A Survey on Deep Learning Language Models (Postprint)

Authors: Wang Sili, Zhang Ling, Yang Heng, b Liu Wei

Date: 2024-04-03T00:00:00+00:00

Abstract

[Purpose/Significance] Deep learning language models represent one of the primary approaches for advancing machine language intelligence and have become indispensable technical instruments for the automatic processing and analysis of data resources as well as the intelligent mining and computation of knowledge and information. Nevertheless, significant challenges persist in leveraging these models for technological development and application services within the domain of library and information science. This study systematically reviews and elucidates the research progress, technical principles, and application development methodologies of deep learning language models, aiming to furnish librarians and fellow practitioners with theoretical foundations and methodological pathways for the in-depth comprehension and application of these models.

[Method/Process] The study systematically investigates and organizes the genesis background, fundamental feature representation algorithms, and representative application development tools of deep learning language models, revealing their evolutionary trajectory and underlying technical principles, while analyzing the merits, limitations, and applicability of various algorithmic models and development tools. Furthermore, it comprehensively synthesizes the critical challenges confronting the application development of deep learning language models and proposes two strategic methodologies for expanding their application capabilities.

[Results/Conclusion] The principal challenges in developing applications for deep learning language models encompass: extensive parameters that are difficult to tune with precision; heavy dependence on large volumes of accurate training data, impeding adaptability; and potential issues related to intellectual property rights and information security. Future endeavors may consider enhancing and expanding their application capabilities through two primary avenues: domain-specific orientation and feature engineering.

Full Text

Preamble

Deep Learning Language Models: A Research Review

Authors: WANG Sili[1], ZHANG Ling[2], YANG Heng[1], LIU Wei[1]
Affiliations:
1. Literature and Information Center, Northwest Institute of Eco-Environment and Resources, Chinese Academy of Sciences, Lanzhou 730000
2. School of Management, Xinxiang Medical University, Xinxiang 453003

Abstract:
[Purpose/Significance] Deep learning language models represent one of the primary methods for enhancing machine language intelligence and have become indispensable technical means for automatic data processing, analysis, and intelligent knowledge mining. However, applying these models for technology development and service provision in the library and information science (LIS) field remains challenging. This study systematically reviews and reveals the research progress, technical principles, and application development methods of deep learning language models to provide theoretical foundations and methodological pathways for librarians and practitioners to deeply understand and apply these models. [Method/Process] We systematically investigated the emergence background, basic feature representation algorithms, and representative application development tools of deep learning language models, revealing their evolutionary trajectory and technical principles while analyzing the advantages, disadvantages, and applicability of each algorithm and tool. We further summarized the key challenges in developing and applying deep learning language models and proposed two strategic approaches to expand their application capabilities. [Results/Conclusions] The major challenges include numerous parameters that are difficult to tune accurately, heavy reliance on large volumes of accurate training data, difficulties in adaptation, and potential intellectual property and information security issues. Future efforts should focus on domain-specific adaptation and feature engineering to expand and enhance application capabilities.

Keywords: deep learning; language model; neural network; pre-trained model; word embedding
Classification Numbers: G202, G250.73, TP391

---

1 Introduction

Deep learning has become a focal point in artificial intelligence and machine learning, and will undoubtedly serve as an essential technological foundation for automatic data processing and intelligent knowledge mining in the LIS field. Building on this understanding, this paper systematically investigates the

emergence background, basic feature representation algorithms, and representative application development tools of deep learning language models, revealing their evolutionary dynamics and technical principles. Deep learning has already demonstrated tremendous potential and broad prospects in natural language processing, computer vision, and related tasks such as text classification and image/speech recognition.

This study further analyzes the advantages and disadvantages of various algorithm models and development tools, and deeply summarizes the challenges faced in developing and applying deep learning language models. We propose two methods to expand their application capabilities. Even well-trained models may still fail to make satisfactory decisions in different application contexts. Machine learning typically achieves great success in relatively specialized domains with limited solution spaces, such as Japanese chess, where numerous algorithmic models have been developed with varying accuracy levels. However, once accuracy reaches a certain saturation point, the ultimate determinant of machine learning performance remains data and features. The ability to abstract features is crucial—machine learning cannot autonomously complete feature engineering yet heavily depends on it, which often becomes a bottleneck in achieving artificial intelligence.

---

2 Emergence Background of Deep Learning Language Models

Language and intelligence are uniquely human capabilities. Enabling machines to understand and express natural language like humans, thereby achieving higher-level intelligent behavior, has always been a primary goal and significant challenge in artificial intelligence. Language models are considered the main approach to enhancing machine language intelligence and have received widespread attention from academia and industry. Before machine learning emerged, machines could hardly process unknown data intelligently without artificial intelligence. Machine learning methods train on large sample datasets to learn knowledge patterns, which are then applied to unknown problems. However, not all data can serve as training samples—only appropriate and relevant data enables machines to make correct predictions. Machines cannot judge the suitability of sample data themselves, nor can they clearly understand what to learn from it. Humans must collect, create, and provide data as input for machine learning.

Deep learning emerged precisely to solve these challenges. Deep learning language models essentially represent an extension and improvement of neural network algorithms in traditional machine learning methods. Conventional neural network models often treat the entire multi-layer network as a single massive neural network for training, which has proven highly limited. When the network has multiple complex layers, errors transmitted backward during training

may gradually diminish and eventually disappear, making it difficult for the top input layer to receive correct error feedback and preventing effective parameter adjustment and optimization. The key to successful deep learning language models lies in allowing each layer to participate in training at its corresponding stage, using the previous layer' s output as the next layer' s input to progressively complete learning from shallow to deep, from simple to complex. Since each layer participates in learning, error feedback can be processed promptly at every level, and different learning methods can be employed according to actual conditions. This enables machines to automatically learn deep, complex features from shallow, simple ones through pre-training.

Deep learning differs significantly from other machine learning methods. Figure 1 illustrates the differences and connections between deep learning and other approaches. Traditional machine learning algorithms such as support vector machines, naive Bayes, and conditional random fields still dominate LIS-related business domains because they are cost-effective, widely validated, and can be quickly updated. Unless supported by sufficient capital infrastructure and guarantee strategies, deep learning cannot completely replace these more economical machine learning methods in the short term.

[Figure 1: see original paper]

---

3 Feature Representation Algorithms for Deep Learning Language Models

Both supervised and unsupervised learning essentially seek optimal feature representation. Supervised learning uses manually or automatically labeled datasets, aiming to minimize feature classification errors. Unsupervised learning uses raw, unprocessed datasets, aiming to preserve original data characteristics as much as possible while reducing feature dimensions. Before deep learning emerged, commonly used language feature representation models included Boolean logic models, vector space models, one-hot representation models, LDA topic models, N-gram statistical language models, and neural network language models (NNLM). However, these models had various limitations and limited feature representation and learning capabilities.

3.1 Deep Belief Network Algorithm

Deep Belief Networks (DBN), first proposed by Hinton et al. from the University of Toronto in 2006, began to develop rapidly as a foundational deep learning direction around 2012. DBN can be used not only for feature representation and data classification but also for generating training data. The core idea is to construct a probabilistic generative model of the joint distribution between observed data and labels. DBN is essentially a deep neural network with multiple hidden layers added between the original input and output layers. The basic

process generally includes pre-training and fine-tuning. Model parameters are learned layer by layer during pre-training and then fine-tuned as a single neural network.

The main difference among deep learning language feature representation algorithms lies in their pre-training and fine-tuning methods. DBN' s pre-training method often uses Restricted Boltzmann Machines (RBM). An RBM is a restricted, binary, undirected graphical model limited to a visible layer and a hidden layer, where neurons in the same layer cannot be connected but neurons between layers have independent random states. The training process of a single RBM essentially seeks the maximum probability distribution of training samples, which depends on weights. Therefore, the ultimate goal of training an RBM is to find optimal weights. The DBN feature learning process can thus be viewed as using greedy algorithms like logistic regression to train RBMs layer by layer to obtain optimal weights.

3.2 Convolutional Neural Network Algorithm

Traditional machine learning algorithms mostly accept one-dimensional input data. However, real-world applications often involve higher-dimensional data such as time series or images, which cannot be simply converted to one-dimensional input without discarding valuable information like temporal or positional data. Convolutional Neural Networks (CNN) were proposed specifically to handle grid-structured data of two or more dimensions, such as time series data with a time axis or image data that can be viewed as two-dimensional pixel grids.

In CNNs, the input layer can receive and process multi-dimensional data such as standardized two-dimensional, three-dimensional, or four-dimensional arrays. The hidden layer contains convolutional layers that perform feature extraction through convolution and pooling operations. A convolutional layer typically contains multiple convolution kernels (filters), where each element has a weight value and bias vector. After convolutional layers extract features, the output is passed to pooling layers for feature selection and filtering (sub-sampling). The general training flow is: convolution → sub-sampling → convolution → sub-sampling → fully connected → output, where the convolution-sub-sampling process can be repeated iteratively as needed. The fully connected layer, built at the end of CNN hidden layers, converts three-dimensional feature data into vectors and passes them through activation functions to the next layer. The output layer typically uses logistic regression or normalization functions to produce final classification labels.

3.3 Recurrent Neural Network Algorithm

While multi-layer perceptrons in general deep learning methods perform well in individual case recognition and classification tasks, they struggle to analyze the overall logical sequence of input data, such as complex temporal sequences with

intricate time dependencies or structural sequence data with variable length. Recurrent Neural Networks (RNN) were proposed specifically to solve sequence-structured data problems and are deep learning models capable of transmitting contextual information. The key difference from traditional neural networks is that RNN hidden layers have weighted connections across time, forming a cycle that allows information from the previous hidden layer (e.g., at time t-1) to be preserved and passed to the current hidden layer (at time t).

RNNs can encode tree or graph structural information into vectors and map them into a semantic vector space, where semantically similar vectors are closer together. The training algorithm for RNNs differs from conventional backpropagation, using Backpropagation Through Time (BPTT), where parameter errors and gradients are propagated backward through the time series to preceding layers. However, in practice, time length must be limited to simplify computational complexity; otherwise, training becomes unmanageable. Researchers have also proposed improved RNN language models (RNNLM) that can adapt to broader contexts, and trained word vectors can reflect word meanings for reasoning tasks (e.g., “woman” + “king” - “man” “queen”).

3.4 Long Short-Term Memory Network Algorithm

Training deep learning models with conventional RNNs often requires truncating time sequences, making it difficult to fully depend on time and reflect complete context. While setting an upper limit on RNN time connections can mitigate gradient explosion, the gradient vanishing problem remains challenging to solve. Long Short-Term Memory Networks (LSTM) were proposed to address RNN’ s long-term dependency issues, particularly gradient vanishing. LSTM is essentially a special type of RNN with time capabilities, primarily used to process time series with relatively long delays or intervals between event contexts.

Unlike conventional RNNs that have only one hidden state h for short-term information, LSTM adds a cell state c, also called a constant error carousel, to preserve long-term input data and gradients. LSTM’ s key innovation is its control mechanism over cell state c, which includes three control gates: an input gate, a forget gate, and an output gate. Each gate is a control switch that determines what information to retain or discard. During training, gate activation functions are typically set as sigmoid functions with value ranges of 0 to 1, while the output activation function is tanh with a value range of -1 to 1. LSTM has achieved good accuracy in many applications, including machine translation, image captioning, and speech recognition.

Since 2014, numerous deep learning language feature representation algorithms have been proposed, including GRU (which simplifies LSTM by combining input and forget gates into an update gate), BLSTM, CNN-RNN, BLSTM-CNN, DRNN, TextCNN, FastText, TextRNN, TextRCNN, DPCNN, MHAN, and AttnConvnet. Despite these variations, they all build upon the foundational concepts and techniques of classic algorithms, representing both a development

trend and a bottleneck that urgently requires new ideas and technological breakthroughs.

---

4 Application Development Tools for Deep Learning Language Models

The industry has provided relatively good tool environments supporting the development of basic deep learning language feature representation algorithms, enabling rapid implementation of common applications at relatively low cost. Most tools have encapsulated mainstream deep learning language models and provide convenient interfaces, allowing users to quickly build various deep learning models without deep algorithmic understanding. Users can focus on text feature analysis and feature engineering implementation without worrying about hardware environments, as most frameworks support convenient switching between different operating systems and CPUs or TPUs.

4.1 Word Embedding Model Generation: Word2Vec as Representative

Word2Vec, released by Google in 2013, is an open-source toolkit for generating word embeddings. Its main function is to train text and convert it into word embedding models, where each word is mapped to a specific vector. Word2Vec primarily includes two model architectures: CBOW and Skip-Gram. The CBOW model uses the context of a target word to predict and calculate its embedding, maximizing the conditional probability of obtaining the target word as output given its context. Essentially, CBOW removes the non-linear hidden layer from the original NNLM model, concentrates all input word vectors in a single embedding layer, and connects it directly to the output layer.

The Skip-Gram model works inversely, using a target word to predict and calculate its context words, minimizing the prediction error of context word embeddings at the output layer. Skip-Gram essentially computes the cosine similarity between input and target word embeddings, followed by normalization. Word2Vec also provides two learning optimization algorithms: hierarchical softmax (which uses a Huffman tree to decompose complex normalization into conditional probability products) and negative sampling (which randomly samples negative examples to establish binary logistic regression likelihood functions). Word2Vec supports multiple programming languages, including C++, Python, and Java, and can efficiently train on text datasets of millions of samples, making it the most widely used word embedding generation tool.

Extended tools based on Word2Vec have also emerged, including Doc2Vec and Sentence2Vec (which learn vector representations for sentences/paragraphs/documents), GloVe (which uses global word co-occurrence statistics rather than neural networks), Topic2Vec (which incorporates topics into the NNLM model), and

Lda2Vec (which combines Word2Vec and LDA). These tools enable measuring similarity through distance calculations and transforming topic relationships into vector-based clustering.

4.2 Pre-trained Language Model Open Source: BERT as Representative

Deep learning pre-trained language models can be trained on one dataset to create a baseline model that can then be directly called or fine-tuned for various preset functions on other datasets—a breakthrough called transfer learning. This approach effectively addresses the increasing costs of completely retraining models as networks deepen and datasets expand, while also helping researchers without time or resources to quickly master relevant technologies.

In 2018, Google released the heavyweight open-source framework BERT, which follows GPT' s basic architecture using Transformer encoders as the main structure. BERT is trained on pure text corpora comprising approximately 3.3 billion words from BooksCorpus and English Wikipedia. Unlike GPT models that only consider single-side context, BERT is the first unsupervised, bidirectional deep pre-training language model that can consider both sides of a word simultaneously and perform multi-task learning through masked language modeling and sentence-level continuity prediction tasks. BERT has achieved remarkable results across multiple NLP tasks.

Following BERT, numerous improved models have been released, including Transformer-XL (which helps machines understand context beyond fixed length limits), XLNet (which integrates Transformer-XL' s segment recurrence mechanism and relative encoding scheme to achieve bidirectional context learning through generalized permutation language modeling), RoBERTa (which optimizes BERT's training), and various Chinese BERT models such as BERT-wwm and RoBERTa-zh-Large. Open-source NLP libraries like Flair and StanfordNLP have integrated multiple pre-trained models (GloVe, BERT, etc.) for convenient invocation, providing one-stop NLP services including named entity recognition, entity relation extraction, and dependency parsing.

4.3 Large-Scale Language Model Applications: GPT as Representative

ChatGPT, launched by OpenAI in November 2022, is an intelligent chatbot program that differs significantly from previous simple chat programs. It can deeply understand and actively learn human language concepts, ideologies, and intentions, engaging in coherent interactive communication based on conversational context and human prompts to complete difficult tasks such as intelligent Q&A, text analysis, and data visualization. ChatGPT has become the fastest-growing application in history.

ChatGPT is essentially an AI-driven natural language processing program and large-scale general language model. Through multi-layer deep neural networks

and a predictable, scalable deep learning stack integrated with human feedback reinforcement learning and supervised fine-tuning mechanisms, the model can sensitively perceive and accurately understand subtle differences in language styles and contextual patterns, then recombine them according to application scenarios to generate more authentic and creative results. ChatGPT has evolved through multiple versions from GPT-1 to GPT-4, with GPT-4 released in March 2023, supporting multimodal functions including text, images, and video input/output processing.

GPT-4' s performance on various professional tests and academic benchmarks is comparable to humans. However, GPT is not open-source in China, and domestic alternatives still lag significantly in capability. Many are either not open-source or remain in trial or confidential R&D stages, making it difficult to embed GPT into LIS institutional knowledge management and service systems to provide convenient development support and popularized intelligent services.

---

5 Challenges and Application Capability Expansion

5.1 Key Challenges

Despite witnessing the powerful capabilities of generative deep learning language models like GPT in automatic data discovery and intelligent knowledge mining, the LIS field still exhibits a wait-and-see attitude with mismatched theoretical and practical capabilities and difficulties in development and application deployment. Deep learning language model development faces several important challenges:

Numerous Parameters: Deep learning language models contain vast numbers of parameters and hyperparameters, including hidden layer counts, neurons per layer, context window sizes, projection matrix dimensions, convolution kernel sizes, learning rates, dropout ratios, and optimization algorithm choices. For example, GPT-1 has approximately 117 million parameters, GPT-2 about 1.5 billion, GPT-3 about 175 billion, and GPT-4 has reached approximately 100 trillion parameters—comparable to the number of neurons in the human brain. Defining and constructing a deep neural network structure requires continuous adjustment and combination of these parameters, with only stable and reasonable parameter tuning ensuring effectiveness and accuracy.

Heavy Reliance on Training Data: Deep learning language models depend on large volumes of accurate training data. Deep and complex neural network structures require substantial data for sufficient weight optimization, with costs increasing as data volumes grow. Training or optimizing large language models like ChatGPT takes months, with GPT-4' s training costing between $2 million and $1.2 million. The massive computational power required may also cause carbon emissions and environmental concerns. Additionally, sample data

distributions often have differences and biases, requiring preprocessing and assumptions that may not generalize well to unknown data.

Intellectual Property and Security Issues: As deep learning language models become popular, concerns about intellectual property, privacy ethics, and environmental issues are growing. These models rely on mining massive text corpora, which may involve copying and using copyrighted works and imitating creative styles, triggering new copyright infringement risks. The highly realistic synthetic content and sensitive private information (medical data, financial data, identity information) generated by models may be used to impersonate or deceive others, causing privacy violations, fraud, and other illegal activities. Model responses may also contain political, gender, or racial biases, misleading information that violates ethical 常识, and other problematic outputs.

Implementation Variability: Since deep neural network algorithms contain many random operations and dropout functions, and different computers have different computational precisions, weight optimization and computed values may vary with implementation methods. Experimental accuracy may depend on specific open-source library implementations, and training the same algorithm with different libraries may yield different results.

5.2 Application Capability Expansion Methods

To expand the application capabilities of deep learning language models, we propose two strategic approaches:

Domain-Specific Expansion: This method starts from the needs of specific domains. Some disciplines are naturally suited for deep learning language models due to their rich professional data resources. Key strategies include: (1) Collecting and preparing large amounts of high-quality domain-specific data while avoiding bias or overfitting; (2) Selecting appropriate model architectures or adapting them to domain features; (3) Involving domain experts to provide professional knowledge, guide data collection, and validate results; (4) Optimizing for specific tasks through transfer learning, fine-tuning, or adjusting outputs.

Feature Engineering Expansion: Feature engineering is the decisive factor for improving prediction accuracy. While deep learning models typically learn features automatically through end-to-end learning, manual feature extraction can still improve performance for specific tasks. Strategies include: (1) Selecting appropriate features based on task requirements; (2) Preprocessing features (e.g., stemming, lemmatization) to reduce interference; (3) Using feature selection techniques (threshold filtering, statistical testing) to prevent overfitting; (4) Applying dimensionality reduction techniques (PCA, kernel functions) to reduce computational complexity while preserving important information.

As generative language models like ChatGPT gain adoption, society faces new demands for prompt engineers and AI trainers, with salaries reaching $175,000-$335,000 annually. This presents opportunities for librarians. However, the

primary challenge in professional/vertical domain applications is the lack of high-quality, compliant annotated training corpora—an area where LIS institutions can serve as developers and providers of high-standard domain-specific training datasets to maintain competitive advantages.

References

[1] ZHAO W X, ZHOU K, LI J Y, et al. A survey of large language models[J]. arXiv Preprint, arXiv: 2303.18223, 2023.
[2] QIU X P, SUN T X, XU Y G, et al. Pre-trained models for natural language processing: A survey[J]. Science China Technological Sciences, 2020, 63(10): 1872-1897.
[3] MAO J, CHEN Z Y. A deep learning based approach to structural function recognition of scientific literature abstracts[J]. Journal of Library and Information Science in Agriculture, 2022, 34(3): 15-27.
[4] KANG M. Deep learning pre-training language model—case: A study on emotion classification of Chinese financial texts[M]. Beijing: Tsinghua University Press, 2022.
[5] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7): 1527-1554.
[6] YUSUKE S. Java deep learning essentials[M]. Beijing: China Machine Press, 2017: 97-113.
[7] IENCO D, GAETANO R, INTERDONATO R, et al. Combining sentinel-1 and sentinel-2 time series via RNN for object-based land cover classification[C]// IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. Piscataway, New Jersey: IEEE, 2019: 4881-4884.
[8] JI S H, VISHWANATHAN S V N, SATISH N, et al. BlackOut: Speeding up recurrent neural network language models with very large vocabularies[J]. arXiv Preprint, arXiv: 1511.06909, 2015.
[9] RNNLM Toolkit[EB/OL]. [2023-02-20]. https://github.com/IntelLabs/rnnlm.
[10] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[J]. arXiv Preprint, arXiv: 1409.3215, 2014.
[11] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv Preprint, arXiv: 1406.1078, 2014.
[12] KIM Y. Convolutional neural networks for sentence classification[J]. arXiv Preprint, arXiv: 1408.5882, 2014.
[13] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[J]. arXiv Preprint, arXiv: 1607.01759, 2016.
[14] LIU P F, QIU X P, HUANG X J. Recurrent neural network for text classification with multi-task learning[J]. arXiv Preprint, arXiv: 1605.05101, 2016.
[15] LAI S W, XU L H, LIU K, et al. Recurrent convolutional neural networks for text classification[C]// Proceedings of the Twenty-Ninth AAAI Conference

on Artificial Intelligence. New York: ACM, 2015: 2267-2273.
[16] JOHNSON R, ZHANG T. Deep pyramid convolutional neural networks for text categorization[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2017: 562-570.
[17] PAPPAS N, POPESCU-BELIS A. Multilingual hierarchical attention networks for document classification[J]. arXiv Preprint, arXiv: 1707.00896, 2017.
[18] KIM Y, LEE H, JUNG K. Attention-based convolutional neural networks for multi-label emotion classification[EB/OL]. [2018-01-01]. http://sciencewise.info/articles/1804.00831/.
[19] TensorFlow[EB/OL]. [2023-02-25]. https://tensorflow.google.cn/.
[20] Deeplearning4j[EB/OL]. [2023-02-25]. https://github.com/deeplearning4j.
[21] PyTorch[EB/OL]. [2023-02-25]. https://pytorch.org/.
[22] Theano[EB/OL]. [2023-02-25]. https://pypi.org/project/Theano/.
[23] Keras[EB/OL]. [2023-02-25]. https://keras.io/.
[24] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv Preprint, arXiv: 1301.3781, 2014.
[25] LE Q V, MIKOLOV T. Distributed representations of sentences and documents[C]//ICML' 14 Proceedings of the 31st International Conference on Machine Learning. Beijing, China: ICML, 2014(32): 1188-1196.
[26] JEFFREY P, RICHARD S, CHRISTOPHER D M. GloVe: Global vectors for word representation[EB/OL]. [2018-12-29]. https://nlp.stanford.edu/projects/glove/.
[27] NIU L Q, DAI X Y, ZHANG J B, et al. Topic2Vec: Learning distributed representations of topics[C]// 2015 International Conference on Asian Language Processing (IALP). Piscataway, New Jersey: IEEE, 2016: 193-196.
[28] MOODY C E. Mixing dirichlet topic models and word embeddings to make lda2vec[J]. arXiv Preprint, arXiv: 1605.02019, 2016.
[29] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, New Jersey: IEEE, 2016: 770-778.
[30] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv Preprint, arXiv: 1706.03762, 2017.
[31] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations[J]. arXiv Preprint, arXiv: 1802.05365, 2018.
[32] REDDY R. Universal language model fine-tuning for text classification[J]. arXiv Preprint, arXiv: 1801.06146, 2018.
[33] GPT-2[EB/OL]. [2023-02-28]. https://github.com/openai/gpt-2.
[34] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv Preprint, arXiv: 1810.04805, 2019.
[35] DAI Z H, YANG Z L, YANG Y M, et al. Transformer-XL: Attentive language models beyond a fixed-length context[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019.
[36] YANG Z L, DAI Z H, YANG Y M, et al. XLNet: Generalized autoregressive

pretraining for language understanding[J]. arXiv Preprint, arXiv: 1906.08237, 2019.
[37] ZHONG H X, ZHANG Z Y, LIU Z Y, et al. Open Chinese language pre-trained model zoo[EB/OL]. [2020-03-18]. https://github.com/thunlp/OpenCLaP.
[38] CUI Y M, CHE W X, LIU T, et al. Pre-training with whole word masking for Chinese BERT[EB/OL]. [2023-03-09]. https://github.com/ymcui/Chinese-BERT-wwm.
[39] XU L. RoBERTa for Chinese[EB/OL]. [2022-06-15]. https://github.com/brightmart/roberta_{zh}.
[40] ALAN A, DUNCAN B, ROLAND V. Contextual string embeddings for sequence labeling[EB/OL]. [2023-03-10]. https://github.com/zalandoresearch/flair.
[41] Stanford NLP[EB/OL]. [2023-03-10]. https://github.com/stanfordnlp.
[42] ChatGPT: Optimizing language models for dialogue[EB/OL]. [2023-03-16]. https://openai.com/blog/chatgpt.
[43] NISAN S, LONG O, JEFFREY W, et al. Learning to summarize with human feedback[C]//Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020: 3008-3021.
[44] LEO G, JOHN S, JACOB H. Scaling laws for reward model overoptimization[J]. arXiv Preprint, arXiv: 2210.10760, 2022.
[45] GPT-4[EB/OL]. [2023-03-16]. https://openai.com/product/gpt-4.
[46] LIU G C, YANG R. How much computing power does ChatGPT require[R/OL]. Beijing: Guosen Securities, 2023.
[47] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15: 1929-1958.
[48] AI text classifier[EB/OL]. [2023-03-16]. https://platform.openai.com/ai-text-classifier.
[49] AIGC-X[EB/OL]. [2023-03-16]. http://ai.sklccc.com.
[50] VAN DIS E A M, BOLLEN J, ZUIDEMA W, et al. ChatGPT: Five priorities for research[J]. Nature, 2023, 614(7947): 224-226.
[51] Prompt engineer and librarian[EB/OL]. [2023-03-31]. https://jobs.lever.co/Anthropic/e3cde481-d446-460f-b576-93cab67bd1ed.
[52] ZHANG Z X, QIAN L, XIE J, et al. The impact of ChatGPT on scientific research and documentation and information work[R/OL]. Beijing: National Science and Technology Library & National Science Library of Chinese Academy of Sciences, 2023.
[53] ZHANG X L. From ape to man: Exploring the phoenix nirvana road of knowledge service[J]. Data Analysis and Knowledge Discovery, 2023, 7(3): 1-4.
[54] CAO S J, CAO R Y. Influence of generative AI on the research and practice of information science from the perspective of ChatGPT[J]. Journal of Modern Information, 2023, 43(4): 3-10.

Note: Figure translations are in progress. See original paper for figures.