机器翻译与自然语言处理机器翻译课程报告

计硕2004-2001831-王裕

概念

机器翻译

机器翻译是自动将一种自然语言翻译为另一种自然语言的技术。随着深度学习 的兴起，当前最流行的机器翻译技术采用的是基于神经网络的编码器-解码器框架， 端到端地学习整个句子翻译过程，也被称为神经机器翻译(Neural Machine Translation, NMT)。

自然语言处理

自然语言处理(Natural Language Processing, NLP)以语言为对象、利用计算机技

术来分析、理解和处理自然语言的一门学科,即把计算机作为语言研究的强大工具，在计算机的支持下对语言信息进行定量化的研究,并提供可供人与计算机之间能共同使用的语言描写。自然语言处理主要应用于机器翻译、舆情监测、自动摘要、观点提取、文本分类、问题回答、文本语义对比、语音识别、中文OCR等方面。

基于统计的机器翻译及自然语言处理

统计机器翻译根据建模单元的不同，可以划分为:基于词[2]、基于短语[3]以 及基于句法[4-7]的统计机器翻译模型，相应的模型表达能力也依次递增。

基于词的统计机器翻译

最早的统计机器翻译模型是在 20 世纪 90 年代由 IBM 的 Brown 等人提出[2]。 通过将翻译任务建模为噪声信道模型，即认为目标语句子在经过一个噪声信道作 用后而成为源语句子，从而利用贝叶斯公式得到统计机器翻译的 基本方程式:

$$Pr(Y|X) = \frac{Pr(Y)Pr(X|Y)}{Pr(X)}$$

其中Pr(Y)是目标语语言模型，Pr($X|Y$)是翻译模型。对于如何建模Pr($X|Y$)， Brown 等人提出了以词为单位的 5 种方法进行建模，也被称为 IBM 模型 1-5，并 使用期望最大化算法(EM)进行参数估计，从而形成了基于词的统计机器翻译 模型。 在 IBM 模型 1-5 中，作者依次考虑了词汇之间的互译概率、在词对齐时同 时考虑词所在的位置与词的内容、一个词翻译为多个词、在翻译过程中词的相对 位置、对于不可能出现的对齐给出非零的概率。 然而，在实际应用过程中，基于词的统计机器翻译系统，由于受限于以词为 建模单元，对上下文感知的能力较弱，难以处理词义消歧问题。例如，“bank”应 该翻译为“银行”还是“河岸”在没有上下文信息的情况下是难以选择的。与此 同时，基于词的翻译系统也难以处理成语、歇后语等多词所组成的特定表达的翻 译，即这些内容有自身独特的语义，无法通过逐词译文的拼接进行翻译.

基于短语的统计机器翻译

不同于基于词的统计机器翻译模型，在基于短语的统计机器翻译中则是以短 语作为翻译的基本单位。值得注意的是，这里的短语定义并不是严格地语言学范 畴上的短语，而是由连续词汇所组成的词串[3]。由于对短语不做语法上的约束， 因此能够更

容易地基于词对齐信息从平行语料中进行自动抽取。相比于基于词的 统计机器翻译，以短语为基本翻译单元能够有效感知到局部上下文，翻译性能因 此得到大幅提升。基于短语的统计机器翻译主要包含三个步骤:短语切分、短 语调序以及短语翻译。

基于句法的统计机器翻译

基于短语的统计机器翻译虽然受益于局部上下文的引入，但也受限于连续短 语的结构，尤其在处理全局范围上的调序问题上存在先天缺陷。此处展示一个非 连续短语翻译的例子:源语是“召开了一次关于...的会议”，而译文“hold a meeting on ...”。为了解决上述问题，研究人员逐渐将目光转向句法结构知识。根据所使 用的句法是否包含语言学知识，又可将基于句法的统计机器翻译系统划分为:基 于形式化语法以及基于语言学语法[145,146]。 在基于形式化语法的统计机器翻译模型中，典型代表为 Chiang 等人提出的 基于层次短语的统计机器翻译[4]。该翻译模型基于带权重的同步上下文无关文法。类似短语翻译规则抽取，层次短语的翻译规则也 可以从平行数据中无监督地自动抽取，而不依赖任何句法分析结果。 而在基于语言学语法的统计机器翻译模型中，主要采用两种句法分析的形式， 分别为基于短语结构的句法树，以及基于依存分析的句法树。所使用的语言学语 法可以作用于源语端、目标语端或者两端均使用，从而形成基于串-树[5]、树-串[6] 以及树-树[7]的统计机器翻译模型。 尽管基于语言学语法的统计机器翻译模型在理论上能够更好地对语义进行 建模，但在实际应用中由于需要事先提供高性能的句法分析器，对于大多数语种 而言是不可行的。同时，基于句法的翻译系统计算复杂度更高，尚不适合大规模 的商业使用。

基于神经网络的机器翻译及自然语言处理

基于神经网络的深度学习技术由于异常优异的性能，在计算机视 觉、语音等领域开始逐渐爆发。受此影响，机器翻译领域开始引入基于神经网络 的建模方法。起初，神经网络被作为统计机器翻译的组件开始发挥效用，如作为 双语语言模型[11]，调序模型[12]，翻译模型[13]，但受制于统计机器翻译框架，上述 的统计机器翻译的痛点依然无法解决。直到 Sutskever 等人[14]以及 Bahdanau 等人 [15]率先提出完全采用神经网络的翻译模型，而不依赖于统计机器翻译的对数线性 模型，神经机器翻译开始崭露头角。

编码器-解码器框架

神经机器翻译通常采用编码器-解码器框架对翻译过程进行建模，其中编码 器和解码器均为独立的一个神经网络。从功能上来看， 编码器网络负责将待翻译句子抽取为相应的语义表示，解码器网络负责根据源语 的语义表示生成对应的译文。

注意力机制

基于编码器-解码器描述的神经机器翻译系统虽然在流畅度上优于统计机器翻译，但整体的翻译性能仍处下风(例如 BLEU 值)。其中的关键问题就是神经机器翻 译常常出现多译及漏译现象，尤其是在翻译长句子时更加明显。造成这一问题的 主要原因是在之前介绍的解码器-解码器框架中，编码器将输入句子统一编码为 一个固定长度的向量作为该句子的语义表征，即总结向量，而忽略了句子的长度。 很明显，句子越长通常包含的信息也越多，但总结向量的维度是固定的，因而导 致源语信息的损失。Bahdanau 等人开创性地提出注意力机制以缓解基于总结向量方式的神经机 器翻译的问题[15]，已成为当今神经机器翻译的标配方法。注意力机制的核心思想 是将编码结果表示为与源语序列长度相同的一组隐藏状态，而非一个向量。而后解码器可以通过基于内容的寻址方式提取上下文表示。

基于循环神经网络的神经编码器

循环神经网络(RNN)能够有效处理任意长度的序列，在语言模型任务上取 得了巨大成功[17]，十分适合对变长的自然语言句子进行建模。因此，最早的神经 机器翻译系统也是基于循环神经网络结构[14,15]。然而，在实际应用时，通常不直接使用公式(2.16)所描述的原生 RNN 的结构， 因为此种形式的 RNN 在应用时容易发生梯度消失或梯度爆炸问题，尤其是处理 较长序列时[18]。对于梯度爆炸问题，可以采用梯度裁剪(gradient clipping)的方法缓解。该 方法的主旨是当梯度的范数大于某一阈值时便对其进行缩放，然后再使用梯度下 降方法进行参数更新。 而对于梯度消失问题，则可以通过使用带有门控机制的 RNN 的变种来缓解。 常见的变种包括:长短时记忆网络19或门循环单元13。相比 于 LSTM，GRU 是其一个简化版本，在性能相近的情况下，GRU 使用更少的参 数量，并且计算速度更快。因此，这里以 GRU 为例来介绍基于 GRU 的循环神 经机器翻译模型[15]，不同于原生GRU 额外使用了两个门，即更新门 和重置门，来控制保留多少过去的信息，以及更新多少当前的信息。相比于原生 RNN，GRU 通过使用门控单元，使得不同时刻之间存在直连的路径，在反向传播时梯度更容易流通，从 而缓解了梯度消失的问题。而基于循环神经网络(GRU)的编码器，是由两个 RNN 层组成的双向编码 器。其中一个 RNN 从左至右对输入序列进行编码，另一个 RNN 则从右至左进 行编码。然后再将两者的编码表示按照相应位置拼接在一起，从而使得输入序列 的每个词都能感知到全部上下文。而解码器端，除了使用一个 GRU 来对已生成的目标语序列进行编码表示外， 还额外引入了注意力机制已从H中提取更准确的上下文信息ci，并 通过输出层进行最终的译文词汇预测。

基于自注意力机制的神经编码器

在之前的研究中、神经机器翻译模型无一例外地或者基于循环神经网络、或 者基于

卷积神经网络，然而 Vaswani 等人提出了一种完全基于自注意力机制的神 经机器翻译模型——Transformer。 所谓自注意力机制，也称为内部注意力(Intra-Attention)，是在句子序列内 部进行注意力计算，而2.1.2.2节介绍的注意力机制作用于两个句子之间(例如， 在源语和目标语之间进行注意力计算)。自注意力最显著的优点是能够以$O(1)$的 代价实现序列中任意两点间的信息交互，而与之对比的循环神经网络需要$O(n)$， 卷积神经网络需要$O(n/k)$，其中n是序列长度，k是卷积核大小。越小的交互代价 意味着越容易学到远距离的依赖关系，这在机器翻译任务中是十分重要的。Transformer 能够出色地在序列上并行计算，因此训练速度相比基于循环神 经网络的系统大幅提高，并实现了顶级地翻译性能，成为当今神经机器翻译模型 的最主流模型。

课程项目

利用基于lstm的语言模型实现了一个唐诗生成的题目,用数据集做一些简单处理后对语言模型进行训练，输入每句的第一个字，以藏头诗的方式预测诗句的生成，也可以不指定输入，随机生成。由于时间有限，所以模型结构简单，最终效果不是特别理想，仅作为课后的入门练手项目。

数据处理：

sentence_parse： 去掉括号中的部分，去掉数字，处理两个句号为1个句号
pad_sequences： 对长的诗句截断，对短的诗句补全
get_data : 构建字典返回保存数据
模型训练：
optimizer用Adam,参数由config文件配置

例子：
月日無人事，山中不可知。的人無事事，不得一年年。
床山水上西山水，天上天中不可知。疑君不得無人事，不得人人不可知。
天中天子，三里無年兮兮兮有神，天兮天子兮天中兮天。西天兮，有天生，。有人不得。不見君子。天子無事兮。君不得兮兮兮兮

```
工具：
    python 3.6.7
    pytorch 1.4.0+cu100
```

```
    pytorch 1.4.0+cu100
```

配置信息：

```
    data_path = "/home/ljy/proj/poetry-generation/chinese-
poetry/json/"
    category = "poet.tang"
    author = None
    constrain = None
    poetry_max_len = 125
    sample_max_len = poetry_max_len-1
    processed_data_path = "data/tang.npz"
    word_dict_path = 'wordDic'
    model_path = 'model/tang_200.pth'
    batch_size = 128
    epoch_num = 201
    embedding_dim = 256
    hidden_dim = 256
    layer_num = 2  # rnn的层数
    lr = 0.01
    weight_decay = 1e-4
    plot_every = 2
    debug_file = '/tmp/debugp'
    env = 'poetry'
    use_gpu = True
    max_gen_len = 200  # 生成诗歌最长长度
```