research studies do not use any sentence embedding learning which means both NSP and SOP decrease the performance of these models.

Different models have different pretraining objectives such as BERT [50] is an autoregressive model without using NSP. Nezha [51] uses autoencoding and autoregressive with NSP while XLNet [28] uses autoencoding and autoregressive without NSP. Another model, UniLMv2 [52], is an autoencoding and partially autoregressive model. Rest of the models, i.e., twenty-seven (27) out of thirty-one (31) models, use autoencoding that is the most used pretraining objective with none sentence learning technique.

RQ2: what is the effect of dataset size with different batch, step, and sequence size on the performance of the pretrained model?

Answer: as shown in Table 6, seven (7) out of thirty-one (31) models, i.e., [15, 33–36, 38, 40], have outperformed RoBERTa. Among these seven models, three models [34, 36, 38] were trained on a larger dataset than RoBERTa while other three models [15, 33, 35] are trained on a smaller dataset whereas [40] trained on 126 GB dataset size which is

Table 6: Effect of training data size with different batch, sequence, and step size.

Training data size	Batch size			Sequence size		Step size			Performance	
	Small	Big	Same	Small	Same	Small	Big	Same	Slow	Better
200 GB-onwards	—	—	[36]	[36]	—	—	[36]	—	—	[36]
150–199 GB	[34]	[38, 50, 52]	[37]	[34, 52]	[37, 38, 50]	—	[34, 38, 50, 52]	[37]	[37, 50, 52]	[34, 38]
100–149 GB	—		[28, 40]	—	[28, 40]	—	[28, 40]	—	[28]	[40]
1–99 GB	[10, 11, 14, 20, 27, 29–32, 39, 41–49]	[15, 33, 35, 51]	—	[10, 11, 14, 15, 27, 32, 39, 42–47, 51]	[20, 29, 30, 33, 35, 41, 48, 49]	[15, 39, 48, 51]	[10, 11, 14, 20, 27, 29, 31, 32, 41–47, 49]	[30, 33, 35]	[10, 11, 14, 20, 27, 29–32, 39, 41–49, 51]	[15, 33, 35]

Table 7: Effect of parameters with different layers, hidden layers, and attention heads.

Parameters	Layers			Hidden			Attention head		
	Less	More	Same	Less	More	Same	Less	More	Same
1–99 M	[31, 41, 42]	—	[48]	[31, 41, 42]	—	[48]	[31, 41, 42]	—	[48]
100–199 M	[14, 27, 43, 45, 47, 52]	—	[15]	[14, 27, 43, 45, 47, 52]	—	[15]	[14, 27, 43, 45, 47, 52]	—	[15]
<200–300>M	[33, 35]	—	—	—	[33, 35]	—	—	[33, 35]	—
301–349M	[20, 39]	—	[10,28–30, 32, 40, 44, 46, 49, 51]	[39]	—	[10, 20, 28–30, 32, 40, 44, 46, 49, 51]	[39]	—	[10, 20, 28–30, 32, 40, 44, 46, 49, 51]
350–399	—	—	[11, 37, 38]	—	—	[11, 37, 38]	—	—	[11, 37, 38]
400–500	[50]	—	—	—	—	[50]	[50]	—	—
501-Ow	[36]	[34]	—	[36]	[34]		[36]	[34]	—

Table 8: Cross-layer parameter sharing.

Cross-layer parameter sharing	Paper	Performance	
		Decrease	Increase
All-shared	[15, 20, 29, 32–36, 40, 44, 46, 47, 51]	[20, 29, 32, 44, 46, 47, 51]	[15, 33–36, 40]
Shared-attention	[28]	[28]	—
Shared-FFN	[48]	[48]	—
Not-shared	[10, 11, 14, 27, 30, 31, 37–39, 41–43, 45, 49, 50, 52]	[10, 11, 14, 27, 30, 31, 37, 41–43, 45, 49, 50, 52]	[38]

close to RoBERTa of 160 GB. However, other three out of thirty-one (31) models [37, 50, 52] trained on the same size of the dataset but perform lesser than RoBERTa due to the large or same step size. Rest of the twenty-four (24) models trained on smaller datasets and among these 23 models, three models outperform the RoBERTa and all of these models use bigger batch size and same or small step size. A model trained on a larger dataset with larger batch size and smaller step size will generate better outcomes and saves pretraining time.

RQ3: what is the effect of parameters with different input layers, hidden layers, and attention heads on the performance of the pretrained model on downstream tasks?

Answer: performance of downstream tasks depends on the number of parameters used in training along with the layers, hidden layers, and attention heads. As shown in Table 7, seven models in [15, 33–36, 38, 40] outperform RoBERTa. Among these seven models, four models [15, 33, 35, 40] have less parameters than RoBERTa such as a model in [15] utilized a very low number of parameters, i.e., 114M parameter w.r.t 360M of RoBERTa. With less parameters, different combinations are also analyzed such as two research studies [33, 35] have less parameters but pretrained with deep hidden layers. On the other side, two out of seven models [34, 36] have very large parameters than RoBERTa. Lastly, one model [38] used the same number of parameters as RoBERTa. In the light of above parameters, it is analyzed that if a model has very large parameters or has large hidden layers and attention heads, it will produce better results.

RQ4: what are the effective techniques for cross-layer parameter sharing in pretraining of models?

Answer: cross-layer sharing helps the models to produce better results. As shown in Table 8, seven models [15, 33–36, 38, 40] produce a better output than RoBERTa and all of these models except for [38] shared both FFN and attention across layers during pretraining. On the other side, almost all the models except for [38] which are not shared during pretraining produce less efficient results. Among these seven models, different combinations with cross-layer parameters are used such as [33, 35] represents deeper model (i.e., including large hidden layers) while [34, 36, 38] are humongous models which means large input layer, hidden layer, and big attention head are involved in these models, whereas [40] is also close to RoBERTa in size, parameters, and other setting, but it shares parameters in pretraining and produces better results. It could be seen that all-shared parameters have a positive effect on the performance of models.

RQ5: what are the leading datasets used in pretraining of models during 2018–2020?

Answer: twenty-eight (28) out of thirty-one (31) models used GLUE for downstream tasks. GLUE consists of ten diversified tasks. These tasks could be seen on https://GLUEbenchmark.com/leadaerboard. As shown in Table 9, the higher number of models, i.e., seventeen (17), uses SQuAD for the downstream task. The SQuAD has two subtasks SQuAD v1.1 and SQuAD v2.0. Six models, [29, 36, 41, 50, 52, 54] research studies, used the BLEU dataset while [27, 28, 33, 34, 38, 43] research studies use the RACE dataset. Only [27] used all of these datasets while [28, 33] used GLUE, SQuAD, and RACE datasets altogether. GLUE and SQuAD are significant datasets for testing new models, and that is the reason that these datasets become the benchmark for future models.

5. Analysis

In this section, we discuss the analysis of the research studies based on the pretraining dataset and different settings such as data size, batch size, and step size.

5.1. Different Ways of Pretraining of BERT Model. Delvin et al. [27] introduced the first deeply bidirectionally pretrained model to train the unlabelled text. BERT also introduced the NSP which is used to predict the next sentence for the Question Answering system. Besides, BERT requires an additional layer to perform any type of downstream task. Wang et al. [30] incorporate the language structures (word and sentence) during pretraining which enable the model to reconstruct the right order of words and sentences. This model extends the NSP by predicting previous values. Joshi et al. [10] used Masked Language Modeling (MLM) at span level. It uses a novel span boundary objective which summarizes as required span as possible and uses a single contiguous segment instead of two segments in pretraining. Zhang et al. [11] introduced the model consisting of out-of-the-shelf labeler, a sentence encoder where semantic labels are mapped into embedding in parallel and semantic integration components to obtain joint representation for fine-tuning. Su et al. [39] presented squeeze and excitation to extract global information between layers and Gaussian blurring to capture the neighbor context in the downstream task. It also uses the Heuristic Analysis for NLI Systems (HANS) dataset which shows SesameBERT adopted shallow heuristic instead of a generalization.

Josefowicz et al. [55] fine-tuned on extreme multilabel text classification. This classification used semantic label

Table 9: Datasets used in selected research studies.

Dataset name	Research studies
GLUE	[10, 11, 14, 15, 20, 27, 29–33, 36–48, 50–52]
BLEU	[28, 29, 36, 41, 50, 52, 54]
SQuAD	[10, 11, 20, 27–31, 33–36, 40, 41, 49, 50, 52]
RACE	[27, 28, 33, 34, 38, 43]

clusters for better model dependencies and both label and input text to build label representation. It consists of semantic label indexing and ensemble ranking component. Jiao et al. [31] propose a transformer distillation method which transfers the linguistic knowledge from teacher BERT to TinyBERT. In distillation methods, we have two methods: first one consists of the big model called the teacher method, and other consists of the small model called the student model. When the teacher model is pretrained, it gains knowledge and transfers its knowledge to the student model. A two-stage learning framework uses transformer distillation on pretraining and fine-tuning and lets the TinyBERT capture general and specific knowledge of teacher BERT with 28% fewer parameters. Xu et al. [42] used progressive model replacing to compress the parameters; it first divides the BERT model and then builds their compact substitute. The probability of replacing was increased through training. Wang et al. [30] trained by the BERT model using stacking algorithm that observes the self-attention at different layers and positions for transferring the knowledge from shallow to deep model. It also finds local attention distribution and start-of sentence distribution. Goyal et al. [43] improved inference time with very little performance loss. PowerBERT removes the word-vector from the encoder pipeline which reduces the computation and directly improves inference time.

Furthermore, Chen et al. [48] task-oriented the compressed BERT model, called AdaBERT which uses differentiable neural architecture searches to automatically compress the BERT model into task-specific small models. AdaBERT incorporates task-oriented knowledge distillation (KD) loss for search hint and efficiency loss as search constraints. Beltagy et al. [26] built a new scientific vocabulary. The trained BERT model on large and in-domain-scientific data shows that the in-domain pretrained model performs better on downstream tasks due to in-domain vocabulary. Lee et al. [8] proposed a first domain-specific pretrained model that trained BERT on a medical dataset. The medical dataset includes PubMed abstracts and PMC full text instead of general datasets such as Wikipedia. Results show that the domain-specific pretrain model outperforms the BERT on Question Answering (QA) (12.24), Relation Extraction (RE) (2.8), and Named Entity Recognition (NER) (0.82). Chadha et al. [49] used set of modified transformer encoder units to add more focused query-to-context (Q2C) and context-to-query (C2Q) attention to BERT architecture. It also adds localized information to self-attention and skips connections in BERT. Kao et al. [35] boosted the BERT by duplicating some layers which makes BERT deeper without extratraining to increase the performance of the BERT model in downstream tasks.

5.2. Different Settings for Models. Liu et al. [17]pre-trained the BERT model on a 12 times bigger and diverse dataset with two changes in hyper-parameters during pre-training. First is a bigger batch size with small step size and second is dropping the NSP.. Lan et al. [33] reduced the size of the parameters of the model during training. Also, it uses two-parameter reduction techniques. The first one is factorized embedding parameterization to separate the size of hidden layers from the size of vocabulary embedding. The second one is cross-layer parameter sharing. It also replaces NSP with SOP. Yang et al. [28] used an autoregressive and autoencoding pretrained model that uses all possible permutations of factorization order. It uses relative positioning and segment recurrence mechanism borrowed from Transformer-XL (extension of transformer architecture with positional encoding). XLNet does not use MLM to remove pretraining and fine-tuning discrepancy and also leave the NSP which decreases the XLNet performance. Zhu et al. [38] presents the adversarial training algorithm which makes the transformer-based models better by adding adversarial perturbation to word embedding and minimize the maximum risk. Bao et al. [52] use Pseudo-Masked Language (PMLN) training procedure combining autoencoding (AE) and partially autoregressive (PAR).it follows BERT for encoding modeling. AE provides global PAR to learn interrelation between masked span. PMLN learns long-distance context better than the BERT.

Moreover, Lewis et al. [50] proposed denoising autoencoder to pretrain sequence-to-sequence models by corrupting the text with arbitrary noisy function and then reconstruct the original text. It proposes a novel in-filling scheme and is best to perform for generalization. It differs from BERT as additional cross-attention by decoder layers perform on the last hidden layer of the encoder. Chen et al. [56] proposed a unified framework that converts language problems into a text-to-text problem for training on a new dataset C4 with 11B parameters. Houlsby et al. [20] presented a novel adaptor tuning that uses only 3.6% of task-specific parameters of BERT instead of 100% use in fine-tuning. It provides a compact and extensible model adding only a small number of additional parameters per task because it remembers the previous values. Chang et al. [54] proposes a novel task conditional masked language to fine-tuned BERT on the text-generation dataset. It improves text generation by providing word probability distribution for every token in the sentence. Xu et al. [57] improved the BERT by using self-ensemble and self-distillation in fine-tuning without using external data. The self-ensemble model is an intermediate model at a different time which has average parameters of base-models. The distillation loss is used as regularization which improves the performance. Jiang et al. [37] overcome the limited downstream resources which make the model overfit, and it forgets the knowledge of the pretraining model. Smoothness-inducing regularization and Bregman proximal point optimization were applied on fine-tuning of models in which SMARTRoBERTa produces the SOTA results for many tasks. Zhang et al. [14] pretrained models on knowledge graphs and

large-scale textual. This model uses lexical, syntactic, and knowledge information with MLM and NSP loss.

Furthermore, Sun et al. [15] presented a pretraining framework that builds the task and then incrementally learn multitask learning. It extracts the lexical, syntactic, and semantic information as named entity, closeness relation from things corpus. Wei et al. [51] presented a new pretrained model trained on large Chinese corpus with functional relative positional encoding whole word masking strategy, LAMB optimizer mixed-precision training, and length of the training sequence. In Clark et al.'s study [40], the pretrained representation masked the input with plausible alternative sampled from a small genitor network. This model predicts whether each of the tokens in corrupted input was replaced by a generator sample or not. The computational speed of Electra was four times faster than RoBERTa and XLNet. Wang et al. [41] presented a compressed and small pretrained language model. This model contains two models: student and teacher. The student model is trained by deeply mimicking the self-attention model in the larger model (the teacher model). It performs distillation of the self-attention model from the last layer of the large model.

Shoeybi et al. [34] used billions of parameters by using efficient intralayer model parallelism attention in the placement of layer normalization in the BERT style model which increases the performance of model. Liu et al. [32] introduced a model that learned from multiple Natural Language Understanding (NLU) tasks. It uses cross-layer sharing and general representation which helps it to adapt to new tasks and domains. Clark et al. [44] introduced a model that uses knowledge distillation in which single-task trains the multitask. It proposes teacher annealing which takes the distillation to supervised learning which helps the multitask model to learn and surpass its teacher model. Dong et al. [29] trained the model on unidirectional, multidirectional, and sequence-to-sequence tasks. It fine-tuned for language understanding and generation tasks. It uses specific self-attention masks and a shared transformer network. Liu et al. [46] presented learning text representation across NLU tasks. An ensemble of teacher models is trained, and the student model is trained on the teacher model via learning distill knowledge.

Table 10 presents the overall data of all the models included in this study, training dataset, dataset size, tokens, model type, sentence learning, and cross-layer parameter sharing of every model. Table 1 shows the parameters and model setting, and Table 11 shows the result of every model. Every model in this study not just pretrained with different hyperparameters, different learning techniques, or sharing techniques. These models also pretrained differently, for example, some use MLM at the token level, some use at span level, some models use annealing, distillation method, and duplication of hidden layers, and others separate the hidden layers from model size. Effect of different ways of pretraining is minimum against the effect of parameters, for example, very few models perform better then RoBERTa. These better models solely depend on techniques which show the effect of hyperparameters, learning, and sharing on the performance of language models.

6. Discussion

In the above section, we have provided the answers to the question in Section 4. There are a few recommendations to existing/new models as follows:

(i) Small + FFN: small models (small input layers, hidden layers, and attention heads) with fewer parameters but with the sharing of both FFN and attention during pretraining improve the performance of the language model.

(ii) Deeper models: small models with very deep hidden layers and bigger attention heads using all-shared cross-layer sharing produce the best result among language models.

(iii) Bigger models: bigger models produce better results except when they are trained with fewer hidden layers and fewer attention heads. To increase hidden layers, we need to use fewer input layers to computationally compatible.

(iv) Dynamic masking: the use of dynamic masking allows changing the masking with every epoch to overcome the limitation of static masking. Static masking only masks the tokens with the same sequence affecting the performance of the model. If dynamic masking is used, then with every epoch, a new token will be masked.

(v) Larger batch: pretraining of the language model with larger batch size learns faster and improves results. It also saves us from large step size. If one increases the size of batch size, the step size decreases.

(vi) Sentence Order Prediction (SOP): use of SOP instead of NSP on other models such as XLNet and RoBERTa is beneficial. The reason is SOP can cover NSP tasks, but NSP cannot cover SOP task which means SOP has higher accuracy.

(vii) Domain-specific dataset: training on domain-specific datasets, such as medical and scientific, produces better results as the model will learn more about the specific domain better than the general domain.

(viii) Adversarial training: use of adversarial training on smaller models with cross-layer sharing is highly recommended. When it is applied on the fine-tuning step, it limits the maximum risks and could be applied to any model built upon the transformer architecture.

(ix) MLM: models can be pretrained with different MLM strategies on the span, lexical, syntactic, semantic, and knowledge information for pretraining the models.

(x) Distillation: the use of distillation methods having a student model and teacher model is highly recommended. The student models learn from the teacher model which saves it to pretrain on the

Table 10: Overall data.

Paper	Name	Training data size	Tokens	Training dataset name	Model type	Sentence learning	Cross-layer parameter sharing
[17]	RoBERTa (large)	160 GB	−2.2T	BooksCorpus + Wikipedia + CC-News + OpenWebText + Stories	Autoencoding	None	False
[10]	SpanBERT	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	False
[11]	SemBERT	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	False
[14]	ERNIE,	9 GB+	4.5B + 140M	English Wikipedia + Wikidata	Autoencoding	NSP	False
[15]	ERNIE2.0	13 GB+	8B	Encyclopedia + BooksCorpus + Dialog + Discourse Relation Data	Autoencoding	None	True
[27]	BERT (base)	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	NSP	False
[28]	XLNet	126 GB	32.89B	BooksCorpus + Wikipedia + Giga5+ ClueWeb 2012B + Common Crawl	Autoencoding + autoregressive	None	True
[29]	UniLM	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	NSP	True
[20]	—	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	True
[30]	StructBERT	16 GB	2.5B+	English Wikipedia + BooksCorpus	Autoencoding	NSP	False
[31]	TinyBERT	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	NSP	False
[32]	MT-DNN	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	NSP	True
[33]	AlBERT (xxlarge)	16 GB	—	BooksCorpus + Wikipedia	Autoencoding	Sop	True
[34]	Megatron-LM	174 GB	—	Wikipedia + CC-Stories + Real News + OpenWebText	Autoencoding	SOP	True
[35]	AlBERT (xxlarge-ensemble)	16 GB	—	BooksCorpus + Wikipedia	Autoencoding	None	True
[36]	T5	29 TB	—	Colossal Clean Crawled Corpus	Autoencoding	None	True
[37]	SMARTRoBERTa	160 GB	—	BooksCorpus + Wikipedia + CC-News + OpenWebText + Stories	Autoencoding	None	False
[38]	FreeLBRoBERTa	160 GB	−2.2T	BooksCorpus + Wikipedia + CC-News + OpenWebText + Stories	Autoencoding	None	False
[39]	SesameBERT	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	False
[40]	Electra1.75M	126 GB	33B	BooksCorpus + Wikipedia + Giga5+ ClueWeb 2012B + Common Crawl	Autoencoding	None	True
[41]	MiniLMa	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	False
[42]	SBERT-WK	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	False
[43]	PowerBERT	11 GB	3.4B	BooksCorpus + Wikipedia	Autoencoding	None	False
[44]	Bam	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	True
[45]	StackBERT	11 GB	3.4B	BooksCorpus + Wikipedia	Autoencoding	None	False
[46]	MT-DNNKD	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	True
[47]	HUBERT	16 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	True
[48]	AdaBERT	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	True
[49]	BERTQA	13 GB	3.8B	BooksCorpus + Wikipedia	Autoencoding	None	False
[50]	BART (large)	160 GB	2.2T	160 GB + Wikipedia	Autoregressive	None	False
[51]	Nezha	—	10.5B	Chinese Wikipedia + Baidu Baike + Chinese News	Autoencoding + autoregressive	NSP	True
[52]	UniLMv2	160 GB	—	BooksCorpus + Wikipedia + CC-News + OpenWebText + Stories	Autoencoding and partially autoregressive	None	False

Table 11: Results.

Paper	Batch size	Max sequence	Learning rate	Step size	Parameters (M)	Layers	Hidden	Attention head
[17]	2K	512	1e − 6	125K	360	24	1024	16
[10]	256	128	1e − 4	2.4M	340	24	1024	16
[11]	32	128	2e − 5	1M	340	24	1024	16
[14]	512	256	5e − 5	1M	114	6	768	12
[15]	400K	256	5e − 5	4K	114	24	1024	16
[27]	256	128	1e − 4	1M	110	12	768	12
[28]	2048	512	1e − 5	500K	340	24	1024	16
[29]	330	512	3e − 5	777K	340	24	1024	16
[20]	32	512	1e − 4	1M	330	24	1024	16
[30]	32	512	1e − 4	1M	340	24	1024	16
[31]	256	128	1	1M	14.5	4	312	12
[32]	256	128	1e − 4	1M	340	24	1024	16
[33]	4096	512	0.00176	125K	233	12	4096	128
[34]	1024	128	1.0e − 4	1M	3.9	48	2560	40
[35]	4096	512	0.00176	125K	233	12	4096	64
[36]	2048	128	0.01	2.1M	11	12	768	12
[37]	2K	512	10^{-3}	125K	356	24	1024	16
[38]	8K	512	1e − 6	500K	360	24	1024	16
[39]	32	128	2e− 5	1M	340	12	768	12
[40]	2048	512	2e − 4	1.75M	335	24	1024	16
[41]	1024	512	5e − 4	400k	33	12	768	12
[42]	32	256	2^{-5} to10^{-5}	1M	66	6	768	12
[43]	256	128	1e − 4	1M	108	12	768	12
[44]	128	128	1e − 4	1M	340	24	1024	16
[45]	256	128	1e − 4	1M	110	12	768	12
[46]	256	128	1e − 4	1M	340	24	1024	16
[47]	256	128	5^{-5} to 10^{-5}	1M	110	12	768	12
[48]	128	512	3e − 4	50K	9.5	24	1024	16
[49]	6	512	1.5e − 5	1M	340	24	1024	16
[50]	8000	512	1e − 6	500K	400	12	1024	12
[51]	5120	128	1.8e − 4	25K	340	24	1024	16
[52]	7680	128	6e − 4	0.5M	110	12	768	12

whole dataset. The student model just needs to learn from the teacher model.

(xi) Duplicating layers: A method of duplicating layers to make models deeper could save a lot of computational power. Duplicating layer is a method in which we keep smaller size of layers, and during the execution of pretraining, we duplicate the layers which directly make the model deep.

It is very hard to say which setting could be used to improve the performance of models due to trade-offs as larger models will use more resources while smaller models will cover fewer data. It is also recommended to use combinations such as bigger batch size with smaller step size, use of fewer input layers with largely hidden layers, and big attention heads. Subsequently, training of the hybrid combination on the domain-specific dataset and use of MLM on the span and lexical level is suggested. By doing so, performance of the bidirectional language models can be improved.

7. Conclusion and Future Work

This paper presents the SLR on a comprehensive study of thirty-one (31) pretrained language models to find the answers to five developed research questions. All models used in this paper are inspired by BERT and have a transformer or Transformer-XL architecture. The significant findings of this SLR are presented in Tables 1, 10, and 11. Table 10 presents the overall data used in these models, Table1shows the hyperparameter setting of these models, and Table 11 highlights the results of these models. These research papers show the effect of sentence embedding learning, size of the dataset, step, batch, parameters, layers, attention heads, and the effect of cross-layer sharing and also provide the most used benchmarks for future models. To conclude, whole focus of our study is about the pretraining of language models covering fine-tuning settings and the downstream task. The tables are created in two ways so that we could depict more accurate data by providing authentic information.

There are different ways to pretrain a model such that MLM on tokens or spans, etc. Besides, many models are pretrained on domain-specific datasets (e.g., business, medical, and physics) to improve the performance of models, but still the impact of these models is minimum when compared by dataset size, objectives, representation, sharing, and parameters. Therefore, it is important to consider these factors for language models before implementation because these factors can affect the performance

of any model. In this study, we only consider the models which were built on the transformer or Transformer-XL architecture and inspired by BERT, but in future, we intend to include models built on other architectures such as RNN.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] T. Mikolov, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 2, pp. 3111–3119, 2013.

[2] S. R. Bowman, "A large annotated corpus for learning natural language inference," 2015, https://arxiv.org/abs/1508.05326.

[3] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," 2017, https://arxiv.org/abs/1704.05426.

[4] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, Jeju Island, South Korea, October 2005.

[5] P. Rajpurkar, "Squad: 100,000+ questions for machine comprehension of text," 2016, https://arxiv.org/abs/1606.05250.

[6] D. Mahajan, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[7] A. Radford, "Improving language understanding by generative pre-training," 2018.

[8] J. Lee, W Yoon, S Kim et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics (Oxford, England)*, vol. 36, no. 4, pp. 1234–1240, 2020.

[9] I. Beltagy, A. Cohan, and K. Lo, "Scibert: pretrained contextualized embeddings for scientific text," 2019, https://arxiv.org/abs/1903.10676.

[10] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[11] Z. Zhang, "Semantics-aware BERT for language understanding," 2019, https://arxiv.org/abs/1909.02209.

[12] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with BERT," *IEEE Access*, vol. 7, pp. 154290–154299, 2019.

[13] X. Yin, Y. Huang, B. Zhou, A. Li, L. Lan, and Y. Jia, "Deep entity linking via eliminating semantic ambiguity with BERT," *IEEE Access*, vol. 7, pp. 169434–169445, 2019.

[14] Z. Zhang, "ERNIE: enhanced language representation with informative entities," 2019, https://arxiv.org/abs/1905.07129.

[15] Y. Sun, "Ernie 2.0: a continual pre-training framework for language understanding," 2019, https://arxiv.org/abs/1907.12412.

[16] R. Zellers, "Defending against neural fake news," *Advances in Neural Information Processing Systems*, vol. 3, pp. 6000–6010, 2019.

[17] Y. Liu, "Roberta: a robustly optimized bert pretraining approach," 2019, https://arxiv.org/abs/1907.11692.

[18] A. Wang, "Superglue: a stickier benchmark for general-purpose language understanding systems," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1–9, 2019.

[19] A. Conneau and D. Kiela, "Senteval: an evaluation toolkit for universal sentence representations," 2018, https://arxiv.org/abs/1803.05449.

[20] N. Houlsby, "Parameter-efficient transfer learning for NLP," 2019, https://arxiv.org/abs/1902.00751.

[21] M. Peters, S. Ruder, and N. A. Smith, "To tune or not to tune? adapting pretrained representations to diverse tasks," 2019, https://arxiv.org/abs/1903.05987.

[22] Y. Iwasaki, "Japanese abstractive text summarization using BERT," in *Proceedings of the 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, IEEE, Kaohsiung, Taiwan, November 2019.

[23] M. G. Sousa, "BERT for stock market sentiment analysis," in *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, Portland, OR, USA, November 2019.

[24] X. Yu, "BioBERT based named entity recognition in electronic medical record," in *Proceedings of the 2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, IEEE, Qingdao, China, August 2019.

[25] S. Jiang, "A BERT-BiLSTM-CRF model for Chinese electronic medical records named entity recognition," in *Proceedings of the 2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, IEEE, Xiangtan, China, October 2019.

[26] I. Beltagy, K. Lo, and A. C.. SciBERT, "A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, November 2019.

[27] J. Devlin, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[28] Z. Yang, "Xlnet: generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, vol. 19, pp. 400–406, 2019.

[29] L. Dong, "Unified language model pre-training for natural language understanding and generation," 2019, https://arxiv.org/abs/1905.03197.

[30] W. Wang, "StructBERT: incorporating language structures into pre-training for deep language understanding," 2019, https://arxiv.org/abs/1908.04577.

[31] X. Jiao, "Tinybert: distilling bert for natural language understanding," 2019, https://arxiv.org/abs/1909.10351.

[32] X. Liu, "Multi-task deep neural networks for natural language understanding," 2019, https://arxiv.org/abs/1901.11504.

[33] Z. Lan, "Albert: a lite bert for self-supervised learning of language representations," 2019, https://arxiv.org/abs/1909.11942.

[34] M. Shoeybi, "Megatron-lm: training multi-billion parameter language models using gpu model parallelism," 2019, https://arxiv.org/abs/1909.08053.

[35] W.-T. Kao, "Further boosting BERT-based models by duplicating existing layers: some intriguing phenomena inside BERT," 2020, https://arxiv.org/pdf/2001.09309.

[36] C. Raffel, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, https://arxiv.org/abs/1910.10683.
[37] H. Jiang, "SMART: robust and efficient fine-tuning for pre-trained Natural Language models through principled regularized optimization," 2019, https://arxiv.org/abs/1911.03437.
[38] C. Zhu, "Freelb: enhanced adversarial training for language understanding," 2019, https://arxiv.org/abs/1909.11764.
[39] T.-C. Su and H.-C. Cheng, "SesameBERT: attention for anywhere," 2019, https://arxiv.org/abs/1910.03176.
[40] K. Clark, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, May 2019.
[41] W. Wang, "MiniLM: deep self-attention distillation for task-agnostic compression of pre-trained transformers," 2020, https://arxiv.org/abs/2002.10957.
[42] C. Xu, "BERT-of-Theseus: compressing BERT by progressive module replacing," 2020, https://arxiv.org/abs/2002.02925.
[43] S. Goyal and PoWER-BERT, "Accelerating BERT inference for classification tasks," 2020, https://arxiv.org/abs/2001.08950.
[44] K. Clark, "Bam! born-again multi-task networks for natural language understanding," 2019, https://arxiv.org/abs/1907.04829.
[45] L. Gong, "Efficient training of bert by progressively stacking," in *Proceedings of the International Conference on Machine Learning*, Long Beach, CA, USA, May 2019.
[46] X. Liu, "Improving multi-task deep neural networks via knowledge distillation for natural language understanding," 2019, https://arxiv.org/abs/1904.09482.
[47] M. Moradshahi, "HUBERT untangles BERT to improve transfer across NLP tasks," 2019, https://arxiv.org/abs/1910.12647.
[48] D. Chen, "AdaBERT: task-adaptive BERT compression with differentiable neural architecture search," 2020, https://arxiv.org/abs/2001.04246.
[49] A. Chadha and R. Sood, "BERTQA_Attention on steroids," 2019, https://arxiv.org/abs/1912.10435.
[50] M. Lewis, "Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019, https://arxiv.org/abs/1910.13461.
[51] J. Wei, "NEZHA: neural contextualized representation for Chinese language understanding," 2019, https://arxiv.org/abs/1909.00204.
[52] H. Bao, "UniLMv2: Pseudo-masked language models for unified language model pre-training," 2020, https://arxiv.org/abs/2002.12804.
[53] B. Kitchenham, *Procedures for Performing Systematic Reviews*, Keele University, Keele, UK, 2004.
[54] Y.-C. Chen, "Distilling the knowledge of BERT for text generation," 2019, https://arxiv.org/abs/1911.03829.
[55] W.-C. Chang and X.- BERT, "eXtreme multi-label text classification using bidirectional encoder representations from transformers," 2019.
[56] R. Jozefowicz, "Exploring the limits of language modeling," 2016, https://arxiv.org/abs/1602.02410.
[57] Y. Xu, "Improving BERT fine-tuning via self-ensemble and self-distillation," 2020, https://arxiv.org/abs/2002.10345.