DOI:10.1145/3490443

A language modeling overview, highlighting basic concepts, intuitive explanations, technical achievements, and fundamental challenges.

BY HANG LI

Language Models: Past, Present, and Future

NATURAL LANGUAGE PROCESSING (NLP) has undergone revolutionary changes in recent years. Thanks to the development and use of pre-trained language models, remarkable achievements have been made in many applications. Pre-trained language models offer two major advantages. One advantage is that they can significantly boost the accuracy of many NLP tasks. For example, one can exploit the BERT model to achieve performances higher than humans in language understanding.[8] One can also leverage the GPT-3 model to generate texts that resemble human writings in language generation.[3] A second advantage of pre-trained language models is that they are universal language processing tools. To conduct a machine learning-based task in traditional NLP, one had to label a large amount of data to train a model. In contrast, one currently needs only to label a small amount of data to fine-tune a pre-trained language model because it has already acquired a significant amount of knowledge necessary for language processing.

This article offers a brief introduction to language modeling, particularly pre-trained language modeling, from the perspectives of historical development and future trends for general readers in computer science. It is not a comprehensive survey but an overview, highlighting the basic concepts, intuitive explanations, technical achievements, and fundamental challenges. While positioned as an introduction, this article also helps knowledgeable readers to deepen their understanding and initiate brainstorming. References on pre-trained language models for beginners are also provided.

NLP is a subfield of computer science (CS), artificial intelligence (AI), and linguistics, with machine translation, reading comprehension, dialogue system, document summarization, text generation, and others, as applications. In recent years, deep learning has become the fundamental technology of NLP.

In our view, there are two main approaches to modeling human languages using mathematical means:

» key insights

- Language modeling research has more than 100 years of history, dating back to Markov.
- Neural language modeling represents a new era of language modeling, in which the model is parameterized by a neural network. Pre-trained languages, as a new type of neural language model, have advanced natural language processing (NLP) technologies to a higher level.
- Neural language models, particularly pre-trained language models, will still be the most powerful tools for NLP in the coming years, and the potential for future technology advancements is still substantial.
- The critical question is how to design neural networks to make the models closer to human language processing in representation ability and computing efficiency. We should seek inspiration from the human brain.

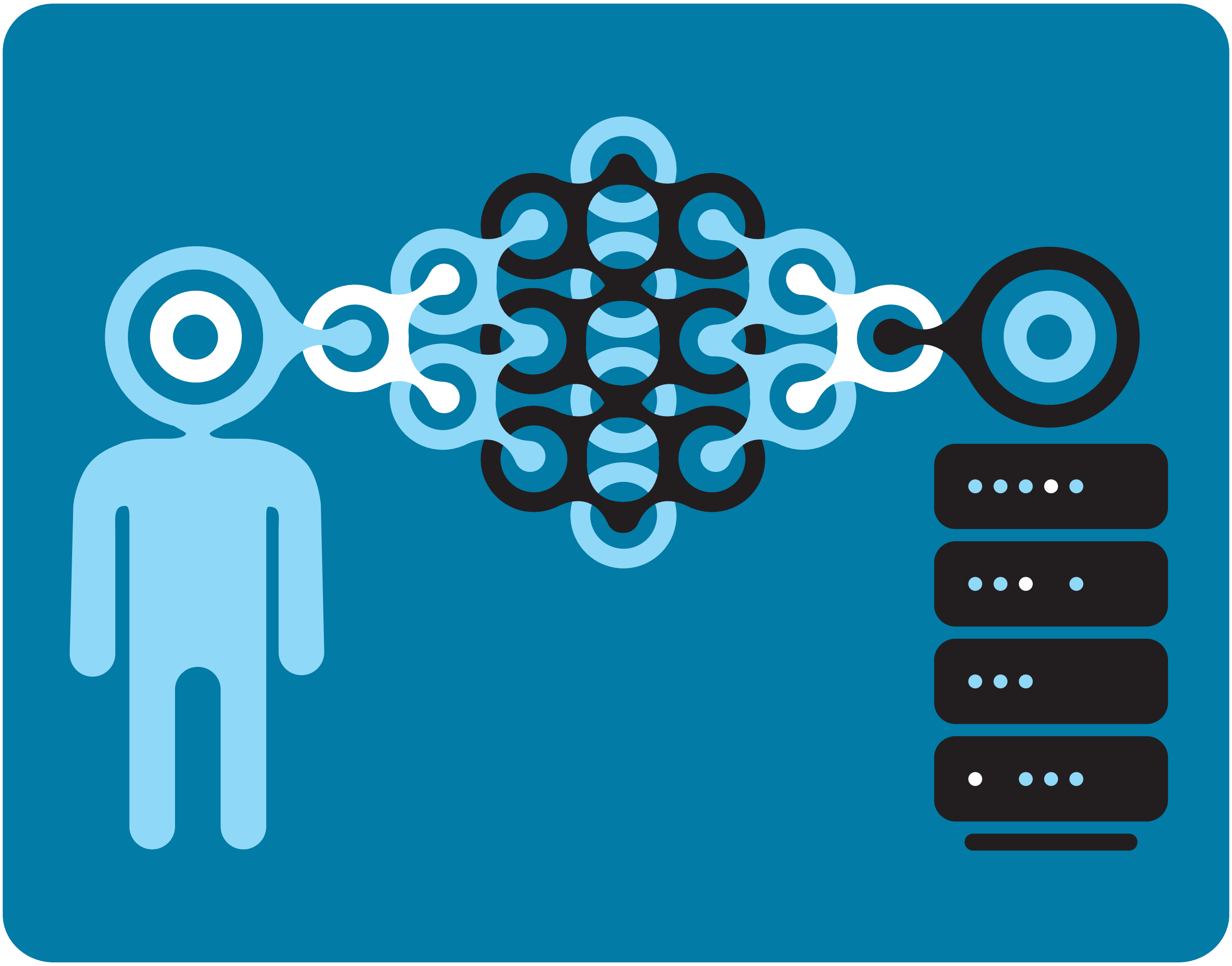


IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

one is based on probability theory and the other on formal language theory. These two approaches can also be combined. Language models fall into the first category from the viewpoint of a fundamental framework.

Formally, a language model is a probability distribution defined on a word sequence (a sentence or a paragraph). Language models amount to important machinery for modeling natural language texts based on probability theory, statistics, information theory, and machine learning. Neural language models empowered by deep learning, especially recently developed pre-trained language models, have become the fundamental technologies of NLP.

In this article, I first introduce the basic concepts of language modeling studied by Markov and Shannon (based on probability theory). Next, I discuss the linguistic models proposed by Chomsky (based on formal language theory), followed by a description of the definitions of neural language models as extensions of traditional language models. Then, I explain the basic ideas of pre-trained language models and follow with a discussion of the advantages and limitations of the neural language modeling approach and a prediction of future trends.

Markov and Language Models

Andrey Markov was perhaps the first scientist who studied language models,[10] although the term "language model" did not exist at the time.

Suppose that w_1, w_2, $\cdots$, w_N is a sequence of words. Then, the probability of the word sequence can be calculated as follows:

$$p(w_1,w_2,\cdots,w_N)=\prod_{i=1}^{N} p(w_i\,|\,w_1,w_2,\cdots,w_{i-1}) \qquad (1)$$

Let $p(w_1|w_0)=p(w_1)$. Different types of language models use different methods to calculate the conditional probabilities $p(w_i|w_1, w_2, \cdots, w_{i-1})$. The process of learning and using a language model is referred to as language modeling. An n-gram model is a basic model that assumes that the word at each position only depends on the words at the $n-1$ previous positions. That is, the model is an $n-1$-order Markov chain.

$$p(w_1,w_2,\cdots,w_N)\approx\prod_{i=1}^{N} p(w_i\,|\,w_{i-n+1},w_{i-n+2},\cdots,w_{i-1})$$

Markov studied the Markov chain in 1906. The model which he first considered was quite simple, with only two states and transition probabilities between those states. Markov proved that if one jumps between the two states according to the transition probabilities, then the frequencies of accessing the two states will converge to the expected values, which is the Ergodic theorem of the Markov chain. In the following years, he expanded the model and proved that the above conclusion still holds in more general settings.

To provide a concrete example, Markov applied his proposed model to Alexander Pushkin's novel in verse, *Eugene Onegin*, in 1913. Removing spaces and punctuation marks and classifying the novel's first 20,000 Russian letters into vowels and consonants, he obtained a sequence of vowels and consonants in the novel. Using paper and pen, Markov then counted the transition probabilities between the vowels and consonants. Then, the data was used to verify the characteristics of the simplest Markov chain.

It is very interesting the initial application area of the Markov chain is language. The example Markov studied is the simplest language model.

It is very interesting the initial application area of the Markov chain is language. The example Markov studied is the simplest language mode.

Shannon and Language Models

In 1948, Claude Shannon published his groundbreaking paper, "The Mathematical Theory of Communication," which pioneered the field of information theory. In the paper, Shannon introduced the notions of entropy and cross-entropy and studied the properties of the n-gram model.[30] (Shannon borrowed the term "entropy" from statistical mechanics based on advice from John von Neumann.)

Entropy represents the uncertainty of one probability distribution, while cross-entropy represents the uncertainty of one probability distribution with respect to the other probability distribution. Entropy is a lower bound of cross-entropy.

Suppose that language (word sequence) is data generated by a stochastic process. The entropy of probability distribution of n-grams is defined as follows:

$$H_n(p) = -\sum_{w_1, w_2, \cdots, w_n} p(w_1, w_2, \cdots, w_n) \cdot \log_2 p(w_1, w_2, \cdots, w_n)$$

where $p(w_1, w_2, \cdots, w_n)$ represents the probability of n-gram $w_1, w_2, \cdots, w_n$. The cross-entropy of probability distribution of n-grams with respect to the "true" probability distribution of data is defined as follows:

$$H_n(p,q) = -\sum_{w_1, w_2, \cdots, w_n} p(w_1, w_2, \cdots, w_n) \cdot \log_2 q(w_1, w_2, \cdots, w_n)$$

where $q(w_1, w_2, \cdots, w_n)$ represents the probability of n-grams $w_1, w_2, \cdots, w_n$ and $p(w_1, w_2, \cdots, w_n)$ represents the true probability of n-gram $w_1, w_2, \cdots, w_n$.

The following relation holds:

$$H_n(p) \leq H_n(p, q)$$

The Shannon-McMillan-Breiman theorem states that when the stochastic process of language satisfies the conditions of stationarity and ergodicity, the following relations hold the following:

$$H(p) = \lim_{n\to\infty} \frac{1}{n} H_n(p) = \lim_{n\to\infty} -\frac{1}{n} \log_2 p(w_1, w_2, \cdots, w_n)$$

$$H(p,q) = \lim_{n\to\infty} \frac{1}{n} H_n(p,q) = \lim_{n\to\infty} -\frac{1}{n} \log_2 q(w_1, w_2, \cdots, w_n)$$

$$H(p) \leq H(p, q)$$

In other words, when the word sequence length goes to infinity, the entropy of the language can be defined. The entropy takes a constant value and can be estimated from the data of the language.

If one language model can more accurately predict a word sequence than the other, then it should have lower cross-entropy. Thus, Shannon's work provides an evaluation tool for language modeling.

Note that language models can model not only natural languages but also formal and semi-formal languages—for example, Peng and Roth.[21]

Chomsky and Language Models

In parallel, Noam Chomsky proposed the Chomsky hierarchy of grammars in 1956, for representing the syntax of a language. He pointed out that finite-state grammars (also n-gram models) have limitations in describing natural languages.[4]

Chomsky's theory asserts that a language consists of a finite or infinite set of sentences, each sentence is a sequence of words of finite length, words come from a finite vocabulary, and a grammar is a set of production rules that can generate all sentences in the language. Different grammars can produce languages in different complexities, and they form a hierarchical structure.

A grammar that can generate sentences acceptable by a finite-state machine is a finite-state grammar or regular grammar, while a grammar that can produce sentences acceptable by a non-deterministic pushdown automaton is a context-free grammar. Finite-state grammars are properly included in context-free grammars.

The "grammar" underlying a finite

Markov chain (or an n-gram model) is a finite-state grammar. A finite-state grammar does have limitations in generating sentences in English. For example, there are grammatical relations between English expressions, such as the following relations in (i) and (ii).

(i) If S1, then S2.
(ii) Either S3, or S4.
(iii) Either if S5, then S6, or if S7, then S8

In principle, the relations can be combined indefinitely to produce correct English expressions (such as in example iii). However, a finite-state grammar cannot describe all the combinations, and, in theory, there are English sentences that cannot be covered. Therefore, Chomsky contended that there are great limitations in describing languages with finite-state grammars, including n-gram models. Instead, he pointed out that context-free grammar can model languages more effectively. Influenced by him, in the following decades, context-free grammars were more commonly used in NLP. (Chomsky's theory is not very influential to NLP now, but it still has important scientific values.)

Neural Language Models

In 2001, Yoshua Bengio and his coauthors proposed one of the first neural language models,[1] which opened a new era of language modeling. (Bengio, Geoffrey Hinton, and Yann LeCun received the 2018 ACM A.M. Turing Award for their conceptual and engineering breakthroughs that have made deep neural networks a critical part of computing, as is well known.)

The n-gram model is limited in its learning ability. The traditional approach is to estimate from the corpus the conditional probabilities $p(w_i|w_{i-n+1}, w_{i-n+2}, \cdots, w_{i-1})$ in the model with a smoothing method. However, the number of parameters in the model is of exponential order $O(V^n)$, where V denotes vocabulary size. When n increases, the parameters of the model cannot be accurately learned, due to the sparsity of training data.

The neural language model proposed by Bengio et al. improves the n-gram model in two ways. First, a real-valued vector, called word embedding, is used to represent a word or a combination of words. (The embedding of a word has much lower dimensionality than the "one-hot vector" of a word, in which the element corresponding to the word is one and the other elements are zero.)

Word embedding, as a type of "distributed representation," can represent a word with better efficiency, generalization ability, robustness, and extensibility than one-hot vector. Second, the language model is represented by a neural network, which greatly reduces the number of parameters in the model. The conditional probability is determined by a neural network:

$$p(w_i|w_{i-n+1}, w_{i-n+2}, \cdots, w_{i-1}) \\ = f_\theta(\boldsymbol{w}_{i-n+1}, \boldsymbol{w}_{i-n+2}, \cdots, \boldsymbol{w}_{i-1})$$

where $(\boldsymbol{w}_{i-n+1}, \boldsymbol{w}_{i-n+2}, \cdots, \boldsymbol{w}_{i-1})$ denote the embeddings of words $w_{i-n+1}, w_{i-n+2}, \cdots, w_{i-1}$; $f(\cdot)$ denotes the neural network; and θ denotes the network parameters. The number of parameters in the model is only of order $O(V)$. Figure 1 shows the relationship between representations in the model. Each position has an intermediate representation that depends on the word embeddings (words) at the previous $n-1$ positions, and this holds for all positions. The intermediate representation at the current position is then used to generate a word for the position.

After the work of Bengio et al., a large number of word-embedding methods and neural language-modeling methods have been developed, bringing improvements from different perspectives.

Representative methods for word embedding include Word2Vec.[18,19] Representative neural language models are recurrent neural network (RNN) language models, including the long short-term memory (LSTM) language models.[9,11] In an RNN language model, the conditional probability at each position is determined by an RNN:

$$p(w_i|w_1, w_2, \cdots, w_{i-1}) = f_\theta(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{i-1})$$

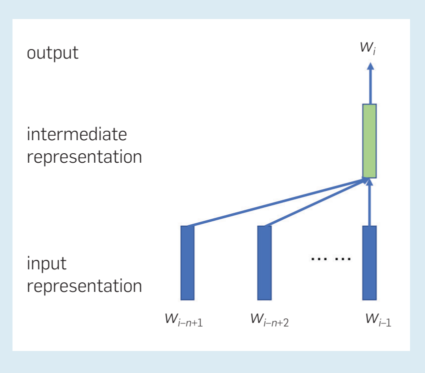
where $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{i-1}$ denote the embeddings of words $w_1, w_2, \cdots, w_{i-1}$; $f(\cdot)$ denotes the RNN; and θ denotes the network parameters. The RNN language model no longer has the Markovian assumption, and the word at each position depends on the words at all previous positions. An important concept in RNN is its intermediate representations or states. The dependencies between words are characterized by the dependencies between states in the RNN model. The model's parameters are shared in different positions, but the obtained representations are different at different positions. (For ease of understanding, we do not give the formal definitions or present the architectures of neural networks in this article.)

Figure 2 shows the relationship between representations in an RNN language model. There is an intermediate representation of each layer at each position that represents the "state" of the word sequence so far. The intermediate representation of the current layer at the current position is determined by the intermediate representation of the same layer at the previous position and the intermediate representation of the layer below at the current position. The final intermediate representation at the current position is used to calculate the probability of the next word.

Language models can be used to calculate the probability of language (word sequence) or to generate language. In the latter case, natural language sentences or articles are generated, for example, by random sampling from language models. It is known that LSTM language models that learn from a large amount of data can generate quite natural sentences.

An extension of a language model is a conditional language model, which calculates the conditional probability of a word sequence under a given condition. If the condition is another word sequence, then the problem becomes transformation from one word sequence to another—that is, the so-called sequence-to-sequence problem. Machine translation,[5,33] text summarization,[20] and

Figure 1. The relationship between representations in the original neural language model.



generative dialogue[31] are such tasks. If the given condition is a picture, then the problem becomes transformation from a picture to a word sequence. Image captioning[35] is such a task.

Conditional language models can be employed in a large variety of applications. In machine translation, the system transforms sentences in one language into sentences in another language, with the same semantics. In dialogue generation, the system generates a response to the user's utterance, and the two messages form one round of dialogue. In text summarization, the system transforms a long text into a short text, making the latter represent the gist of the former. The semantics represented by the conditional probability distributions of the models vary from application to application and are learned from the data in the applications.

The study of sequence-to-sequence models has contributed to the development of new technologies. A representative sequence-to-sequence model is a transformer developed by Vaswani et al.[34] The transformer is entirely based on the attention mechanism[5] and exploits attention to conduct encoding, decoding, and information exchange between encoder and decoder. At present, almost all machine translation systems employ the transformer model, and machine translation has reached the level that can almost meet the needs in practice. The architecture of the transformer is now adopted in almost all pre-trained language models because of its superior power in language representation.

Figure 2. The relationship between representations in an RNN language model. Here, <bos> denotes the beginning of a sentence and <eos> denotes the end of a sentence.

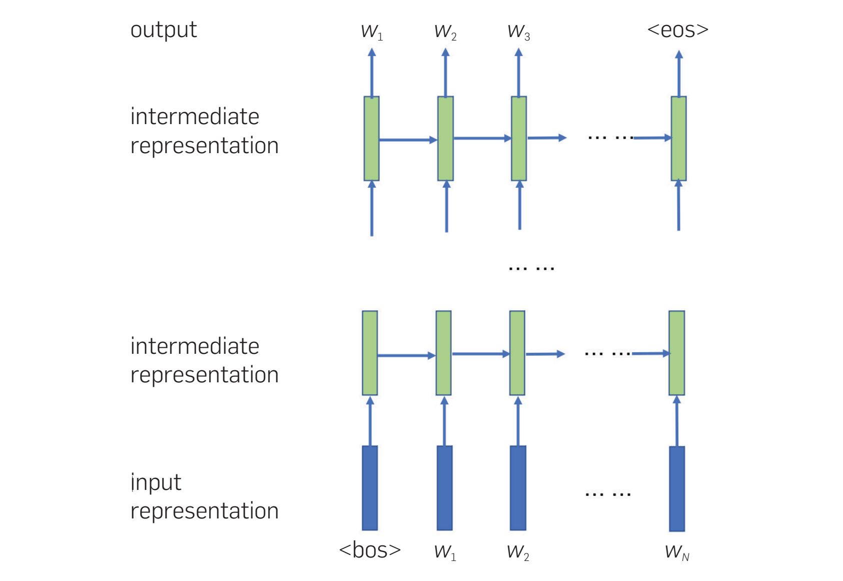


Table 1. Resources for learning and using pre-trained language models.

Software of pre-trained language models	**Huggingface Transformer**: A large number of pre-trained models are available. https://github.com/huggingface/transformers
Tutorial on fine-tuning of pre-trained models	**Huggingface Tutorial**: https://huggingface.co/transformers/training.html
Benchmark	GLUE, a benchmark for language understanding tasks in English. https://gluebenchmark.com/ CLUE, a collection of datasets for language understanding tasks in Chinese. https://www.cluebenchmarks.com/

Table 2. Summary of existing pre-trained language models.

	Unidirectional language model	**Bidirectional language model**	**Sequence-to-sequence model**
Architecture	Transformer decoder	Transformer encoder	Transformer
Pre-training	Language modeling (2)	Mask language modeling (3)	Sequence-to-sequence learning
Tasks	Language generation	Language understanding	Sequence-to-sequence
Models	GPTs[3,25,26]	BERT,[8] RoBERTa,[17] ALBERT,[14] XLNet,[36] Electra[7]	BART,[15] T5[24]

Pre-Trained Language Models

The basic idea of a pre-trained language model is as follows. First, one implements the language model based on, for example, the transformer's encoder or decoder. The model learns in two phases: pre-training, where one trains the parameters of the model using a very large corpus via unsupervised learning (also called self-supervised learning), and fine-tuning, where one applies the pre-trained model to a specific task and further adjusts the model's parameters using a small amount of labeled data via supervised learning.[3,7,8,14,16,24–26,36] The links in Table 1 offer resources for learning and using pre-trained language models.

There are three types of pre-trained language models: unidirectional, bidirectional, and sequence-to-sequence. Due to space limitations, this paper covers only the first two types. All the major pre-trained language models adopt the transformer's architecture. Table 2 offers a summary of existing pre-trained language models.

A transformer has strong language representation ability; a very large corpus contains rich language expressions (such unlabeled data can be easily obtained) and training large-scale deep learning models has become more efficient. Therefore, pre-trained language models can effectively represent a language's lexical, syntactic, and semantic features. Pre-trained language models, such as BERT and GPTs (GPT-1, GPT-2, and GPT-3), have become the core technologies of current NLP.

Pre-trained language model applications have brought great success to NLP. "Fine-tuned" BERT has outperformed humans in terms of accuracy in language-understanding tasks, such as reading comprehension.[8,17] "Fine-tuned" GPT-3 has also reached an astonishing level of fluency in text-generation tasks.[3] (Note that the results solely indicate machines' higher performance in those tasks; one should not simply interpret that BERT and GPT-3 can understand languages better than humans, because this also depends on how benchmarking is conducted.[6] Having the proper understanding and expectation of the capabilities of AI technologies is critical to the healthy growth and development of the area, as is learned from history.)

GPTs developed by Radford et al.[25,26] and Brown et al.[3] have the following architecture. The input is a sequence of words $w_1, w_2, \cdots, w_N$. First, through the input layer, a sequence of input representations is created, denoted as a matrix $\boldsymbol{H}^{(0)}$. After passing L transformer decoder layers, a sequence of intermediate representations is created, denoted as a matrix $\boldsymbol{H}^{(L)}$

$$\boldsymbol{H}^{(L)} = \text{transformer\_decoder}(\boldsymbol{H}^{(0)})$$

Finally, a probability distribution of words is calculated at each position based on the final intermediate representation at the position. The pre-training of GPTs is the same as conventional language modeling. The objective is to predict the likelihood of a word sequence. For a given word sequence $\boldsymbol{w} = w_1, w_2, \cdots, w_N$, we calculate and minimize the cross-entropy or the negative log-likelihood to estimate the parameters:

$$-\log p(\boldsymbol{w}) = -\sum_{i=1}^{N} \log p_{\theta}\left(w_i \middle| w_1, \cdots, w_{i-1}\right) \tag{2}$$

where $\boldsymbol{\theta}$ denotes the parameters of the GPTs model.

Figure 3 shows the relationship between the representations in the GPTs model. The input representation at each position is composed of the word embedding and the "position embedding." The intermediate representation of each layer at each position is created from the intermediate representations of the layer below at the previous positions. The prediction or generation of a word is performed at each position repeatedly from left to right—Cf. (1) and (2). In other words, GPTs are a unidirectional language model in which the word sequence is modeled from one direction. (Note that an RNN language model is also a unidirectional language model.) Therefore, GPTs are better suited to solving language-generation problems that automatically produce sentences.

BERT, developed by Devlin et al.,[8] has the following architecture. The input is a sequence of words, which can be consecutive sentences from a single document or a concatenation of consecutive sentences from two documents. This makes the model applicable to tasks with one text as input (such as text classification), as well as to tasks with two texts as input (such as answering questions). First, through the input layer, a sequence of input representations is created, denoted as a matrix $\boldsymbol{H}^{(0)}$. After passing L transformer encoder layers, a sequence of intermediate representations is created, denoted as $\boldsymbol{H}^{(L)}$

$$\boldsymbol{H}^{(L)} = \text{transformer\_encoder}(\boldsymbol{H}^{(0)})$$

Finally, a probability distribution of words can be calculated at each position based on the final intermediate representation at the position. Pre-training of BERT is performed as the so-called mask language modeling. Suppose that the word sequence is $\boldsymbol{w} = w_1, w_2, \cdots, w_N$. Several words in the sequence are randomly masked—that is, changed to a special symbol [mask]—yielding a new sequence of words $\tilde{\boldsymbol{w}}$, where the set of masked words is denoted as $\bar{\boldsymbol{w}}$. The objective of learning is to recover the masked words by calculating and minimizing the following negative log-likelihood to estimate the parameters:

$$-\log p(\bar{\boldsymbol{w}} | \tilde{\boldsymbol{w}}) \approx -\sum_{i=1}^{N} \delta_i \log p_{\theta}\left(w_i \middle| \tilde{\boldsymbol{w}}\right) \tag{3}$$

where $\boldsymbol{\theta}$ denotes the parameters of the BERT model and δ_i takes a value of 1 or 0, indicating whether the word at position i is masked or not masked. Note that mask-language modeling is already a technique that differs from traditional language modeling.

Figure 4 shows the relationship between the representations in the BERT model. The input representation at each position is composed of word embeddings, "position embeddings,"

Figure 3. The relationship between representations in the GPTs model. Here, <bos> denotes the beginning of a sentence and <eos> denotes the end of a sentence.

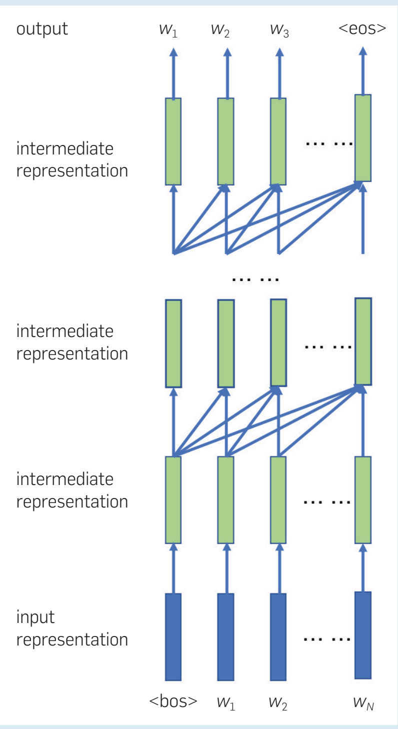

output
w_1
w_2
w_3
<eos>
intermediate representation
… …
… …
intermediate representation
… …
intermediate representation
… …
input representation
… …
<bos>
w_1
w_2
w_N

Figure 4. The relationship between representations in the BERT model. Here, <cls> denotes a special symbol representing the entire input sequence.

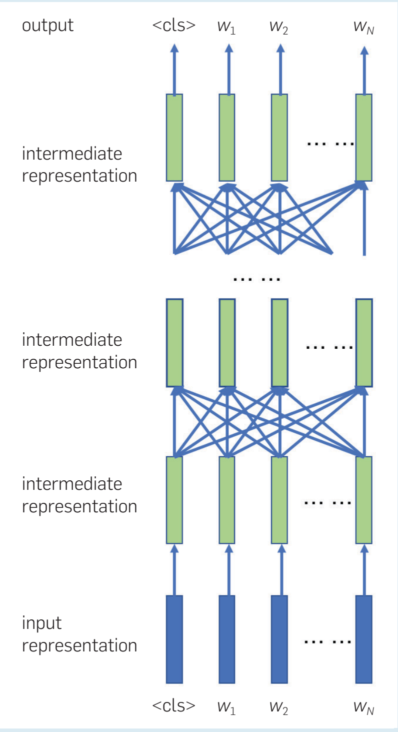
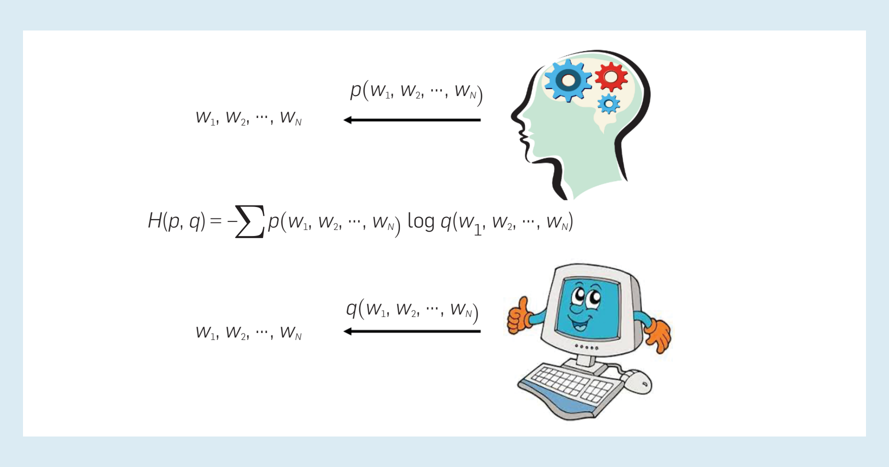

output
<cls>
w_1
w_2
w_N
intermediate representation
… …
… …
intermediate representation
… …
intermediate representation
… …
input representation
… …
<cls>
w_1
w_2
w_N

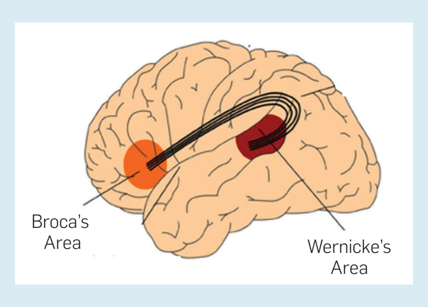
Figure 5. The machine manages to mimic human language behavior by tuning the parameters of the neural network inside its "brain." Eventually, it can process language like a human.


$p(w_1, w_2, \cdots, w_N)$
$w_1, w_2, \cdots, w_N$
$H(p, q) = -\sum p(w_1, w_2, \cdots, w_N) \log q(w_1, w_2, \cdots, w_N)$
$q(w_1, w_2, \cdots, w_N)$
$w_1, w_2, \cdots, w_N$

etc. The intermediate representation of each layer at each position is created from the intermediate representations of the layer below at all positions. The prediction or generation of a word is independently performed at each masked position—Cf. (3). That is to say, BERT is a bidirectional language model in which the word sequence is modeled from two directions. Therefore, BERT can be naturally employed in language understanding problems whose input is a whole word sequence and whose output is usually a label or a label sequence.

An intuitive explanation of pre-training language models is that the machine has performed a lot of *word solitaire* (GPTs) or *word cloze* (BERT) exercises based on a large corpus in pre-training, capturing various patterns of composing sentences from words, then composing articles from sentences, and expressing and memorizing the patterns in the model. A text is not randomly created with words and sentences, but constructed based on lexical, syntactic, and semantic rules. GPTs and BERT can use a transformer's decoder and encoder, respectively, to realize the compositionality of language. (Compositionality is the most fundamental feature of language, which is also modeled by grammars in the Chomsky hierarchy.) In other words, GPTs and BERT have acquired a considerable amount of lexical, syntactic, and semantic knowledge in pre-training. Consequently, when adapted to a specific task in fine-tuning, the models can be refined with only a small amount of labeled data to achieve high performance. It is found, for example, that different layers of BERT have different characteristics. The bottom layers mainly represent lexical knowledge, the middle layers mainly represent syntactic knowledge, and the top layers mainly represent semantic knowledge.[13,16,29]

Figure 6. Areas in the human brain responsible for language processing.



Pre-trained language models (without fine-tuning), such as BERT and GPT-3, contain a large amount of factual knowledge. For example, they can be used to answer questions such as, "Where was Dante born?" and conduct simple reasoning such as, "What is 48 plus 76?," as long as they have acquired the knowledge from the training data.[3,22] However, the language models themselves do not have a reasoning mechanism. Their "reasoning" ability is based on association instead of genuine logical reasoning. As a result, they fail to show high performance on problems that need complex reasoning, including argument reasoning,[38] numerical and temporal reasoning,[37] and discourse reasoning.[32] Integrating reasoning ability and language ability into an NLP system will be an important topic in the future.

Future Outlook

Contemporary sciences (brain science and cognitive science) have limited understanding of the mechanism of human language processing (language understanding and language generation). It is difficult to see a major breakthrough happening in the foreseeable future, and the possibility of never breaking through exists. On the other hand, we hope to continuously promote the development of AI technologies and develop machines of language processing that are useful for human beings.

It seems that neural language modeling is by far the most successful approach. The essential characteristic of language modeling has not changed—that is, it relies on the probability distribution defined in a discrete space containing all word sequences. The learning process is to find the optimal model so that the accuracy of predicting language data in terms of cross-entropy is the highest (see Figure 5). Neural language modeling constructs models through neural networks. The advantage is that it can very accurately simulate human language behaviors by leveraging complex models, big data, and powerful computing. From the original model proposed by Bengio et al. to RNN language models and pre-trained language models such as GPTs and BERT, the architectures of neural networks have become increasingly complex (Cf., Figures 1–4), while the ability to predict languages has become higher and higher (cross-entropy gets smaller and smaller). However, this does not necessarily mean that the models have the same language ability as humans, and the limitations of the approach are also self-evident.

Are there other possible development paths? It is not yet clear. It can be predicted that there are still many opportunities for improvement with the approach of neural language modeling. There is still a big gap between the current neural language models and human brains in representation ability and computing efficiency (in terms of power consumption). An adult human brain operates on only 12 W;[12] in striking contrast, training the GPT-3 model has consumed several thousand Petaflop/s-day, according to the authors.[3] Whether a better language model can be developed to be closer to human language processing is an important direction for future research. There are still many opportunities for technology enhancement. We can still learn from the limited discoveries in brain science.

Human language processing is believed to be carried out mainly at two brain regions in the cerebral cortex: Broca's area and Wernicke's area (Figure 6). The former is responsible for grammar, and the latter is responsible for vocabulary.[23] There are two typical cases of aphasia due to brain injuries. Patients who suffer from injuries in Broca's area can only speak in sporadic words instead of sentences, while patients who suffer from injuries in Wernicke's area can construct grammatically correct sentences, but the words often lack meaning. A natural hypothesis is that human language processing is carried out in both brain regions in parallel. Whether it is necessary to adopt a more human-like processing mechanism is a topic worth studying. Language models do not explicitly use grammars and cannot infinitely compose languages, which is an important property of human language, as pointed out by Chomsky. The ability to incorporate grammars more directly into language models will be a problem that needs to be investigated.

Brain scientists believe that human language understanding is a process of activating representations of relevant concepts in the sub-conscious and generating relevant images in the conscious. The representations include visual, auditory, tactile, olfactory, and gustatory representations. They are the visual, auditory, tactile, olfactory, and gustatory contents of concepts remembered in various parts of the brain through one's experiences during growth and development. Therefore, language understanding is closely related to the experiences of people.[2] Basic concepts in life, such as cat and dog, are learned from the input of sensors through seeing, hearing, touching, and so forth. Hearing or seeing the words "cat" and "dog" also reactivates the relevant visual, auditory, and tactile representations in people's brains. Can machines learn better models from a large amount of multimodal data (language, vision, speech) so that they can more intelligently process language, vision, and speech? Multimodal language models will be an important topic for future exploration. Most recently, there has been some progress in research on the topic—for example, Ramesh et al.[28] or Radford et al.[27]

Conclusion

Language models have a history that dates back more than 100 years. Markov, Shannon, and others could not have foreseen that the models and theories they studied would have such a great impact later; it might even be unexpected for Bengio. How will the language models develop over the next 100 years? Will they still be an essential part of AI technologies? This is beyond our imagination and prediction. What we can see is that language modeling technologies are continuously evolving. It is highly likely that more powerful models will replace BERT and GPTs in the years to come. For us, we are lucky enough to be the first generation to see the great achievements of the technologies and to participate in the research and development.

Acknowledgments

The author thanks the two anonymous reviewers for their critical and constructive reviews. He also thanks Yan Zeng and Quan Wang for their useful references and crucial suggestions, and Jun Xu, Xinsong Zhang, and Xiangyu Li for their helpful comments.

References

1. Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. In *Advances in Neural Information Processing Systems* (2001), 932–938.
2. Bergen, B. *Louder Than Words: The New Science of How the Mind Makes Meaning*. Basic Books, New York, NY, (2012).
3. Brown, T.B. et al. Language models are few-shot learners. arXiv:2005.14165 (2020).
4. Chomsky, N. Three models for the description of language. *IEEE Transactions on Information Theory 2*, 3 (1956), 113–124.
5. Cho, K. et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *The Conf. on Empirical Methods in Natural Language Processing* (2014), 1724–1734.
6. Church, K., Liberman, M., and Kordoni, V. *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*. Association for Computational Linguistics (2021).
7. Clark, K., Luong, M.T., Le, Q.V., and Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. arXiv:2003.10555 (2020).
8. Devlin, J., Chang, M.W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language*.
9. Elman, J.L. Finding structure in time. *Cognitive Science 14*, 2 (1990), 179–211.
10. Hayes, B. First links in the Markov chain. *American Scientist 101*, 2 (2013), 92.
11. Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.
12. Jabr, F. Does thinking really hard burn more calories. *Scientific American* (July 2012), 18.
13. Jawahar, G., Sagot, B., and Seddah, D. What does BERT learn about the structure of language?. In *57th Annual Meeting of the Association for Computational Linguistics* (July 2019).
14. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. arXiv:1909.11942 (2019).
15. Lewis, M. et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461 (2019).
16. Liu, N.F., Gardner, M., Belinkov, Y., Peters, M.E., and Smith, N.A. Linguistic knowledge and transferability of contextual representations. arXiv:1903.08855 (2019).
17. Liu, Y. et al. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 (2019).
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (2013), 3111–3119.
19. Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013).
20. Nallapati, R., Zhou, B., Gulcehre, C., and Xiang, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv:1602.06023 (2016).
21. Peng, H. and Roth, D. Two discourse driven language models for semantics. arXiv:1606.05679 (2016).
22. Petroni, F. et al. Language models as knowledge bases? arXiv:1909.01066 (2019).
23. Pinker, S. *The Language Instinct*, William Morrow and Company, New York, NY (1994), Chapter 9.
24. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683 (2019).
25. Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training (2018).
26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *Open AI Blog 1*, 8 (2019).
27. Radford, A. et al. Learning transferable visual models from natural language supervision. arXiv:2103.00020 (2021).
28. Ramesh, A. et al. Zero-shot text-to-image generation. arXiv:2102.12092 (2021).
29. Rogers, A., Kovaleva, O., and Rumshisky, A. A primer in Bertology: What we know about how BERT works. *Transactions of the Association for Compututational Linguistics 8* (2020), 842–866.
30. Shannon, C. A mathematical theory of communication. *The Bell System Technical J. 27* (July 1948), 379–423.
31. Shang, L., Lu, Z., and Li, H. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Assoc. for Computational Linguistics and the 7th International Joint Conf. on Natural Language Processing* (2015), 1577–1586.
32. Shen, A., Mistica, M., Salehi, B., Li, H., Baldwin, T., and Qi, J. Evaluating document coherence modelling. arXiv:2103.10133 (2021).
33. Sutskever, I., Vinyals, O., and Le, Q.V. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems 2014*, 3104–3112.
34. Vaswani, A. et al. Attention is all you need. *Advances in Neural Information Processing Systems 2017*, 5998–6008.
35. Xu, K. et al. Show, attend and tell: Neural image caption generation with visual attention. In *Intern. Conf. on Machine Learning, PMLR* (June 2015), 2048–2057.
36. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems* 2019, 5754–5764.
37. Zhang, X., Ramachandran, D., Tenney, I., Elazar, Y., and Roth, D. Do language embeddings capture scales? arXiv:2010.05345 (2020).
38. Zhou, X., Zhang, Y., Cui, L., and Huang, D. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI Conf. on Artificial Intelligence 34*, 05 (April 2020), 9733–9740.

Hang Li (lihang.lh@bytedance.com) is director of the AI Lab, Bytedance, Beijing, China.