第三讲
大语言模型基础
Lecture 3
Basics of LLM

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

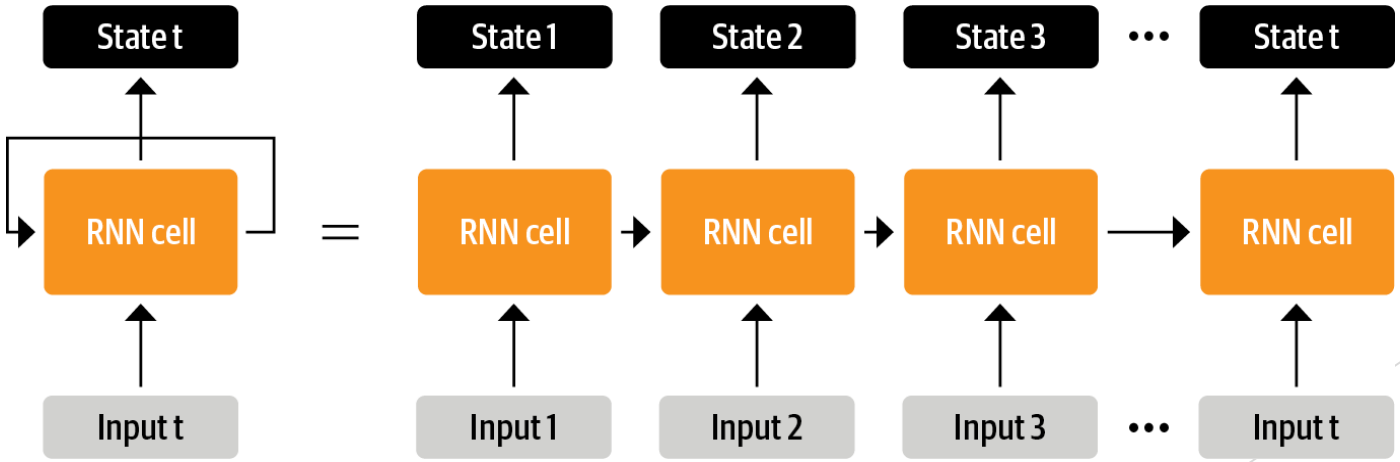
- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit for the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for remedy including deletion have you had any concern.

循环神经网络
Recurrent Neural Network

- 我们首先回顾一下循环神经网络，下图是其按时间步骤展开的图示。从图中可以看出，这种架构较为容易捕获时序数据的潜在的依赖关系，并且在对一些时序数据的处理中，也证实了循环结构的有效性。

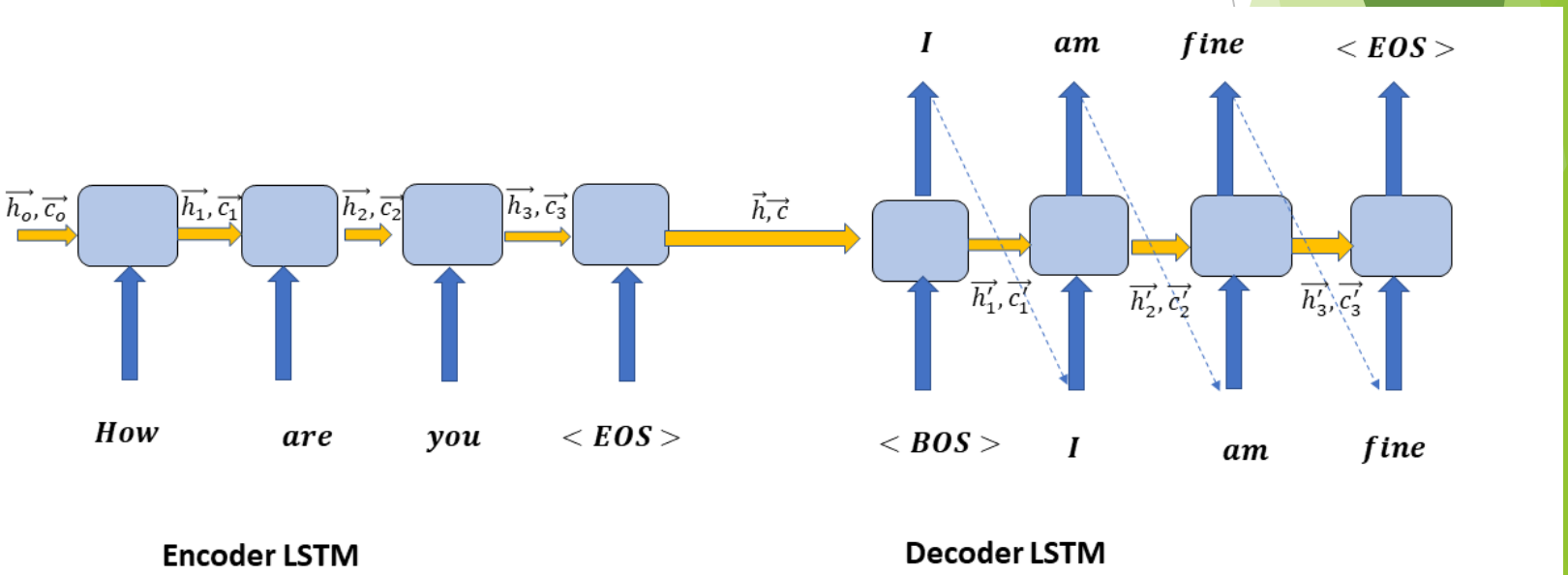
  Let‘s first review the recurrent neural network. The following diagram illustrates its unfolding over time steps. From the diagram, it’s evident that this architecture is quite adept at capturing the temporal correlation within time-series data. Applications of various types of sequential data also confirm its effectiveness.



序列对序列模型
Sequence-to-sequence Model

- Naturally, people apply recurrent neural networks to process natural language which is obviously sequential data. Additionally, for different natural language tasks, network variants have been crafted. For instance, for language translation, the sequence-to-sequence (Seq2Seq) model has been devised, illustrated below. Here, $\vec{h}$ is referred to as the context vector, EOS signifies the input end-of-sequence token, and BOS represents the output start token.

- 自然地，人们应用循环神经网络来处理作为序列数据的语言，并且针对不同的自然语言任务，产生了一些变体。例如，对于语言翻译，人们发明了序列对序列模型，其如图所示。其中$\vec{h}$被称为上下文向量，EOS称为输入结束表示，BOS称为输出开始标识。



序列对序列模型
Sequence-to-sequence Model

- 尽管基于深度学习的方法，例如循环神经网络，在处理一些自然语言问题的尝试中，比基于统计的方法得到了更好的效果，但循环网络本身也存在着一些问题。
  - 在循环神经网络的计算中，尽管可以捕获时序数据内蕴的依赖关系，但这种依赖往往假设在相邻的步骤中。如果序列太长，则这种时序依赖关系可能不能够被捕获。并且，缺乏有效的方法来直观展示这种捕获的依赖关系，因此，很难验证捕获的依赖关系的质量，从而缺乏对优化模型的依赖指标。例如，在seq2seq模型中，上下文向量$\vec{h}$是最后一步的隐向量，从该量很难得出步骤t_i的数据点与步骤t_j的数据点的依赖关系。
  - 由于循环神经网络结构上的顺序性质以及时间步之间的相互依赖，其在计算中难以并行化。与卷积神经网络不同，可以同时在图像的不同空间位置进行计算，而RNN依赖于先前时间步的信息，这使得很难进行并行计算。每个时间步的计算都依赖于前一个时间步的结果，这导致了顺序流，阻碍了并行化。

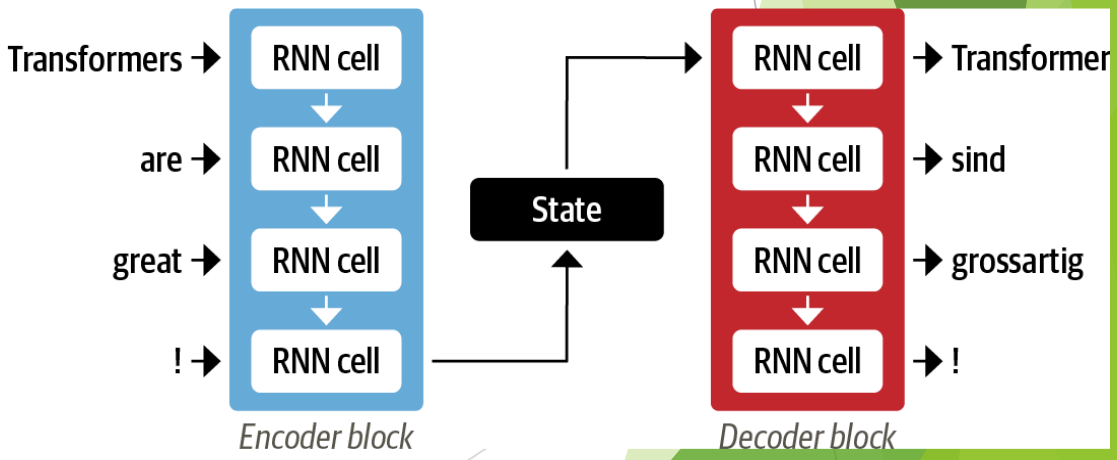
序列对序列模型
Sequence-to-sequence Model

- Although deep learning models, such as recurrent neural networks, have shown better performance compared to statistical methods in handling some natural language problems, recurrent networks themselves present certain limits.
  - Although recurrent neural networks can capture inherent correlations within sequential data, it often assumes that these correlations exist in adjacent steps. If the sequence is too long, these temporal correlations might not be effectively captured. Moreover, there are lack of effective methods to visually represent these captured dependencies, making it challenging to verify the quality of captured dependencies and therefore lacking reliable metrics for optimizing models. For example, in Seq2Seq models, the context vector $\vec{h}$ is the latent vector out of the last step, making it difficult to infer the dependencies between data points at step t_i and step t_j.
  - Recurrent neural networks are challenging to parallelize due to their sequential nature and the interdependence between time steps. Unlike convolutional neural networks, where computations across different spatial locations in an image can be performed simultaneously, RNNs rely on information from previous time steps, making it difficult to compute them in parallel. Each time step's computation depends on the result of the previous step, leading to a sequential flow that impedes parallelization.

编码—解码框架
Encoder-Decoder Framework

- 序列对序列模型有时也称为编码解码框架。其中，编码器的作用是将输入序列的信息编码成某种向量形式的表示，称为最终隐含状态。这个状态随后传递给解码器，用于解码器生成输出序列。从下图容易看出，这个隐含状态往往成为整个模型的性能瓶颈。因此，寻求一种可以让解码器有效访问编码器所有隐含状态的方法是潜在的努力方向：

- Sequence-to-sequence model is also referred to as encoder-decoder framework, under which the encoder is to encode information from the input sequence into some vector representation, known as the final hidden state. This state is subsequently passed to the decoder, which generates the output sequence based on it. As shown in the right diagram, it is manifest that the hidden state tends to be the performance bottleneck for the whole model. Therefore, seeking a method that allows the decoder to effectively access all hidden states of the encoder is a potential direction.



注意力机制
Attention Mechanism

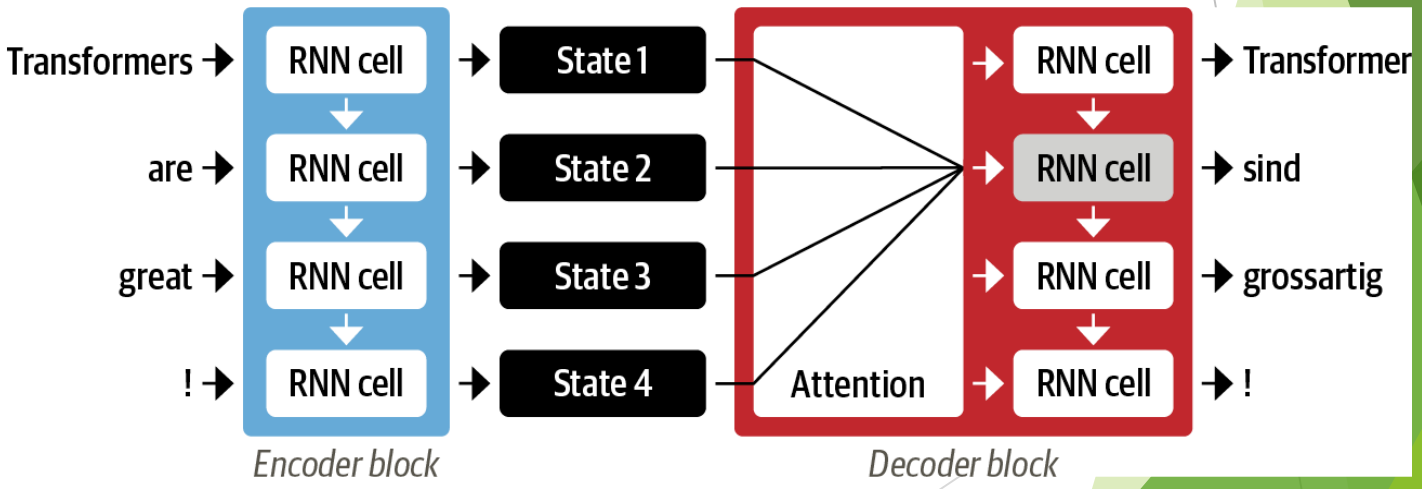
- 我们知道编码器或循环神经网络在每一步都能输出一个隐含状态，既然仅利用输入序列的最后的单一的隐含状态是有问题的，有没有可能解码器可以同时访问这些所有的隐含状态？但这会引入新的问题：对于较长的序列，同时使用所有状态会为解码器创建一个巨大的输入，可能导致训练需要的算力剧增，因此需要某种机制来优先考虑使用哪些状态。

  We know that the encoder or recurrent neural network can output a hidden state at each step. Since relying solely on the final single hidden state of the input sequence is problematic, is it possible for the decoder to access all of these hidden states simultaneously? However, this introduces new challenges: for longer sequences, using all states simultaneously would create a massive input for the decoder, potentially leading to a significant increase in computational power required for training. Therefore, there is a need for some mechanism to prioritize which states to use.

注意力机制
Attention Mechanism

- 一个直观的想法是让解码器在每一个解码时间步分配不同的权重给单独的编码器状态。这个机制称为注意力机制，其相关过程如下图所示，其中展示了在预测输出序列中特定词时注意力分配或关注的情况：

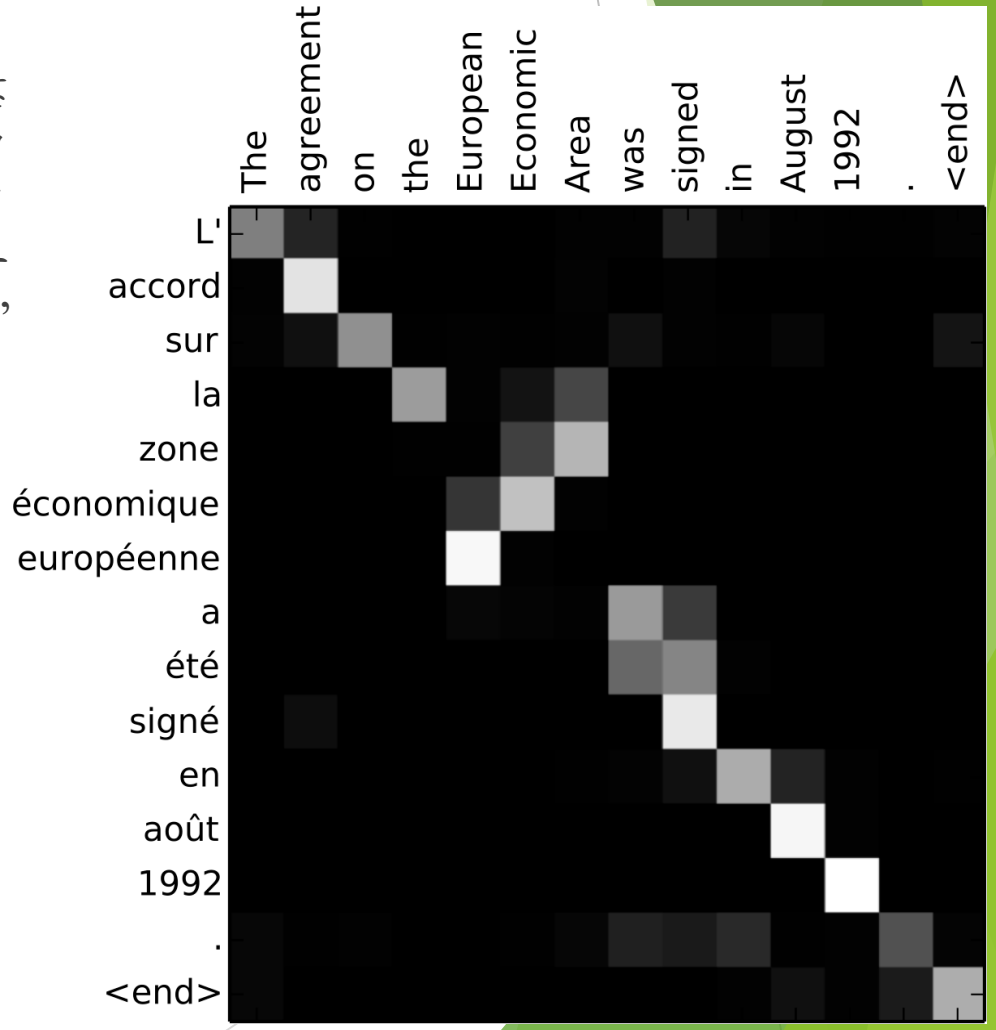
- An intuitive idea is to allow the decoder to assign different weights to individual encoder states at each decoding time step. This mechanism is called the attention mechanism, and its associated process is illustrated in the diagram below, showing the attention allocation or focus when predicting the specific word in the output sequence:



注意力机制
Attention Mechanism

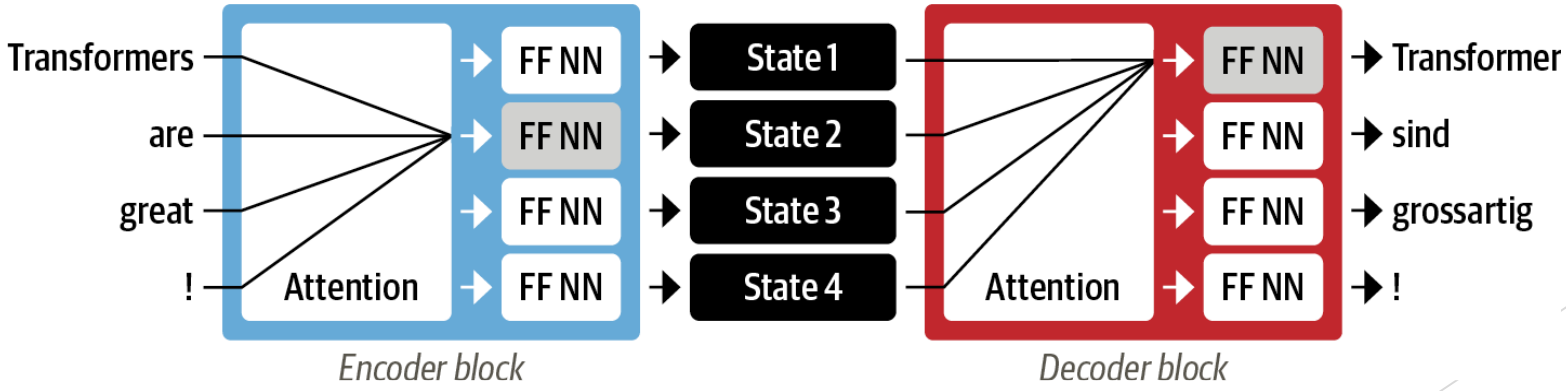
- 另外，通过查看每个时间步最相关的输入单词，基于注意力的模型能够很好地展示生成的翻译中的单词与源句子中的单词之间的非平凡的关系或对齐。例如，右图可视化了英语到法语翻译模型的注意力权重，其中每个像素表示一个权重。该图展示了解码器如何正确地对齐单词“zone”和“Area”，尽管这两种语言中单词的顺序是不同的：

- Additionally, by examining the most relevant input words at each time step, attention-based models can effectively demonstrate non-trivial relationships or alignments between words in the generated translation and words in the source sentence. For example, the figure below visualizes the attention weights of an English-to-French translation model, where each pixel represents a weight. The illustration demonstrates how the decoder correctly aligns the words 'zone' and 'Area,' despite the different word orders in two languages:



注意力机制
Attention Mechanism

- 虽然引入的注意力机制能够使得基于循环网络的编码器和解码器框架有更好的性能表现，但正如前所述，这种架构存在一个主要缺点：计算本质上是顺序的，无法在输入序列上并行。为了解决这个问题，Google公司的科学家引入了一种称为Transformer的架构，其最具创新性的建模范式是摒弃了递归，而完全依赖于一种称为自注意力的特殊形式的注意力。该架构的基本思想是允许注意力在同一层神经网络的所有状态上运行，如下图所示。其中，编码器和解码器都有自己的自注意力机制，其相关计算均通过全连接网络进行。由于这种架构可以比递归模型快得多，为自然语言处理领域的许多突破铺平了道路。



注意力机制
Attention Mechanism

- While by introducing the attention mechanism, scholars improve the performance of encoder-decoder framework based on recurrent networks, as mentioned earlier, this architecture has a major drawback: computations are inherently sequential and cannot be parallelized over the input sequence. To address this issue, scientists at Google introduced an architecture called Transformer. The most innovative modeling paradigm of it is to completely abandon recursion and to rely entirely on a special form of attention called self-attention. The fundamental idea of this architecture is to allow attention to operate over all states in the same layer of the neural network simultaneously. Both the encoder and decoder have their own self-attention mechanisms, and the relevant computations are performed through fully-connected networks. Because this architecture can be much faster than recursive models, it paved the way for many breakthroughs in the field of natural language processing since its proposition.

迁移学习
Transfer Learning

- 迁移学习是机器学习方法的一种，其通过在一个任务或领域上训练过的模型重新用于另外的任务或领域，从而节省训练时间并提高泛化性能。迁移学习的实施依赖于如下前提，即在不同任务或领域之间，存在一些通用的、可迁移的表示和学习方法。

  Transfer learning is one of approaches in machine learning. By reusing a model, which has been trained on a task or domain, for another task or domain, it can save the time for training from scratch and improve generalization performance. The implementation of transfer learning relies on the premise that there are common, transferrable representations and learning methods across different tasks or domains.

- 迁移学习使得机器学习模型能够更快速地适应新任务、处理大规模数据集、处理复杂问题，并且减少了对大量标注数据的需求。因此，迁移学习在许多领域中都得到了广泛的应用。

  Transfer learning allows machine learning models to quickly adapt to new tasks, handle large-scale datasets, address complex problems, and reduce the need for a large amount of labeled data. Therefore, transfer learning has been widely used in many fields.

迁移学习
Transfer Learning

- 推动深度神经网络发展的一个因素，特别是在计算机视觉领域，便是迁移学习。人们可以在一个较大的数据集（如ImageNet）上训练一个神经网络（如ResNet），完成分类或侦测任务，然后基于新的可能较小的数据集将模型微调，以完成新的任务。与传统的基于监督学习的全新训练的方式比，这种方法通常会在较小数据量的情况下，依然产生高质量的模型。这种基于上游预训练的模型解决多种下游任务的迁移学习的方式，慢慢地由图像视频领域，扩展到其他领域。

  One factor driving the popularity of deep neural networks, especially in the field of computer vision, is transfer learning. People can train a neural network (such as ResNet) on a large dataset, such as ImageNet, to perform classification or detection tasks, and then fine-tune the model based on a new, possibly smaller dataset to complete a new task. Compared with the traditional approach of training from scratch based on supervised learning, this method typically produces high-quality models even with smaller amounts of data. This approach of using upstream pre-trained models to solve various downstream tasks through transfer learning has gradually expanded from the image and video field to other fields.

迁移学习
Transfer Learning

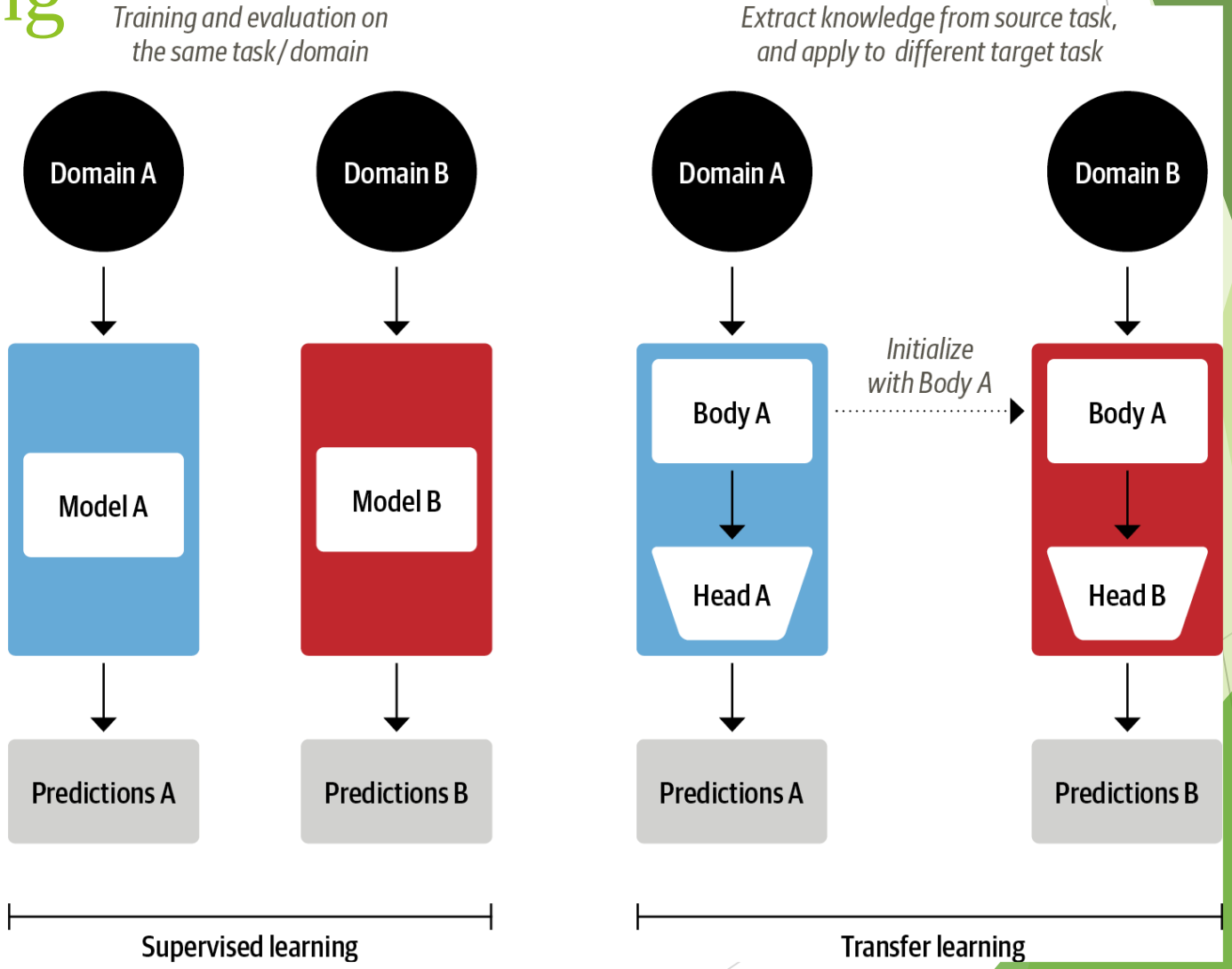
- 基于深度学习的迁移学习往往具体采用如下方式。在网络结构上，一般将模型分成身体和头部，其中身体往往是基于原来的大型数据集训练的原先模型的底部或基座部分，而头部是特定于任务的神经网络。有时，为适配新的任务，头部的网络结构可能需要做出调整。在训练期间，基于原来源领域的学到的基本性的特征，即身体的权重保持不变，可以认为这些权重用于初始化针对新任务的新模型。然后会根据有限的数据集，训练新的头部结构的权重。由于头部一般比较简单，很好地避免了过拟合的情况。这些是高质量模型产生的原因。

  Neural network based transfer learning often adopts the following approach. In terms of network structure, the model is generally divided into a body and a head, where the body is often the bottom or base part of the original model trained on the large dataset, and the head is a neural network specific to the task. Sometimes, the network structure of the head may need to be adjusted to adapt to the new task. During training, based on the features learned from the source domain, the weights of the body are kept unchanged and can be considered as initializing the weights for the new model for the new task. Then, the weights of the new head structure are trained based on a limited dataset. Since the head is usually relatively simple, overfitting is generally avoided, resulting in high-quality models.

迁移学习
Transfer Learning

- 右图展示了传统基于单问题的解决方案与基于神经网络的迁移学习的实施过程。

  The figure on the right shows the traditional per problem based solutions and the neural network-based transfer learning.



迁移学习
Transfer Learning

- 尽管迁移学习在图像处理与计算机视觉领域取得了很大的成功，但是在自然语言处理领域，其在一段时间内效果并不好。特别的，不同自然语言处理问题的异质性比图像视频领域来的要大，数据本身的差异也可能较大，因此曾有迁移学习在自然语言处理领域有效性的疑问。但后来通过增加称为领域适应的阶段，人们发现迁移学习也能很好地用于自然语言处理领域。下面以情感分类为例说明：
  - 预训练

    预训练的目标通常称为语言建模，其通常在于以最小的成本利用大体量的语料库。其做法通常非常简单，例如，根据前几个单词预测下一个单词。这种方法的优势在于它不需要标注数据。例如我们可以利用如维基百科等大量可用文本进行训练。
  - 领域使用

    一旦在大型语料库上完成了语言模型的预训练，下一步就是将其适应到目标语料库中（例如，从维基百科到电影评论IMDB语料库）。这个阶段仍然使用语言建模方式，但现在的模型预测的是目标语料库中的下一个单词。

迁移学习
Transfer Learning

- Although transfer learning has achieved great success in image processing and computer vision, for a period of time, it was not efficient in the field of natural language processing. Due to the fact that the heterogeneity of different natural language processing problems is greater than that in the image and video field, and the differences in the data itself may also be large, there were doubts about the transfer learning efficacy in the field of NLP. However, later on, by adding an extra stage called domain adaptation, it was found that transfer learning can also be well applied to NLP. Take sentiment classification as an example:
  - Pre-training

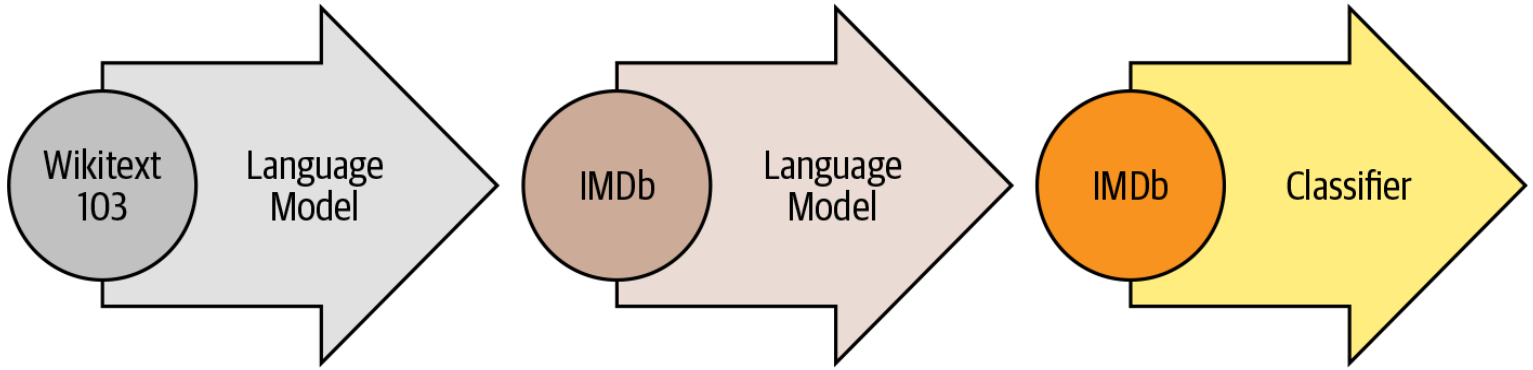
    The goal of pre-training is usually called language modelling, which usually aims to use a large corpus at minimum cost. The method is usually very simple, such as predicting the next word based on the previous few words. The advantage of this method is that it does not require labelled data. For example, we can use a lot of available text, such as Wikipedia, for training.
  - Domain adaptation

    Once the pre-training of the language model is completed on a large corpus, the next step is to adapt it to the target corpus (for example, from Wikipedia to the movie review IMDb corpus). Language modelling is still used in this stage, but now the model predicts the next word in the target corpus.

迁移学习
Transfer Learning

- 微调

在这一步中，语言模型通过为目标任务添加分类层来进行微调（例如，对IMDB的电影评论进行情感分类）。

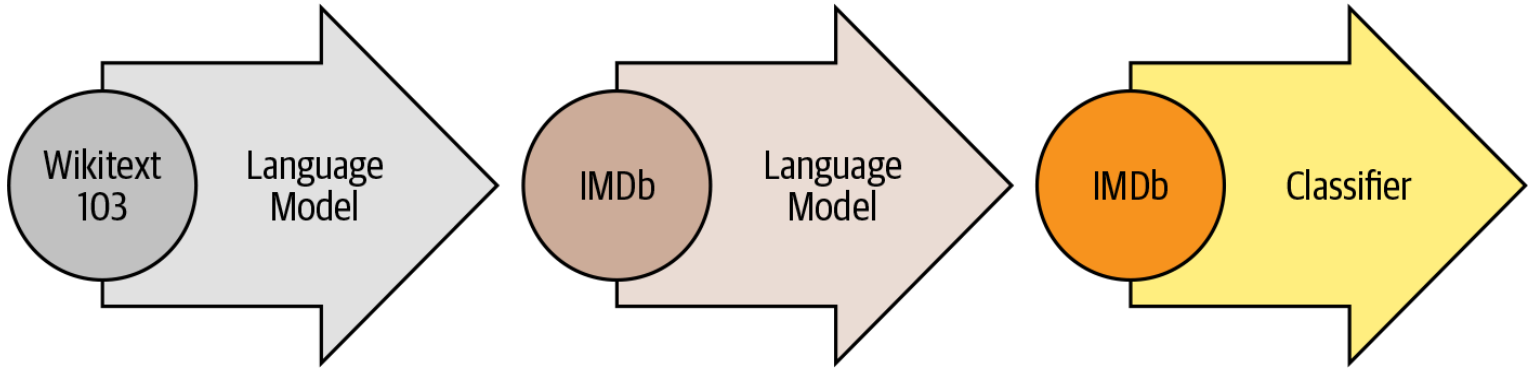


- 上述方式引入了一个可行的解决NLP问题的一个预训练和迁移学习的框架，这为结合基于自注意力机制的Transformer，创造性地、系统性地将深度学习用于一般性的生成式语言任务提供了基础。

迁移学习
Transfer Learning

- Fine-tuning

  In this step, the language model is fine-tuned by adding a classification layer for the target task (for example, sentiment classification of movie reviews on IMDB).



- The above method introduces a feasible framework for pre-training and transfer learning to solve NLP problems. This provides a foundation for creatively and systematically applying deep learning to general generative language tasks, in combination with Transformer-based self-attention mechanism.

架构初探
Framework Exploration

- 为了解决上述的序列数据之间的依赖或耦合问题，我们知道语言模型引入了注意力机制。而注意力的实现则依赖于如下一些概念，如查询（Query），键（Key）与值（Value）。这些在语言模型如ChatGPT中，均是向量的形式。
- 查询、键和值向量的概念在第一次遇到时可能有些晦涩难懂。这些概念和信息检索系统中的一些概念有类似之处，但也借鉴了人们对内存网络的探索的一些经验与做法。
- 我们可以用一个简单的比喻来理解它们的含义。假设我们去超市购买晚餐所需的食材，假设我们知道某道菜的食谱，则每个需要的食材都可以被视为一个查询。当我们在货架之间穿行寻找时，我们会查看标签（键）并检查它们是否匹配清单上的成分（相似性函数）。如果找到了匹配项，那么我们就会从货架上把东西（值）拿走。
- 在上面的比喻中，每个与成分匹配的标签只能得到一件商品。而自注意力是一种更抽象和“平滑”的版本，即超市里的每个标签与成分的匹配程度取决于每个键与查询的匹配程度。所以如果我们的清单上有鸡蛋，那么你可能会主要拿鸡蛋，捎带拿几个鸭蛋和鹌鹑蛋等。但通常在这种情况下，我们不会拿肉或其他的，因为其与我们所要的食材匹配程度太低了。

架构初探
Architecture Exploration

- We know that to solve the dependency or coupling problems between sequence data, language models introduce attention mechanisms. The implementation of attention relies on some concepts such as Query, Key, and Value. These are all in vector form in language models such as ChatGPT.
- The concept of query, key and value vectors may be a bit obscure when encountered for the first time. These concepts share some similarities with concepts in information retrieval systems, but they are also drawn on some experiences and practices from people's exploration of memory networks.
- We can make a simple analogy to understand their meanings. Suppose we go to the supermarket to buy ingredients for dinner, and suppose we know the recipe for a dish, then each ingredient can be regarded as a query. As we walk among the shelves, we look at the labels (keys) and check if they match the ingredients on the list (similarity function). If a match is found, we take the thing (value) off the shelf.
- In the above analogy, each label matching an ingredient can only get one item. Self-attention is a more abstract and "smooth" version, where the match between each label and ingredient depends on the match between each key and query. So if we have eggs on our list, you may mainly take eggs and also get a few duck eggs and quail eggs. But usually in this case, we won't take meat or other things because they don't match what we want.

架构初探
Architecture Exploration

- 我们知道，我们的需要的食材名称（即对应的查询），相应的标签，以及食材本身是内蕴于食材的，并且和食谱有密不可分的关系。我们可以将食材对应于词，食谱对应于句子。因此，我们知道，既然有这种内蕴关系，即查询、键、值是可以由语料生成的。在实际中，我们也确实是这么做的，如下页图所示。

  We know that the names of the ingredients we need (i.e., the corresponding queries), the corresponding labels, and the ingredients themselves are intrinsic to the ingredients and are dependent on the recipe. We can correspond the ingredients to words and the recipes to sentences. Therefore, since there is such an intrinsic relationship, i.e., queries, keys, and values can be generated from the corpus. In practice, this is indeed what we do, as shown in the figure on the next page.

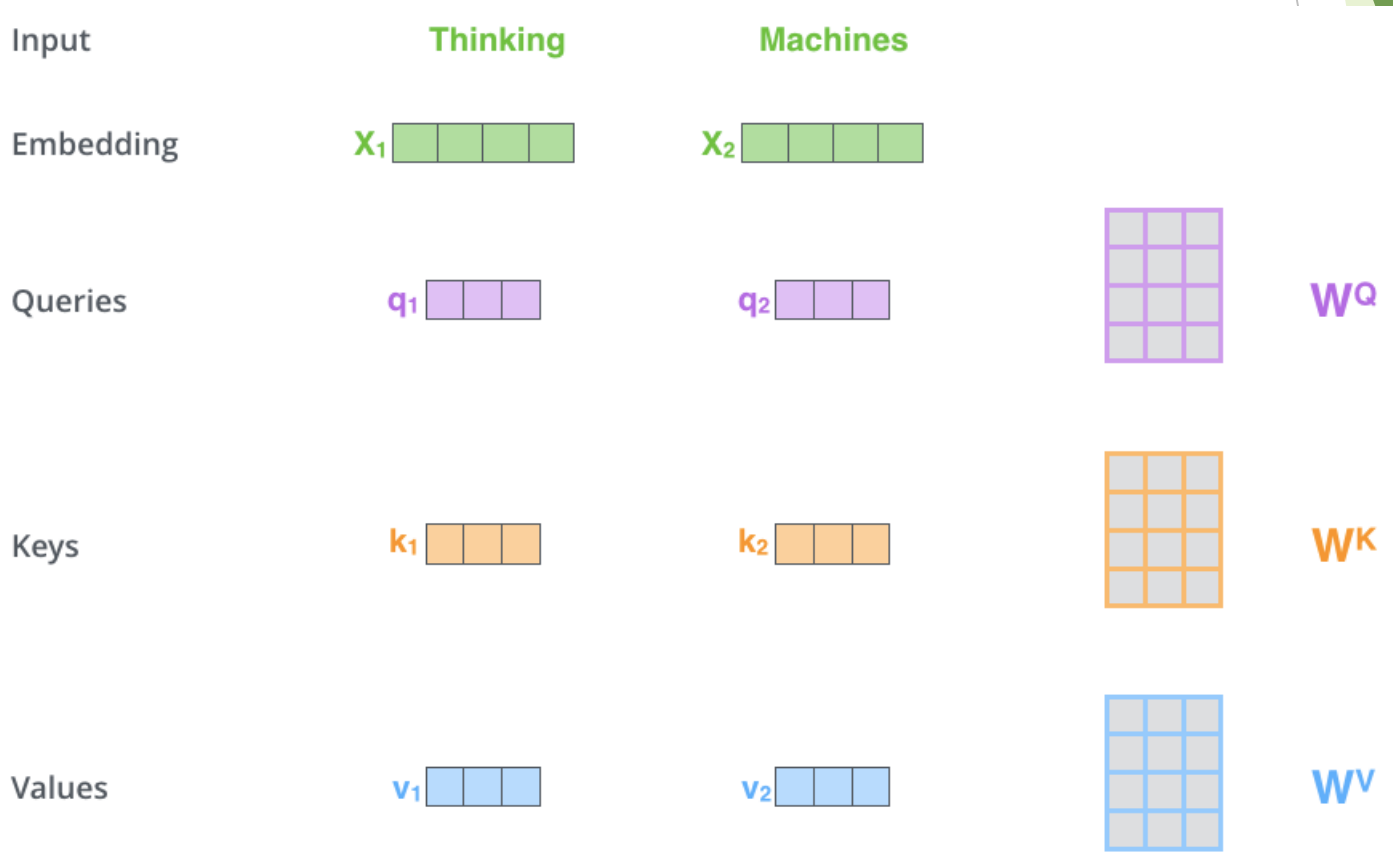
- 即对于每个输入的单词，在做了词嵌入之后，我们可以将其输入全连接网络，得到相应的查询、键与值向量。

  That is, for each input word, after doing word embedding, we can input it into a fully connected network to obtain the corresponding query, key, and value vectors.

架构初探
Architecture Exploration

- 该图展示了如何通过全连接网络计算相应的查询、键、值。

  This figure shows how to calculate the corresponding query, key, and value vectors through a fully connected network.



架构初探
Architecture Exploration

- 在有了查询、键与值向量之后，便可以按照既定的思路进行计算，得到在每一步的隐向量，在更高层的结构中使用。右图展示了相关的计算。

  After obtaining the query, key, and value vectors, we can follow the established method to calculate the hidden vectors at each step and use them in higher-level structures. The figure on the right shows the relevant calculation.

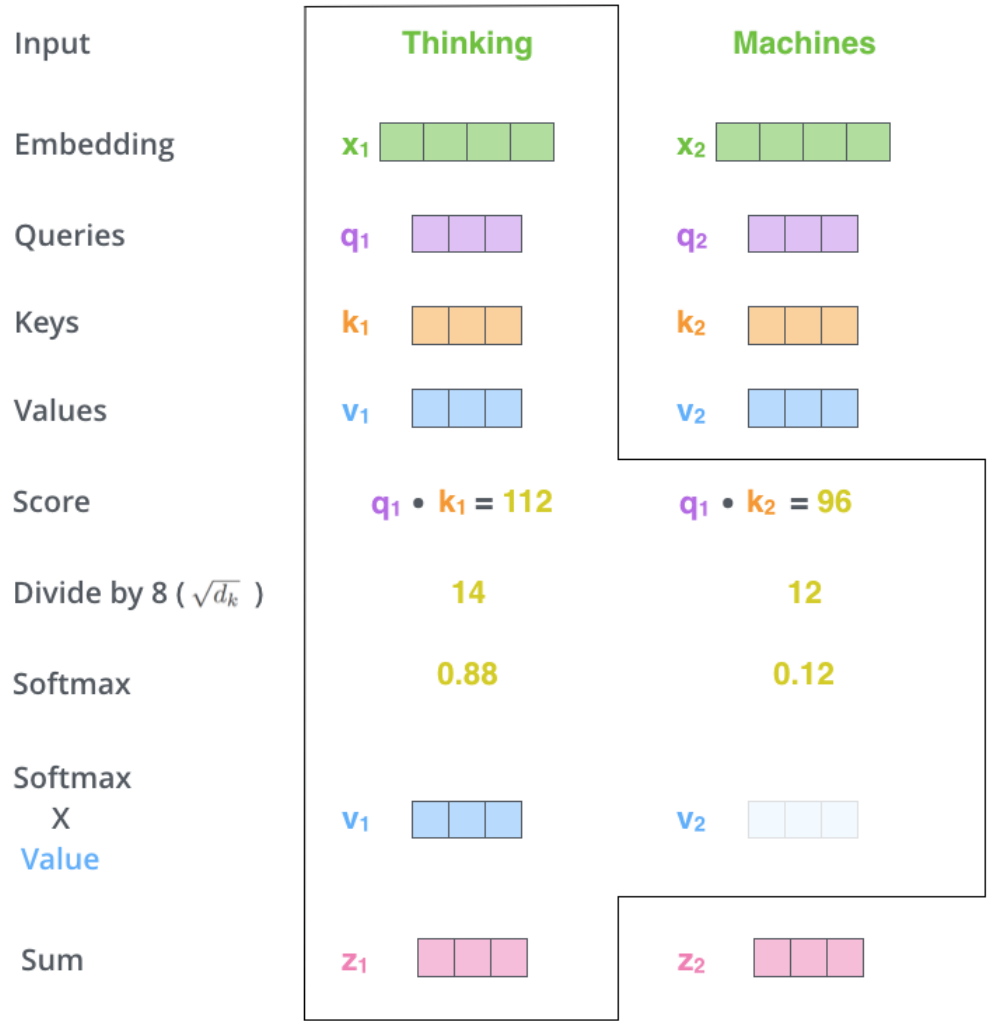
- 而用于计算这些向量的的全连接网络，我们只需要定义这些网络的结构，然后相应的权重，根据语言任务，由语料库根据端到端训练即可。最基本的任务即语言建模任务，基于大体量的语料库，进行单词预测。

  To calculate these vectors using a fully connected network, we only need to define the structure of these networks and the corresponding weights, which can be trained end-to-end by the corpus according to the language task. The most basic task is the language modeling task, which predicts words based on a large corpus.

架构初探
Architecture Exploration

- 右图展示了如何利用查询、键与值向量计算隐向量。

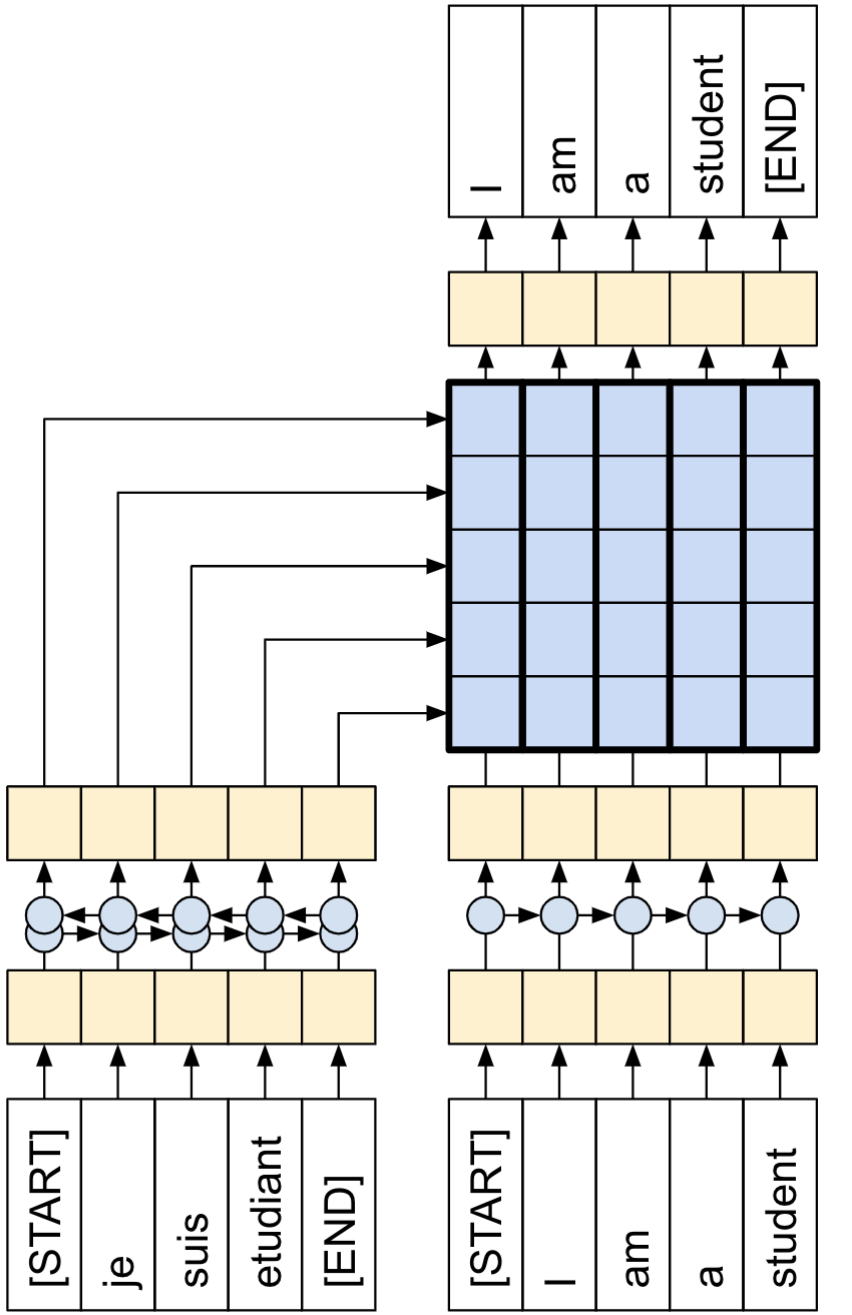
  The figure on the right shows how to use the query, key, and value vectors to calculate the hidden vectors.



架构初探
Architecture Exploration

- 一般地说，在用注意力或自注意力机制解决了传统的循环神经网络存在的并行性差与局部耦合性高问题之后，则可以认为在整体架构上，序列对序列模型，与Transformer模型，没有特别实质的差别，其均属于编码与解码架构。

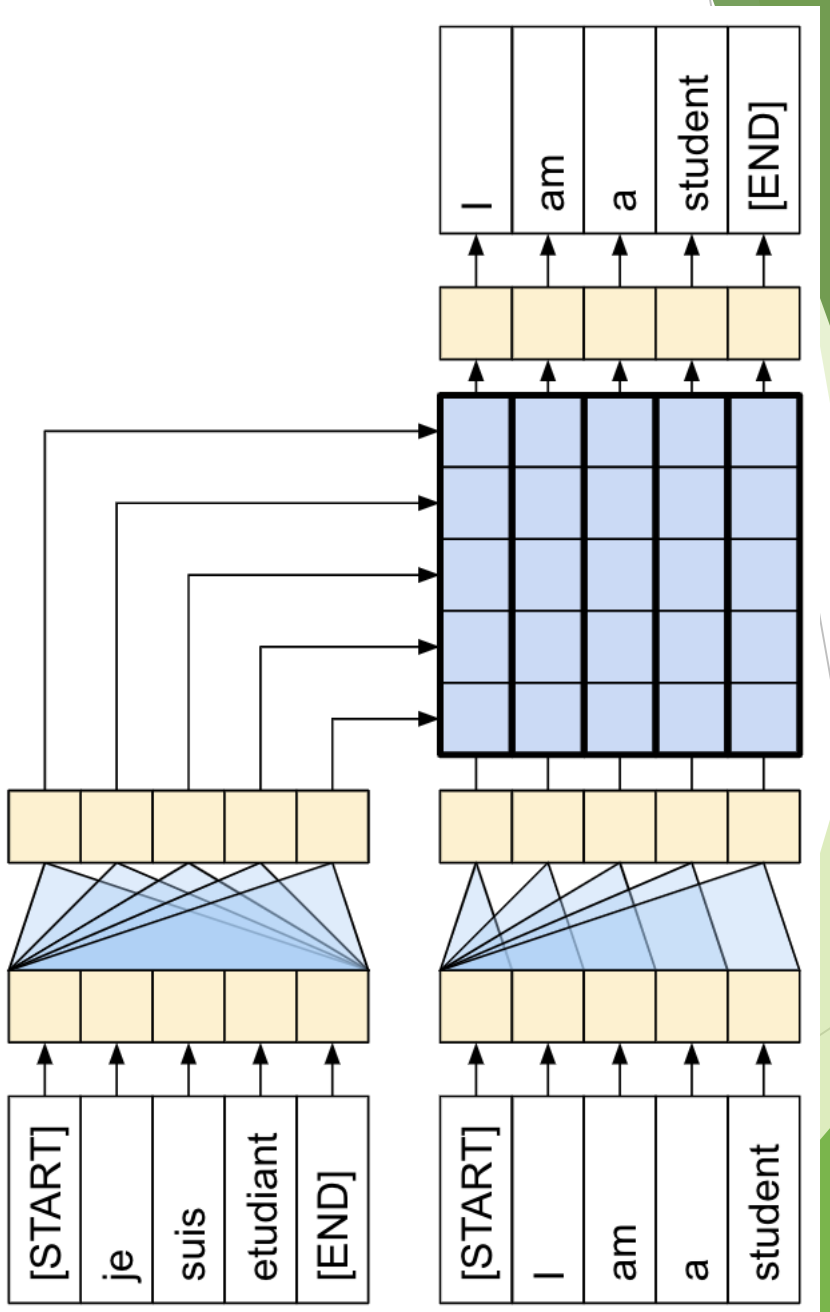
  In general, after using the attention or self-attention mechanism to solve the problem of poor parallelism and high local coupling in traditional recurrent neural networks, it can be considered that there is no significant difference in the overall architecture between sequence-to-sequence models and Transformer models. They are all encoding and decoding architectures.



架构初探
Architecture Exploration

- 上页右图与本页右图，分别展示了基于循环神经网络，与基于自注意力，针对语言翻译任务的计算范式的对比：

  The figure on the right of the previous page and the figure on the right of this page respectively show the comparison of the computing paradigm for language translation tasks based on recurrent neural networks and self-attention mechanisms.



架构对比
Architecture Comparison

- 在某些自然语言处理问题如语言生成，基于神经网络的语言模型，特别是Transformer，取得了比基于规则与基于统计的方法的更好的结果，这可能会引发人们比较同样基于神经网络的GAN与Transformer的区别与联系。
- 尽管GAN与Transformer在训练上，粗略地说，都可以认为是基于概率的游戏，但从整体上，差别还是比较大的。尽管从整体上看，无论是GAN还是Transformer都主要包含两个子网络，但GAN是生成器与鉴别器的博弈，需要两个损失函数；而Transformer则是编码器与解码器的合作，一个损失函数就好。
- 同时，在学习上，GAN要求的是群体的概率分布一致，因此，如果batch size过小，如单个样本，这样的训练对GAN没有意义。而对于Transformer来说，其基础任务是基于语料库的预测（无论语料库打不打标），单个样本除了训练不太稳定之外，并不会实质性的问题。
- 此外，除非利用更为高级的技巧，基于初始向量，GAN的生成与初始向量的相关性可能不大；而Transformer应用于语言生成任务时，虽然不能完全定向，但生成的内容网初始向量往往是相关的。

架构对比
Architecture Comparison

- In some natural language processing tasks such as language generation, neural network-based language models, especially Transformers, have achieved better results than rule-based and statistical methods. This may lead to people comparing the differences and relationships between GANs and Transformers, both of which are based on neural networks.
- Although GANs and Transformers can be roughly regarded as probability-based games in training, there are still significant differences overall. Although both GANs and Transformers mainly consist of two sub-networks overall, GANs require a game between generator and discriminator, with two loss functions, while Transformers cooperate between encoder and decoder with only one loss function.
- In learning, GANs require consistency of the distribution of the population probabilities. Therefore, training is meaningless for GANs if the batch size is too small. For Transformers, the basic task is prediction based on a corpus (whether labeled or not), and a single sample does not present a significant problem except for unstable training.
- In addition, unless more advanced techniques are used, the correlation between the initial vector and the generated result of GAN may not be significant, while in the case of using Transformer for language generation tasks, although it is not completely directed, the generated content is often related to the initial vector.