

RESEARCH ARTICLE

Can Bidirectional Encoder Become the Ultimate Winner for Downstream Applications of Foundation Models?

Lewen Yang¹, Xuanyu Zhou¹, Juao Fan¹, Xinyi Xie¹, Shengxin Zhu¹

¹ Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, Hong Kong Baptist University, Hong Kong

Funding: This work was funded by the Natural Science Foundation of China (12271047); Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, BNU-HKBU United International College (2022B1212010006); UIC research grant (R0400001-22; UICR0400008-21; UICR0400036-21CTL; UICR04202405-21); Guangdong College Enhancement and Innovation Program (2021ZDZX1046).

Potential competing interests: No potential competing interests to declare.

Abstract

Over the past few decades, Artificial Intelligence(AI) has progressed from the initial machine learning stage to the deep learning stage, and now to the stage of foundational models. Foundational models have the characteristics of pre-training, transfer learning, and self-supervised learning, and pre-trained models can be fine-tuned and applied to various downstream tasks. Under the framework of foundational models, models such as Bidirectional Encoder Representations from Transformers(BERT) and Generative Pre-trained Transformer(GPT) have greatly advanced the development of natural language processing(NLP), especially the emergence of many models based on BERT. BERT broke through the limitation of only using one-way methods for language modeling in pre-training by using a masked language model. It can capture bidirectional context information to predict the masked words in the sequence, this can improve the feature extraction ability of the model. This makes the model very useful for downstream tasks, especially for specialized applications. The model using the bidirectional encoder can better understand the domain knowledge and be better applied to these downstream tasks. So we hope to help understand how this technology has evolved and improved model performance in various natural language processing tasks under the background of foundational models and reveal its importance in capturing context information and improving the model's performance on downstream tasks.

This article analyzes one-way and bidirectional models based on GPT and BERT and compares their differences based on the purpose of the model. It also briefly analyzes BERT and the improvements of some models based on BERT. The model's performance on the Stanford Question Answering Dataset(SQuAD) and General Language Understanding Evaluation(GLUE) was compared.

I. Introduction

In 2021, models with high transferability and self-supervised learning were redefined as foundational models (Refer to the image in Figure 1), marking a new stage in the development of AI^[1]. The popularity of GPT^[2] also made more people see

the potential of foundational models. The high transferability of foundational models is analogous to the smartphone, where developers create an initial yet well-architected model that is ready for immediate use, while users have the flexibility to customize it according to their specific needs. Clearly, this is a highly efficient model, where people don't need to build a house from scratch based on their needs. Moreover, self-supervised learning not only makes the model more transferable, but also greatly reduces the cost of using data. Current general large models can be viewed as foundational models, they have very extensive knowledge and can be used directly or slightly adjusted to achieve higher accuracy. There are two important branches: the one-way model led by GPT and the bidirectional model led by BERT^[3]. GPT has achieved great success in practical applications in recent years and has gained a lot of traffic. However, we are currently in an era where larger models tend to perform better. Simply increasing the size of a model does not necessarily mean that it understands what it is generating. Describing GPT-like one-way generative models as having “learned a language” may be more appropriate^[4]. Foundational one-way models generate things more like the most likely word following the previous word.

On the other hand, models like BERT are different. We can simply differentiate them based on their purpose. One-way models aim to predict what will come next, while bidirectional models are more about extracting features, in other words, understanding the meaning of each word. Of course, one-way models also extract features, but they always do so in a sequential order. However, real humans do not completely process information in a one-way manner. For example, “I am happy because the weather is good.” A one-way model cannot understand that the reason why I am happy is the weather is good, because the one-way model predicts the next word and gives the word with the highest probability. But bidirectional models can process information from both directions, allowing the model to better understand the whole sentence and context, and achieve better results.

Better results also make models with bidirectional encoders have a variety of applications and improvements. But as the models become larger, due to the inherent complexity of bidirectional models, they require more time and computing power than one-way models, and driven by interests, it is clear that pure one-way models have a higher cost performance at present. Of course, understanding and being able to do are certainly different. With the improvement of computing power, the situation may change in the future.

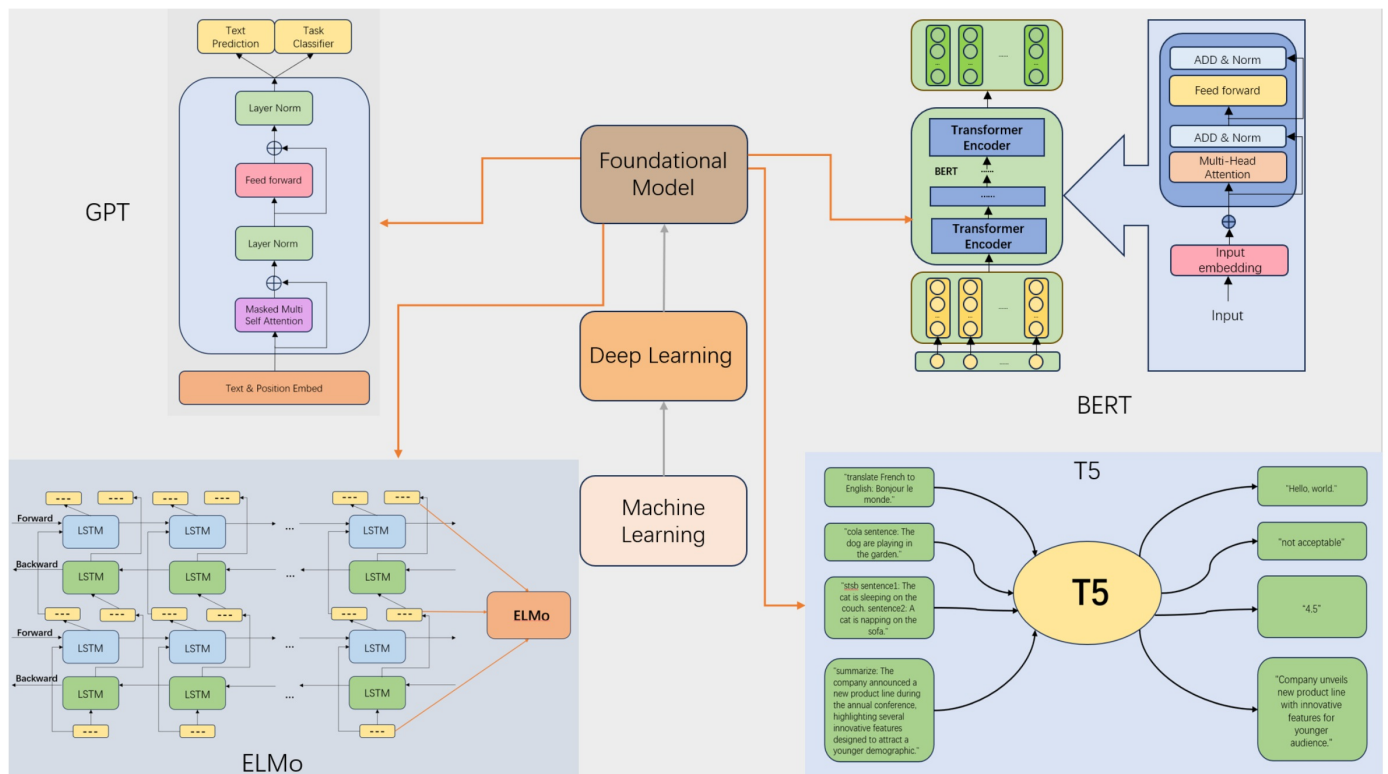


Figure 1. Foundational Model Introduction

II. The Origin of Bidirectional Encoders and BERT

A. Two sites of foundational model

GPT and BERT can be considered among the earliest foundational models, representing two different approaches. GPT is usually called a one-way language model, also known as a causal language model, and it is a recurrent model. This type of model calculates the probability of the next word in a single direction. In contrast, BERT is considered a bidirectional language model, despite the lack of a unified definition for bidirectional language models in the field. There is a consensus that a bidirectional language model means considering both forward and backward information when processing current words, and its training goal is to obtain better representational capacity (better feature extraction ability). The birth of GPT and BERT starts with Transformer^[5], which is mainly composed of two parts: encoder and decoder. Transformer introduces a self-attention mechanism. For each position in the sequence, the self-attention mechanism computes the query, the key and value vectors, and the attention weight, and uses the attention weight to weigh the sum of the value vectors to get the final output representation. In the Transformer encoder architecture, each encoding layer's multi-head self-attention sublayer will interact with each position in the input sequence with all other positions in the sequence, so that the self-attention layer when encoding a word will consider the entire sentence's words when encoding a word. However, in the decoder, the self-attention layer uses a mask matrix so that each position in the sequence can only see the sequence before it and the positions behind it will be hidden. GPT uses the decoder part of the Transformer, while BERT uses the encoder part. Because the encoder of self-attention mechanism considers both forward and backward

information, we also call it bidirectional encoder.

B. BERT and its applications

Both encoders and decoders have very strong representation capabilities thanks to the multi-head attention mechanism of Transformer, which makes it possible to train models on unlabeled datasets. The pre-training and fine-tuning paradigm then emerged in the NLP field. GPT is the earliest one, which introduced the concept of semi-supervised learning, later also known as self-supervised learning. This mainly refers to training the model on unlabeled datasets during the pre-training stage, which is unsupervised learning, allowing the model to extract more effective features^[2]. Then, the model is fine-tuned using labeled datasets specific to the application scenarios, which is supervised learning, allowing the model to have better performance in the corresponding application scenario. For example, researchers can use large amounts of unlabeled data from various fields for pre-training, and then use labeled data from the medical field to fine-tune the model so that it can be better applied to medical problems, which researchers will explain in detail later. BERT, as a later model, inherited the experience of its predecessors and brought the development of the foundational model to a new stage^[1].

BERT builds on ideas from GPT by using the Transformer encoder and the pre-training and fine-tuning approach. Additionally, it incorporates concepts from Embeddings from Language Models (ELMo)^[6], but in a different way. model's bidirectional approach. The method used by ELMo is to pre-train word vectors using two one-way Long Short-Term Memory networks(LSTMs)^[7], thus changing the traditional word embedding representation. In traditional word embedding representations, such as Word2Vec^[8] and Global Vectors for Word Representation(GloVe)^[9], each word has a fixed vector and does not consider the different meanings of words in different contexts. ELMo, on the other hand, assigns a vector to each word that varies with the context of the entire sentence by combining the hidden states of a forward LSTM and a backward LSTM. Therefore, the representation of each word is based on the current entire sentence, resulting in a dynamic word vector. This solves the problem of ambiguity of words in different sentences. Meanwhile, BERT has an excellent deeply bidirectional mechanism, unlike ELMo which only performs bidirectional operation at the surface level. BERT considers contextual information from both directions at every layer of the network, which can capture more comprehensive contextual information. Therefore, compared to sequential models such as GPT, BERT has a significant advantage in feature extraction. Excellent feature extraction ability allows the model to be deeper, which means the model will have stronger generalization ability and can solve multiple different types of problems with a single model^[3].

Although BERT is not the first to use pre-training and fine-tuning in NLP, it's the one that made it popular. Because of its powerful representation ability, BERT has significantly better pre-training and fine-tuning results than GPT. At the same time, BERT also marks the beginning of the model size competition, and the trend of better results with larger models can be seen from here. Later, when GPT-2^[10] was developed, the research team also found that even with significant improvements to the model, the results were not much better than those of BERT, so they took a different approach which led to "*Language Models are Few-Shot Learners*"(GPT-3)^[11].

BERT achieves such good results not only because of its bidirectional encoder, but also because of two important tasks it performs during pre-training: Mask Language Model (MLM) and Next Sentence Prediction (NSP)^[3].

- **MLM:** Its working principle is to randomly mask certain words in a sentence and then train the model to predict these masked words based on the context (similar to filling in the missing parts of a sentence). This approach allows the model to learn bidirectional information by considering the context simultaneously, whereas traditional models only process text in one direction.
- **NSP:** And the goal of this task is to enable the model to understand the relationships between sentences. Simply put, it achieves this by predicting whether the following sentence is the logical next sentence of the preceding one. This helps the model grasp relational information between sentences, thereby optimizing its performance.

During fine-tuning, the model only needs to be trained briefly to adapt it to a specific NLP task. The main process of fine-tuning is to initialize the model with learned parameters from pre-training and further train it on labeled data of the target task. For each specific task, a simple output layer is usually added to the BERT model, which can transform the pre-trained universal language representation into the prediction results (or convert the language which is more suitable for machines into the language that is more suitable for humans and meets the requirements). During fine-tuning, the model will use the labeled data from the target task to train, and adjust all parameters and then added output layer. It allows the model to focus more attention on the required task while retaining the broad language understanding acquired during pre-training.

That's why BERT fine-tuning has many applications. In the professional field, for example, SciBERT^[12] is trained on a large scientific text corpus and used for NLP tasks in scientific literature. ClinicalBERT^[13] is pre-trained on clinical text for processing electronic health records and other medical texts. BioBERT^[14] is pre-trained on biomedical text (PubMed abstracts and PMC full-text articles) for biomedical NLP tasks. BERTweet^[15] is the first large-scale pre-trained language model for English tweets, pre-trained on social media text (such as Twitter) for processing informal texts. Of course, there are also fine-tuned models for various languages, such as CamemBERT^[16] and FlauBERT^[17] for French, BERTje^[18] for Dutch, and AraBERT^[19] for Arabic, etc.

C. Disadvantages of Bidirectional Model

Of course, higher representational ability and stronger understanding ability also come with a cost. Because of utilizing contextual information simultaneously, bidirectional models actually require more computation and time resources at every step of training than one-way models^[20]. Additionally, due to the complex structure of bidirectional models, which handle information in both directions, it is more difficult to explain the operation of the model^[21]. Furthermore, since bidirectional models do more word embedding and feature extraction, training is done using a method similar to cloze tasks, and it seems difficult to make it perform generative tasks at present. It is more suitable for classification and judgment tasks^[22].

III. BERT and Its Evolution

Within just a few years, the BERT model has become a foundational tool in NLP. However, as mentioned in section II.C,

while the traditional BERT model performs exceptionally well in NLP tasks, it has several limitations such as high computational resource requirements, difficulty in interpretability, and unsuitability for generative tasks.

To address these limitations, researchers have proposed a series of improved models. Here, we list some models that excel in terms of computational resource optimization, interpretability, or performance in generative tasks, aiming to facilitate understanding for researchers.

A. *DistilBERT and TinyBERT*

A distil version of BERT (DistilBERT)^[23] employs knowledge distillation to condense the original BERT model into a lighter one while retaining most of its performance. This model significantly reduces the number of parameters and computational load, thus speeding up inference and lowering memory usage, making it ideal for resource-constrained environments.

TinyBERT^[24], through distillation techniques, compresses the BERT model into a smaller version. This model maintains high accuracy while offering fast inference speed and low resource consumption, making it suitable for real-time or embedded systems.

B. *ALBERT*

A lite BERT (ALBERT)^[25] uses parameter sharing and factorized embedding parameterization to drastically reduce the number of parameters. This approach decreases the demand for computational resources while maintaining performance comparable to BERT, and in some tasks, even surpassing the original BERT model.

C. *ERNIE*

Enhanced Representation through Knowledge Integration (ERNIE)^[26] improves semantic understanding through combining external knowledge bases. This knowledge injection not only boosts performance in specific domain tasks but also lets the decision-making process of the model clearer and easier to interpret.

D. *T5*

Text-To-Text Transfer Transformer(T5)^[27] transforms all kinds of NLP tasks into text-to-text format, enabling it to handle various generative tasks. It excels in translation, summarization, and question answering, demonstrating strong capabilities in generative tasks.

E. *SpanBERT*

SpanBERT^[28] focuses on learning span-based representations, capturing relationships and entity information in sentences better than traditional BERT. It performs exceptionally well in information extraction and relation detection

tasks, offering higher interpretability.

These models can be applied to many downstream applications, like CycleTrans^[29] and EchoMamba4Rec^[30] which can expand BERT application idea. For example, CycleTrans effectively integrates patients' EMR data through the combination of cross-attention mechanisms and bidirectional encoders, achieving bidirectional transmission between symptoms and medications, and attaining multiple State-Of-The-Art (SOTA) performances. Besides, EchoMamba4Rec combines bidirectional State Space Models (SSM) and frequency-domain filtering techniques. Its architecture includes bidirectional processing modules, Fast Fourier Transform (FFT), and more, enabling the model to capture complex dependencies. Following CycleTrans, EchoMamba4Rec has also achieved SOTA performances.

Additionally, we have sorted papers on Google Scholar that cited the foundational BERT model by citation count and selected some as representatives of BERT's wide application in various fields. These papers are then arranged chronologically, summarizing each model's main contributions in terms of approach and performance. This compilation aims to provide researchers with useful references. Figure 2 is the timeline of these model Details can be found in Appendix 1.

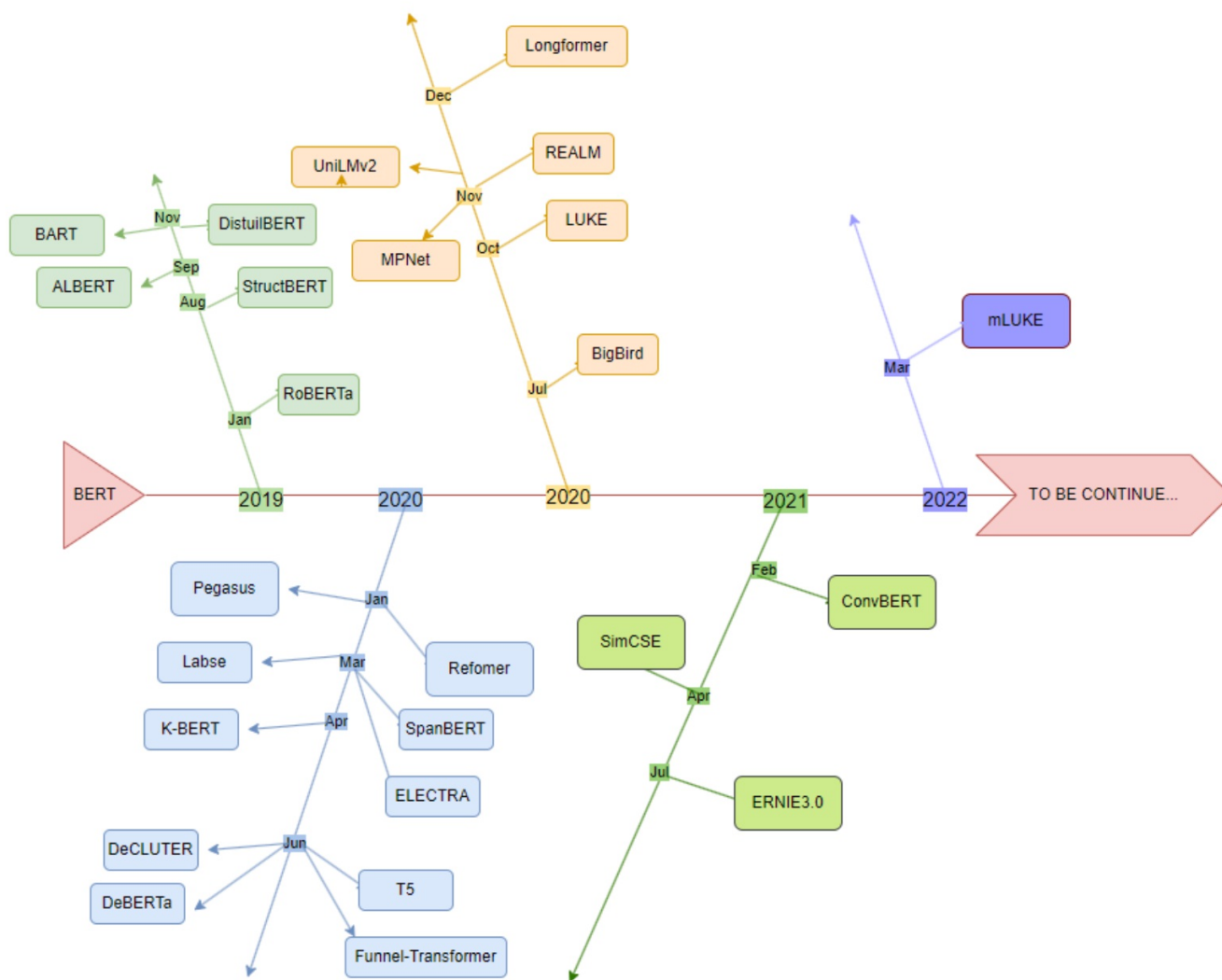


Figure 2. BERT Model Timeline

Table I. Model Performance Comparison

Model	Params	GLUE Avg.	SQuAD 1.1 F1/EM	SQuAD 2.0 F1/EM
BERT (Devlin et al., 2018) ^[3]	340M(334M)	82.1	93.2/87.4	83.1/80.0
XLNet (Yang et al., 2019) ^[31]	355M	90.5	*95.1/*89.7	90.7/87.9
RoBERTa (Liu et al., 2019) ^[20]	355M	88.5	*94.6/*88.9	89.8/86.8
StructBERT (Wang et al., 2019) ^[32]	340M	83.9	*85.2/*92.0	-
ALBERT (Lan et al., 2019) ^[25]	235M	89.4	95.5/90.1	91.4/88.9
DistilBERT (SanH et al., 2020) ^[23]	66M	77	*86.9/*79.1	-
BART (Lewis et al., 2020) ^[33]	374M	-	94.6/88.8	89.2/86.1
ELECTRA (Clark et al., 2020) ^[34]	335M	89.5	*94.9/*89.7	91.4/88.7
Funnel-Transformer (Dai et al., 2020) ^[35]	488M	89.7	*94.7/*89.0	*90.4/*87.6
SpanBERT (Joshi et al., 2020) ^[28]	340M	82.8	94.6/88.8	88.7/85.7
ConvBERT (Jiang et al., 2020) ^[36]	106M	86.4	90.0/84.7	83.1/80.6
MPNet (Song et al., 2020) ^[37]	110M	86.5	*92.7/*86.9	85.8/82.8
LUKE (Yamada et al., 2020) ^[38]	483M	-	95.4/90.2	-
UNILMv2 (Bao et al., 2020) ^[39]	110M	87.3	92.0/85.6	83.6/80.9
DeBERTa (He et al., 2021) ^[40]	433M	90.0	95.5/90.1	90.7/88.0

^aThe dev test is annotated with “**”

^bNote: Missing results in literature are signified by “-”

IV. Model Performance Comparison

Since most models are evaluated using GLUE and SQuAD, we use these two methods to compare some models[Table I].

A. GLUE

Human language understanding ability is universal, flexible, and robust. In contrast, most word-level and above NLU models are designed for specific tasks and are difficult to handle out-of-domain data.

Wang et al.(2019) proposed the GLUE benchmark^[41], which covers a set of natural language understanding (NLU) tasks, including question answering, sentiment analysis, and semantic inference, and also provides an online platform for evaluating, comparing, and analyzing models. The GLUE framework prioritizes models that can efficiently learn from samples and effectively transfer knowledge between tasks, thus representing language knowledge in the best possible way. Since GLUE’s goal is to advance general NLU systems, we designed this benchmark test to require models to share a large amount of knowledge (e.g., trained parameters) across all tasks while still retaining some task-specific components.

GLUE includes the following tasks:

- **Corpus of Linguistic Acceptability (CoLA)**: judging the grammatical correctness of sentences^[42].
- **Stanford Sentiment Treebank 2 (SST-2)**: single-sentence sentiment classification task^[43].
- **Microsoft Research Paraphrase Corpus (MRP)**: sentence pair semantic equivalence judgment^[44].
- **Semantic Textual Similarity Benchmark (STS-B)**: sentence pair semantic similarity score^[45].
- **Quora Question Pairs (QQP)**: question pair semantic equivalence judgment.
- **Multi-Genre Natural Language Inference (MNLI)**: judging the inference relationship between sentence pairs^[46].
- **Question Natural Language Inference (QNLI)**: inference task based on question and sentence^[41].
- **Recognizing Textual Entailment (RTE)**: text entailment task^[47].
- **Winograd Natural Language Inference (WNLI)**: inference task based on Winograd schema^[48].

The GLUE benchmark provides a diverse set of tasks, allowing researchers to comprehensively evaluate the performance of models on different types of tasks and thus gain insights into their general language understanding ability.

B. SQuAD

SQuAD tests the models' abilities in reading comprehension and information extraction, which is an important indicator of the models' understanding and generation capabilities^[49].

a) *SQuAD 1.1*: Known as Reading Comprehension (RC), comprehending written text and responding to questions about it, presents a significant challenge for machines. This task demands a deep understanding of natural language and worldly knowledge. SQuAD comprises 107,785 pairs of questions and answers based on 536 articles. Systems are required to choose the answer from numerous potential spans within the passage, necessitating the ability to handle a large number of candidates. The constraint of selecting from specific spans also brings the important advantage that evaluating span-based answers is simpler than assessing freeform answers.

b) *SQuAD 2.0*: This challenge involves the precise formulation of various types of natural language representation and comprehension to facilitate the processing of a question and its corresponding context. Subsequently, it entails selecting an appropriate correct answer that is deemed satisfactory by humans or indicating the absence of such an answer. Each question may have a definite answer, or none at all. The unanswerable questions should appear relevant to the topic of the context paragraph. The system needs to be able to not only find the correct answer but also identify what questions are without answers^[50]. Evaluation is done using F1-score and Exact Match(EM), with higher scores indicating better performance. With these features and methods, SQuAD2.0 has become an important benchmark for evaluating and improving question-answering systems.

V. Future Work

At present, it seems that models using bidirectional encoders have not produced a game-changing product like GPT. Many large companies have temporarily abandoned bidirectional feature extraction models and moved towards one-way generative models, such as Google switching to one-way generative models after completing T5. Of course, while bidirectional models have slowed down in recent years, their potential remains huge. Because in most cases, the understanding of each word by a one-way model is generally not as good as that of a bidirectional model, simplified, GPT-like models are aimed at generating, and they may not care about the intermediate process, because the training goal is to generate the correct answer. However, BERT-like models are different, these models aim to extract features as much as possible, that is, to understand the meaning of each word, which is obviously a more human-like way of thinking.

Of course, as mentioned earlier, higher computing cost and time consumption are still serious problems for bidirectional models at present, which is also the price of higher effectiveness. Only large companies have the ability to independently discover game-changing bidirectional large models. However, we believe that bidirectional models still have great potential, and we can use new acceleration inference and energy-saving training methods such as pruning, distillation, and quantization or continue researching small models to achieve this. Additionally, the recent popularity of Mamba2^[51], which has made improvements to self-attention, may also be used to speed up training time.

Another aspect is explainability, traditional explainability often visualizes the process of the model's output to explain the model, but this is difficult for bidirectional models. So, perhaps a way for the model to explain the process itself is more suitable for bidirectional models. Because the focus of bidirectional models is understanding, if they understand the meaning of words and sentences, then it is possible for the model to explain the origin of the answer on its own.

Furthermore, in today's trend towards multi-modal large models, it seems that the need for adding task-specific classification heads to BERT models to achieve better performance on specific tasks has become no longer popular. However, this does not imply that bidirectional encoders are obsolete. For instance, Google Brain introduced Unifying Language Learning Paradigms (UL2), which integrates various language model training tasks, including span masking, enabling UL2 to possess bidirectional context understanding capabilities.^[52] In the future, there may also emerge improved models using bidirectional encoders.

VI. Conclusion

The development of AI has experienced from the initial machine learning stage to the deep learning stage, and then to the foundational model stage today. The foundation model has the characteristics of pre-training, transfer learning and self-supervised learning. The pre-trained model can be applied to various downstream tasks after fine tuning. Within the realm of foundation models, models such as BERT and GPT have greatly advanced the development of natural language processing. At the same time, due to the excellent portability of BERT, the model based on BERT continue to emerge. BERT breaks through the limitation of using only one-way method for feature extraction in pre-training by using bidirectional encoder, that is, the model can capture bidirectional context information to predict mask words in the sequence, which improves the feature extraction capability of the model. The good feature extraction capability also

makes subsequent improvements possible. In this paper, a brief comparison of GPT and BERT is made, and some BERT-based models and their improvements are analyzed. Finally, the performance of the bidirectional encoder-based model on SQuAD and GLUE where data can be found is collected. The models' performance in various natural language processing tasks reveals the importance of bidirectional encoders in improving the model's ability to generalize, as well as their potential for downstream applications in foundational models.

Table II. Model Contributions and Performance

Model	Time	Contribution (approach)	Contribution (performance)
BERT (Devlin et al., 2018) ^[3]	10/2018	Pre-training with bidirectional Transformer for the first time	Significantly improve the performance of natural language understanding tasks
RoBERTa (Liu et al., 2019) ^[20]	1/2019	Proposed an optimized pre-training method with better hyperparameter tuning	Enhanced BERT's overall performance
DistilBERT (Sanh et al., 2019) ^[23]	10/2019	Applied knowledge distillation to create a lightweight BERT version	Reduced model size and computation
ALBERT (Lan et al., 2019) ^[25]	9/2019	Used parameter sharing and factorization techniques	Reduced model parameters and improved parameter efficiency
StructBERT (Wang et al., 2019) ^[32]	8/2019	Introduced word order and sentence order prediction tasks	Enhanced bidirectional representation capability
BART (Lewis et al., 2019) ^[33]	10/2019	Introduced a method combining bidirectional encoder and autoregressive decoder	Applied for generative tasks and text reconstruction
Reformer (Kitaev et al., 2020) ^[53]	1/2020	Used locality-sensitive hashing and reversible residual layers	Reduced computational complexity of bidirectional Transformer
Pegasus (Kitaev et al., 2020) ^[54]	1/2020	Designed pre-training tasks to simulate abstractive summarization	Improved performance on abstractive summarization tasks
SpanBERT (Joshi et al., 2020) ^[28]	3/2020	Improved masking scheme and training objectives	Better representation and prediction of text spans
ELECTRA (Clark et al., 2020) ^[34]	3/2020	Introduced a generator-discriminator framework	Increased pre-training efficiency
LaBSE (Feng et al., 2020) ^[55]	3/2020	Introduced a new pre-training and dual-encoder fine-tuning method	Achieved state-of-the-art performance in bi-text mining
K-BERT (Liu et al., 2020) ^[56]	4/2020	Enhanced pre-trained language models with entity-aware mechanisms <pre>[pre-train]</pre>	Improved performance on tasks involving entity information understanding <pre>[performance]</pre>
DeCLUTR (Giorgi et al., 2020) ^[57]	6/2020	Proposed a self-supervised sentence-level objective <pre>[architecture]</pre>	Generated generalized embeddings for sentences and paragraphs without labeled data <pre>[application]</pre>
T5 (Raffel et al., 2020) ^[27]	6/2020	Introduced a unified text-to-text framework for NLP tasks <pre>[architecture]</pre>	Enhanced task generalization <pre>[performance]</pre>
Funnel-Transformer (Dai et al., 2020) ^[35]	6/2020	Employed a funnel structure <pre>[architecture]</pre>	Reduced computational complexity while maintaining bidirectional representation capability <pre>[efficiency]</pre>
DeBERTa (He et al., 2020) ^[40]	6/2020	Enhanced decoding and disentangled attention mechanism <pre>[architecture]</pre>	Improved contextual information understanding <pre>[performance]</pre>
BigBird (Zaheer et al., 2020) ^[58]	7/2020	Utilized sparse attention mechanisms <pre>[architecture]</pre>	Enabled processing of longer text sequences <pre>[performance]</pre>
LUKE (Yamada et al., 2020) ^[38]	10/2020	Combined knowledge graph information with self-attention <pre>[architecture]</pre>	Enhanced performance in knowledge-intensive tasks <pre>[performance]</pre>
REALM (Guu et al., 2020) ^[59]	11/2020	Incorporated a latent knowledge retriever <pre>[pre-train]</pre>	Captured knowledge in a modular and interpretable way <pre>[performance]</pre>

MPNet (Song et al., 2020) ^[37]	11/2020	Combined masked language modeling and permuted language modeling[pre-train]	Enhanced representation capability[performance]
UniLMv2 (Bao et al., 2020) ^[39]	11/2020	Integrated multi-task learning and cross-task consistency[architecture]	Enhanced bidirectional Transformer representation capability[performance]
Longformer (Beltagy et al., 2020) ^[60]	12/2020	Combined local attention and global attention mechanisms[architecture]	Efficiently processed long texts[performance,efficiency]
ConvBERT (Jiang et al., 2021) ^[36]	2/2021	Combined convolutional neural networks with bidirectional Transformer[architecture]	Reduced computational complexity and improved representation capability[performance,efficiency]
SimCSE (Gao et al., 2021) ^[61]	4/2021	Applied simple contrastive learning[architecture]	Improved performance of bidirectional Transformer in sentence embedding tasks[performance]
ERNIE 3.0 (Sun et al., 2021) ^[26]	7/2021	Proposed a unified framework combining auto-regressive and auto-encoding networks[architecture]	Improved knowledge representation in multilingual settings[performance]
mLUKE (Ri et al., 2022) ^[62]	3/2022	Extended LUKE to multilingual versions[architecture]	Enhanced knowledge representation in multilingual environments[performance]

Acknowledgments

This work was funded by the Natural Science Foundation of China (12271047); Guangdong Provincial Key Laboratory of Interdisciplinary Research and Application for Data Science, BNU-HKBU United International College (2022B1212010006); UIC research grant (R0400001-22; UICR0400008-21; UICR0400036-21CTL; UICR04202405-21); Guangdong College Enhancement and Innovation Program (2021ZDZX1046).

References

- ^{a, b} Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, Bernstein H, Bohg J, Bosselut A, Brunskill E, Brynjolfsson S, Buch S, Cardie A, Cerezo MS, Chatterji N, Clark KL, D'Amour K, Deaton J, Durmus Z, Dwight T, Fei-Fei JE, Hsu WG, Jacob HI, Jurafsky DL, Kann SP, Krishna RM, Malik TA, Manning Z, Mittal CL, Moritz N, Russell J, Sahai CS, Stangl A, Talati K, Tesauro J, Weber PW, Liang P. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258*, 2021. Available from: <https://arxiv.org/abs/2108.07258>.
- ^{a, b} Radford A, Narasimhan K, Salimans T, Sutskever I (2018). "Improving language understanding by generative pre-training."
- ^{a, b, c, d, e} Devlin J, Chang M-W, Lee K, Toutanova K. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv:1810.04805 [cs.CL]*, Oct. 2018. Available from: <https://arxiv.org/abs/1810.04805>.
- [^] Fedorenko E, Piantadosi ST, Gibson EAF. "Language is primarily a tool for communication rather than thought." *Nature*. **630**:575–586, 2024.
- [^] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems* **30**.
- [^] Sarzynska-Wawer J, Wawer A, Pawlak A, Szymanowska J, Stefaniak I, Jarkiewicz M, Okruszek L (2021). "Detecting formal thought disorder by deep contextualized word representations." *Psychiatry Research*. **304**: 114135.

7. [^] Hochreiter S, Schmidhuber J (1997). "Long short-term memory." *Neural Computation*. **9** (8): 1735–1780.
8. [^] Mikolov T, Chen K, Corrado G, Dean J (2013). "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*. Available from: <https://arxiv.org/abs/1301.3781>.
9. [^] Pennington J, Socher R, Manning CD. "Glove: Global vectors for word representation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; 2014 Oct. p. 1532-1543.
10. [^] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019). "Language models are unsupervised multitask learners." *OpenAI Blog*. **1** (8): 9.
11. [^] Brown T, et al. "Language models are few-shot learners." in *Advances in Neural Information Processing Systems* **33**: 1877-1901, 2020.
12. [^] Beltagy I, Lo K, Cohan A (2019). "SciBERT: A pretrained language model for scientific text." *arXiv preprint arXiv:1903.10676*. Available from: <https://arxiv.org/abs/1903.10676>.
13. [^] Alsentzer E, Murphy JR, Boag W, Weng WH, Jin D, Naumann T, McDermott M (2019). "Publicly available clinical BERT embeddings." *arXiv preprint arXiv:1904.03323*. Available from: <https://arxiv.org/abs/1904.03323>.
14. [^] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." *Bioinformatics*. **36** (4): 1234–1240.
15. [^] Nguyen DQ, Vu T, Nguyen AT (2020). "BERTweet: A pre-trained language model for English Tweets." *arXiv preprint arXiv:2005.10200*. Available from: <https://arxiv.org/abs/2005.10200>.
16. [^] Martin L, Muller B, Suárez PJO, Dupont Y, Romary L, de La Clergerie ÉV, Seddah D, Sagot B (2019). "CamemBERT: A tasty French language model." *arXiv preprint arXiv:1911.03894*. Available from: <https://arxiv.org/abs/1911.03894>.
17. [^] Le H, Vial L, Frej J, Segonne V, Coavoux M, Lecouteux B, Allauzen A, Crabbe B, Besacier L, Schwab D. "FlauBERT: Unsupervised language model pre-training for French." *arXiv preprint arXiv:1912.05372*, Dec. 2019. Available from: <https://arxiv.org/abs/1912.05372>.
18. [^] de Vries W, van Cranenburgh A, Bisazza A, Caselli T, van Noord G, Nissim M. "BERTje: A Dutch BERT model." *arXiv preprint arXiv:1912.09582*, Dec. 2019. Available from: <https://arxiv.org/abs/1912.09582>.
19. [^] Antoun W, Baly F, Hajj H (2020). "AraBERT: Transformer-based model for Arabic language understanding." *arXiv preprint arXiv:2003.00104*. Available from: <https://arxiv.org/abs/2003.00104>.
20. ^{a, b, c} Liu M, Ott N, Goyal J, Du M, Joshi D, Chen O, Levy M, Lewis L, Zettlemoyer L, Stoyanov V. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv:1907.11692 [cs.CL]*, Jul. 2019. Available from: <https://arxiv.org/abs/1907.11692>.
21. [^] Tenney I, Das D, Pavlick E (2019). "BERT rediscovers the classical NLP pipeline." *arXiv preprint arXiv:1905.05950*. Available from: <https://arxiv.org/abs/1905.05950>.
22. [^] Chen H, Du W, Ji Y (2022). "Explaining predictive uncertainty by looking back at model explanations." *arXiv preprint arXiv:2201.03742*. Available from: <https://arxiv.org/abs/2201.03742>.
23. ^{a, b, c} Sanh V, Debut L, Chaumond J, Wolf T. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv:1910.01108 [cs.CL]*. Oct 2019. Available from: <https://arxiv.org/abs/1910.01108>.

24. ^a Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, ..., Liu Q (2019). "TinyBERT: Distilling BERT for natural language understanding." *arXiv preprint arXiv:1909.10351*. Available from: <https://arxiv.org/abs/1909.10351>.
25. ^{a, b, c} Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." *arXiv:1909.11942 [cs.CL]*, Sep. 2019. Available from: <https://arxiv.org/abs/1909.11942>.
26. ^{a, b} Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q (2019). "ERNIE: Enhanced language representation with informative entities." *arXiv preprint arXiv:1905.07129*.
27. ^{a, b} Raffel C, et al. (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer." *Journal of Machine Learning Research*. **21** (140): 1–67.
28. ^{a, b, c} Joshi D, Chen Y, Liu D, Weld DS, Zettlemoyer L, Levy O (2020). "SpanBERT: Improving Pre-training by Representing and Predicting Spans." *Transactions of the Association for Computational Linguistics*. **8**: 64-77.
29. ^a Zheng Y, Lin X, Chen K, Zhu S. "CycleTrans: A Transformer-Based Clinical Foundation Model for Safer Prescription." In: *AAAI 2024 Spring Symposium on Clinical Foundation Models*; Mar 2024.
30. ^a Wang Y, He X, Zhu S (2024). "EchoMamba4Rec: Harmonizing Bidirectional State Space Models with Spectral Filtering for Advanced Sequential Recommendation." *arXiv preprint arXiv:2406.02638*. Available from: <https://arxiv.org/abs/2406.02638>.
31. ^a Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. "XLNet: Generalized Autoregressive Pretraining for Language Understanding." *arXiv preprint arXiv:1906.08237*, June 2019. Available from: <https://arxiv.org/abs/1906.08237>.
32. ^{a, b} Wang W, Bi B, Yan M, Wu C, Bao Z, Xia J, Peng L, Si L. "StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding." *arXiv:1908.04577 [cs.CL]*, Aug. 2019. Available from: <https://arxiv.org/abs/1908.04577>.
33. ^{a, b} Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." *arXiv:1910.13461 [cs.CL]*. Available from: <https://arxiv.org/abs/1910.13461>.
34. ^{a, b} Clark K, Luong M-T, Le QV, Manning CD. "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." *arXiv:2003.10555 [cs.CL]*, Mar. 2020. Available from: <https://arxiv.org/abs/2003.10555>.
35. ^{a, b} Dai G, Lai Y, Yang, Q, Le Q (2020). "Funnel-transformer: Filtering out sequential redundancy for efficient language processing." *Advances in Neural Information Processing Systems* **33**: 4271–4282.
36. ^{a, b} Jiang ZH, Yu W, Zhou D, Chen Y, Feng J, Yan S (2020). "ConvBERT: Improving BERT with span-based dynamic convolution." *Advances in Neural Information Processing Systems* **33**: 12837-12848.
37. ^{a, b} Song K, Tan X, Qin T, Lu J, Liu TY (2020). "Mpnet: Masked and permuted pre-training for language understanding." *Advances in Neural Information Processing Systems* **33**: 16857-16867.
38. ^{a, b} Yamada I, Asai A, Shindo H, Takeda H, Matsumoto Y. "LUKE: Deep contextualized entity representations with entity-aware self-attention." *arXiv:2010.01057 [cs.CL]*, Oct. 2020. Available from: <https://arxiv.org/abs/2010.01057>.

39. ^{a, b}Bao H, Dong L, Wei F, Wang W, Yang N, Liu X, Hon HW. "Unilmv2: Pseudo-masked language models for unified language model pre-training." In: *Proceedings of the International Conference on Machine Learning PMLR*; Nov 2020. p. 642-652.
40. ^{a, b}He P, Liu X, Gao J, Chen W (2020). "DeBERTa: Decoding-enhanced BERT with Disentangled Attention." *arXiv:2006.03654 [cs.CL]*. Available from: <https://arxiv.org/abs/2006.03654>.
41. ^{a, b}Wang A, Singh A, Michael J, Hill F, Levy O, Bowman SR (2018). "GLUE: A multi-task benchmark and analysis platform for natural language understanding." *arXiv preprint arXiv:1804.07461*. Available from: <https://arxiv.org/abs/1804.07461>.
42. [^]Liu NF, Gardner M, Belinkov Y, Peters ME, Smith NA (2019). "Linguistic knowledge and transferability of contextual representations." *arXiv preprint arXiv:1903.08855*. Available from: <https://arxiv.org/abs/1903.08855>.
43. [^]Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C. "Recursive deep models for semantic compositionality over a sentiment treebank." In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*; 2013 Oct; p. 1631-1642.
44. [^]Dolan B, Brockett C. "Automatically constructing a corpus of sentential paraphrases." In: *Third International Workshop on Paraphrasing (IWP2005)*; 2005 Jan.
45. [^]Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L (2017). "Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation." *arXiv preprint arXiv:1708.00055*. Available from: <https://arxiv.org/abs/1708.00055>.
46. [^]Williams A, Nangia N, Bowman SR (2017). "A broad-coverage challenge corpus for sentence understanding through inference." *arXiv preprint arXiv:1704.05426*. Available from: <https://arxiv.org/abs/1704.05426>.
47. [^]Dagan I, Glickman O, Magnini B. "The PASCAL recognising textual entailment challenge." In: *Machine Learning Challenges Workshop*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005 Apr. p. 177-190.
48. [^]Levesque H, Davis E, Morgenstern L. "The Winograd schema challenge." In: *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*; May 2012.
49. [^]Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016). "SQuAD: 100,000+ questions for machine comprehension of text." *arXiv preprint arXiv:1606.05250*. Available from: <https://arxiv.org/abs/1606.05250>.
50. [^]Rajpurkar P, Jia R, Liang P (2018). "Know what you don't know: Unanswerable questions for SQuAD." *arXiv preprint arXiv:1806.03822*. Available from: <https://arxiv.org/abs/1806.03822>.
51. [^]Dao T, Gu A. "Transformers are SSMS: Generalized models and efficient algorithms through structured state space duality." *arXiv preprint arXiv:2405.21060*, 2024. Available from: <https://arxiv.org/abs/2405.21060>.
52. [^]Tay Y, Dehghani M, Tran VQ, Garcia X, Wei J, Wang X, ... Metzler D. "UL2: Unifying Language Learning Paradigms." *arXiv preprint arXiv:2205.05131*, 2022. Available from: <https://arxiv.org/abs/2205.05131>.
53. [^]Kitaev N, Kaiser Ł, Levskaya A (2020). "Reformer: The Efficient Transformer." *arXiv preprint arXiv:2001.04451*. Available from: <https://arxiv.org/abs/2001.04451>.
54. [^]Zhang J, Zhao Y, Saleh M, Liu P. "Pegasus: Pre-training with extracted gap-sentences for abstractive

- summarization." In: *International Conference on Machine Learning*; Nov 2020. p. 11328-11339. PMLR.
55. [^]Feng F, Yang Y, Cer D, Arivazhagan N, Wang W (2020). "Language-agnostic BERT sentence embedding." *arXiv preprint arXiv:2007.01852*. Available from: <https://arxiv.org/abs/2007.01852>.
 56. [^]Liu W, Zhou P, Zhao Z, Wang Z, Ju Q, Deng H, Wang P. "K-BERT: Enabling language representation with knowledge graph." In: *Proceedings of the AAAI Conference on Artificial Intelligence 2020 Apr*; **34**(03):2901-2908.
 57. [^]Giorgi J, Nitski O, Wang B, Bader G (2020). "Declutr: Deep contrastive learning for unsupervised textual representations." *arXiv preprint arXiv:2006.03659*. Available from: <https://arxiv.org/abs/2006.03659>.
 58. [^]Zaheer M, Guruganesh G, Dubey KA, Ainslie J, Alberti C, Ontanon S, ..., Ahmed A. "Big Bird: Transformers for longer sequences." *Advances in Neural Information Processing Systems* **33**: 17283-17297, 2020.
 59. [^]Guu K, Lee K, Tung Z, Pasupat P, Chang M. "Retrieval Augmented Language Model Pre-training." In: *International Conference on Machine Learning*; Nov 2020. p. 3929-3938. PMLR.
 60. [^]Beltagy I, Peters ME, Cohan A (2020). "Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150*. Available from: <https://arxiv.org/abs/2004.05150>.
 61. [^]Gao T, Yao X, Chen D (2021). "SimCSE: Simple Contrastive Learning of Sentence Embeddings." *arXiv preprint arXiv:2104.08821*. Available from: <https://arxiv.org/abs/2104.08821>.
 62. [^]Ri R, Yamada I, Tsuruoka Y (2021). "mLUKE: The Power of Entity Representations in Multilingual Pretrained Language Models." *arXiv preprint arXiv:2110.08151*. Available from: <https://arxiv.org/abs/2110.08151>.