深度学习在自然语言处理中的发展和应用

万贤林 [1)]

[1)](华中科技大学计算机科学与技术学院　武汉　430074

摘　要　自然语言处理（Natural Language Processing, NLP）是人工智能的一个重要分支，NLP 的目的在于设计和实现一个算法使机器能够像人一样理解和处理自然语言。近年来，随着深度学习在 NLP 中的发展和应用，人们不断突破各种常见 NLP 任务的最好成绩。在智能问答、谎言侦测等任务上，机器的表现甚至超过人类。早期的自然语言处理领域长期使用 Word2Vec 等词向量方法对文本进行编码, 这些词向量方法也可看作静态的预训练技术。然而, 这种上下文无关的文本表示给其后的自然语言处理任务带来的提升非常有限, 并且无法解决一词多义问题。ELMo 提出了一种上下文相关的文本表示方法, 可有效处理多义词问题。其后,GPT 和 BERT 等预训练语言模型相继被提出, 其中 BERT 模型在多个典型下游任务上有了显著的效果提升, 极大地推动了自然语言处理领域的技术发展, 自此便进入了动态预训练技术的时代。此后, 基于 BERT 的改进模型、XLNet 等大量预训练语言模型不断涌现, 预训练技术已成为自然语言处理领域不可或缺的主流技术。介绍自然语言处理的相关背景，综述了当前研究人员在自然语言处理热点领域上所使用的最新深度学习方法及所取得成果，分析并总结了深度学习方法在当前自然语言处理研究应用中所遇到的瓶颈，最后对未来可能的研究重点做出展望。

关键词　深度学习；自然语言处理；预训练模型; 词向量；语言模型

The development and application of deep learning in natural language processing

WAN Xian-lin[1)]

[1)](Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract　Natural Language Processing is an important branch of artificial intelligence. The purpose of NLP is to design and implement an algorithm that enables machines to understand and process natural language like humans. In the early days of natural language processing ,the word embedding methods such as Word2Vec were used to encode text. These word embedding methods can also be regarded as static pre-training techniques. How-ever , the context-independent text representation has limitation and cannot solve the polysemy problem. The ELMo pre-training language model gives a context-dependent method that can effectively handle polysemy problems. Later, GPT ,BERT and other pretraining language models have been proposed, especially the BERT model . which significantly improves the effect on many typical downstream tasks,greatly promotes the technical development in the field of natural language processing , and thus initiates the age of dynamic pre-training. Since then,a number of pre-training language models such as BERT-based improved models and XLNet have emerged , and pre-training techniques have become an indispensable mainstream technology in the field of natural language processing. In this paper, we introduce the relevant background of natural language processing, summarize the latest deep learning methods and results obtained by current researchers in the hot areas of natural language processing, analyze the bottlenecks encountered by deep learning in natural language processing and make an outlook on possible future research priorities.

Keywords:　Deep learning; Natural language processing; Pre-training model; Word embedding; Language model

1 引言

自然语言处理是人工智能和语言学领域的分支学科, 主要探讨如何处理及运用自然语言。近年来, 随着深度学习方法的快速发展, 自然语言处理领域中的机器翻译、机器阅读理解、命名实体识别等技术都取得了重要突破。借助于深度学习技术, 面向自然语言处理领域的预训练技术也获得了长足的进步。

在自然语言处理领域的背景下, 预训练技术通过使用大规模无标注的文本语料来训练深层网络结构, 从而得到一组模型参数, 这种深层网络结构通常被称为 “预训练模型” ; 将预训练好的模型参数应用到后续的其他特定任务上, 这些特定任务通常被称为 “下游任务”。

通常来说, 大多数基于深度学习的自然语言处理任务可以分为以下 3 个模块: 数据处理、文本表征和特定任务模型。其中, 数据处理模块和特定任务模型模块需要根据具体任务的不同做相应设计, 而文本表征模块则可以作为一个相对通用的模块来使用。类似于计算机视觉领域中基于 Image-Net 预训练模型的做法, 自然语言处理领域也可以预训练一个通用的文本表征模块, 这种通用的文本表征模块对于文本的迁移学习具有重要意义。

以 Word2Vec 为代表的词向量技术是自然语言处理领域一直以来最常用的文本表征方法, 但这种方法仅学习了文本的浅层表征, 并且这种浅层表征是上下文无关的文本表示, 对于后续任务的效果提升非常有限。直到 ELMo 提出了一种上下文相关的文本表示方法, 并在多个典型下游任务上表现惊艳, 才使得预训练一个通用的文本表征模块成为可能。随后,GPT 和 BERT 等预训练语言模型相继被提出, 自此便进入了动态预训练技术的时代。其中,BERT 在击败 11 个典型下游任务的 State-of-the-art 结果之后, 成为了自然语言处理领域预训练技术的重要里程碑, 极大地推动了自然语言处理领域的发展。此后, 基于 BERT 的改进模型、XL-Net 等大量预训练语言模型涌出, 预训练技术逐渐发展成了自然语言处理领域不可或缺的主流技术。

预训练技术取得的巨大成功, 很大程度上归功于其实现了迁移学习的概念。迁移学习本质上是在一个数据集上训练基础模型, 通过微调等方式, 使得模型可以在其他不同的数据集上处理不同的任务。预训练的过程如上文所述, 是将预训练好的模型的相应结构和权重直接应用到下游任务上, 从而实现 “迁移学习” 的概念, 即将预训练模型 “迁移” 到下游任务。

预训练技术最早被应用于计算机视觉领域, 自 ResNet 出现后, 便开始在视觉领域广泛应用。大量实验证实, 使用预训练技术可以大幅提升下游任务的效果。更重要的是, 充分使用预训练模型极大地改善了下游任务模型对标注数据数量的要求, 从而可以很好地处理一些难以获得大量标注数据的新场景。因此, 在计算机视觉领域, 使用训练好的预训练模型, 再用特定下游任务数据微调模型, 已经成为惯例。

借鉴视觉领域的做法, 自然语言处理领域开始尝试使用预训练技术实现迁移学习。一般来说, 自然语言处理领域使用语言模型来做预训练。语言模型能够量化一个句子近似人类自然表达的概率。大量研究表明: 语言模型可以捕获与下游任务相关的许多知识, 例如长期依赖、层次关系和情绪。语言模型的最大优势之一是训练数据可以来自任意的无监督文本语料, 这意味着可以获得无限量的训练数据。

早期的预训练技术是一种静态技术。2003 年 Bengio 提出的 NNLM 是使用神经网络实现语言模型的经典范例。2013 年,Word2Vec 借鉴 NNLM 的思想, 提出使用语言模型得到词向量。随后,GloVe 和 FastText 等相继被提出。这种词向量的方法作为早期的预训练技术, 逐渐成为了最常用的文本表征技术, 对大多数任务是有帮助的, 但其本质是一种静态的预训练技术, 即不同上下文中的同一词语具有相同的词向量, 因而无法解决自然语言中经常出现的多义词问题, 且其给下游任务带来的提升也非常有限。

对此, 预训练语言模型提供了一种动态的预训练技术方案。2018 年,ELMo 提出了一种上下文相关的文本表示方法, 并在多个典型任务上表现惊艳, 能有效处理一词多义问题。其后,GPT,BERT,XLNet 等预训练语言模型相继被提出, 预训练技术开始在自然语言处理领域大放异彩。从实验效果来看, 预训练语言模型在诸多下游任务上的表现较传统词向量方法取得了很大的提高, 此类下游任务几乎涵盖了自然语言处理领域的典型任务, 例如句子语义关系判断、命名实体识别、阅读理解等, 这充分说明了预训练模型的普适性。此外, 这些模型已经被证明具有极高的采样效率, 只需数百个样本就可以取得很好的性能, 甚至可以实现零样本学习。

预训练语言模型的核心在于关键范式的转变: 从只初始化模型的第一层, 转向了预训练一个多层网络结构。传统的词向量方法只使用预训练好的静态文本表示, 初始化下游任务模型的第一层, 而下游任务模型的其余网络结构仍然需要从头开始训练。这是一种以效率优先而牺牲表达力的浅层方法, 无法捕捉到那些也许更有用的深层信息; 更重要的是, 其本质上是一种静态的方式, 无法消除词语歧义。而

预训练语言模型是预训练一个多层网络结构，用以初始化下游任务模型的多层网络结构，可以同时学到浅层信息和深层信息。此外，预训练语言模型是一种动态的文本表示方法，会根据当前上下文对文本表征进行动态调整，经过调整后的文本表征更能表达词语在该上下文中的具体含义，能有效处理一词多义的问题。

2 原理和优势

以 NNLM，Word2Vec，GloVe 和 FastText 为代表的静态预训练技术推动了自然语言处理领域的快速发展，然而这种静态的词向量技术无法较好地处理一词多义问题。对此，预训练语言模型提供了一种动态的预训练技术方案。2018 年，ELMo 提出了一种上下文相关的文本表示方法，能够有效处理一词多义问题。其后，GPT 和 BERT 等预训练语言模型相继被提出，尤其是 BERT 模型横扫自然语言处理领域的诸多典型任务，成为了自然语言处理领域的一个重要里程碑。

2.1 ELMo 模型

静态的词向量方法存在一个重要缺陷，即无法较好地处理一词多义问题；而 ELMo 通过使用针对语言模型训练好的双向 LSTM 来构建文本表示，由此捕捉上下文相关的词义信息，因而可以更好地处理一词多义问题。

为了使用大规模无监督语料，ELMo 使用两层带残差的双向 LSTM 来训练语言模型，如图 1 所示。此外，ELMo 借鉴了 Jozefowicz 等的做法，针对英文形态学上的特点，在预训练模型的输入层和输出层使用了字符级的 CNN 结构。这种结构大幅减小了词表的规模，很好地解决了未登录词的问题；卷积操作也可以捕获一些英文中的形态学信息；同时，训练双向的 LSTM，不仅考虑了上文信息，也融合了下文信息。

从预训练模型的迁移方式来看，ELMo 是一种特征抽取式的预训练模型。对于第 k 个词来说，ELMo 有 3 层的文本表示可以利用：输入层 CNN 的输出 $h_{k,0}$、第一层双向 LSTM 的输出 $h_{k,1}$ 和第二层双向 LSTM 的输出 $h_{k,2}$。设 3 层文本表示如下：

$$R_k = h_{k,j}(j = 0, 1, 2) \tag{1}$$

则第 k 个词经过预训练模型得到的文本表示为：

$$ELMo_k^{task} = \gamma^{task} \Sigma s_j^{task} h_{k,j}(j = 0, 1, 2) \tag{2}$$

其中，γ^{task} 是一个缩放因子，用以将 ELMo 输出的向量与下游任务的向量拉到同一分布；s_j^{task} 是针对每一层的输出向量设置的不同权值参数，用以组合不同层次的语义信息。

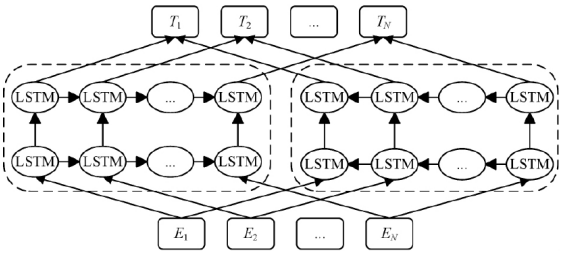

图 1 ELMo 模型的结构

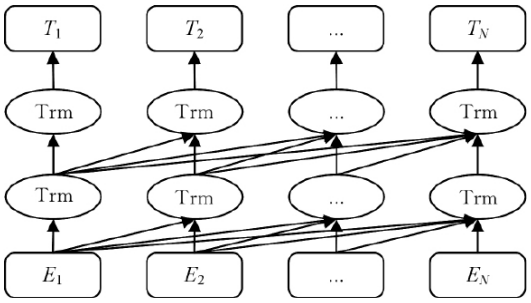

图 2 GPT 模型的结构

ELMo 模型不仅简单，而且表现出众，在自然语言处理领域的 6 个典型下游任务的数据集上全面刷新了最优成绩，尤其在阅读理解任务上提高了 4.7 个点。其主要贡献是提供了一种新的文本表征的思路：在大规模无监督数据上训练预训练语言模型，并将其迁移到下游特定任务中使用。

2.2 GPT 模型

ELMo 使业界意识到了基于大规模语料集预训练的语言模型的威力；同期，ULMFiT 提出的多阶段迁移方法和微调预训练模型的技巧为后来预训练技术的发展提供了重要指导意义；与此同时，Transformer 在处理长期依赖性方面比 LSTM 有更好的表现，它在机器翻译等任务上取得的成果也使一些业内人士开始认为其是 LSTM 的替代品。在此背景下，OpenAI 的 GPT 预训练模型应运而生。GPT 主要借鉴了谷歌 Liu 等的工作，使用生成式方法来训练语言模型。该工作中的解码器在逐字生成翻译的过程中屏蔽了后续的词语序列，天然适合语言建模，因此 GPT 采用了 Transformer 中的解码器结构，并没有使用一个完整的 Transformer 来构建网络。GPT 模型堆叠了 12 个 Transformer 子层，并用语言建模的目标函数来进行优化和训练。GPT 模型的结构如图 2 所示。在迁移学习的模型设计方面，GPT 同样借鉴了 Liu 等的做法，

巧妙地将整个迁移学习的框架做到非常精简和通用。在输入层，若输入只有一个序列，则直接在原序列的首尾添加表示开始和末尾的符号；若输入有两个序列，则通过一个中间分隔符“$”将其连接成一个序列，然后同样在开头和末尾添加标记符号。这套输入的表示方法，基本可以使用同一个输入框架来表征大多数文本问题。除此之外，在输出层，只需要接入一个全连接层或其他简单结构，一般不需要非常复杂的模型设计。基于这种输入层和输出层的通用化设计，只要中间多层解码器层的表征能力足够强，迁移学习在下游任务中的威力就会变得非常强大。GPT 在公布的结果中，一举刷新了自然语言处理领域中的 9 项典型任务，效果不可谓不惊艳。GPT 模型使用的是 Transformer 的解码器结构，正是 Transformer 强大的表征能力，为最终的模型表现奠定了坚实的基础。

2.3 BERT 模型

GPT 模型虽然达到了很好的效果，但本质上仍是一种单向语言模型，对语义信息的建模能力有限。因此，建立一个基于 Transformer 的双向预训练语言模型是一种重要的研究思路。BERT 使用了一种特别的预训练任务来解决这个问题。与 GPT 相同，BERT 同样通过堆叠 Transformer 子结构来构建基础模型，模型结构如图 3 所示，但通过 Masked-LM 这个特别的预训练方式达到了真双向语言模型的效果。Masked-LM 预训练类似于一种完形填空的任务，即在预训练时，随机遮盖输入文本序列的部分词语，在输出层获得该位置的概率分布，进而极大化似然概率来调整模型参数。在模型训练过程中，随机选择文本序列中 15% 的词用于后续替换，但这些词也并非全部被替换为 [MASK]，其中 10% 替换为随机词，10% 保持不变。这种操作可以理解为通过引入噪声来增强模型的鲁棒性。与此同时，为了更好地处理多个句子之间的关系，BERT 还利用和借鉴了 Skip-thoughts 中预测下一句的任务来学习句子级别的语义关系。具体做法是：按照 GPT 提出的组合方式将两个句子组合成一个序列，模型预测后面句子是否为前面句子的下文，也就是建模预测下一句的任务。因此，BERT 的预训练过程实质上是一个多任务学习的过程，同时完成训练 Masked-LM 和预测下一句这两个任务，损失函数也由这两个任务的损失组成。在预训练细节上，BERT 借鉴了 ULMFiT 的一系列策略，使模型更易于训练。在如何迁移到下游任务方面，BERT 主要借鉴了 GPT 的迁移学习框架的思想，并设计了更通用的输入层和输出层。此外，在预训练数据、预训练模型参数量和计算资源上，BERT 也远多于早期的 ELMo

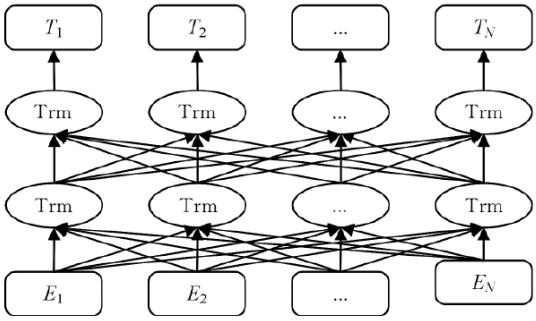

图 3 BERT 模型的结构

和 GPT。BERT 的表现是里程碑式的，在自然语言处理领域的 11 项基本任务中获得了显著的效果提升。而自然语言处理领域的许多后续研究一般也以 BERT 模型为基础进行改进，学界普遍认为，从 BERT 模型开始，自然语言处理领域终于找到了一种方法可以像计算机视觉那样进行迁移学习。总而言之，BERT 的出现是建立在前期很多重要工作之上的，包括 ELMo，ULMFiT，GPT，Transformer 以及 Skip-thoughts 等，是一个集大成者。BERT 的出现极大地推动了自然语言处理领域的发展，凡需要构建自然语言处理模型者，均可将这个强大的预训练模型作为现成的组件使用，从而节省了从头开始训练模型所需的时间、精力、知识和资源。

2.4 小结

虽然静态的预训练技术带来了一定程度的性能提升，但是这种提升非常有限；更重要的是，这种静态的词向量技术无法解决一词多义问题。ELMo 的出现开创了一种上下文相关的文本表示方法，很好地处理了一词多义问题，并在多个典型任务上有了显著的效果提升。其后，GPT 和 BERT 等预训练语言模型相继被提出，自此便进入了动态预训练技术的时代。尤其是 BERT 的出现，横扫了自然语言处理领域的多个典型任务，极大地推动了自然语言处理领域的发展，成为预训练史上一个重要的里程碑模型。此后，基于 BERT 的改进模型、XLNet 等大量新式预训练语言模型涌出，预训练技术在自然语言处理领域蓬勃发展。在预训练模型的基础上，针对下游任务进行微调，已成为自然语言处理领域的一个新范式。

3 研究进展

BERT 的出现开启了一个新时代，此后涌现出了大量的预训练语言模型。这些新式的预训练语言

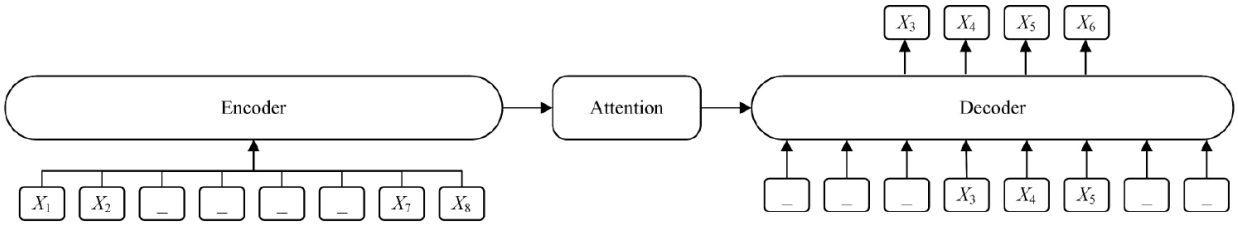

图 4 MASS 模型的结构

模型从模型结构上主要分为两大类：基于 BERT 的改进模型和 XLNet。基于 BERT 的改进模型主要是针对原生的 BERT 模型进行改进，主要改进方向包括：改进生成任务、引入知识、引入多任务、改进掩码方式，以及改进训练方法。基于 BERT 的改进模型都是自编码语言模型；而 XLNet 与 BERT 模型区别较大，是自回归语言模型的一个典型范例。

3.1 基于 BERT 的改进模型

本节主要介绍了在各个方向上基于 BERT 的改进工作。改进的方向主要包括：改进生成任务、引入知识、引入多任务、改进掩码方式，以及改进训练方法。

3.1.1 改进生成任务

由于 BERT 本身在预训练过程和使用过程中不一致，并且没有为生成任务设计相应的机制，导致其在生成任务上效果不佳。本节主要介绍 MASS 和 UNILM 两个模型，这两个模型基于 BERT 改进了其在生成任务上的表现。MASS 的模型结构如图 4 所示。MASS 使用 4 层的 Transformer 结构，训练数据是单句话，编码器模块会随机掩码连续的个词，然后把这些词放入解码器模块的相应位置。MASS 期望解码器模块利用编码器编码的信息和解码前面的词，来预测这些被掩码的词。BERT 和 GPT 都是 MASS 的特例。当 k=1，即随机掩码单个词时，MASS 就退化成 BERT；当与句子长度相等，即掩码所有词时，MASS 就退化成 GPT，即标准的单向语言模型。MASS 把 BERT 推广到生成任务，并设计统一了 BERT 和传统单向语言模型框架 BERT+LM，使用 BERT 作为编码器，使用标准单向语言模型作为解码器。在 WMT14 英语-法语、WMT16 英语-德语和 WMT16 英语-罗马尼亚语的机器翻译数据上进行对比实验，实验结果均为 BLEU 评测值，如表 1 所列。从表中数据可以看出，MASS 优于对比框架 BERT+LM。UNILM 模型的核心框架也是 Transformer，但同时以双向语言模型、单向语言模型和 Seq2Seq 语言模型为目标函数。这些目标函数共享一个网络结构，训练也都使用了类似 BERT 中的掩码机制。与 BERT 的双向语言模型不同，单向语言模型在训练时不能使用下文的信息。Seq2Seq 语言模型在解码器预测时也有与单向语言模型类似的约束，UNILM 使用掩码机制来满足这些约束。UNILM 模型训练时目标函数的设定也参照 BERT，但要兼顾双向语言模型、单向语言模型和 Seq2Seq 语言模型；使用的模型大小与 BERT-large 相同，即 24 层的 Transformer。在微调阶段，对于自然语言理解任务，UNILM 和 BERT 的处理相同；对于自然语言生成任务，UNILM 随机掩码解码器中的词，再进行预测。UNILM 通过在预训练阶段同时训练双向语言模型、单向语言模型和 Seq2Seq 语言模型，使用掩码机制解决了语言模型中的约束问题，可以更好地处理自然语言理解和自然语言生成的各种任务。从实验结果来看，UNILM 在摘要生成任务、问题生成任务、生成式问答任务和对话回复生成任务上的效果提升明显；同时，在其他自然语言处理任务上也有较大的效果提升，如在 GLUE 上首次不加外部数据的情况下的实验效果优于 BERT。

表 1 BERT+LM 与 MASS 的对比

Dataset	BERT+LM	MASS
en-fr	33.4	37.5
fr-en	32.3	34.9
en-de	24.9	28.3
de-en	32.9	35.2
en-ro	31.7	35.2
ro-en	30.4	33.1

3.1.2 引入知识

BERT 模型通过训练 Masked-LM，利用多层双向 Transformer 的建模能力，取得了很好的效果。但是，BERT 模型主要建模原始的语言内部的信号，较少利用语义知识单元建模，导致模型很难学出语义知识单元的完整语义表示。这个问题在中文方面尤为明显，例如，对于“乒 [X] 球”“清明上 [X]

图”等词，BERT 模型可以通过字的搭配推测出掩码的字信息，但没有显式地对语义知识单元（如乒乓球、清明上河图）及其对应的语义关系进行建模。本节主要介绍 ERNIE1.0 和 ERNIE(THU) 两个模型，它们通过引入知识，使预训练模型学习到海量文本中蕴含的潜在知识，进一步提升了预训练语言模型在各个下游任务中的效果。针对 BERT 模型的不足，百度提出基于知识增强的 ERNIE1.0 模型，通过建模海量数据中的实体概念等先验语义知识，学习语义知识单元的完整语义表示。ERNIE1.0 的模型结构与 BERT 基本一致，不同点在于 BERT 是对字进行随机掩码，而 ERNIE1.0 通过掩码词和实体概念等完整语义单元来训练 Masked-LM，从而使得模型对语义知识单元的表示更贴近真实世界。相较于 BERT 基于局部词语共现学习的语义表示，ERNIE1.0 直接对语义知识单元进行建模，增强了模型的语义表示能力。表 2 对 BERT 与 ERNIE1.0 进行了对比说明。在 BERT 模型中，通过『哈』与

表 2 BERT 与 ERNIE1.0 的数据对比

模型	哈	尔	滨	是	黑	龙	江	省	会
BERT	哈	X	滨	是	X	龙	江	省	会
ERNIE1.0	X	X	X	是	黑	龙	江	省	会

『滨』的局部共现，可以判断出『尔』字，然而模型没有学习与『哈尔滨』相关的知识。而 ERNIE1.0 通过学习词与实体的表达，使模型能够建模出『哈尔滨』与『黑龙江』的关系。此外，在训练数据方面，BERT 仅使用百科类语料训练模型；而 ERNIE1.0 则对此进行了改进，使用包括百科类、新闻资讯类、论坛对话类语料来训练模型，进一步提升了模型的语义表示能力。相比 BERT 而言，ERNIE1.0 的优势在于：通过学习实体概念知识，可以获得知识单元的完整语义表示；对训练语料的扩展，尤其是论坛对话类语料的引入，增强了模型的语义表示能力。如表 3 所列，在自然语言推断、语义相似度、命名实体识别、情感分析、问答匹配任务的公开中文数据集上的实验结果表明：ERNIE1.0 模型较 BERT 取得了更好的效果。BERT 存在只学习语言相关的信息，而忽略了将知识信息整合到语言理解中的缺陷。清华大学与华为的研究者认为，知识图谱中的多信息实体可以作为外部知识改善语言表征，并提出 ERNIE(THU) 模型，该模型通过使用知识图谱增强 BERT 的预训练效果。ERNIE(THU) 主要分为抽取知识信息与训练语言模型两大步骤。在抽取知识信息部分，研究者首先识别文本中的命名实体，将识别到的实体与知识图谱中的实体进行匹配。在训练语

表 3 ERNIE1.0 与 BERT 的对比实验

（单位：%）

Task	Metrics	BERT		ERNIE1.0	
		dev	test	dev	test
XNLI	*acc*	78.1	77.2	79.9	78.4
LCQMC	*acc*	88.8	87.0	89.7	87.4
MSRA-NER	*f*1	94.0	92.6	95.0	93.8
ChnSentiCorp	*acc*	94.6	94.3	95.2	95.4
	mrr	94.7	94.6	95.0	95.1
Nlpcc-DBQA	*f*1	80.7	80.8	82.3	82.7

言模型部分，与 BERT 类似，ERNIE(THU) 采用 Masked-LM 任务以及预测下一句任务作为预训练的目标。为了更好地融合文本和知识特征，研究者设计了新的预训练目标，即随机掩码掉一些对齐了输入文本的命名实体，并要求模型从知识图谱中选择合适的实体以完成对齐。ERNIE(THU) 是结合大规模语料库和知识图谱训练出的增强版的语言表征模型，新的预训练目标要求模型同时聚合上下文和知识事实的信息，充分利用词汇、句法和知识信息，从而构建一种知识化的语言表征模型。ERNIE(THU) 针对知识驱动型任务进行了实验，实验结果表明，ERNIE(THU) 在知识驱动型任务中效果显著，超过当前最佳的 BERT，如表 4 所列。此外，在其他类型的自然语言处理任务上，ERNIE(THU) 也能获得与 BERT 相媲美的性能。

表 4 ERNIE(THU) 与 BERT 的对比实验

（单位：%）

Dataset	Metrics	BERT	ERNIE(THU)
FIGER	*acc*	52.04	57.19
Open Entity	*f*1	76.37	78.42
FewRel	*f*1	84.89	88.32
TACRED	*f*1	66.00	67.97

3.1.3 引入多任务学习

在预训练模型背景下，引入多任务学习指在预训练模型过程中同时学习多个任务，这些任务在训练过程中共享预训练模型的结构和参数，利用多个任务之间的相关性来改进预训练模型的性能和泛化能力。BERT 的预训练过程实质上是一个多任务学习的过程，通过同时训练 Masked-LM 和预测下一句两个任务，提高了预训练模型的语义表达能力。本节主要介绍了 MT-DNN 和 ERNIE2.0 两个模型，它们通过引入多任务学习来提升预训练语言模型的表现。MT-DNN 的模型架构主要包括输入层、文本编码层和任务特定层。文本编码层采用与 BERT 相同的机制，并在后续学习具体任务时共享参数；任务特定层是特定于具体任务的，例如单句分类、文

本相似性、成对文本分类等任务。MT-DNN 的训练过程分为两个阶段：预训练阶段和微调阶段。预训练阶段与 BERT 相同，通过训练 Masked-LM 和预测下一句两个无监督任务学习共享的文本编码层的参数；不同点在于，MT-DNN 的微调阶段引入了多任务学习机制，使用多个任务来微调共享的文本编码层和任务特定层的参数，这种微调的方法使得预训练模型能在更多的数据上进行训练，同时还能获得更好的泛化能力。MT-DNN 具有良好的迁移能力，在训练数据很少的情况下，较 BERT 可以获得更好的性能。MT-DNN 可以较好地处理 BERT 在一些小数据集上微调可能存在无法收敛而表现很差的问题，同时节省新任务上标注数据和微调的成本。MT-DNN 可被看作一个集成学习的过程，因此可以用知识蒸馏进行优化。采用知识蒸馏后，模型在 GLUE 中的表现有了明显提升，如表 4 所列。BERT 主要通过词或句子的共现信号预训练语言模型。然而，除语言共现信息之外，语料中还包含词法、语法、语义等更多有价值的信息。基于此，百度团队提出可持续学习的语义理解框架 ERNIE2.0，在预训练阶段引入多任务学习机制，这也是对其早期发布的 ERNIE1.0 的改进。ERNIE2.0 框架支持增量地引入词汇、语法、语义多个层次的自定义预训练任务，能全面捕捉训练语料中的词法、语法、语义等潜在信息。在预训练阶段，通过交替学习这些不同种类的任务，对模型不断训练更新，这种连续交替的学习范式使模型不会忘记之前学到的语言知识，持续提升模型效果。依托 ERNIE2.0 框架，百度团队充分借助飞桨多机分布式训练的优势，使用 79 亿词语的训练数据（约 1/4 的 XLNet 数据）和 64 张 V100（约 1/8 的 XLNet 算力）训练预训练模型。该团队还试验了此预训练语言模型在中英文领域的效果：在英文领域，ERNIE2.0 在 GLUE 的 7 个任务上的表现超越了 BERT 和 XLNet；在中文领域，其在包括阅读理解、情感分析、问答等 9 个不同类型的数据集上的表现超越了 BERT 并刷新了最佳成绩，如表 5 所列。ERNIE2.0 的工作表明，在预训练阶段，通过构建多个训练任务可以显著提升模型效果。

表 5　MT-DNN 和 ERNIE2.0 在 GLUE 数据集上的实验结果

（单位：%）

Dataset	BERT	MT-DNN	ERNIE2.0
CoLA	60.5	62.5	63.5
SST-2	94.9	95.6	95.6
MRPC	89.3	91.1	90.2
STS-B	86.5	88.8	90.6
QQP	72.1	72.7	73.8
MNLI-m/mm	86.7/85.9	86.7/86.0	88.7/88.8
QNLI	91.1	93.1	94.6
RTE	70.1	81.4	80.2
WNLI	65.1	65.1	67.8

3.1.4　改进掩码方式

BERT 模型使用 Masked-LM 任务进行训练，按照字粒度进行掩码，这种掩码方式不利于学习到完整的词义表示。本节介绍 BERT WWM(Whole Word Masking) 系列模型和 SpanBERT 模型，它们改进了原生 BERT 模型的掩码方式，进一步提升了模型性能。BERT WWM 是一种全词掩码方式，是谷歌发布的一项 BERT 的升级版本，主要更改了预训练阶段 Masked-LM 的掩码策略。BERT 采用 WordPiece 的分词方法，把一个完整的词切分成若干个子词。最初的 BERT 的掩码策略是随机掩码一个句子中的部分子词，而在全词掩码中，如果一个完整词的部分子词被掩码，则同属该词的其他部分也会被掩码。这种全词掩码的策略使得预训练模型在训练 MaskedLM 的过程中将恢复整个词语作为训练目标，而不是仅恢复部分子词。全词掩码策略克服了原生 BERT 模型掩码部分子词的缺点，进一步提升了 BERT 模型的性能水平。谷歌现已发布了基于全词掩码方式训练好的预训练模型（BERT-large-wwm）。此外，在中文领域，哈工大讯飞联合实验室发布了基于全词掩码的中文 BERT 预训练模型 BERT-wwm-ext，其在多个中文数据集上取得了当前中文预训练模型的最佳水平，实验效果甚至超过了原生 BERT 和 ERNIE 等中文预训练模型。SpanBert 是一个新的分词级别的预训练模型，能够对分词进行更好地表示和预测。该模型与 BERT 的差别主要体现在掩码机制和训练目标上。在掩码机制方面，与 BERT 团队的全词掩码类似，SpanBERT 不是随机地对单个子词进行掩码，而是对随机的邻接分词添加掩码。每次掩码的过程是先从一个几何分布中采样得到需要掩码的分词的长度，并在此分词级别上进行掩码。在训练目标方面，SpanBERT 提出了一个新的模型训练目标 Span Boundary Object(SBO)，通过使用分词边界的表示来预测被添加掩码的分词的内容，不再依赖分词内单个子词的表示。这种训练目标能使模型在边界词中存储其分词级别的信息，有助于模型的调优。从实验效果来看，SpanBERT 在多个任务中的表现都超越了所有的 BERT 基线模型，且在问答任务、指代消解等分词选择类任务中均取得了重要的性能提升。特别地，在使用与 BERT 相同的训练数据和模型大小时，SpanBERT 在 SQuAD1.0 和 2.0 中的 F1 值分别为 94.6% 和 88.7%。此外，SpanBERT 在不涉及分词选择的任务中也取得了进展，在 GLUE 数据集上的表现亦有所提升。

3.1.5 改进训练方法

本节主要介绍 RoBERTa 模型，该模型与 BERT 基本一致，改进之处在于设计了更加精细的训练方法，提高了模型性能。RoBERTa 是 Facebook AI 联合 UW 发布的基于 BERT 改进的预训练模型，在模型结构层面上较 BERT 并没有较大的改变，其改进主要体现在以下 4 个预训练的方法：动态掩码机制、移除预测下一句的任务、更大的批大小、更多的数据和更长的训练时间。BERT 的掩码机制是一种静态的掩码策略，对于每一个序列来说，掩码的词语一旦选定，在之后的整个训练过程中都不会再发生改变。而 RoBERTa 提出了一种动态的掩码机制，即一开始把预训练的数据复制 10 份，每一份都随机进行掩码，则同一个序列会有 10 种不同的掩码方式，因而在模型预训练的过程中，每个序列被掩码的词语是会变化的。RoBERTa 在只将静态掩码改成动态掩码而其他训练方法不变的情况下进行实验，结果表明动态掩码机制确实能提高性能。为了捕捉句子之间的关系，BERT 除了使用 Masked-LM 任务外，还使用了预测下一句的任务来预训练模型，在预训练阶段每次拼接两个句子作为输入数据。而 RoBERTa 在预训练阶段去除了预测下一句的任务，改为每次输入连续的多个句子，直到序列达到最大长度，这种训练方式也叫作全句模式。实验表明，在推断句子关系的任务上，RoBERTa 也能有更好的性能。RoBERTa 还在批大小上设计了实验探索：BERT 的批大小是 256，RoBERTa 探索了 2k 和 8k 的批大小。这一思想主要借鉴了在机器翻译中，使用更大的批大小并配合更大的学习率能加快模型优化速率并提升模型性能，对比实验也证明了更大的批大小可以给模型性能带来一定程度的提升。借鉴 XLNet 用了比 BERT 多 10 倍的训练数据的思想，RoBERTa 也使用了更多的训练数据，同时需要训练更长的时间。从实验效果来看，更多的训练数据配合更长的训练时间，确实可以带来模型性能的提高。这种思路一定程度上与 GPT2.0 扩充数据的方法类似，需要消耗大量的计算资源。RoBERTa 模型主要在以上 4 个方面对 BERT 进行精细调参，在 GLUE 上对比当时最先进的 XLNet 模型，其在多个任务上获得了超越 XLNet 的表现，如表 6 所列。

表 6 RoBERTa 与 XLNet 在 GLUE 数据集上的实验结果

（单位：%）

Dataset	XLNet	RoBERTa
CoLA	67.8	67.8
SST-2	96.8	96.7
MRPC	93.0	92.3
STS-B	91.6	92.2
QQP	90.3	90.2
MNLI-m/mm	90.2/89.8	90.8/90.2
QNLI	98.6	98.9
RTE	86.3	88.2
WNLI	90.4	89.0

3.2 XLNet 模型

BERT 是典型的自编码模型，旨在从引入噪声的数据中恢复出原数据。BERT 的预训练过程采用了降噪自编码思想，提出了 Masked-LM 预训练任务，该任务的最大贡献在于使模型获得了真正的双向上下文信息，但是也带来了一些问题：首先，预训练时使用的掩码机制在下游任务微调时并不会使用，导致训练和使用两个过程存在数据偏差，对实际效果有一定影响；其次，BERT 中每个单词的预测是相互独立的，而类似于 “New York” 这样的实体，“New” 和 “York” 是存在关联的，这个假设忽略了这样的情况。自回归模型一般不存在第二个问题，但传统的自回归模型本质上是单向的，无法建模双向信息。XLNet 的贡献在于提出了一种可以获得真双向的上下文信息的自回归语言模型，进而避免了第一个问题。XLNet 主要使用 3 种机制来解决上述问题：排列语言模型、双流自注意力和循环机制。排列语言模型是指预测某个单词时，XLNet 使用原始输入次序的随机排列来获取双向的上下文信息，同时维持自回归模型原有的单向形式。它采用了一种比较巧妙的实现方式：使用单词在排列中的位置计算上下文信息。如对于一个 2 → 4 → 3 → 1 的排列，单词 2 和单词 4 就可以作为上文的输入来预测单词 3 。当原句的所有排列都取完时，就能获得所有的上下文信息。为了考虑位置因素对预测结果的影响，引入了要预测单词的位置信息。此外，为了降低模型的优化难度，XLNet 使用了部分预测的方式，最终优化目标如式（3）所示：

$$\max_{\theta} E_{z\sim Z_T}[\Sigma_{t=1}^{T} log\ p_\theta(x_{z_t}|X_{z<t})] \tag{3}$$

其中，Z_T 表示长度为 T 的序列的所有排列组成的集合，z 是一种排列方法，x_{z_t} 表示排列的第 t 个元素，$X_{z<t}$ 表示排列的第 1 到第 t-1 个元素。双流自注意力机制要解决的问题是，当获得考虑了位置因素的向量表示后，只能获得该位置信息以及上文信息，不足以预测该位置后的单词；而原来的向量表示则因为获取不到位置信息，依然不足以预测该位置后的单词。因此，XLNet 引入了双流自注意力机制，将两者结合起来。循环机制借鉴了 Transformer-XL 的思想，即在处理下一个单词时结合上个单词的隐层表示，使得模型能够获得更长距离的上下文

信息。XLNet 虽然在前端采用了相对位置编码，但在隐层表示时涉及到的处理与排列独立，因此还可以沿用这个循环机制。该机制使得 XLNet 在处理长文档时具有较好的优势。相比 BERT，XLNet 采用自回归语言模型解决了单词之间预测不独立的问题，同时采用了排列语言模型等机制使自回归模型也可以获得真双向的上下文信息。XLNet 的最终结果与 BERT 进行了较为公平的比较，在模型的训练数据、超参数以及网格搜索空间等与 BERT 一致的情况下，使用单模型在 GLUE 的 dev 上进行对比实验。如表 7 所列，XLNet 的实验结果优于 BERT。

表 7　BERT 与 XLNet 在 GLUE 数据集上的实验结果

（单位：%）

Dataset	BERT	XLNet
CoLA	60.6	63.6
SST-2	93.2	95.6
MRPC	88.0	89.2
STS-B	90.0	91.8
QQP	91.3	91.8
MNLI-m/mm	86.6/—	89.8/—
QNLI	92.3	93.9
RTE	70.4	83.8
WNLI	—	—

3.3 小结

BERT 的出现开启了一个新时代，此后涌现出了大量的预训练语言模型。以上是依据模型结构，分为基于 BERT 的改进模型和 XLNet 进行的讨论。此外，预训练语言模型还可以从特征抽取、语言模型目标、特征表示 3 个方面进行划分。特征抽取方面，主要分为 RNNs，Transformer 和 Transformer-XL 3 种。ELMo 和 ULMFiT 使用 RNNs 作为特征抽取器，自谷歌提出 Transformer 后，GPT 和 BERT 系列模型就使用 Transformer 的相关结构进行特征抽取，XLNet 则使用 Transformer-XL。语言模型目标方面，主要分为自编码语言模型和自回归语言模型。BERT 系列模型均使用自编码语言模型和单向语言模型，包括 ELMo，ULMFiT，GPT 等；XLNet 则使用自回归语言模型。特征表示方面，主要分为单向特征表示和双向特征表示。单向语言模型使用单向特征表示，BERT 系列模型和 XLNet 均使用真双向的特征表示。

4 总结与展望

本文主要概述了面向自然语言处理领域的预训练技术及其发展历史。以 BERT 为分界点，预训练技术的发展历史大致可以分为 3 个阶段：早期的静态预训练技术、经典的动态预训练技术以及新式的动态预训练技术。早期的静态预训练技术主要是以 Word2Vec 为代表的词向量技术；经典的动态预训练技术主要是 ELMo，GPT 和 BERT 等；新式的动态预训练技术主要包括基于 BERT 的改进模型和 XLNet。目前，预训练技术已经在自然语言处理领域取得了很大进展，但同时也面临诸多挑战：无法处理常识和推理问题，在生成任务中表现逊色，以及资源消耗过大等。未来预训练技术可能会致力于解决上述问题，重点研究如何处理常识和推理问题，如何改善生成任务，以及如何降低训练成本等。

参 考 文 献

[1] Poth, Clifton, et al. "What to Pre-Train on? Efficient Intermediate Task Selection." arXiv preprint arXiv:2104.08247 (2021).

[2] Koto, Fajri, Jey Han Lau, and Timothy Baldwin. "Discourse Probing of Pretrained Language Models." arXiv preprint arXiv:2104.05882 (2021).

[3] Wang, Sinong, Madian Khabsa, and Hao Ma. "To Pretrain or Not to Pretrain: Examining the Benefits of Pretraining on Resource Rich Tasks." arXiv preprint arXiv:2006.08671 (2020).