**基于Transformer的机器翻译算法研究**

*Abstract*

机器翻译领域中占据主导地位的深度学习架构主要是Sequence-to-sequence架构，风靡一时的Transformer架构也是如此，但是其采取的Multi-head Attention并剔除以往CNN、RNN的结构导致其在模型的泛化能力、语料理解能力上更胜一筹。本文在实现Transformer模型和剖析各个部分算法原理的同时，选取了Tatoeba Project精选的English作为源语言，Deutsch、French、Chinese作为目标语言的双语句子对数据集进行实验，最终得到目标语言为Deutsch的BLEU score为49.67，French为52.07，Chinese为31.25。

1. Introduction

随着互联网快速发展，人们聊天不再局限于国内，各种社交软件的诞生使得与外国友人聊天的机会越来越多，一定程度上促进了对异国风情、文化的兴趣和学习。对于某国语言精通者，交流和学习并不是一件困难的事情；但是对于普通人来说不同语种可能导致传递信息时产生歧义，因此如Google翻译和Microsoft翻译等翻译软件逐渐问世，解决了多数人不同语言的沟通问题。早期这些翻译软件存在一定通病，如单纯翻译字符或词语，并没有考虑上下文、语序、时态等问题，这就导致了翻译结果与理想产生了一定程度的偏差。由此，科学家们希望借助计算机来解决这些问题，机器翻译的概念逐渐浮现在人们的眼中。

自然语言处理(NLP)是机器学习中深度学习分支的一个重要分支，旨在让计算机理解、生成和处理人类生成的自然语言，机器翻译属于NLP的一个重要应用，旨在自动将源语言文本翻译成另一种目标语言文本的技术。其研究历史最早可以追溯到20世纪30年代。1933年，法国科学家G.B.阿尔楚尼提出了使用机器来完成自然语言翻译任务的想法，但这时候并没有正式确定机器翻译的思想。1946年，世界上第一台现代电子计算机ENIAC诞生。随后不久，信息论的先驱、美国科学家Warren Weaver于1947年提出了计算机进行语言自动翻译的想法，紧接着在1949年发表《翻译备忘录》，里面正式提出机器翻译的概念和思想，后续学者们纷纷加入讨论和研究中。1954年，美国乔治敦大学在IBM公司协同下，使用IBM-701计算机首次完成了英语转俄语的机器翻译试验。这次实验中IBM-701计算机有史以来第一次自动将60个俄语句子翻译成了英语，但是没人提到这些翻译得到的样本是经过人工多重挑选和测试的，从而排除了歧义性。正当机器翻译的话题热火朝天时，1964年美国科学院成立了Auto Language Processing Advisory Committee，其经过两年的研究，于1966年11月公布了一份ALPAC报告，里面明确且全面否定了机器翻译的可行性，因此各大公司停止了机器翻译研究的资金支持，从而开始了大约11年的受挫时间，各公司的机器翻译项目数量锐减。

1970年中后期，计算机技术和语言学的快速发展，社会信息服务的需求提高，之前的限制条件被逐一打破，促进了机器翻译的研究复苏。随着计算机相关领域的发展，20世纪末，IBM公司的**Brown *et al*., 1993**基于统计学首次提出了基于词对齐的翻译模型，这是一个重要的里程碑，标志着现代机器翻译方法的诞生。Och, 2003基于机器学习思想提出对数线性模型及其权重训练方法，提出了基于短语的翻译模型和最小错误率训练方法，与**Koehn *et al.,* 2003**发表的论文标志着机器翻译的真正崛起。2006年，Google翻译作为一个免费服务正式发布，拥有连接互联网设备的人们纷纷享受到不同语言快速翻译的便利性，也为Google公司提供了更多翻译数据，便于展开更多的机器翻译研究工作。随着数据量的增加，出现许许多多的生僻词、超长句、特殊单词等传统机器学习算法无法较好解决的问题，如果增加算法的步骤和复杂度，计算机无法在短期内能够完成权重的训练。随着计算机硬件的发展，显卡GPU计算技术逐渐优化改良，这很好的解决了权重训练时间长的问题，学者们在算法、模型架构上能够更大胆地增加计算步骤以优化传统机器学习算法的缺陷。

**Kalchbrenner and Blunsom, 2013**提出了Recurrent Continuous Translation Models，这是一种用于机器翻译的新型End-to-end Encoder-Decoder Architecture，它使用卷积神经网络（CNN）将给定源文本编码为连续向量，紧接着使用循环神经网络（RNN）作为解码器，将状态向量转换为目标语言，这是一个重要的研究，后续出现的各种架构都是基于Encoder-Decoder Architecture的改良，如**Sutskever, 2014**提出的Sequence to sequence learning with neural networks，该网络架构与前者不同，其Encoder和Decoder均采用RNN结构的LSTM，并且实验表示该架构上线Google翻译后，翻译的质量已经超越了部分人类。**Bahdanau, 2015**等人提出在Encoder-Decoder结构中加入对其的Attention权重，**Vaswani *et al.*, 2017**基于Attention权重的思想发表了一篇撼动机器翻译、乃至整个智能技术领域的论文，该论文中提出的Encoder-Decoder结构Sequence-to-sequence模型——Transformer，与以往的Sequence-to-sequence模型不同，该模型抛弃了一切的CNN和RNN，整个网络结构几乎都是由Attention机制组成，主要创新点在于Multi-Head、Self-Attention，这些设计使得该模型训练的收敛时间大大减少，并且实验结果表明，它的性能和鲁棒性远超当时所有模型架构。

本文将逐一剖析Transformer模型Encoder-Decoder架构，介绍和复现原文Transformer模型的各处细节和训练策略，在其英语转德语和英语转法语的基础上，新增英语转简体中文的数据集，验证复现的性能。

1. Related studies
2. *Model Architecture*

Transformer和大多数competitive neural sequence transduction models相同，均为Encoder-Decoder架构，其整体架构如Fig. 1.

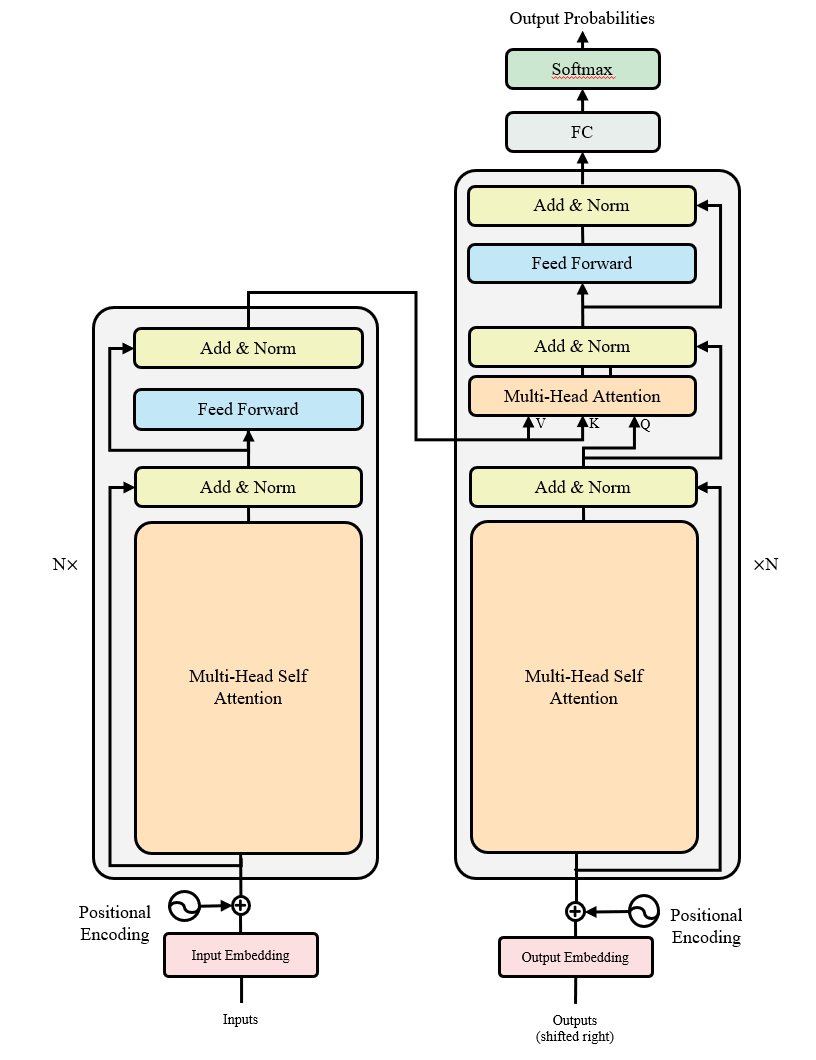


Figure 1. The Transformer model architecture.

其中，左侧为Encoder，左侧为Decoder，每个Encoder/Decoder都由个Encoder/Decoder blocks“夹心式”组成。青色方块FC表示Fully Connected Layer，也称Linear Layer；紫色小方块MatMul表示Matrix Multiplication；黄色小方块Concat表示矩阵连接；黄色方块Add & Norm表示残差连接和Layer Normalization (LN)。

1. *Encoder/Decoder block*

在整体架构上，encoder和decoder均是采取“夹心式“设计的个blocks组成。假设encoder由个encoder blocks组成，形状（表示source语言的句子长度，表示维度）的词向量Inputs经过第一个encoder block前向计算后，得到的outputs与inputs是相同形状的，即均为形状，详细结构如Fig. 2。

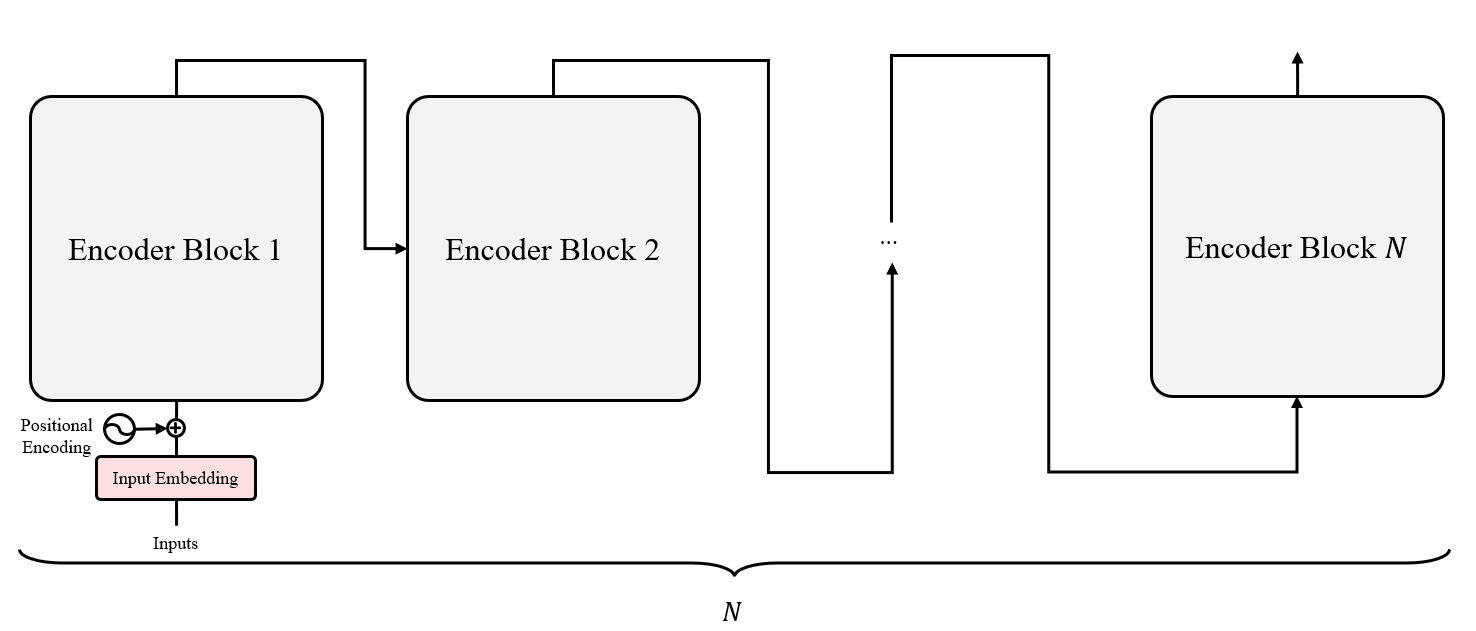


Figure 2. The Encoder forward calculation detailed architecture.

Decoder同理，但是其与Encoder block不同的是在Multi-head Self Attention (MHSA)后增加了一个MHA，用于接受Encoder block的输出和解码器内部的MHSA输出，从而完成注意力机制计算，详细结构如Fig. 3.

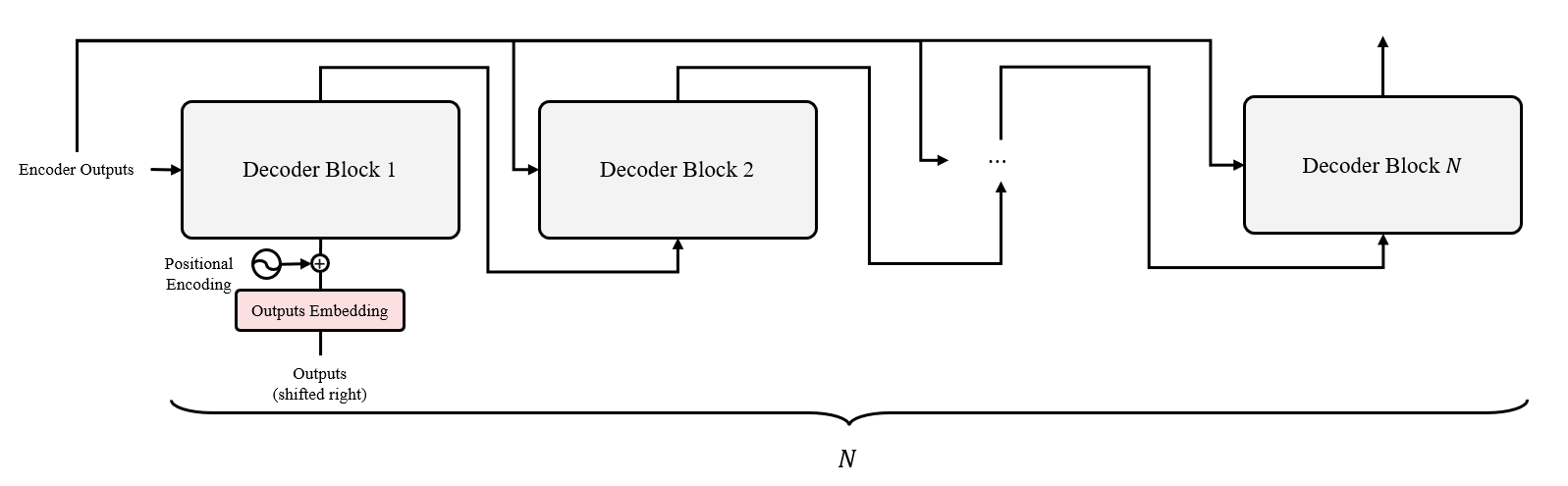


Figure 3. The Decoder forward calculation detailed architecture.

在训练过程中会要求输入形状的目标语言词向量进行训练，这里的shifted right表示target语言词向量的向右移动。由于数据预处理部分会为句子增加“<BOS>”、”<EOS>”标记句子起始和结束的特殊符号，在预测部分会根据第一个词”<BOS>”和输入source语言句子去预测下一个词即句子的真正第一个词。这个操作的目的是为了保证解码过程，每一个位置的输出都是基于前一个位置输入的信息，因此在训练过程需要对target语言句子在长度维度上向右移动一次。

1. *Multi-Head Attention*

上文提到的MHSA和MHA均是相同的计算过程，由于输入不同导致其计算的注意力矩阵具有不同的解释。原文中描述MHA由个Single Head Attention组成，将inputs拆分成了份，即每个头输入的词向量形状均是，根据FC线性映射从而实现并行加速计算，最后Concat得到与inputs相同的形状，其中单个头的注意力计算采用Scaled Dot-product Attention，MHA详细结构如Fig. 4.

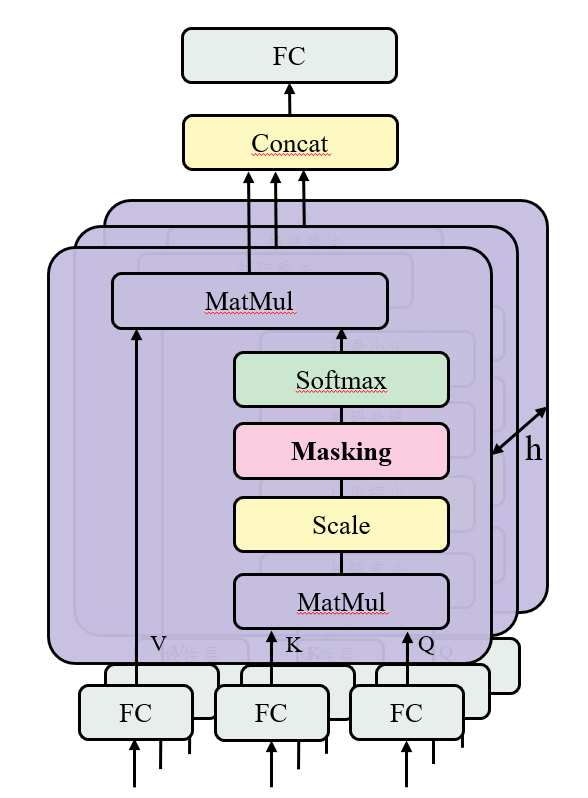


Figure 4. Multi-head Attention architecture.

在单个头的Scaled Dot-product Attention中，inputs经过三个FC层映射得到的分别表示“Query”、“Key”、“Value”，其中和的矩阵维度均为，的矩阵维度为。显而易见，MHSA的均来自inputs的复制，而MHA会出现差异，具体体现在Decoder block的MHSA的下一层MHA中，其和来自于Encoder outputs。计算公式如Eq. (1)：

根据Fig 4.和Eq. (1)，可以得知是是的方形矩阵。在encoder中和均是来源于inputs的线性映射，因此该方阵经过Scaled、Masking、Softmax等运算后可以表示inputs对自己的注意力权重矩阵；在decoder中Q来源于outputs，属于outputs对于inputs的一种软查询方式，使得预测当前位置的词语是基于上一个位置词语解码得到。最终该注意力矩阵与 Matmul后得到一个注意力“处理”后的词向量，一定程度上能够使得模型理解上下文关系，从而得到更优质的翻译结果。

在Attention的计算当中，涉及了一个至关重要的运算，便是Masking。在庞大的数据中，不可能保证每一条句子长短相同，因此在训练前会对数据进行padding操作，填充空白字符使得在当前批次下句子长度是相同的。而在Scaled Dot-product Attention中，计算注意力权重矩阵时，不希望空白字符对它自己产生更多“关注”，因此需要一个0-1二值化的Mask来“遮住”填充字符。Masking本质上是对填充字符位置的词向量加上一个无限小的数，因此在填充字符处对应的Mask值为True（即1），其余字符为False（即0），在True的位置上将会被累加一个无限小的数，源语言词向量的mask可视化如Fig. 5.

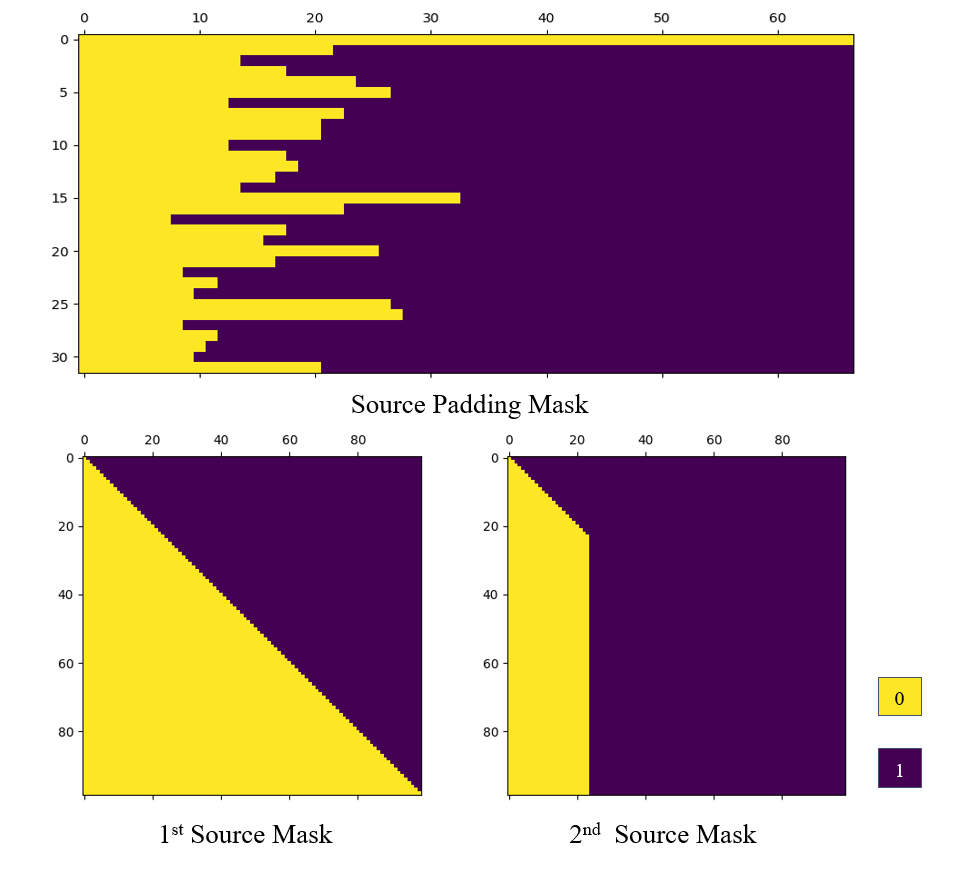


Figure 5. Visualzation of the generated source mask.

对于每一条句子，都会产生一个方阵mask。在批数据训练模式中，若再生成batch size数量的mask，无疑会增加GPU的运算消耗，因此根据Source/Target Padding Mask来调整最终的Source/Target Mask。

1. *Feed Forward Network*

在每一个Encoder/Decoder block中都会包含一个Feed Forward Network，由FC-ReLU-Dropout-FC组成，其计算如Eq. (2).

其中，，并且要远大于。原文中提到encoder or decoder blocks的可能不相同，本文后续实验将统一是相同的。Dropout是一种防止过拟合手段，即前向计算过程中随机剔除一定比例的连接，一定程度上能够减少模型参数过拟合。

1. *Add & Norm*

在每一个MHA或FFN后，都会进行一个Residual Add和Layer Normalization (LN)。Residual Add是**K. He *et al.,* 2015**发表的文章中提出模型结构中，将相同形状的inputs和outputs进行累加，一定程度上可以避免前向计算中特征的损失。

在神经网络中引入Normalization是一种有效减少模型训练时长的方法，其中最常用的便是Batch Normalization (BN)。BN能够在batch训练中，通过batch数据的均值和方差对数据进行当前batch下的约束，从而加快模型收敛速度。但是这种方式并不适用于句子长短不一的机器翻译模型中，不同句子的同一个单词位置的分布概率是不同的，并且每次step都要保存和计算Batch统计量，所以用BN来约束是没有意义的并且耗时又费力。由于机器翻译数据中Testing中可能会出现比Training最大长度还要长的句子，对于多余部分的词向量没有训练的参数可用，所以和CNN一样使用Batch Normalization是没有任何意义的。在RNN的相关研究中，**Ba, 2016**发现使用LN的效果比BN的要好，所以LN就变成了后续相关工作的default config. LN和BN的对比可视化如Fig. 6.

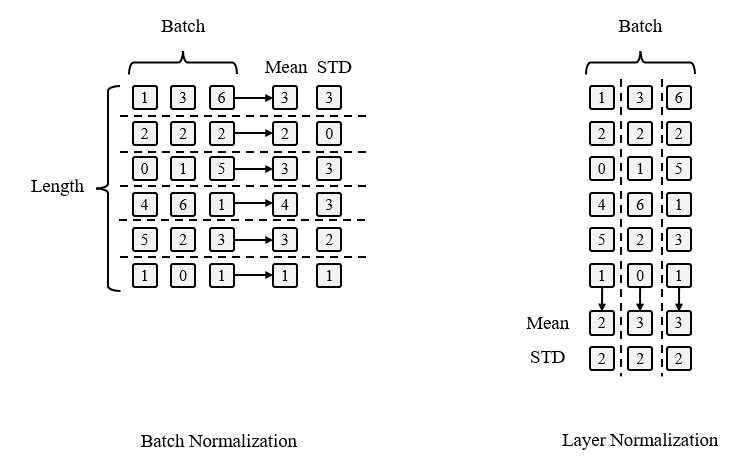


Figure 6. Batch Normalization vs Layer Normalization.

显而易见，BN是对batch纬度做归一化，针对不同样本的同一特征做操作。LN是对Hidden的维度做归一化，针对单个样本的不同特征做操作。由此可得，LN可以不受样本数量的限制，保证每个样本的分布是稳定的，因此非常适用于句子长短不一的机器翻译中。

令输入的表示MHA或FFN的运算结果，由此可以得到Add & Norm的计算结果如Eq. (3)。

1. *Positional Encoding*

一条句子是带有上下文关系的，但是经过Embedding后得到的词向量元素之间是相互独立的，没有办法能够得到具体的上下文信息。因此在进入Encoder/Decoder前，会进行Positional Encoding的累加，来使词向量得到位置特征。原文中提供了两种positional encoding的方法：

1. 根据人为经验公式得到，通常采取的是正余弦函数如下。

其中，表示矩阵中词索引，表示维度。即位置编码的每个维度对应于正弦曲线。波长形成从到的几何级数字。选择正余弦函数是因为假设它可以让模型轻松学习相对位置，根据经验公式可视化其位置编码矩阵如Fig. 7。

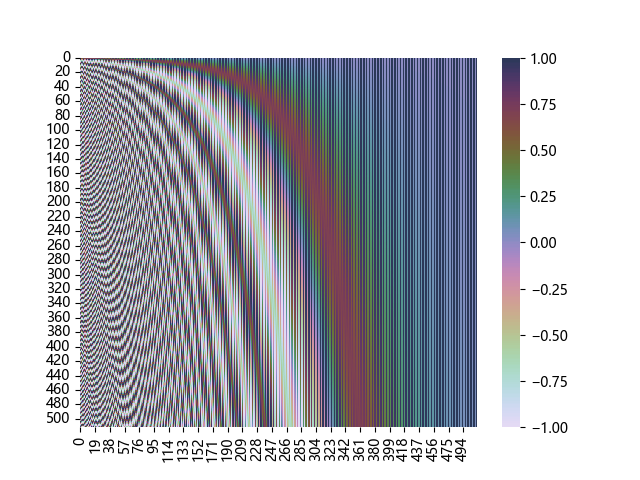


Figure 7. Positional Encoding matrix visualization

1. 使用Embedding层的权重与词向量做Matmul来生成最终输入的词向量。

第一种方法的好处是能够在短时间内使模型收敛，但是模型翻译质量的上限一定程度上取决于经验公式的选取；而第二种方法的好处是整体模型一旦收敛，其翻译质量的上限将远高于第一种，缺点是模型训练过程收敛困难，可能花很长一段时间都无法收敛。

1. *Loss function*

原文中并没有提及使用的损失函数，根据机器翻译任务的要求，最后输出结果的处理相当于是一个多分类任务，在多分类任务上诸多学者采取的是Cross Entropy损失函数，如Eq. (4)。

其中，表示类别总数量，在这里相当于目标语言的词典数量。由于在机器翻译数据中，对长短不一句子进行了Padding操作，因此填充字符是没必要计算损失的。

1. *Evaluation metric*

为了评估模型的好坏，因此需要一个评价指标。本文是采取了Accuracy和BLEU两种评价指标。使用Accuracy的原因是因为损失函数使用了Cross Entropy，因此希望添加多一个评价指标来衡量模型性能。

1. *Accuracy*

Accuracy是最简单且最直接的分类任务评价指标。假设一个positive和negative的二分类任务中，对于类别预测分类标签，与其对应的原始分类标签为，即 在的基础上预测成功，用True Positive (TP)表示。反之若，即 在的基础上预测失败，用False Negative (FN)表示。同理可得True Negative (TN)和False Positive (FP)，通常采取表格或矩阵的形式来表示TP、TN、FP、FN这四种数值，即混淆矩阵，如表1.

Table 1. Binary classification confusion matrix

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Predict Positive | TP | FP |
| Predict Negative | FN | TN |

由此可得Accuracy的公式如Eq. (5)。

对于多分类任务，将二分类应用在每一类上即可。

1. *BELU score*

BLEU是**Papineni, 2002**提出的用于机器翻译任务的评价指标。假设给定参考翻译文本reference是“there is a cat on the mat”，模型预测句子候选翻译文本candidate是”the the the the the the the“，那么计算BLEU分数的原始思想是对candidate的每个词在reference出现的次数作为分子，reference的词数量作为分母，求得上述reference和candidate的BLEU分数为7/7=1，显然这是不合理的。因此定义多元精度n-gram，称上述单个单词的准确率分数计算为uni-gram或1-gram，这种指标只能衡量翻译的充分性。如果将上述candidate拆成{“the the”, “the the”, …., “the the”}；reference拆成{“there is”, “is a”, “a cat”, “on the”, “the mat”}这种两个相邻词一组叫做bi-gram或2-gram，以此类推有三元组、四元组、多元组，再进行BLEU分数的计算就可以更全面的衡量机器翻译模型的质量。一般取1-4元组的BLEU分数，对于BLEU的n-gram计算公式如Eq. (6)。

其中，表示某一个n-gram在reference出现的个数；表示翻译文本的n-gram。在一般的机器翻译任务中，candidate和reference都是一一对应，因此最终对每个对照组的进行求和平均，即模型在当前数据集的分数。最后组合这些分数来综合衡量模型质量，即对个进行加权平均。

当候选翻译文本和参考翻译文本的长短不对等时，根据上述继续计算BLEU分数可能会导致较短的模型翻译句子得到较高的分数，因此原文定义了“best match length”：假如候选翻译长度大于或等于参考翻译文本的长度，就认为它满足最佳匹配长度，否则进行一个较短惩罚(Brevity Penalty, BP)，如Eq. (7)。

其中，表示reference的n-gram个数，表示candidate的n-gram个数。由此可得最终BLEU分数的计算如Eq. (8).

其中，表示当前n-gram的权重，定义，即加权平均。

1. Analysis, discussion, and future work
2. *Bilingual Sentence Pairs Dataset*

本文将采用Tatoeba Project中精选的双语句子对数据集，来验证Transformer模型的可行性。其中一共采取了三种双语数据：English-to-Deutsch、English-to-French、English-to-Chinese，详情如表2.

Table 2. Dataset details

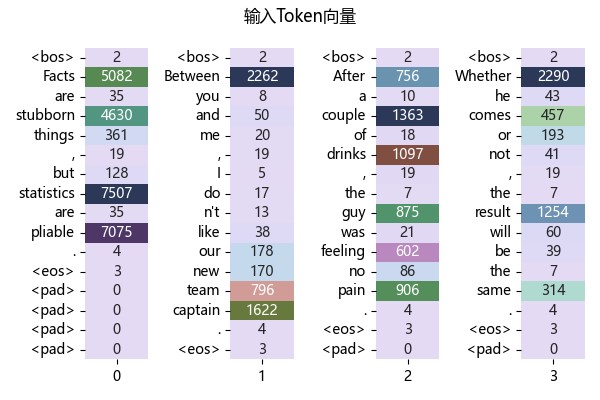
|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Sentences** | **Source vocabulary** | **Target vocabulary** |
| English-to-Deutsch | 255,817 | 18,973 | 41,042 |
| English-to-French | 197,463 | 17,473 | 28,283 |
| English-to-Chinese | 29,155 | 7,899 | 11,332 |

由于这些数据集没有划分Training、Validation、Testing，因此本文将对这些数据集进行划分，其中随机打乱后挑选100条作为Testing，1000条作为validation。

1. *Data preprocessing and post-processing*

模型训练前，对句子进行分词、索引化，索引化之前需要对句子插入起始和结束符号，以标记长短不一句子的起始和结束位置。本文采取的是”<BOS>”和”<EOS>”作为起始和结束符号，除此之外采取”<PAD>”作为填充符号，”<UNK>”作为词频数量少的替换符号。由于部分词在整个数据中出现次数较少的原因，类似于剔除数据异常值的操作，将词频极低的词语替换成”<UNK>”，防止模型训练过程收到生僻词的影响导致收敛时间过长。

在对句子进行词索引化时，先统计整个文本中全部词语和标点符号的频次，根据频次从大到小进行排序，根据位置将词语一一转换成索引。由于Transformer中采取的Encoder-Decoder结构，因此不需要对每一条句子都按照最大长度进行切片或填充，因此只需要在batch中按照其中最大长度进行填充即可，比如在batch size为4的情况下如Fig. 8.

  
Figure 8. Batch word index vector

其中，索引为1的句子的长度为16（包括起始和终止符号），因此在当前batch下以16为最大长度进行填充即可，也可以在数据预处理时加入人为设置的最大长度max\_len，当批次中最长句子的长度大于max\_len时，将按照max\_len对最长句子进行切片操作，此时不需要插入终止符号，其他句子进行填充即可。

模型训练结束后，测试模型时需要对源文本的词向量进行机器翻译，因此在翻译推理时，和Transformer前向计算时有一定差异，即对最后推理结果采取 greedy algorithm进行句子翻译。实际上只是选取输出概率矩阵中概率最高的位置作为词索引，称为greedy decode，伪代码如下。

|  |
| --- |
| Algorithm 1. Greedy Decode |
| Input：source token vector  1 Get the outputs of Encoder inference : ;  2 Initialize prediction sequence ;  3 for i = 1 to target max length do  4 generate square subsequent mask ;  5 Usefor Decoder inference and get the probability outputs: ;  6 Get the index of maximum probability position of dimension at position ;  7 Let the index obtained in the previous step be added to the end of ;  8 if is “<EOS>” do   9 break;  10 return  ; |

1. *Optimizer*

为了还原Transformer原文的实验，同样采取的Adam optimizer。

1. *Results*

在Tatoeba Project精选的双语句子对数据集中三种双语句子对数据进行实验，训练硬件为13th Gen Intel® Core™ i7-13700K、NVIDIA GeForce RTX 2060 SUPER 8GB，因此能设置的参数有限如表4.

Table 4. Hyperparameter details.

|  |  |
| --- | --- |
| **Hyperparameter** | **Value** |
| max\_len | 40 |
| N | 6 |
|  | 8 |
|  | 512 |
|  | 2048 |
| dropout | 0.1 |
|  | 0.0001 |

其中，N表示Encoder和Decoder的block数量，统一设置为6。表示MHA中head的数量。Dropout表示其在Add & Norm中剔除的比例，即在Residual Add时随机选择10%的部分不Add。表示Adam optimizer的初始学习率。使用上述超参数，使用三种双语句子对数据集训练100个周期（batch size is 64），Validation上得到最优结果如表5.

Table 5. Optimal results of the training process.

|  |  |  |
| --- | --- | --- |
| **Language** | **BLEU/%** | **Accuracy/%** |
| English-to-Deutsch | 49.67 | 78.90 |
| English-to-French | 52.07 | 80.38 |
| English-to-Chinese | 31.25 | 65.95 |

根据实验结果，English to French的效果最好BLEU score达到了52.07，其次English to Deutsch的49.67，最后是English to Chinese的31.25。French目标语言的BLEU score最高可能一部分原因是因为target language词典数量相较Deutsch而言较少，即生僻词部分较少。而Chinese数据量相较Deutsch和French偏少，且Chinese的词典特殊性较大，English to Chinese是word-to-character的模式，希望English-to-Chinese收敛可能得制定更详细的训练策略。其中这三种语言训练的损失函数和BLEU score变化如Fig. 9.

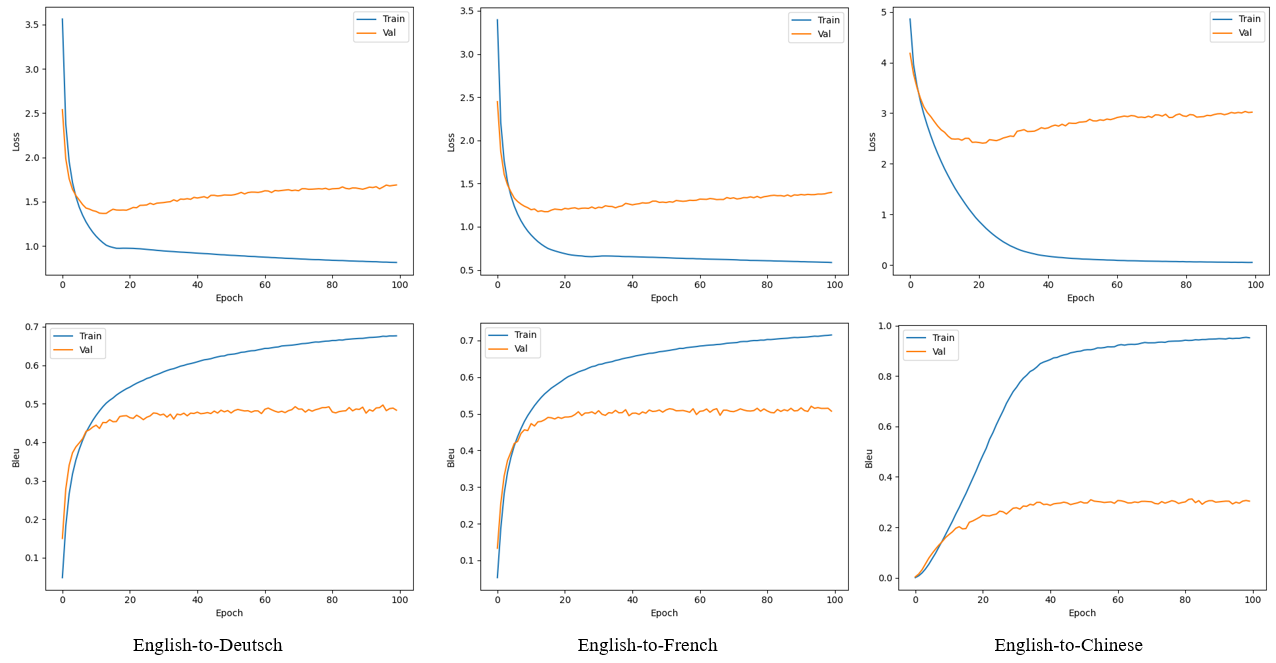


Figure 9. Loss function and BLEU score changes.

可以发现，Deutsch和French在20个epochs左右已经收敛，而Chinese需要在40个epochs左右才进行收敛，并且后续已经出现过拟合的情况，validation的损失逐渐上升。因此对于像Chinese这种特殊语种，需要做更多的工作。

1. *Advantages and Disadvantages*

本文对Transformer模型的复现和实验，一定程度上能够体现出该模型在机器翻译领域的重要性，开创了不使用CNN架构的神经网络，性能和鲁棒性比同类型中使用了CNN的架构更优秀，推理结果更加可靠。在Transformer的核心MHA中，根据不同输入来形成对词向量的一种“注意力”权重，使得输入词向量能够在一定程度上放大特征值，句子能够出现类似位置编码的上下文关系，因此模型更加关注整个句子的全局性和连贯性，不再是像CNN一样关注部分特征。

由于Transformer没有使用CNN等结构，因此在模型训练上参数量是巨大的，训练时间长是必然的。但是在模型推理过程，由于其庞大的结构即使是base架构推理速度仍然较慢。并且由于设备限制，无法完美地再现Transformer训练过程，即仅仅复现了base。因此后续希望能够在多GPU服务器上进行训练，以更接近其原文的实验复现。并且在English-to-Chinese的实验上并没有得到令人满意的结果，很大部分原因来源于其word-to-character的模式，导致识别效果并不是特别理想。

1. Conclusions

整体下来Transformer的模型复现，是使用了PyTorch深度学习框架，因此在描述向量形状时均基于PyTorch的style进行赘述。模型训练、验证部分是采用了单机单卡的训练方式，受限于设备只能对base架构进行training和validation。数据集上选用了Tatoeba Project中精心挑选的双语数据集，其中选取English-to-French、English-to-Deutsch是参考了原文作者实验的数据语料，English-to-Chinese是出于对English to Chinese的机器翻译任务探讨和研究所添加的语料。实验结果证明，即使是Transformer也未能很好的解决word-to-character的问题，还需要定制更详细的数据处理方案或者模型训练方案。

Reference

1. T. Poibeau, “The 1966 ALPAC report and its consequences,” in Machine Translation, MIT Press, pp.75-89, 2017
2. P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and et al., “The mathematics of statistical machine translation: Parameter estimation,” *Computational Linguistics,* vol. 19, no. 2, pp. 263-311, 1993.
3. P. Koehn, F. J. Och, and D. Marcu, “Statistical Phrase-Based Translation,” *In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 127-133, 2003.
4. F. J. Och, "Minimum Error Rate Training for Statistical Machine Translation," *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL, 2003.
5. N. Kalchbrenner and P. Blunsom, “Recurrent Continuous Translation Models,” *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700-1709, 2013.
6. I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14)*, pp. 3104-3112, 2014.
7. D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *presented at 3rd International Conference on Learning Representations*, 2015.
8. A. Vaswani, et al., "Attention is all you need," Advances in neural information processing systems, 30, 2017.
9. K. He, X. Zhang, S. Ren, and et al., “Deep residual learning for image recognition,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2017.
10. J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
11. K. Papineni, S. Roukos, T. Ward, and et al., “BLEU: a method for automatic evaluation of machine translation,” *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. pp. 311-318, 2002.