

单位代码: 10293 密 级:

南京邮电大学

专 业 学 位 硕 士 论 文



论文题目: 基于深度神经网络的对联生成系统的
研究与实现

学 号 1217012225

姓 名 张江

导 师 王玉峰

专业学位类别 工程硕士

类 型 全 日 制

专业（领域） 电子与通信工程

论文提交日期 二零二零年四月

Research and implementation of Chinese Couplet Generation System Based on Deep Neural Network

Thesis Submitted to Nanjing University of Posts and
Telecommunications for the Degree of
Master of Engineering



By

Zhang Jiang

Supervisor: Prof. Yufeng Wang

April 2020

南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生学号： 1217012225 研究生签名： 张江 日期： 2020年5月6日

南京邮电大学学位论文使用授权声明

本人承诺所呈交的学位论文不涉及任何国家秘密，本人及导师为本论文的涉密责任并列第一责任人。

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

非国家秘密类涉密学位论文在解密后适用本授权书。

研究生签名： 张江 导师签名： 张江 日期： 2020年5月6日

摘要

对联是中华优秀传统文化中一种独特的艺术形式，其要求上下联之间长度结构相同、语义相似、对仗工整和平仄和谐，体现了中华语言的美感，在众多节日场合承担了表达情绪、烘托气氛的重要作用，备受人们喜爱。但是，正是由于对联严格的格式和内容要求，创作对联对普通人来说是一项较为困难的任务。因此，使用计算机进行对联的自动生成让大众都有创作对联的机会。但是，由于自然语言的含义和语境十分复杂，即使十分简单的语言，电脑也无法准确理解，因此，自动生成对联对计算机来说也是一项富有挑战性的工作。

针对已有的对联生成方案中没有考虑对联词语的词性信息和未登录词及低频词处理等问题，论文基于深度神经网络的相关技术，对基于注意力机制搭建的 Transformer 模型进行了改进，并基于改进的模型实现了中文对联自动生成系统。论文使用机器翻译中流行的评价标准 BLEU、困惑度(Perplexity)和人工评价三种方式对本文提出改进的对联生成模型进行评价，BLEU 评分越高、困惑度(Perplexity)评分越低，说明模型的性能越好。主要贡献如下：

一、将基于 Transformer 的对联生成模型、已有的研究工作中使用的基于编码-解码框架的对联生成模型及其改进形式，结合注意力机制的编码-解码框架的对联生成模型进行实验比较，实验结果证实了注意力机制在对联生成任务中的有效性。将基于 Transformer 的模型作为本文的基线模型，与本文提出的三种改进策略进行比较。

二、为了充分利用中文的语言学知识，将对联的词性信息引入模型。对联要求对仗工整，其上下联对应位置的词语的词性一般是相同的。已有的研究工作没有显式地考虑这个约束条件。本文使用了一种融合词性特征信息的词向量训练方法，将进行了词性标注后的语料和原语料分开进行词向量训练，再将得到的词性向量和词向量以一定的方式融合，使用融合后的词向量进行神经网络的训练。融合词性信息特征后的对联生成模型和基准模型相比，在测试集上的 BLEU 评分提高了 0.059，困惑度降低了 2.51，模型的性能获得了一定的提升；

三、为了减轻模型计算过程中词典的未登录词和低频词对模型造成的影响，提出了一种低频词处理方法。针对模型训练及预测过程中遇到的未登录词及低频词问题，论文基于词向量的相似度计算方法，使用与未登录词和低频词相似度较高的高频词对其进行替换，设计了一种加入未登录词和低频词处理的对联生成模型。改进后的模型和基准模型相比，目标词典的规模减小了约 16%，系统在测试集上的 BLEU 评分提高了 0.004；

四、为了进一步改善系统生成下联的质量，论文借鉴了诗人创作诗歌时反复修改的创作方式，提出了一种对联生成的润色机制。将解码器端生成的下联再经过一轮注意力计算，其中包括自注意力计算和上下文注意力计算两部分。实验证明，加入润色机制的模型和基准的

Transformer 模型相比,在测试集上的 BLEU 评分提高了 0.038,困惑度评分降低了 3.6,证实了润色机制对模型有积极作用。在将三种改进策略都应用到对联生成模型中,改进后的模型和基准 Transformer 模型相比。在测试集上的 BLEU 评分提升了 0.066,困惑度评分降低了 5.33,实验结果证实了本实验提出的方法的有效性。

关键词: 对联生成; 词性特征; 未登录词; 低频词; 润色机制; **Transformer**

ABSTRACT

The couplet is a unique art form in the Chinese traditional culture, which requires the antecedent clause (the first sentence in the couplet) and the subsequent clause (the second sentence in the couplet) follow some strict restrictions and reflects the beauty of Chinese language. The couplet plays an important role in expressing emotions on many festival occasions, which is loved by Chinese people. Creating couplets is a difficult task for ordinary people due to the strict format and content requirements of couplets. Therefore, generating couplets using computers gives the public the opportunity to create couplets. However, the meaning and context of natural language are very complex. For computers, even a very simple language cannot be accurately understood, thus automatic generation of couplets is a challenging task. With the technology of deep neural network, this thesis improves the attention mechanism-based Transformer model, and realizes the generation system of Chinese couplets based on the improved model. For performance evaluation, this thesis refers to three evaluation methods including BLEU, perplexity and artificial evaluation commonly used in machine translation to evaluate the results of all the models proposed. The performance of the model is directly proportional to the BLEU score and inversely proportional to the perplexity score. The main contributions are as follows:

- 1) First, this thesis compares the Transformer based couplet generation model with the existing two schemes: the model based on Encoder-Decoder framework and the Encoder-Decoder framework with attention mechanism. Experimental results confirm the effectiveness of attention mechanism in couplet generation task. The model using attention-based Transformer is used as the baseline model and compared with the three improvement ways proposed in this thesis.
- 2) Then, in order to utilize the linguistic knowledge of Chinese language, this thesis introduces pos (part-of-speech) features into the model. A couplet should be neat in antithesis, and the pos of words in the same position of the two sentences should be consistent. Existing work does not explicitly consider this constraint. In this thesis, a word vector training method combined with pos information is used. Specifically, the tapped pos corpus and original corpus are separated for word vector training. Then the obtained pos vector and word vector are fused in a certain way, and the neural network is trained using the fused word vector. Compared with the baseline model, the improved model improves the BLEU score on the test set by 0.059, and reduces the perplexity by 2.51.

3) This thesis proposes a low-frequency word processing method, aiming at solving the problem of unregistered words and low-frequency words encountered in the process of model training and prediction. Specifically, through inferring the similarity between words, high-frequency words similar to unregistered words and low-frequency words are used to replace them. Under the condition of using this substitution mechanism, the size of the target dictionary is reduced by about 16%. Compared with the baseline model, the improved model improves the BLEU score on the test set by 0.004.

4) Finally, to further improve the quality of the subsequent clause generated by the system. This thesis proposes a couplet polish-up mechanism, inspired by the way that poets iteratively modify when creating poetry. Specifically, in our work, the subsequent clause generated by the decoder is re-processed by attention mechanism, which includes self-attention calculation and contextual attention calculation. The results of experiments show that, compared to the baseline Transformer model, the model with polish-up mechanism improves the BLEU score on the test set by 0.038 and the perplexity decreases by 3.6. The results prove that the polish mechanism has a positive effect on the model. After adding three methods to the model at the same time, the BLEU score of the improved model was increased by 0.066, and the perplexity score was reduced by 5.33, which confirms the effectiveness of the method proposed in this thesis.

Keywords: couplet generation; Transformer; part-of-speech features; unregistered words; low-frequency words; polish-up mechanism

目录

专用术语注释表.....	VII
第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 基于规则模板或模式的方法.....	3
1.2.2 基于实例推理的方法.....	4
1.2.3 基于进化算法的方法.....	5
1.2.4 基于统计机器翻译的方法.....	5
1.2.5 基于深度学习的方法.....	6
1.3 论文研究内容及贡献.....	10
1.4 论文组织结构.....	10
第二章 相关理论及技术概述.....	12
2.1 循环神经网络简介.....	12
2.1.1 标准循环神经网络.....	12
2.1.2 长短时记忆网络.....	13
2.2 编码-解码框架.....	14
2.3 注意力机制.....	16
2.3.1 局部注意力机制.....	17
2.3.2 全局注意力机制.....	17
2.4 结合注意力机制的编码-解码神经网络.....	18
2.5 本章小结.....	20
第三章 中文对联生成模型研究.....	21
3.1 基于编码-解码结构的对联生成系统.....	21
3.2 结合注意力机制的编码-解码框架的对联生成方法.....	22
3.3 基于注意力机制的中文对联生成系统.....	23
3.3.1 Transformer 结构介绍.....	23
3.3.2 基于 Transformer 框架的对联生成模型.....	27
3.4 实验与结果分析.....	28
3.4.1 实验数据.....	28
3.4.2 实验环境.....	28
3.4.3 参数设置.....	29
3.4.4 评价标准.....	29
3.4.5 结果分析.....	31
3.4.6 结果示例.....	32
3.5 本章小结.....	33
第四章 加入词性特征和罕见词处理的中文对联生成模型.....	35
4.1 词向量概述.....	35
4.1.1 词向量表示.....	35
4.1.2 词向量模型.....	37
4.1.3 词向量模型对比.....	42
4.1.4 词向量模型参数选择.....	43
4.2 融合词性信息的对联生成模型框架.....	44
4.2.1 词性序列信息.....	44

4.2.2 加入词性特征的对联生成模型	45
4.3 引入低频词处理的对联生成模型框	46
4.3.1 未登录词和低频词问题	46
4.3.2 未登录词和低频词的解决方法	47
4.3.3 基于词向量相似度的低频词处理方法	48
4.4 实验与结果分析	50
4.4.1 词性特征实验	51
4.4.2 低频词处理实验	51
4.5 本章小结	52
第五章 加入润色机制的中文对联生成模型	55
5.1 问题描述	55
5.2 润色机制概述	55
5.3 加入润色机制的对联生成模型	57
5.4 实验与结果分析	58
5.5 本章小结	59
第六章 总结与展望	61
6.1 总结	61
6.2 展望	62
参考文献	63
附录 1 攻读硕士学位期间撰写的论文	66
附录 2 攻读硕士学位期间申请的专利	67
附录 3 攻读硕士学位期间参加的科研项目	68
致谢	69

专用术语注释表

缩略词说明:

SVM	Support Vector Machine	支持向量机
BLEU	Bilingual Evaluation Understudy	双语评估替换
RNN	Recurrent Neural Network	循环神经网络
CNN	Convolutional Neural Networks	卷积神经网络
LSTM	Long Short-Term Memory	长短期记忆网络
BiLSTM	Bi-directional Long Short-Term Memory	双向长短期记忆网络
GRU	Gate Recurrent Unit	门循环单元
SeqGAN	Sequence Generative Adversarial Nets	序列生成对抗网络
Seq2Seq	Sequence to Sequence	序列到序列
NNLM	Neural Network Language Model	神经网络语言模型
CBOW	Continuous Bag-of-Words Model	连续词袋模型
Skip-Gram	Skip-Gram Model	跳字模型

第一章 绪论

1.1 研究背景及意义

深度学习的概念最早由 Geoffrey Hinton 在 2006 年提出，其兴起于图像识别领域，在之后的很短时间内，深度学习技术广泛应用于机器学习的各个领域。如今，随着计算机硬件性能的快速发展，制约深度学习发展的瓶颈如计算机性能不足等问题被逐步解决，深度学习在图像和语音识别、机器人技术、生物信息处理、自然语言处理等领域都取得显著效果。苹果的 Siri、谷歌的 AlphaGo、特斯拉的无人驾驶等，都是深度学习技术的发展成果，在一些领域内，基于深度学习技术的人工智能的表现甚至已经超越人类，如 AlphaGo 战胜围棋世界冠军李世石和柯洁等。

自然语言处理是深度学习和语言学领域的分支科学，涉及机器翻译、句法分析、信息检索等诸多研究方向。与计算机视觉和语音识别领域的发展类似，在自然语言处理领域，深度学习也是向更智能、更自动提取复杂特征方向发展。

本文研究的对联是起源于我国古代诗词歌赋中对偶句的一种文学体裁，其包含上联和下联两个句子。它融合百姓生活和文学精华于一体，是中华优秀传统文化的重要组成部分。凭借独特的艺术形式以及包罗万象的丰富内容，对联已经深入了人们生活的各个方面、各个领域。在节日场合，人们借用对联表达自己的个人情感或传递信息。对联也有着严格的创作要求，其要求上联和下联两个句子长度相等，断句相同。上联与下联相同位置的字符要遵循语义或语法相关性的某些约束。除此以外，一副优秀的对联的上下联的意境要有联系，这些规则都无法用逻辑表达式规范化表述。如图 1.1 中是民族英雄林则徐所作的对联：“海到无边天作岸”；“山登绝顶我为峰”。意境深远，堪称千古名联。同时这副对联严格遵从了对联的一些约束，如“海”和“山”对仗，“无边”和“绝顶”对仗。创作一副这样的对联需要专业的知识和深厚的文学功底，一般人很难做到。对于中文的自然语言处理，对联在语义上的丰富多样、韵律上的平仄规律对于计算机来说都是不可理解的，因此使用计算机创作对联也具有很大挑战性。随着深度神经网络技术的发展，基于深度神经网络的对联生成获得研究者的关注。

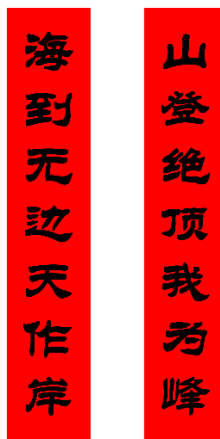


图 1.1 经典对联示例图

使用深度神经网络进行对联的生成研究是一项十分具有意义的工作，主要表现在以下几个方面：对联要求严格对仗、平仄相合、词性相对、内容相关，要求创作者有深厚的文学功底，这对普通人来说过于困难，使用深度神经网络进行对联的自动生成可以帮助普通人也能有创作对联的良好体验；使用深度神经网络进行对联的自动生成，也是对自然语言处理领域中一个具体方面的深入探索和研究，对自然语言处理领域内的其他任务也有一定的借鉴参考价值；使用机器学习方法进行对联的创作，可以将其生成方法拓宽到其他场景下，促进中国传统文化的传播和发展。

1.2 国内外研究现状

对于计算机而言，对联的自动生成和机器翻译任务有相似之处：机器翻译任务是将一种语言映射为含义相同的另一种语言，对联的自动生成则是完成中文内一些词语到另一些关联词语的映射。两者逻辑有相似之处。在使用深度学习技术进行这两种任务时，可以认为对联的生成任务是机器翻译任务的一个特例，从而可以借鉴机器翻译领域的相关成熟的技术。

机器翻译的概念始于 1949 年，1954 年美国 Georgetown-IBM 实验室第一次完成英语和俄语间的机器翻译实验，证明了机器翻译的可行性。但是在后面一段时间内，由于速度慢、消耗计算资源高、准确性低等缺点，机器翻译的发展一度停滞。直到 20 世纪 80 年代，随着社会信息服务需求的扩大，机器翻译技术在处理大量文本翻译任务的优势逐渐凸显，机器翻译的研究开始复苏。

随着机器翻译的发展，关于计算机创作文学诗歌的研究也在进行。计算机创作诗歌文学的研究开始于 20 世纪 60 年代，并且在最近的若干年间迅速发展。由于本文研究的对联是中国特有的传统文学形式，因此相关研究十分有限。由于对联脱胎于诗歌中的对偶句，可以将

对联看作是仅仅有两句的诗歌,但是同时也要注意对联的生成和诗歌的生成存在的一些差异,相较于生成一首诗的所有句子,在给定上联的情况下,生成与之相匹配的下联的任务来说更加明确。而且,一首诗中并不是所有句子都要遵循和对联一样的约束。尽管有一些细小的差异,对计算机程序而言,自动生成诗句的任务和自动生成对联本质上是相同的。因此,在诸多研究中,虽然大部分工作都是针对诗歌的自动生成所做的研究,仅有少数工作是针对对联的自动生成,但是他们采用的方法大致相同,研究成果可以相互借鉴。在诗歌的自动生成任务中,Manurung 提出使用计算机创作诗歌的三个要素^[1]:可读性、可理解性和韵律性。可读性是指自动生成的诗歌需要服从语法规则;可理解性是指自动生成的诗歌需要契合主题,而不仅是词语简单堆砌;韵律性是指生成的诗歌需要有一定的韵律。这些规则同样适用对联。目前主要的诗歌或对联生成方法有传统生成方法和深度神经网络生成方法两大类,其中传统生成方法有基于规则模板的自动生成方法;基于实例推理的自动生成方法;基于遗传算法的自动生成方法;基于统计机器翻译的自动生成方法。

1.2.1 基于规则模板或模式的方法

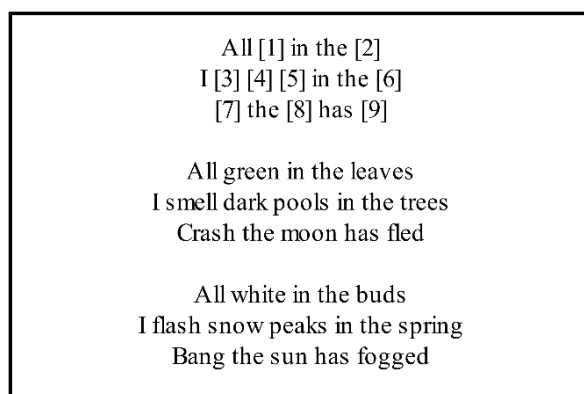


图 1.2 Masterman 系统使用的模板及其生成的诗歌

早期针对诗歌对联自动生成的研究有使用规则模板或模式匹配的方法。这个方法的思路类似于选词填空,其主要通过去掉已有的诗歌中的部分词语,重新选择词典中与去掉的词语意思相近的词对其进行替换,从而产生新的诗歌。该方法具有较高的语法上的可靠性,但灵活性欠佳,文献[2][3][4]均采用基于规则模板的方法生成诗歌。除了基于规则模板的方法,还有基于模式的方法,基于模式的方法在基于规则模板的基础上,对每个位置的词的词性、韵律平仄进行限制,提高了生成诗歌的质量。PaBlo Gervás 提出的西班牙语诗歌生成系统 WASP^[5]在生成诗歌时,根据模式预先设定的句子长度、单词数目、各词性词语所占比例,使用贪心算法从词典中选择满足约束条件的词语填入指定位置。此外,还可以加入短语语法规则约束

来保证诗歌或对联语法上的要求，如 Xiaofeng Wu 提出从语料库的数据中提取语法规则，并将其用于诗歌的生成^[6]。其他具有代表性的基于模板的诗歌生成系统还有 RETURNER、Masterman 和 PROSE 等，图 1.2 中是 Masterman 系统使用的模板及其生成的诗歌。

基于规则模板或模式的诗歌或对联生成方法，对于在一定程度上保证生成诗歌或对联的质量，美国的《Hartman》杂志甚至曾经收录过这些机器生成的作品。但是，采用这样的方法也有一定的局限性，生成的诗歌灵活性较差，诗歌的质量依赖于模板的质量，离真正的使用计算机自动生成诗歌或对联的目标相去较远。

1.2.2 基于实例推理的方法

基于实例推理的诗歌或对联生成方法即根据用户需要，检索已有诗句，依据用户所描述的目标信息对已有诗句作内容上的调整，基于实例推理的具有代表性的诗歌生成系统有 ASPERA^[7]和 COLIBRI^[8]。Gervás 提出的 ASPERA 以一种半自动交互的方式进行西班牙语诗歌的撰写，用户输入诗歌风格参数及其他约束，ASPERA 搜索知识库确定符合要求的诗歌，对所选的诗歌按照用户需求进行调整改编，完成从诗歌到诗歌的“翻译”。COLIBRI 系统则通过对现有数据库进行检索，找出匹配当前描述的案例，将案例运用的解决方法应用到新的问题上，使用 COLIBRI 进行诗歌对联的自动生成主要包括图 1.3 中四个步骤。

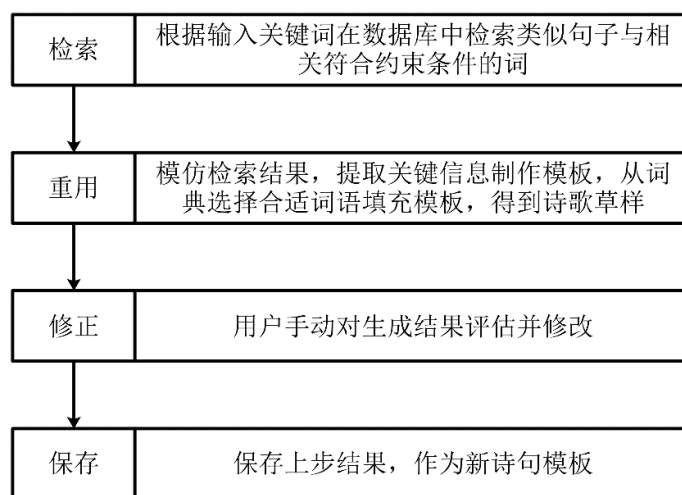
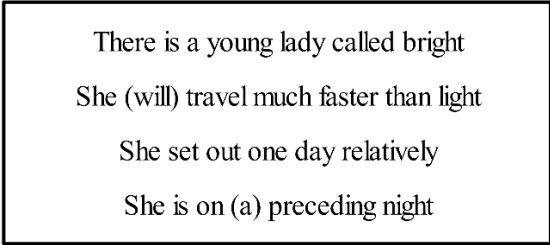


图 1.3 COLIBRI 系统生成诗歌过程

基于实例推理的方法相比基于规则模板或模式的方法在生成的诗歌质量上有了一定的提升，但是也存在一些不足，如只考虑了浅层的语义信息而没有完全满足诗歌对意义内涵的要求，修正步骤如何自动处理也是机器自动创作诗句对联的障碍。

1.2.3 基于进化算法的方法

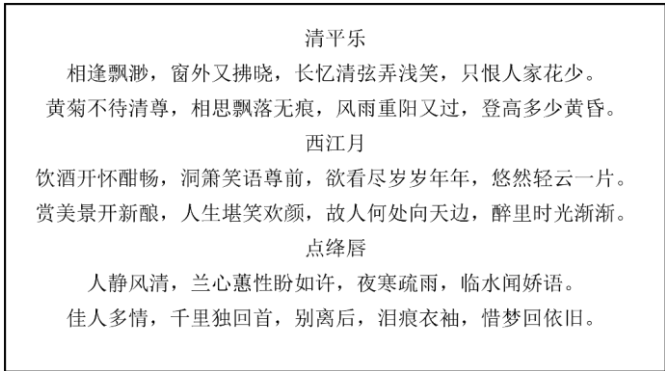
还有一种基于进化算法的对联自动生成方法。进化算法也被称为“演化算法”。其灵感来自于大自然的生物进化，是一种成熟的高鲁棒性且广泛适用的全局优化方法，用来解决全局最优化问题。基于进化算法自动生成诗歌或对联就是通过择优进化的思想不断对生成的诗句进行优化，直到其最优地满足某个约束条件或评价标准^[9]。使用进化算法生成诗歌包含两个部分：生成和评估，先按照简单的准则随机生成诗句，再按照事先制定的标准对其打分，随后生成模块根据评估模块的打分对诗句进行优化，不断迭代优化，最终得到评分最高的结果。基于进化算法设计的诗歌生成系统有 POEVOLVE 和 MCGONAGALL 等。图 1.4 为 MCGONAGALL 生成的诗句。



There is a young lady called bright
She (will) travel much faster than light
She set out one day relatively
She is on (a) preceding night

图 1.4 MCGONAGALL 系统生成诗句

周昌乐^[10]使用进化算法生成宋词也取得不错的效果，他根据宋词的特点，按照平仄从词典中随机选取若干词语初始化诗句，再对句法合法性、主题相关性、词句搭配的适当性和风格情感的统一性四个评价指标加权求和作为适应度函数，结合精英主义和轮盘赌算法对生成的诗句迭代优化，得到适应度函数取值最大的诗句作为最终结果。图 1.5 为该基于进化算法的宋词自动生成系统典型示例。



清平乐
相逢飘渺，窗外又拂晓，长忆清弦弄浅笑，只恨人家花少。
黄菊不待清尊，相思飘落无痕，风雨重阳又过，登高多少黄昏。

西江月
饮酒开怀酣畅，洞箫笑语尊前，欲看尽岁岁年年，悠然轻云一片。
赏美景开新酿，人生堪笑欢颜，故人何处向天边，醉里时光渐渐。

点绛唇
人静风清，兰心蕙性盼如许，夜寒疏雨，临水闻娇语。
佳人多情，千里独回首，别离后，泪痕衣袖，惜梦回依旧。

图 1.5 基于进化算法的宋词自动生成系统示例

1.2.4 基于统计机器翻译的方法

统计机器翻译的基本思想是通过对大量平行语料的统计分析，构建统计翻译模型，使用

该模型进行翻译任务。微软亚洲研究院的 Long Jiang 将机器自动创作对联视为一种机器翻译的过程^[11]，提出了一种基于短语的统计机器翻译方法来生成对联：首先，对联生成系统接收上联作为输入，根据上联翻译下联，使用翻译解码器生成 N 条下联的候选句子；然后使用一组过滤器去掉候选句子中违反语法规则的部分；最后，使用 SVM 算法对候选句子按照质量从高到低排序，再辅以人工评价和 BLEU 评分机制进行综合评估。实验结果显示性能优秀。

Ming Zhou 将这种方法进一步拓展，用于中国古典诗词中绝句的生成^[12]，按照用户输入的关键词和词典中词语的相关性使用语言模型进行打分，产生第一行诗句，之后根据第一行诗句利用统计机器翻译模型生成第二行诗句，根据第二行诗句生成第三行诗句，依此类推，直到生成一首完整的绝句。图 1.6 是使用统计机器翻译方法生成诗歌的示例。

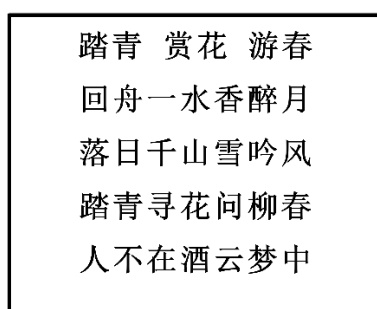


图 1.6 统计机器翻译生成诗歌示例

统计机器翻译模型能够学习语料中的规律，保证了对联内部的语义平衡，但是生成的诗歌对联包含的句法成分较低。在给出上联，生成下联的情况下，可能生成结果都是“词词对仗”，但是组合起来却难以理解。同时，统计机器翻译模型依赖庞大的语料库，处理这些语料库所需的计算资源也十分庞大，这些都是制约统计机器翻译模型进一步发展的障碍。如何将句法信息引入诗歌对联生成模型，使诗歌生成模型更加“智能”中是目前的热点。

1.2.5 基于深度学习的方法

以上的传统方法都依赖于人为设计的大量规则对诗歌对联韵律和质量进行约束，迁移能力较差。而近来发展迅速的深度学习技术，不仅推动图像处理、语音识别等应用领域快速发展，针对诗歌对联生成任务，深度学习也有用武之地。

Xingxing Zhang 提出一种基于循环神经网络的中文诗歌生成模型 RNNPG^[13]。RNNPG 系统框图如图 1.7 所示，首先根据用户给定的关键词产生第一句诗，过程和上述基于统计机器翻译的方法相似，加入一定的约束使得第一行诗句符合规范，之后系统根据历史生成的诗句生成下一行诗句，直到诗歌生成完成。RNNPG 模型主要包含三个部分：CSM(Convolutional

Sentence Model)、RCM(Recurrent Context Model)和 RGM(Recurrent Generation Model)。CSM 是一个 CNN 模型,用来获取一句话的向量表示;RCM 是句子级别的 RNN 模型,根据历史生成的句子向量,输出下一个待生成诗句的背景向量;RGM 是字符级别的 RNN,根据 RCM 的输出的背景向量和该行诗句已经产生的字符,得到待生成字符的概率分布。图 1.8 为 RNNPG 系统生成的诗歌示例。

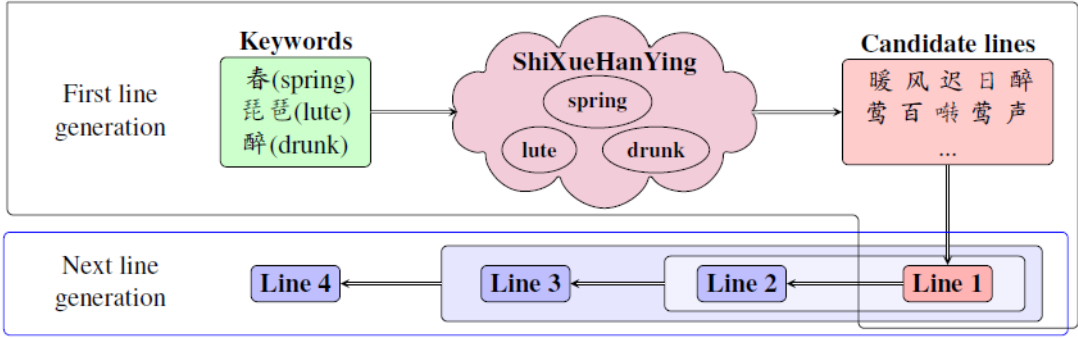


图 1.7 RNNPG 系统框图

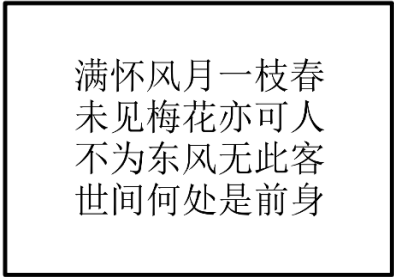


图 1.8 RNNPG 系统生成诗句实例

RNNPG 是一种基于深度神经网络的诗歌对联生成技术,不同于传统方法,RNNPG 可以自动从训练语料中学习到文本特征,效果更好。但是其局限性在于产生第一行诗句时仍然使用类似基于统计机器翻译的方法,且用户输入的关键词仅与生成的第一句诗有关,造成主题漂移。

2013 年,N Kalchbrenner 等人首先提出了基于神经网络的“Encoder-Decoder”的模型框架^[14],也称为“编码-解码”框架,这个框架的提出引起了众多学者的关注,“编码-解码”框架开始广泛应用于各种领域,如场景分割^[15]、对话生成^[16]、媒体内容的描述^[17]、物体轮廓检测^[18]和机器翻译^[19]等等。编码和解码的部分可以是任意的文字、语音、图像和视频数据。在自然语言处理领域,编码-解码框架有重要的应用价值,2014 年由 Google 的 Sutskever、O Vinyals 和 QV Le 提出使用基于长短时记忆网络 LSTM(Long Short-Term Memory)的“编码-解码”框架用于序列到序列(Sequence-to-Sequence)的生成任务^[20]。具体而言,对于输入语言序列,使用编码网络将其映射为一个连续稠密向量,再通过解码网络将其转化为目标语言序列。

对联自动生成任务就是解决上联到下联的映射问题。但是在运用“编码-解码”框架解决这个映射问题的过程中，如果上联过长，编码器在将上联的所有信息都压缩在一个固定长度的上下文向量(context vector)的过程中不可避免会出现信息损失，为了解决这个问题，Bengio 等人提出了基于“Attention”机制的神经网络结构^[21]，“Attention”机制即注意力机制，最早是由 DeepMind 团队为解决图像分类问题提出来的^[22]，目的是让神经网络计算时分配更多的“注意力”给输入序列中和待生成的目标相关度较高的部分，而更少地关注相关度较低的输入。

Qixin Wang 将基于注意力机制的“编码-解码”框架用于宋词的生成研究^[23]，编码器 Encoder 使用双向长短时记忆网络 BiLSTM，解码器 Decoder 使用了长短时记忆网络 LSTM。使用 LSTM 是为了解决普通 RNN 处理长距离依赖问题时产生的“梯度爆炸/消失”问题。图 1.9 为该宋词生成模型的整体结构。该系统将已经生成的内容作为源句，生成下一行宋词。即用户提供宋词的第一句，系统根据第一句生成第二句，根据第一句和第二句生成第三句，不断重复，直到生成整首宋词。LSTM 配合注意力机制，使模型可以学习到更长诗句中的语义关系，也可以保证语意连贯性。但是同时，这样的结构依然存在主题漂移的缺点。由于对联仅有上下联两个句子，不存在主题漂移的问题，因此该结构可以用来生成对联，本文将该方案作为对照方案之一，与本文提出的方法进行比较。

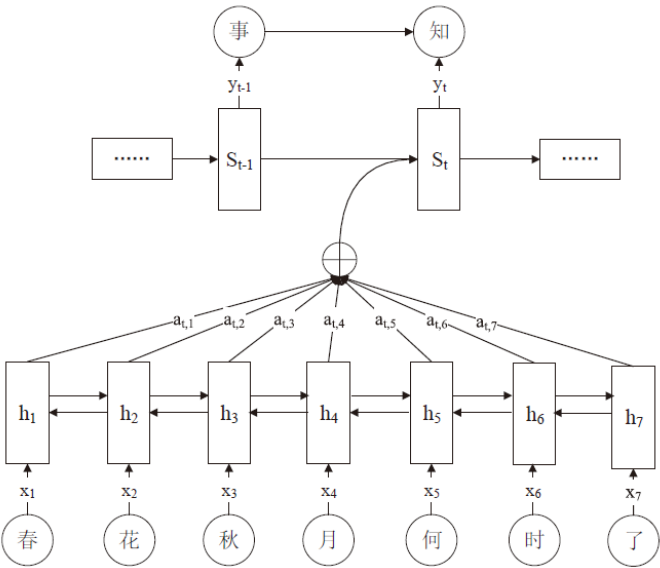


图 1.9 基于注意力机制的“编码-解码”宋词生成系统框架图

Wang Zhe 在结合注意力机制的“编码-解码”框架的基础上，提出了一种两阶段式的诗歌生成方法^[24]。第一阶段，用户输入关键词，模型生成诗歌的子主题词序列，子主题词序列长度和待生成的诗歌行数相等，每个子主题词代表对应各行的主题；第二阶段，基于“编码-

解码”框架，根据子主题词生成诗歌。该框架由两个编码器和一个解码器组成，其中一个编码器以子主题词作为输入，另一个编码器以历史生成的句子作为输入，将两个编码器的输出拼接作为解码器的输入，由解码器生成下一个句子。解码器生成下个句子时，由注意力机制对主题词和历史生成句子向量加权求和，模型决定生成过程中各个部分的权重大小。图 1.10 为该系统模型整体框架图。该模型结构可以使用户写作意图影响整首诗的生成，避免了主题漂移现象，使生成作品的语义更加连贯。

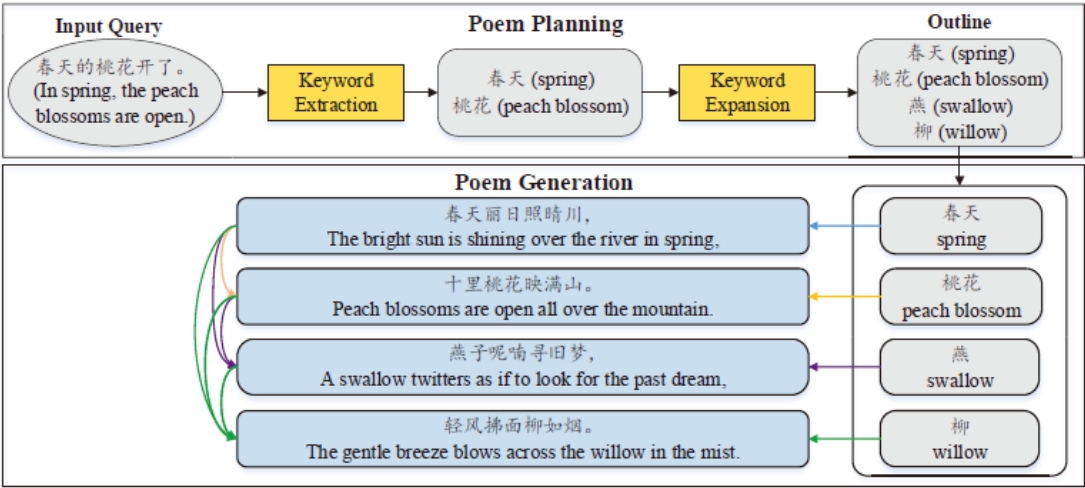


图 1.10 两阶段式诗歌生成方法整体框架图

Rui Yan 提出了一种诗歌生成模型 iPoet^[25]。iPoet 在“编码-解码”框架的基础上引入了一种诗歌润色机制，该模型模仿人类创作诗歌时不断润色修改的创作方式。iPoet 诗歌生成模型接受用户输入的写作意图关键字并将其表示为向量形式；在获得的写作意图向量的基础上，使用层次化的 RNN 网络逐个字符地生成诗句。与其他的对联生成模型直接输出诗歌不同，iPoet 将已经生成的完整的诗歌作为“草稿”，附加到用户意图向量上，再次输入 RNN 网络，通过迭代产生新的诗歌，iPoet 提出的这种润色机制是提高机器创作诗歌对联任务质量的一种有效尝试。

除了传统的神经网络之外，Lantao Yu 还将图像处理中的对抗生成网络应用到自然语言处理中的文本生成任务上^[26]，提出了 SeqGAN 的模型结构。SeqGAN 的生成网络使用循环神经网络，网络结构可以是 RNN 或 LSTM 等，直接生成整首诗歌；判别网络是一个卷积神经网络，用于判断诗歌是人还是机器创作的，并使用强化学习的方式，将梯度回传给生成网络。将对抗网络用于序列文本的生成为机器自动创作诗歌对联任务提供了全新思路。

尽管深度学习技术在对联和诗歌的生成任务上获得广泛的应用，但是，其无法显式地满足对联的上下联对仗的要求。如何对基于深度神经网络的语言模型进一步改进，是本文研究的重点。

1.3 论文研究内容及贡献

本文针对自然语言处理领域的计算机自动生成对联任务，将深度学习与自然语言处理问题结合进行研究。重点关注对联自动生成中的如下问题：上联和下联之间的对仗问题，根据对联的格式要求，上联和下联对应位置的词语需要严格对仗，词性相同，如何在神经网络模型中显式的加入这个约束；未登录词及低频词对模型性能的影响，在使用神经网络进行对联生成的过程中，输入的上联中包含词典中未登录词时，会造成输入句子语义的不完整，进而导致语言模型的表现会显著下降，同时，过多的低频词汇很大程度上增大了词汇表的规模，从而降低语言模型的运算速度，如何对其进行改进；神经网络的结构改进，对于大规模的训练语料，如何改进神经网络的结构，增强下联和上联之间的语义相关性，提升模型表现。

主要贡献在于：

(1) 本文提出使用完全基于注意力机制的 Transformer 模型进行对联的自动生成。和传统的基于“编码-解码”框架的语言模型比较，提高了计算速度，表现也有一定提升。相较于传统的基于 CNN 的语言模型，在获取上下文依赖关系时可以保持较少的参数量^{[27][28]}；相较于 RNN 或 LSTM 网络结构，可以使模型的训练和预测过程保持并行性。

(2) 引入词性特征信息到对联生成模型之中。提取训练语料中所有词的词性信息，将生成的词性向量与原语料中各个词的初始词向量以一定的方式相结合，使用经过处理后的词向量对神经网络进行训练，将中文语言学的先验知识融合到模型当中，使模型可以学习到对联中的对仗关系。

(3) 解决对联生成任务中的词汇表受限问题，基于词向量的相似度算法，使用高频词对未登录词和词典中的低频词进行替换，避免了对联语义的缺失，减小词汇表规模，提升了语言模型的计算速度。

(4) 借鉴诗人创作诗歌时反复修改的创作方式，在解码器生成下联的词向量序列后，将所生成的词向量序列作为“草稿”，重新输入模型解码器进行计算，再次生成下联，提高下联和上联的语义连贯性。

1.4 论文组织结构

本文将完全基于注意力机制的 Transformer 神经网络框架应用于对联的自动生成任务，并对其进行改进，进一步提升生成的对联的质量。论文的组织结构如下：

第一章：绪论。介绍论文的研究背景，多个角度归纳目前应用于诗歌或对联自动生成任

务的国内外研究工作，最后介绍了本文提出的改进策略。

第二章：本章介绍了深度学习的基础知识。包括本文使用到的循环神经网络几种常见的网络单元结构如循环神经网络 RNN 及其变种 LSTM 等，接下来介绍了最常用的自然语言处理模型，如编码-解码网络框架和注意力机制的基本概念。

第三章：本章将基于 Transformer 的对联生成模型和已有的方案相比较，证实注意力机制的在对联生成上的优越性。已有的对联生成模型主要包括：基于“编码-解码”框架的对联生成模型；结合注意力机制的“编码-解码”框架的对联生成模型。论文使用 BLEU 评测方法、困惑度(Perplexity)方法和人工评价三种方式对三种模型的性能进行比较，保证评价的公正性和合理性。

第四章：本章主要研究对联生成任务中的两个关键问题：上下联对应位置词语的对仗问题；输入对联中的未登录词和词典中的低频词的问题。本章在 Transformer 模型的基础上做出两个重要改进：将对联语料的词性信息引入模型，使语言模型可以学习到上下联之间对应位置的词语的词性应该保持相同；根据词向量的相似度计算算法，建立了一种对未登录词和低频词进行替换的机制，最后设计实验将结合改进机制的模型和第三章中的基线 Transformer 模型进行比较，给出实验结果，证实本文提出的改进策略的有效性。

第五章：本章借鉴诗人创作诗歌时反复修改润色的创作方式，在基线对联生成模型的基础上引入了一种润色机制，并与第三章、第四章中的多种模型的性能进行比较，给出了实验结果，证实了本章提出的润色机制对模型的性能有一定的提升作用。

第六章：总结与展望，对本文中搭建的各类对联生成模型进行总结并指出不足。最后分析了使用深度学习技术进行对联生成的未来的研究方向。

第二章 相关理论及技术概述

2.1 循环神经网络简介

深度神经网络(Deep Neural Networks)指具有深层网络结构的神经网络,其在语音识别^{[29][30]}和计算机视觉^{[31][32][33][34]}等领域取得突出成就。深度神经网络因为较多的神经网络层数,所以能从输入中提取更多特征,对现实有更强的刻画能力。按照采用的核心网络,深度神经网络可分为:深度卷积神经网络(Deep CNN)和深度循环神经网络(Deep RNN)等。在计算机视觉和图像视频处理领域,卷积神经网络相较于其他神经网络有较大优势。而在语言建模、文本生成和机器翻译等领域,由于输入数据都具有依赖性且是序列模式, CNN 的前后输入之间没有任何关联,所有输出相互独立,因此 CNN 的性能并不好。对于本文研究的中文对联自动生成任务,所有的输出都与之前的输出有一定的关联,需要有一些基于之前输出信息的偏向,因此循环神经网络 RNN 更为合适。

2.1.1 标准循环神经网络

基础的神经网络包括输入层、隐藏层和输出层三层结构,其只在层与层之间建立连接,而标准循环神经网络 RNN(Recurrent Neural Network)在此基础上,在同层之间的神经元之间也建立了连接,RNN 的神经网络结构如图 2.1 所示,等号右边为神经网络按时间展开图,等号左边是其简化图。假设 x_t 是序列中时间步 t 的输入, h_t 是该时间步的隐藏状态向量,根据图 2.1 中的神经网络结构,当前的 h_t 为:

$$h_t = \mathcal{F}(x_t W_{xh} + h_{t-1} W_{hh} + b_h) \quad (2.1)$$

其中 h_{t-1} 是上一时间步的隐藏状态向量。简单来说,时间步 t 的隐藏状态向量由当前时间步的输入 x_t 和上一时间步的隐藏状态向量 h_{t-1} 共同决定, \mathcal{F} 是神经网络的激活函数, W_{xh} 和 W_{hh} 是神经网络的权重矩阵。RNN 特殊的网络结构有其独特优势,RNN 能够将历史信息和当前输入相结合来预测当前输出,当输出依赖于距离较近的历史信息时性能尚可,但是实验证明^[35],当输出依赖于距离较远的历史信息时,即存在长距离依赖问题(Long-Term Dependencies)时,RNN 表现并不好。此外,RNN 神经网络训练采用基于时间的反向传播算法(Back Propagation Through Time, BPTT),进行链式求导时,会出现梯度消失和梯度爆炸问题。为了解决梯度爆炸,可以使用“截断梯度”和添加正则项的做法^[36];而针对梯度消失,可以优化激活函数,如使用 Relu(Rectified linear units)函数替代标准 RNN 使用的 sigmoid 和 tanh。但是,针对梯度

消失最常用的解决方法还是使用长短时记忆网络 LSTM 代替一般 RNN 网络。

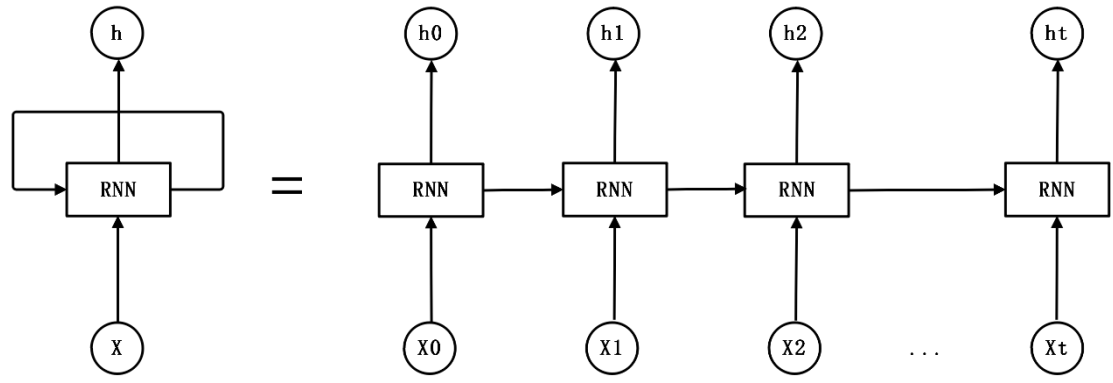


图 2.1 标准循环神经网络基本结构

2.1.2 长短时记忆网络

长短时记忆网络 LSTM 更适用于解决存在长期依赖关系的时间序列问题，其就是为了解决一般 RNN 网络的缺陷。如图 2.2 所示，和普通的 RNN 相比，LSTM 中增加了一个记忆单元和三个控制器：输入控制、遗忘控制和输出控制。记忆单元作用是存储网络状态；输入控制器决定保留多少当前时刻的输入；遗忘控制器决定上一时刻神经网络状态在当前时刻的保留程度；输出控制器根据当前时刻的神经网络状态决定输出信息。

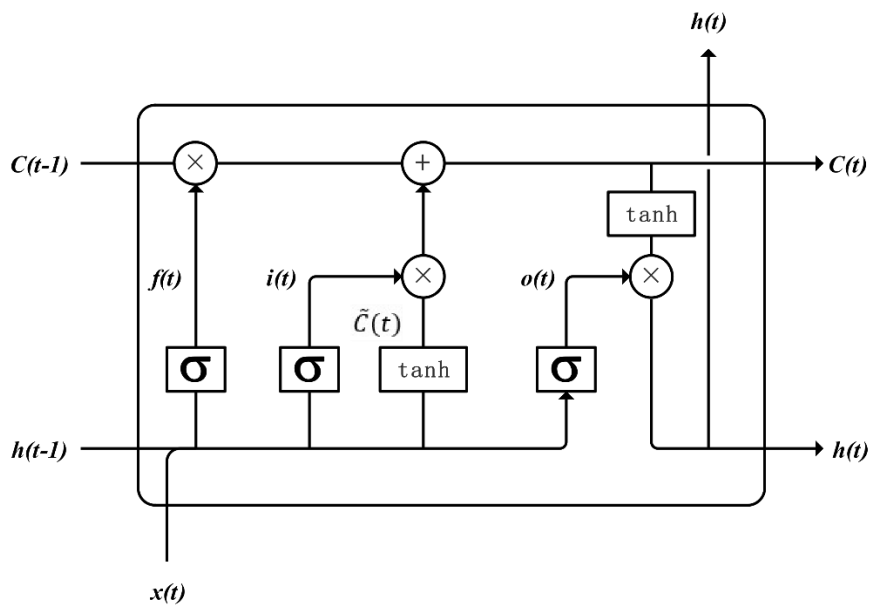


图 2.2 长短时记忆网络 LSTM 基本结构

LSTM 的核心是记忆单元状态，由 sigmoid 网络和相乘器组成，使用这种结构可以增加或删除记忆单元中的信息。sigmoid 网络输出范围在[0,1]的数字，表示有多少信息可以通过记忆单元，0 表示都不能通过，1 表示全都可以通过。LSTM 神经网络单元对数据的处理主要包括

四个阶段：

1) 决定记忆单元丢弃哪些历史信息。sigmoid 网络层通过查看 h_{t-1} 和 x_t 输出一个 0~1 的向量 f_t ，决定保留多少上一个神经单元的状态 C_{t-1} ，公式表示如下式 2.2 所示：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

2) 决定给记忆单元增加哪些新的信息，分为两个步骤。首先，输入控制器查看 h_{t-1} 和 x_t 输出一个 0~1 的向量 i_t ，决定更新哪些信息；然后， h_{t-1} 和 x_t 通过 tanh 网络层得到候选记忆单元信息 \tilde{C}_t ， \tilde{C}_t 可能会更新到记忆单元信息中。公式表示如下式 2.3 和 2.4 所示：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

3) 更新记忆单元信息 C_{t-1} ，得到新的记忆单元信息 C_t 。遗忘控制器选择忘记历史单元信息的部分，输入控制器选择添加候选记忆单元信息的部分。更新操作如下式 2.5 所示：

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.5)$$

4) 输出神经网络单元特征信息 h_t 。sigmoid 网络层查看 h_{t-1} 和 x_t 输出一个 -1~1 的向量 o_t ， C_t 通过 tanh 网络层后与该向量相乘，最终得到当前神经网络单元输出，计算过程如下：

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.6)$$

$$h_t = o_t * \tanh(C_t) \quad (2.7)$$

这是最基本的 LSTM 结构，目前有 LSTM 的变种形式如 GRU 等等，但是基本思想都是统一的。LSTM 和 GRU 是目前比较常见的神经网络隐藏层单元结构，除此以外，还有 Koutnik 提出的 Clockwork RNN 结构^[37]，Yao 提出的 Depth Gated RNN 结构等，对于具体的任务，可以适当变形以适应任务需要。

2.2 编码-解码框架

编码-解码框架是深度学习中的一种重要的模型框架。框架中的编码器与解码器处理的数据类型可以是文字、图片和语音等，为了处理不同数据，编码器与解码器模块可以采用上节中提到的 RNN、LSTM 和 GRU 等网络结构或多种神经网络结构的组合，多变的应用方式使其适用于多种应用场景。

编码-解码框架主要解决序列到序列的映射问题^[19]，即 Seq2Seq 问题。Seq2Seq 问题，即输入一个序列，输出另一个序列的问题。如机器翻译、语音识别和自动问答系统等，都可以视作从一个序列映射到另一个序列的问题，这类问题的典型特征是输入和输出的长度都无法确定。使用编码-解码框架搭建模型可以解决这类问题，编码-解码框架的简单示意图如图 2.3

所示。其中，编码器接收输入序列，解码器输出目标序列。编码阶段将整个输入词序列映射为一个固定大小的背景(context)向量 c ，解码阶段根据背景向量 c ，通过最大化预测序列概率解码出整个目标序列。

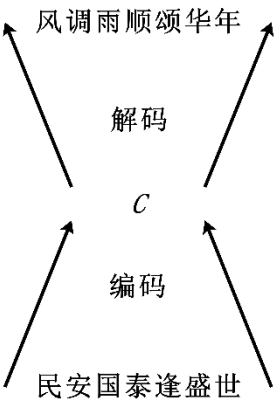


图 2.3 编码-解码框架

近年来，图像处理、语音识别和机器翻译领域的众多工作都是围绕编码-解码框架展开的。其中机器翻译任务和对联生成任务相似度较高，都是从一个文本序列映射到另一个文本序列。用于机器翻译的编码-解码框架可以看作完成这类问题的通用解决方案。其基本思想是：给定源词序列 $X = \{x_1, x_2, \dots, x_m\}$ ，目标词序列 $Y = \{y_1, y_2, \dots, y_n\}$ 。编码-解码框架的目标是计算如式 2.8 所示的条件概率密度函数：

$$p(y_1, y_2, \dots, y_n | x_1, x_2, \dots, x_m) = \prod_{t=1}^{t=n} p(y_t | c, y_1, \dots, y_{t-1}) \tag{2.8}$$

其中 c 表示编码器生成的背景向量，包含了输入序列的全部信息。每一个 $p(y_t | c, y_1, \dots, y_{t-1})$ 通过 SoftMax 的计算方式得到。

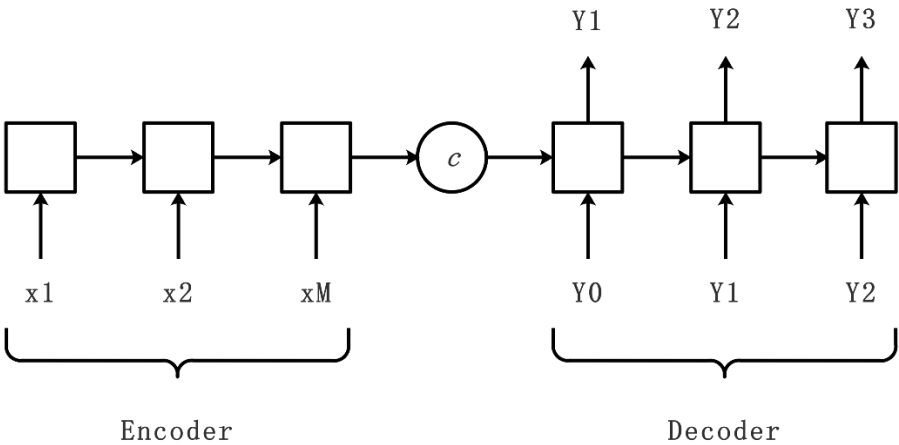


图 2.4 编码器-解码器模型示意图

如图 2.4 所示，编码阶段通过非线性变换 Enc 将输入词序列 $X = \{x_1, x_2, \dots, x_m\}$ 映射为背景

向量 c :

$$c = Enc\{x_1, x_2, \dots, x_m\} \quad (2.9)$$

解码阶段根据背景向量 c 和 t 时刻之前已经生成的历史信息 $\{y_1, y_2, \dots, y_{t-1}\}$ 进行解码, 通过非线性变换 Dec 生成 t 时刻的目标词 y_t :

$$y_t = Dec\{c, y_1, y_2, \dots, y_{t-1}\} \quad (2.10)$$

每个 y_t 都如此依次产生, 整个编码-解码框架就是根据输入词序列 X 生成目标词序列 Y , 如果 X 是中文词序列, Y 是英文词序列, 那么此编码-解码框架就是解决机器翻译问题; 如果 X 是一个文段, Y 是概括性描述语句, 那么此编码-解码框架就是解决文本摘要生成问题; 如果 X 是对联的上联, Y 是对联的下联, 那么此框架就可以用来完成中文对联的自动生成任务。

2.3 注意力机制

注意力机制, 自提出起就广泛应用于深度学习的各大领域。注意力机制借鉴了人类视觉上的选择性注意力机制。人类使用眼睛快速扫描全局图像, 经过大脑处理后, 会自发地选择需要重点关注的区域, 这就是注意力焦点, 在随后的观察中, 大脑会将更多的注意力资源放到注意力焦点上, 来获取目标更多的细节, 这就是注意力机制。注意力机制最早应用在计算机视觉领域^{[22][38]}, 之后, 其在自然语言处理领域也获得了巨大成功^{[39][40]}。

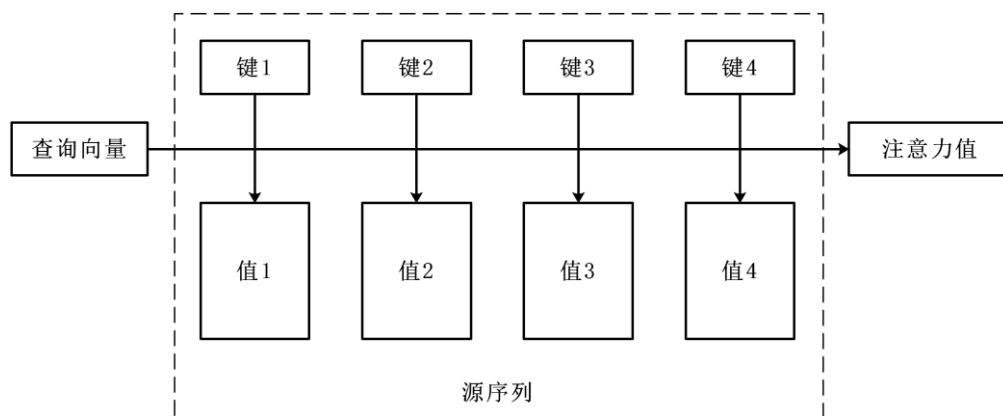


图 2.5 注意力机制结构示意图

可以对注意力机制进一步抽象。如图 2.5 所示, 源序列由一系列的键值对向量 Key 、 $Value$ 构成, 给定目标序列中的某个元素的查询向量 $Query$, 计算 $Query$ 和每个键 Key 的相似性, 得到每个键 Key 对应的权重, 根据得到的权重对 $Value$ 加权求和, 得到注意力计算的值。所以注意力机制的本质理解为: 对源序列中的特征值加权求和。其中, 每个特征值的权重由源序列和目标序列的相似度决定。即如式 2.11 所示:

$$Attention = \sum_{i=1}^{L_x} Similarity(Query, Key_i) \times Value_i \quad (2.11)$$

其中， L_x 是源序列的长度，公式含义如上所述。对于自然语言处理问题，源序列的键值对向量 Key、Value 是相同的，都是输入句子中每个词对应的语义编码。在深度学习领域，根据注意力分配在全部输入还是部分输入上，注意力机制又可以分为全局注意力机制(Global Attention)和局部注意力机制(Local Attention)。

2.3.1 局部注意力机制

局部注意力机制，指有选择地选取部分特征，而完全忽略未被选择的特征。如图 2.6 所示，局部注意力机制中，目标序列仅与源序列中的部分特征进行相似度计算。举例来讲，在图像处理任务中^[41]，对图像中的数字进行识别。神经网络需要在每个时刻提取原图像中的特征信息，识别出图像中的数字。该任务中，有用信息仅存在于数字部分的像素点，其余部分是冗余的，因此可以采用局部注意力机制，只选择图像中需要关注的像素点，减少网络的计算量。

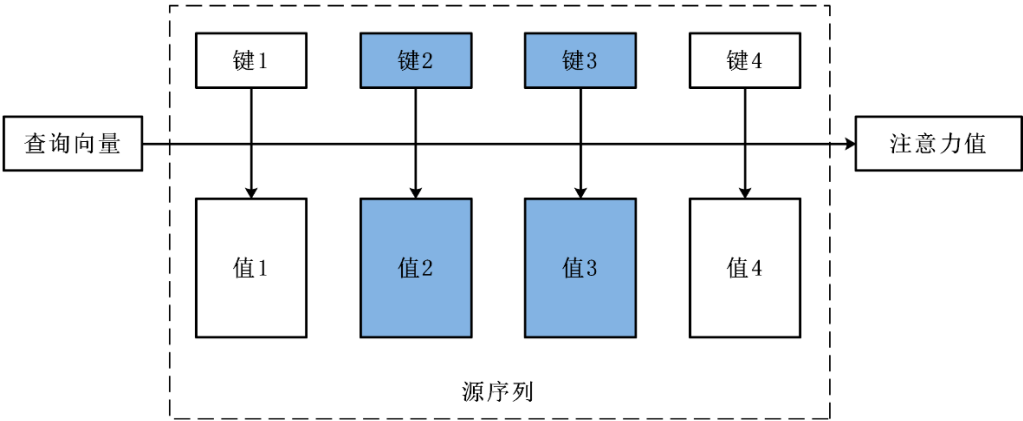


图 2.6 局部注意力机制结构示意图

2.3.2 全局注意力机制

全局注意力机制，是对输入的全部特征分配注意力再进行加权的方法，不同于局部注意力机制忽略了部分的输入特征。全局注意力机制考虑全部的输入，但是降低不重要输入特征信息的权重。这种方式使全局注意力机制具备可微分性，可以与神经网络一起获得训练。全局注意力机制对所有特征进行加权处理，符合中文对联的整体性和连贯性的要求，同时突出重要特征，又符合对联要求的上下联对仗的要求。且在所有的输入上都分配注意力，有利于增强上下联的语义相关性。

2.4 结合注意力机制的编码-解码神经网络

经典的编码-解码模型用来解决大多数问题都十分有效，但是也有不足之处：编码神经网络需要将输入序列的所有相关信息都压缩到固定的背景向量 c 中。如果输入序列过长，则背景向量 c 会无法完全包含整个输入序列的全部信息，解码器根据背景向量 c 和历史信息生成当前时刻输出时，本应该和当前时刻输出强相关的输入信息可能会被严重“稀释”，导致输出信息的不准确。实验[19]表明，随着输入句子长度的增加，经典编码-解码框架的性能确实迅速下降。

为了解决传统编码-解码框架的局限性，Dzmitry Bahdanau 等人对其进行改进^[21]，引入了上一节提到的注意力机制，提出了结合注意力机制的编码-解码模型。结合注意力机制的编码-解码模型与传统编码-解码模型最大的区别在于，前者将输入序列映射为一个背景向量序列 $C = \{c_1, c_2, \dots, c_m\}$ 而非一个固定大小的背景向量，这个背景向量序列中的元素包含输入序列不同部分的信息，使解码网络在解码时，可以选择性地关注背景向量序列 C 中的部分元素。注意力机制赋予模型动态选择输入向量的某个特定部分的能力，提升语言模型处理长距离依赖的能力。

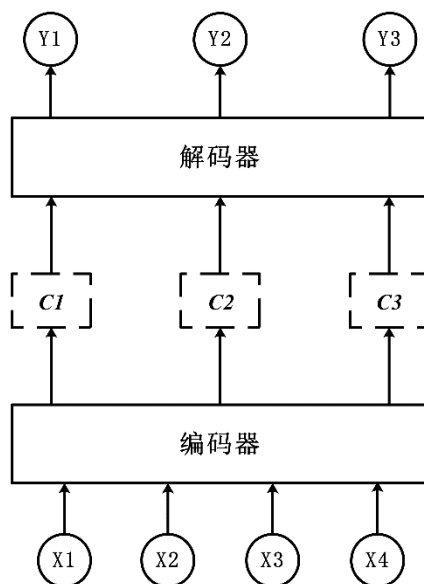


图 2.7 结合注意力机制的编码-解码模型

如图 2.7 所示是结合注意力机制的编码-解码模型结构，假设输入序列 $X = \{x_1, x_2, \dots, x_m\}$ ，输出目标序列 $Y = \{y_1, y_2, \dots, y_n\}$ ，则 t 时刻解码器的输出为：

$$y_t = Dec\{c_t, y_1, y_2, \dots, y_{t-1}\} \quad (2.12)$$

其中， c_t 不是一个固定的向量，而是根据待输出目标的变化而变化。对图 2.7 中的模型进行细化，编码器与解码器均使用 RNN 神经网络，得到如图 2.8 所示的编码-解码模型框架，其中

$\{h_1, h_2, \dots, h_m\}$ 和 $\{s_1, s_2, \dots, s_n\}$ 分别为编码器和解码器的神经网络隐藏状态向量。

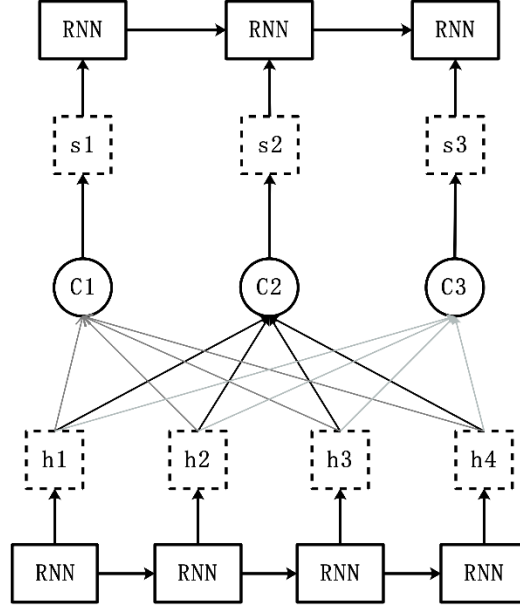


图 2.8 基于注意力机制和 RNN 网络的编码-解码模型

第一阶段，假设当前时刻为 i ，计算上一个时刻的解码器隐藏状态向量 s_{i-1} 和编码器隐藏状态向量 $\{h_1, h_2, \dots, h_m\}$ 的相似度，最常见的计算方法有：向量点积；向量余弦相似度；引入额外的神经网络，计算公式分别如下式 2.13、2.14 和 2.15 所示，其中 $j = 1, 2, \dots, m$ ：

$$Sim_{ij} = Similarity(s_{i-1}, h_j) = s_{i-1} \cdot h_j \quad (2.13)$$

$$Sim_{ij} = Similarity(s_{i-1}, h_j) = \frac{s_{i-1} \cdot h_j}{\|s_{i-1}\| \cdot \|h_j\|} \quad (2.14)$$

$$Sim_{ij} = Similarity(s_{i-1}, h_j) = MLP(s_{i-1} \cdot h_j) \quad (2.15)$$

使用不同的相似度计算方法，获得的相似度得分的取值范围也不同。无论使用哪种相似度计算方法，第二阶段都需要对获得的相似度得分进行数值转换，论文使用 SoftMax 的计算方式，其作用是将原始的相似度得分归一化，得到权重之和为 1 的概率分布；利用 SoftMax 的机制突出相关度更高的元素的权重。计算公式如下式 2.16 所示：

$$\alpha_{ij} = SoftMax(Sim_{ij}) = \frac{e^{Sim_{ij}}}{\sum_{k=1}^m e^{Sim_{ik}}} \quad (2.16)$$

计算结果 α_{ij} 为时刻 i 编码器隐藏状态向量 h_j 对应的权重。如式 2.17 所示，对 $\{h_1, h_2, \dots, h_m\}$ 进行加权求和得到当前时刻的背景向量 c_i ，解码神经网络根据这个背景向量和历史隐藏层状态向量输出目标元素。

$$c_i = \sum_{j=1}^m \alpha_{ij} h_j \quad (2.17)$$

通过如上三个阶段的计算，即可得到不同时刻的解码器的背景向量，结合注意力机制的编码-解码框架都采用了如上三个阶段的计算过程。

2.5 本章小结

本章介绍了深度学习的基础理论知识，其中详细分析了标准循环神经网络 RNN 和长短时记忆网络 LSTM 两种神经网络结构。LSTM 作为 RNN 的改进形式，一定程度上弥补了 RNN 不能处理长期依赖的缺陷，但是随着序列长度增加，LSTM 的表现也会明显下降。在此背景下，本文介绍了经典的编码-解码框架，还介绍了其改进形式：结合注意力机制的编码-解码框架。注意力机制使编码器将更多的“注意力”集中在和待生成的目标词语相关度高的输入元素上，进一步解决了长期依赖的问题。

第三章 中文对联生成模型研究

中文对联生成任务与机器翻译任务类似，都可以看作是一个序列映射到另一个序列的问题。对于机器翻译任务来说，输入是一种语言的序列，输出的是另一种语言的序列。对于对联生成任务，输入上联，输出与之对仗的下联。

本章首先对已有的研究工作进行总结和模型实现。目前，使用神经网络进行对联生成的主要方法包括 Sutskever 使用的基于编码-解码框架的方法^[20]，因此本章将传统的编码-解码框架用于对联生成，编码器与解码器均使用 LSTM 作为神经网络单元结构。使用 LSTM 构建模型是因为标准 RNN 的梯度爆炸和梯度消失问题。此外，论文还将在机器翻译任务上大获成功的注意力机制和经典编码-解码框架相结合，以加强上下联之间对应位置词语之间的联系，结合注意力机制的编码-解码框架的编解码器同样使用 LSTM 作为神经网络单元结构。

最后，本章使用完全基于注意力机制的 Transformer 模型用于对联的自动生成。设计了相应的实验，将基于 Transformer 的对联生成模型和已有的对联生成模型进行比较，证实了注意力机制应用于对联自动生成任务比传统的编码-解码效果更好。

3.1 基于编码-解码结构的对联生成系统

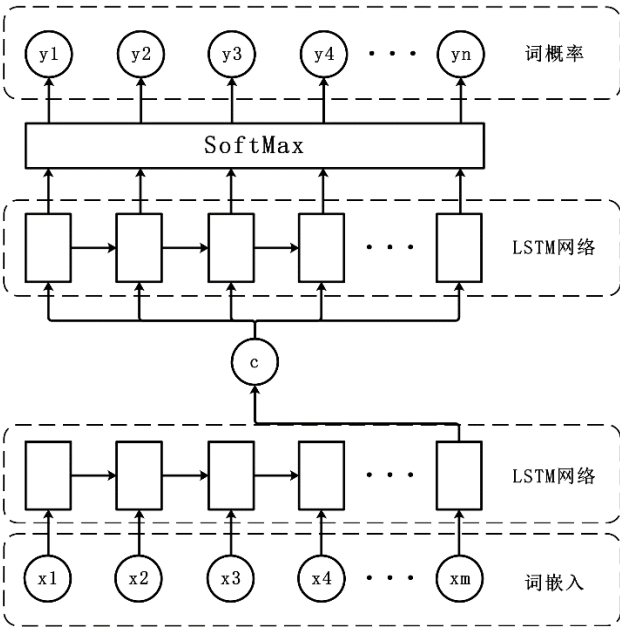


图 3.1 基础编码-解码结构对联生成模型

传统的基于编码-解码结构的对联生成模型框架如图 3.1 所示，首先需要对上联进行词嵌入(Word Embedding)。对联生成任务要转化为机器学习问题，首先要将对联中的文字数学化，

即将文字转化为计算机可以处理的向量。

因此,在将上联输入对联生成模型之前,先要将语料库中的每个词表示为一个向量的形式。在对神经网络进行训练时,需要在输入序列的头部和尾部分别加上开始标志“SOS”(Start Of Sequence)和结束标志“EOS”(End Of Sequence),开始标志和结束标志可以帮助模型在判断当前序列的开始条件和终止条件,随后将分布式的词向量数据输入编码器端神经网络。关于词向量的相关知识和背景将在第四章详细介绍。

上联的词序列在通过词嵌入层之后,形成词向量序列 $\{x_1, x_2, \dots, x_m\}$ (x_m 为表示上联各词的向量),词向量序列到达编码器 LSTM 神经网络层, LSTM 逐步处理序列中的每一项,在 t 时刻,编码网络根据输入 x_t 以及前一时刻的神经网络隐藏状态 h_{t-1} ,计算出当前时刻的隐藏状态 h_t ,重复该步骤直至完成所有输入。将最后一个 LSTM 网络单元输出的隐藏状态 h_m 作为表示输入序列所有信息的背景向量 c ,输入解码神经网络,解码器通过背景向量 c 、前一时刻的神经网络隐藏状态 s_{t-1} 和已经生成的历史序列来预测 t 时刻的下联词语,解码器各个神经网络单元的输出经过 SoftMax 层,得到下联对应词语的概率分布。

3.2 结合注意力机制的编码-解码框架的对联生成方法

结合注意力机制的编码-解码框架的对联生成模型结构如图 3.2 所示,与传统的基于编码-解码框架的对联生成模型不同之处在于,本方法使用双向神经网络 BiLSTM 组成编码器,同时解码器结合了注意力机制。

同样,首先需要对上联进行词嵌入,得到词向量序列 $\{x_1, x_2, \dots, x_m\}$,将词向量序列 $\{x_1, x_2, \dots, x_m\}$ 输入编码器。本模型中的编码器由 BiLSTM 神经网络组成,是一个双向编码器,由一个前向 LSTM 和一个后向 LSTM 组成。前向 LSTM 按照 $\{x_1, x_2, \dots, x_m\}$ 的顺序处理词向量序列,得到神经网络隐藏状态序列 $\{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_m\}$ 。后向 LSTM 按照 $\{x_m, x_{m-1}, \dots, x_1\}$ 的顺序处理词向量序列,得到神经网络隐藏状态序列 $\{\tilde{h}_1, \tilde{h}_2, \dots, \tilde{h}_m\}$ 。因此对于上联序列中的词向量 x_i , $i = 1, 2, \dots, m$,其对应的编码器神经网络隐藏状态为 $h_i = [\vec{h}_i; \tilde{h}_i]$,双向编码器从两个方向处理上联,神经网络隐藏状态包含了两个方向的依赖关系,加强了上下文之间的联系。

之后,根据编码器输出的神经网络隐藏状态,结合注意力机制的解码器开始进行解码。假设当前时刻为 t ,解码器根据时刻 $t-1$ 的解码神经网络隐藏状态 s_{t-1} 和编码器每个神经网络单元的隐藏状态向量 h 的相关性,赋予每个隐藏状态向量不同的权重,对编码器的隐藏状态向量序列按此权重加权求和得到背景向量 C ,之后的解码器处理和上一小节中传统的编码-解

码框架相同。在解码器输出每一个目标下联的元素时都要更新权重。简而言之，每生成下联的一个词语的时候关注上联的部分信息，比如生成下联“鸟”时，关注权重值比较大的上联中的“鱼”。

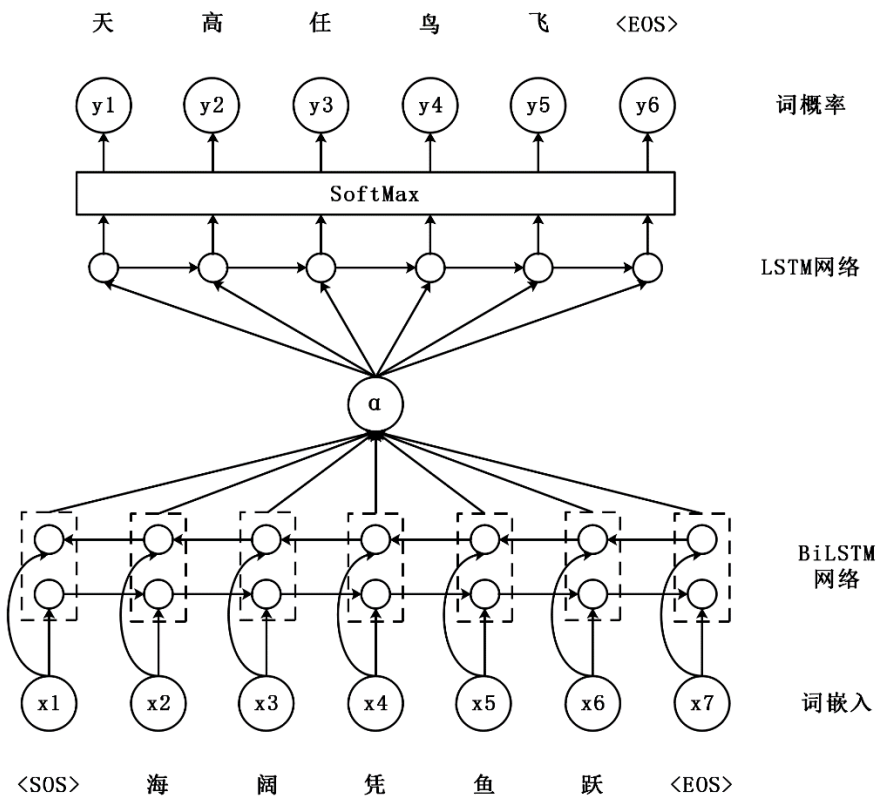


图 3.2 基于双向编码器和注意力机制的对联生成模型框架

3.3 基于注意力机制的中文对联生成系统

3.3.1 Transformer 结构介绍

采用 LSTM 等循环神经网络作为编码-解码框架的内部结构，在 t 时刻，神经网络的隐藏状态 h_t 都依赖于前一个时刻的隐藏状态 h_{t-1} 和神经网络的输入 x_t ，这种顺序计算的性质严重限制了神经网络训练的并行性。尽管有些研究者使用分解技巧^[42]和条件计算^[43]提升了计算效率，但是计算依然还是顺序执行的。

此外，尽管 LSTM 及其各种改进形式一定程度上改善了普通 RNN 网络的梯度消失和梯度爆炸问题，但是对于超长距离依赖问题依然无法解决。如何对模型进一步改进，需要进一步的研究。

自然语言处理任务，本质上主要还是需要解决三个问题：源语句内部的关系；目标语句内部的关系；源语句和目标语句之间的关系。仅仅在编码器与解码器之间使用注意力机制辅

助处理任务，依然无法完全捕捉这三种关系。

为了解决这些问题,Ashish Vaswani 等人提出了完全基于注意力机制的 Transformer 模型^[40]。Transformer 与传统的编码-解码模型最大的不同点是,传统的编码-解码模型总是依赖于 RNN 或 CNN 的神经网络结构和注意力机制的结合;而 Transformer 则完全使用注意力机制获取输入与输出之间的全局依赖关系。注意力机制计算的并行性保证了 Transformer 模型计算的并行性,缩短了神经网络的训练时间,提高了计算效率。Transformer 的结构如图 3.3 所示。

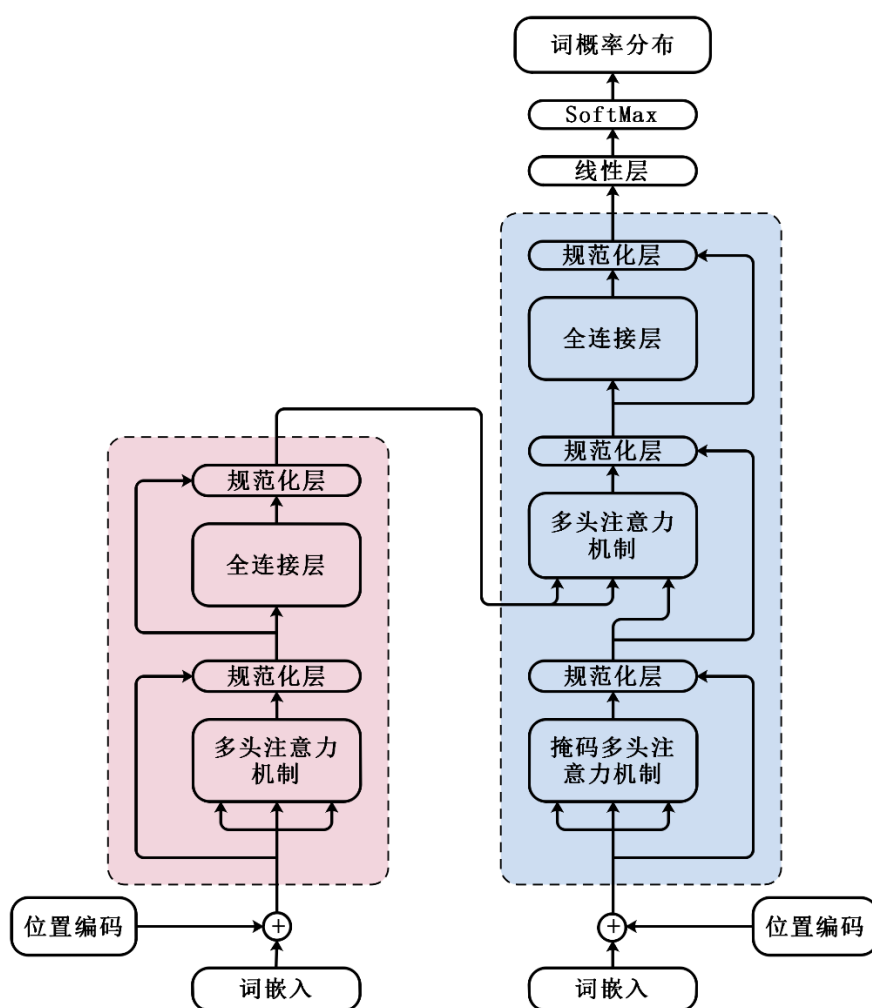


图 3.3 Transformer 结构图

Transformer 本质上依然是经典的编码-解码框架,如上图 3.3 所示,左边虚线框为编码器,右边虚线框内为解码器, $N \times$ 表示编码器和解码器都由 N 层如图所示的结构堆叠而成。其核心是多头注意力机制(Multi-Head Attention)。

(1) 多头注意力机制

如 2.3 节中介绍,注意力机制可以抽象表述为编码器的输出经过加权求和,再输入到解码器中。具体来说,目标输出序列的查询向量 *Query* 和输入序列中的 *Key* 向量进行相似度计算,根据得到的权重,对源序列中的 *Value* 向量加权求和。在实际的应用中,由于注意力机制计

算的并行性，可以同时计算一组 *Query* 值的注意力函数，这一组 *Query* 的值可以用矩阵 Q 表示，同理，源序列中 *Key* 和 *Value* 的值可以用矩阵 K 和 V 表示。传统的注意力机制的公式即可表示为如下 3.1 所示的形式：

$$Attention(Q, K, V) = SoftMax(QK^T)V \quad (3.1)$$

Transformer 没有使用普通的注意力机制，而是采用缩放点乘的注意力机制。缩放点乘注意力机制在传统注意力机制的基础上引入了一个缩放因子 $\sqrt{d_k}$ ，其具体的表达式如下式 3.2 所示：

$$Attention(Q, K, V) = SoftMax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3.2)$$

其中，在自然语言处理任务中， Q 为目标文本序列当前的隐藏状态矩阵，维度为 d_q ； K 为源文本序列中用于相似度计算的隐藏状态矩阵，维度为 d_k ； V 是待加权的输入文本特征矩阵，维度为 d_v 。使用 $\sqrt{d_k}$ 作为缩放因子的作用在于，当 d_k 很大的时候， Q 和 K 进行点乘计算得到的结果维度很大，使得结果位于 SoftMax 函数梯度很小的区域，而除以一个缩放因子后，可以改善这种情况。缩放点乘注意力机制的计算结构示意图如图 3.4 所示。

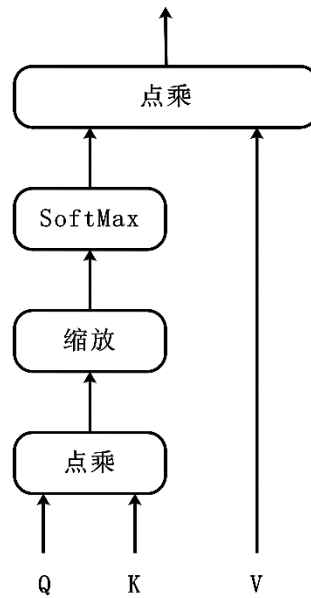


图 3.4 缩放注意力机制网络结构图

注意力机制还有一种特殊的应用形式，即自注意力机制。注意力机制通常涉及编码网络和解码网络两个神经网络的隐藏状态 h_i 和 s_j ，前者是输入序列 i 位置的隐藏状态向量，后者是输出序列在 j 位置的隐藏状态向量。而自注意力机制，本质上即输出序列和输入序列是相同的，即计算同一个语句内部元素之间的相关程度。

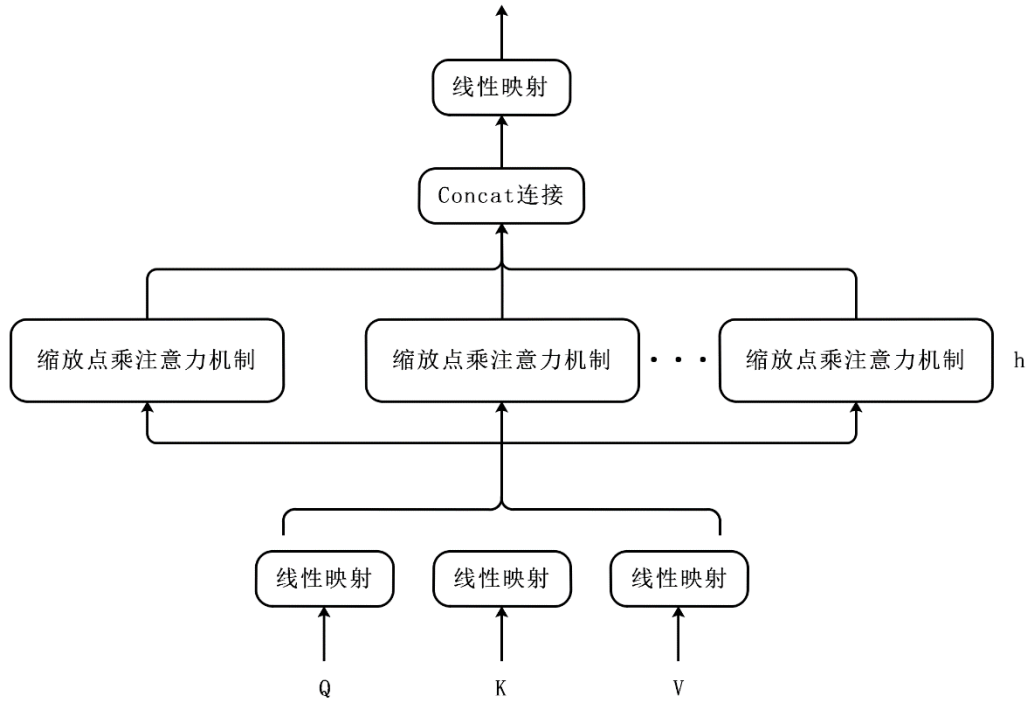


图 3.5 多头注意力机制结构

Transformer 没有直接对 *Query*、*Key* 和 *Values* 直接进行注意力计算，而是采用了多头注意力机制。多头注意力机制，顾名思义，由多个缩放点乘注意力机制组成，将 *Query*、*Key* 和 *Values* 分别通过一个线性映射 Linear 网络层之后，分成 h 份，对每一份都分别进行缩放点乘注意力计算， h 是多头注意力机制的头数。之后将 h 个头的注意力计算的结果进行合并，最后通过线性层输出。使用多头注意力机制的目的是从多个维度，更充分地获得句子内部和句子之间的依赖关系。多头注意力机制的结构图如图 3.5 所示。计算形式如下式 3.3, 3.4 所示：

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^o \quad (3.3)$$

其中

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3.4)$$

W_i^Q 、 W_i^K 和 W_i^V 是第 i 个缩放点乘注意力机制的 Linear 映射函数， W^o 是对各个头的注意力输出进行拼接后的向量的 Linear 映射函数。

(2) 编码器

Transformer 的编码器由若干个结构完全相同的网络堆叠而成。如图 3.3 所示，每个网络包括多头自注意力机制 (multi-head self-attention) 和一个前馈神经网络 (position-wise feed-forward network) 两个部分。此外，在多头自注意力机制和前馈神经网络层的后面都附加了一个残差连接网络 (Residual connection)^[44] 和一个规范化层 (Layer normalization)。

多头自注意力机制在句子内部的每个词之间建立依赖关系，捕获句子的内部结构，生成

各个词的背景向量；全连接的前馈神经网络的作用在于将多头注意力生成的源文本序列的背景向量与当前词语信息进行整合，生成包含整个上下文的隐藏状态序列。

残差网络的作用在于解决随着网络层数的增加产生的退化问题^[44]，实验表明，残差网络更容易优化，并且能够通过增加深度提高神经网络的准确率。层规范化有很多种类型，其目的都是一致的，就是将输入转化为 0 均值且方差为 1 的数据。在将数据输入激活函数前进行归一化，可以避免数据位于激活函数的饱和区。

(3) 解码器

Transformer 的解码器与编码器类似，由若干个完全相同结构的网络组成，但是解码器在编码器结构的基础上又增加了一个上下文注意力机制层，上下文注意力机制层即进行下联和上联之间的注意力计算，和传统编码-解码模型中使用的注意力机制作用相同，解码器的每个组成部分包含三个网络子层。

第一个子层是多头自注意力机制层，与编码器中结构类似，此注意力机制负责计算目标句子内部各词之间的相关程度，获取目标句子内部各个词之间的依赖关系，生成目标句子的背景向量。但是与编码器中的自注意力机制也有不同之处，解码器生成目标句子是一个时间步生成一个词，还是一个自回归模型，因此生成当前时刻的目标词时只能看到历史信息，因此在进行网络的训练时需要使用一定的方法“挡住”后面的信息。

第二个子层是多头上下文注意力机制层(multi-head context-attention mechanism)，这个子层是负责编码器与解码器之间的注意力计算，与传统的编码-解码框架中的注意力机制类似，用于目标文本序列隐藏状态与源文本序列隐藏状态之间的相似度计算。

第三个子层和编码器中的第二个子层作用相同，是一个全连接的前馈神经网络，负责整合自注意力网络生成的目标文本的背景向量、注意力机制生成的源文本的背景向量和当前词语的信息，提升预测下一个词的准确性。

3.3.2 基于 Transformer 框架的对联生成模型

使用基于 Transformer 的模型进行对联的自动生成任务，主要包括如下步骤：

- 1) 对待处理的上联进行预处理。首先将上联分词，将经过分词后的数据使用词向量生成方法生成可以被计算机处理的词向量序列。

- 2) 将上联词向量序列输入编码器，得到包含上联全部信息的背景向量。

- 3) 将上联的背景向量输入解码器，得到下联的背景向量，该向量包含了下联的全部信息和上联中的相关信息。

4) 将下联的背景向量表达输入一个线性层和一个 SoftMax 层, 得到下联各词的词概率分布, 最终转换为文本形式, 得到下联。

3.4 实验与结果分析

3.4.1 实验数据

本节将本章所述三种方法都应用于中文对联的自动生成任务。实验所使用的数据来自于一位名叫冯重朴_梨味斋散叶的博主的新浪博客。该数据集包含了 770491 条中文对联数据, 其中字数多于四个的对联有 740032 条, 随机选取 4000 句作为测试集, 2000 句作为验证集, 剩余 734032 作为训练集。

在进行模型训练之前, 首先需要对所有数据进行分词处理, 将对联词与词之间、词与标点符号之间做分割, 对联包含了大量的古汉语词语和平常书面口语表达中不经常用到的词语, 因此将对联进行分词会大幅度扩大词典规模, 且分词的准确性较差, 因此本文对对联语料采取了逐字分割的方式。

本章进行比较的对联生成系统中编码器和解码器两端都使用固定大小的词典。本文统计了对联语料中所有词的出现频次, 按照频率从高到低的顺序构成词典, 加上标点逗号和句号, 以及开始标志 SOS、结束标志 EOS 和填充符号 PAD, 不在词典中的字用 UNK 表示, 词典共有 9121 个元素。将上联与下联中的所有元素数字化, 即将语料中的每一行转化为词典中的索引值, 使用空格隔开。

3.4.2 实验环境

由于本文实验需要训练深度神经网络, 语料规模较大, 结构比较复杂, 计算规模较大, 神经网络训练过程需要使用 GPU 加速计算, 实验环境配置如下表 3.1 所示。

表 3.1 实验环境配置

操作系统	Windows
CPU 型号	Intel(R) Core (TM)i7-8700 CPU @3.20GHz
GPU 型号	NVIDIA GeForce GTX 1070
内存	16GB
主要工具	TensorFlow; Python; Spyder

本文的实验均在此机器上完成,其中神经网络的训练均使用 GPU 加速,使用的编程语言为 Python,版本为 3.6,使用的深度学习框架为 TensorFlow12.0,程序部署的 IDE 为 Spyder。

3.4.3 参数设置

本章将基于 Transformer 的模型用于中文对联生成任务时的性能和已有的对联生成方案的性能进行比较,包括传统的基于编码-解码框架的模型、结合注意力机制的编码-解码框架的模型两种。模型的输入都是基于 Word2vec 模型生成的词向量。都采用交叉熵作为神经网络训练的损失函数,用来刻画预测下联的概率分布和真实下联的概率分布之间的距离。都采用随机梯度下降算法(Stochastic Gradient Descent)作为优化器。

在训练传统的基于编码-解码框架的模型时,使用 LSTM 作为编码器和解码器的神经网络结构,并且在 SoftMax 层和词向量层间共享参数以减少参数数量^{[45][46]},LSTM 隐藏层的大小为 1024,编码与解码神经网络中 LSTM 结构层数都为 6 层,Batch size 的大小为 128,节点被 dropout 的概率设为 0.1,训练每 2000 步保存一次模型。

结合注意力机制的编码-解码模型神经网络训练参数与传统的基于编码-解码框架的模型相同,其与传统的基于编码-解码框架的模型的不同之处在于,编码器使用双向的 BiLSTM 作为神经网络结构,解码器依然使用 LSTM 作为神经网络结构。

训练基于 Transformer 的模型时,编码器与解码器的层数也都设为 6 层,即图 3.3 中的 N 为 6,全连接前馈神经网络的单元数为 1024,多头注意力机制的头数设为 8,Batch size 的大小为 128,节点被 dropout 的概率设为 0.1。

3.4.4 评价标准

由于中文语言含义的多样性,对联的生成并没有标准的答案。评价一副对联的好坏,需要从多个维度进行,评判者需要有专业的知识,因此评价一副对联的好坏具有很大难度,目前缺少一种可以客观公正评价的方法。实际上评价生成的对联的质量应该由专家来完成才能保证公正性,但是人工评价方法耗时耗力,且主观色彩较重,因此,也出现了一些自动化的评价方法,虽然相较于人工评价准确性略低,但是成本低,速度快,客观性强。因此,本文采用人工评价与机器评价相结合的方法对所生成的对联进行评估,主要评价方法包括以下几种。

(1) BLEU 测评方法

BLEU(Bilingual Evaluation understudy)测评方式由 Papineni 等人在 2002 年提出的, 广泛应用于机器翻译等领域, 是一种基于精确度的相似性度量方法, 用来衡量生成的译文和参考译文之间 n 元组同时出现的程度。由于机器翻译和对联生成都可以看作一个文本序列到另一个文本序列的映射, 可以认为在对联生成任务上, BLEU 方法也同样适用。

BLEU 的核心思想是比较机器翻译结果和人工翻译结果的相似性, BLEU 的得分位于 0 到 1 之间, BLEU 得分越高表示机器翻译的性能越好。具体表达式如下, 其中 BP 是一个惩罚因子, 根据系统生成句子的长度和正确文本的长度确定, l 表示系统生成的句子长度, r 表示参考句子的长度, 其作用在于调节系统生成的句子对于参考句子的完整性; w_n 表示各个 n -gram 元组的权重; p_n 表示 n -gram 元组的匹配精度, 即 n -gram 匹配的词数占总词数的比例。

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^N w_n \log p_n\right) \quad (3.5)$$

$$BP = \begin{cases} 1 & l > r \\ e^{(1-\frac{r}{l})} & l < r \end{cases} \quad (3.6)$$

BLEU 从本质上来看就是 n -gram 元组精确度的加权平均, N 可取 1 到 4 的任意整数, 而 w_n 一般取 $\frac{1}{n}$ 。随着 n 的增大, BLEU 在句子层级上的匹配度变差, 在个别句子上表现不佳也是常见的。

对于本文研究的中文对联生成问题, 采用的词汇一般都是单个汉字或者两个汉字的词组, 因此 N 取 2, 计算公式简化如下。

$$BLEU = BP \cdot \exp(0.5 \cdot \log p_1 + 0.5 \cdot \log p_2) \quad (3.7)$$

BLEU 只有在生成的下联和测试集参考下联完全一致时值才会为 1, 该方法是基于字符的完全匹配, 实际上无法理解对联的含义。

(2) 困惑度(Perplexity)

对于大多数语言模型, 评价模型性能最常用的指标是困惑度, 也称为复杂度(Perplexity)。一个语言模型在测试集上的困惑度评分越低, 说明性能越好, 困惑度的计算公式如下。

$$\begin{aligned} \text{Perplexity} &= p(w_1, w_2, w_3, \dots, w_m)^{-\frac{1}{m}} = \sqrt[m]{\frac{1}{p(w_1, w_2, w_3, \dots, w_m)}} \\ &= \sqrt[m]{\prod_{i=1}^m \frac{1}{p(w_i | w_1, \dots, w_{i-1})}} \end{aligned} \quad (3.8)$$

困惑度衡量的是语言模型对语言样本进行预测的能力, 当句子 $\{w_1, w_2, w_3, \dots, w_m\}$ 出现在语料库中的条件下, 语言模型计算这句话出现的概率越高, 说明语言模型对语料库拟合的越

好。从公式定义中可以得出,困惑度指标实际计算的是每一个单词的概率倒数的几何平均,所以,困惑的还可以表示为平均分支系数(average branching factor),即语言模型预测下一个词时候选词的可选择数量。举个例子,考虑一个由 0 到 9 的整数随机组成的长度为 10 的句子,由于每个数字出现的概率随机,因此每个数字的概率都是 $\frac{1}{10}$,在任意时刻语言生成模型都有 10 个候选词,因此该模型的困惑度为 10。在神经网络的训练中,适用的损失函数交叉熵采用的就是困惑度的对数形式。

(3) 人工评价

除了客观的自动化标准外,对联生成系统的评价还可以采用人工评价的方法。微软在 VTTChallenge(MSR Video to Language Challenge)2016 中提到了由视频生成文字的机器翻译任务人工评价的三个指标:流畅性、相关性和助盲性。流畅性:生成语句的逻辑和可读性;相关性:生成语句是否包含与原输入相关;助盲性:生成语句对视力缺陷者理解视频的帮助程度。借助微软在 VTTChallenge 上的标准和文献[13]的思路,本文提出从句法和语义两个角度对系统生成的对联进行评测。对于句法角度,评估者考虑下联和上联是否长度相等、对应位置词语是否符合对仗关系;对于语法角度,评估者考虑下联和上联在语义上是否有意义、是否连贯。表 3.2 展示了具体的评判标准,每一项标准最高设置为 5 分,得分越高性能越好。

表 3.2 人工评价标准

评判标准	标准说明	评分细则
句法	上下联对应位置词语是否对仗	1-5 分
语法	下联是否语言流畅	1-5 分
	下联和上联是否主题内容一致	
总体效果	系统生成下联的总体效果	1-5 分

3.4.5 结果分析

本节将基于 Transformer 的对联生成方法、传统的基于编码-解码框架的方法、结合注意力机制的编码-解码框架的方法比较,其中传统的基于编码-解码框架的方法、结合注意力机制的编码-解码框架的方法都是已有的相关工作中使用的^[55]。一共做了三组实验,第一组实验:基于编码-解码框架的对联生成模型;第二组实验:基于注意力机制和编码-解码框架的对联生成模型;第三组实验:基于 Transformer 的对联生成模型。

为了公平起见,三种方法使用完全相同的数据预处理方法,使用本文训练的词向量作为

模型的输入。本文采用自动评价方法和人工评价方法测试模型生成结果的性能，自动评价方法包括 BLEU 评测和 Perplexity 评测，具体的各项指标结果如表 3.3 所示。

表 3.3 对联模型综合性能指标比较

模型名称	Perplexity	BLEU	人工评价		
			句法	语法	整体
传统编码-解码框架	83.62	0.238	3.39	3.14	3.27
结合注意力机制的 编码-解码框架	79.53	0.243	3.46	3.54	3.5
Transformer	73.43*	0.272*	3.67*	3.77*	3.72*

由于人工评价费时费力，本文在测试结果中随机选取了 50 组结果，请了 20 位评价者对三种不同结果进行打分评价，这 20 位评价者当中大多数是在校学生，有 12 位是南京邮电大学的在校研究生，主修工科专业，有 6 位是南京林业大学的在校学术，主修艺术学专业，有 2 位是中学语文老师，最终结果取的是 20 位评价者打分的平均分。

如表 3.3 所示，加星号的数据表示是最优的。可以看出，结合注意力机制的编码-解码框架用于对联生成时的表现要优于传统的编码-解码框架，虽然注意力机制为编码-解码模型在 BLEU 得分上带来的提升有限，仅 0.005，但是带来了 4.09 的困惑度的下降。而完全基于注意力机制的 Transformer 模型在此基础上表现得更优秀，带来了 0.029 得 BLEU 提升和 6.1 的困惑度的下降，模型的性能得到了很大的改善。人工评分也显示，基于 Transformer 的对联生成模型表现要明显由于其他两种模型。可以得出如下结论：注意力机制可以明显改善对联生成模型的性能，基于实验结果，本文选择基于 Transformer 的模型作为基准模型，并用于后续的实验。

3.4.6 结果示例

如下表 3.4 所示，为基于 Transformer 的对联生成模型的生成示例。从表 3.4 中可以发现，尽管 Transformer 模型生成的部分结果和真实结果相差较大，甚至完全不同，但是生成的下联语言保持流畅，与上联之间对仗工整，依然是很优秀的下联。

表 3.4 Transformer 模型生成对联示例

上联	模型生成下联	真实下联
晚风摇树树还挺	春雨润花花自香	晨露润花花更红
隔岸春云邀翰墨	临江春水伴诗篇	绕城波色动楼台
露浥杏花红欲滴	风吹柳絮绿初来	春辉杨柳绿含烟
医卜精编松隐集	文章妙笔墨书香	尚书封赐益阳侯
刊花滋雨开三载	把酒迎春醉九州	喜讯乘风上九天

3.5 本章小结

本章将完全基于注意力机制的 Transformer 模型应用于对联的自动生成任务，并将其与已有两种基准方案相比较。两种基准方案包括：首先使用传统的基于编码-解码框架的神经网络模型用于对联的生成任务，接下来在此模型的基础上结合了注意力机制，并使用双向的 BiLSTM 网络替代原来的单向 LSTM 作为解码器。最后，本章针对这三种模型，在相同的数据处理方法下进行了实验，并采用多元化的评价方法对其进行比较，证实了注意力机制在对联生成任务上对模型的产生的积极作用。

第四章 加入词性特征和罕见词处理的中文对联生成模型

编码-解码框架由于其新颖的模型架构而受到学界的广泛关注,相较于传统的基于统计机器翻译的方法有更优秀的表现。但是非线性的神经网络结构使基于神经网络的对联生成模型难以利用汉语的先验知识。

同时,中文所包含的词语数目十分庞大,常见的对联中仅仅使用了很小一部分常用的词语,而绝大部分词语都只是偶尔使用。因此词典仅收录了有限的词语,如果输入对联中含有词典中不包含的未登录词,会造成对联语义的缺失。同时很少出现的低频词又会很大增加词典的规模,降低模型计算速度。如何解决系统的未登录词和低频词问题也需要进一步研究。

本章在上一章提出的完全基于注意力机制的 Transformer 的模型基础上,将汉语包含的先验知识融合到对联生成模型中。具体做法是,首先提取训练语料中每个词语的词性信息,将语料的词性信息和语料分开进行词向量训练,分别得到语料的词性向量和词向量;然后,将语料的词性向量和词向量以一定的方式结合,通过这种方法将上联的词性信息加入对联生成模型之中。将汉语的先验信息融入模型当中,一定程度上可以提升模型的性能。

本章针对模型训练和预测过程中可能遇到的未登录词和低频词,使用基于词向量相似度计算的方法,寻找与之相似的高频词语对其进行替换,采用这种做法可以一定程度上减小模型的搜索空间,对模型性能也不会造成影响。

4.1 词向量概述

对联的自动生成任务需要转化为机器学习任务,首先需要将对联使用的文字符号数字化,词向量为计算机处理自然语言提供了桥梁。词向量的基本思想是将一个词映射为一个几十到几百维的向量,所有词向量组成一个共同的向量空间,在这个向量空间中,各个词向量之间的距离和这些词语的相似度成反比,距离越近,相似度越高。本文在构建对联生成模型时,同样是将文本中的各个词语转化为词向量的形式而不是使用文本的形式,实际训练中,输入的是文本的词向量组。

4.1.1 词向量表示

将中文词语用词向量的方法表示为计算机可以处理的向量形式,是计算机自动生成对联的关键步骤,常用的词向量表示方法包括 One-Hot 表示和分布式表示两种。

(1) One-Hot 表示方法

在自然语言处理中，One-Hot 的词向量表示是最直观的词向量表示方法。One-Hot 表示方法将词语映射为一个一维向量，该向量只有一个元素为 1，其余所有元素均为 0。具体表示方法为：统计训练语料中不同词语的个数，将其个数记为 N ，并以出现频次由高到低排序作为词典，也可以采用其他排序方式；使用一个 N 维的稀疏向量表示词语，将词在词典中的位置与稀疏向量中的对应位置设为 1，其余位置为 0。

举个例子，对联中经常看到的“月”和“花”两个字，使用 One-Hot 的方法表示，假设“月”和“花”在词典中的索引分别为 3 和 9，则这两个字的 One-Hot 向量表示为 4.1 和 4.2 的形式：

$$\text{月: } [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \dots] \quad (4.1)$$

$$\text{花: } [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \dots] \quad (4.2)$$

这种词向量的表示方法有些简单粗暴，存在明显的缺陷。词向量的维度仅取决于词典规模的大小。复杂的语言模型的词典规模都很庞大，即使本文所构建的不复杂对联生成系统的词典规模也在一万左右，每个词语都表示为一个一万维的向量导致向量空间的维数灾难。这种对词汇特征的编码方式没有考虑词语之间的相关性，任何两个词的词向量之间都相互独立没有任何关联，无法体现词之间的相关性。

(2) 分布式表示方法

为克服 One-Hot 词向量表示方法的缺点，Hinton^[47]提出了分布式的词向量表示方法。分布式的词向量表示方法将词语表示为一个定长的稠密向量，向量空间中两个词语的词向量之间的距离可以反映它们之间的相似性。

如图 4.1 所示为词向量空间示意图，表示“遇”的词向量与表示“见”的词向量间距离很近，表示他们的相似性很高。将意思相近的词语的向量空间映射到二维空间后，他们是聚集在一起的。此外，采用分布式表示的词向量的维度也是可控的，通常在几十到几百维之间，不会造成维数灾难，适当降低了神经网络复杂度。但是词向量的可解释性较弱，词向量各个维度的含义是无法确切知道的，类似黑箱操作。

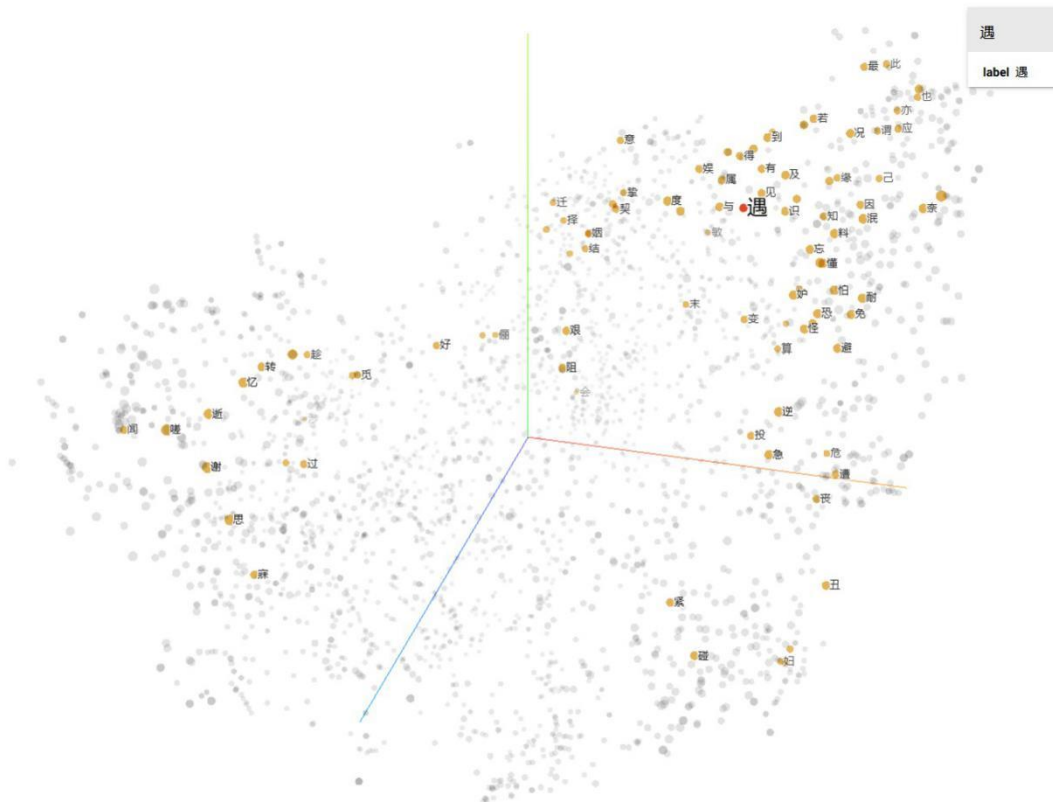


图 4.1 词向量空间示意图

4.1.2 词向量模型

(1) 统计语言模型

统计语言模型(Statistical Language Model)是描述词、语句乃至整个文档不同语法单元的概率分布模型,由其计算得到的概率分布表示该语法单元存在的可能性^[56]。对于包含 m 个词语的语句,其语言模型可以表示为如下式 4.3 的形式:

$$P(w_1, w_2, \dots, w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_m|w_1, w_2, \dots, w_{m-1}) \quad (4.3)$$

可以使用最大似然估计计算条件概率 $P(w_i|w_1, \dots, w_{i-1})$, 计算形式如下式 4.4 所示:

$$P(w_i|w_1, \dots, w_{i-1}) = \frac{N(w_1, \dots, w_{i-1}, w_i)}{N(w_1, \dots, w_{i-1})} \quad (4.4)$$

$N(w_1, \dots, w_{i-1}, w_i)$ 表示词串 $(w_1, \dots, w_{i-1}, w_i)$ 在语料中出现的次数。在实际使用中为了减少参数数量,基于马尔可夫假设,一般使用 n 元语言模型,通常可以认为一个词的出现仅依赖于其前面 n 个词,模型因此可以简化为式 4.5 的形式:

$$P(w_1, w_2, \dots, w_m) = \prod_{i=1}^m P(w_i|w_{i-n+1}, \dots, w_{i-2}, w_{i-1}) \quad (4.5)$$

n -gram 模型的参数同样可以使用最大似然估计方法计算:

$$P(w_i|w_{i-n+1}, \dots, w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-n+1}, \dots, w_{i-1}, w_i)}{\text{Count}(w_{i-n+1}, \dots, w_{i-1})} \quad (4.6)$$

n -gram 语言模型将语料的词序信息考虑在内, 其能够捕捉目标词语前 $n-1$ 个词的信息, 但又存在如下缺点: 1) 使用的数据规模有限, 用于训练模型参数的语料存在稀疏性问题, 数据稀疏性又会导致零概率问题; 2) 无法建模出词语之间的相似性, 导致泛化能力弱。为了弥补统计语言模型的缺陷, 现有的工作大多采用基于神经网络的词向量训练方法。

(2) 神经网络语言模型

在自然语言处理领域, 深度学习的重要应用之一就是挖掘文本特征的分布式表示。神经网络语言模型从计算概率角度出发, 针对目标函数建立神经网络优化模型, 实现预测词语的任务, 优化过程后模型的副产品就是词向量。

Xu 等人在 2000 年最先提出利用神经网络建模二元语言模型^[48], 而同年 Bengio 提出神经网络语言模型 NNLM(Neural Network Language Model)^[49]的概念, 神经网络语言模型才广为人知。

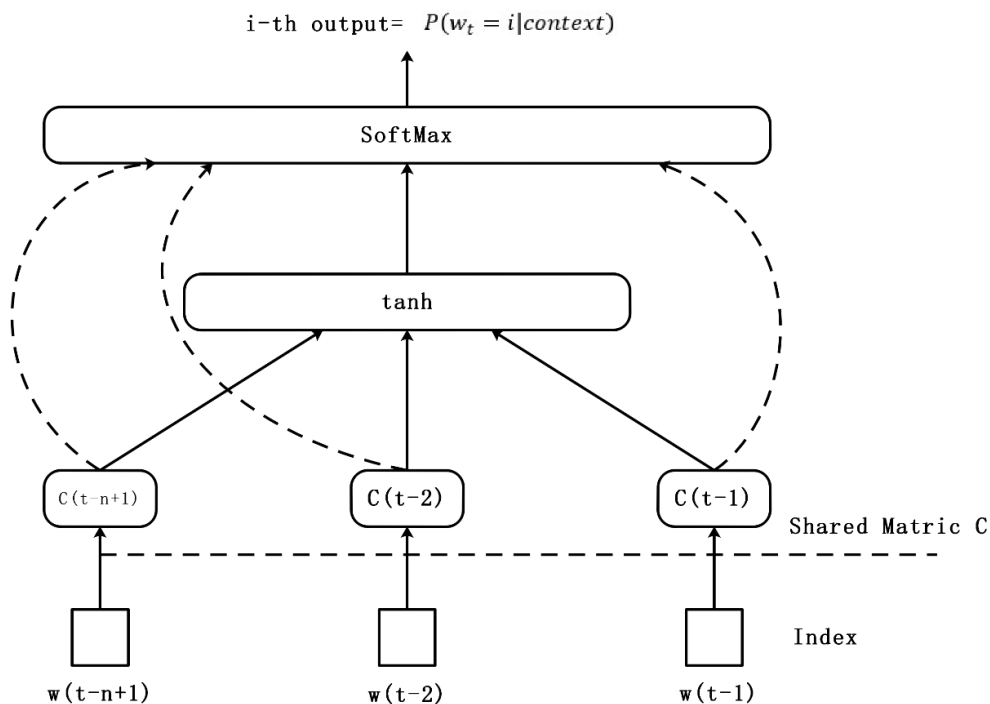


图 4.2 n 元神经网络语言模型结构

NNLM 对 n 元语言模型建模, 输入词串 $(w_{i-n+1}, \dots, w_{i-2}, w_{i-1})$, 输出单词 w_i 在给定上述词串条件下的概率分布 $P(w_i|w_{i-n+1}, \dots, w_{i-2}, w_{i-1})$, 图 4.2 为神经网络语言模型的基本结构。如图, NNLM 包括输入层、词嵌入层、隐藏层和输出层四层结构。模型接收的输入是长度为 n 的词序列, 输出的是下一个词的条件概率。首先, 输入词序列的 One-Hot 向量序列, 维度为 $n \times V$; 词嵌入层是一个大小为 $V \times K$ 的共享矩阵 C , 与输入的 One-Hot 向量序列相乘后得到的矩阵作

为词嵌入层的输出；词嵌入层的输出作为隐藏层的输入，以 \tanh 为激活函数，最后送入带 SoftMax 的输出层，输出目标词的条件概率。

该模型的计算瓶颈在于输出层的矩阵运算，Bengio 将输入层和输出层直接相连，目的是减少迭代次数，但是会对模型的表现造成一定的影响，因此，在实际使用中，输入层与输出层的直连边通常被忽略。

NNLM 最大的缺点就是参数量多，训练慢，此外，NNLM 要求输入定长序列，限制了模型灵活性。

(3) C&W 模型

Collobert 和 Weston 提出了一种以直接生成词向量为目标的 C&W 模型^[50]。语言模型的目标都是求解 $P(w_i | w_{i-n+1}, \dots, w_{i-2}, w_{i-1})$ ，其中隐藏层到输出层的矩阵运算是十分耗时的。C&W 模型为了快速且高效地生成词向量，直接对 n 元短语 $(w_{i-n+1}, \dots, w_{i-2}, w_{i-1}, w_i)$ 进行打分，而没有预测目标词 w_i 的条件概率分布 $P(w_i | w_{i-n+1}, \dots, w_{i-2}, w_{i-1})$ 。具体而言，C&W 模型对给定训练语料库中固定奇数长度的任意短语 $s = (w_{i-\frac{n-1}{2}}, \dots, w_{i-1}, w_i, w_{i+1}, \dots, w_{i+\frac{n-1}{2}})$ ，将其中心位置词 w_i 随机替换为其他词语 w_i' ，则 $s' = (w_{i-\frac{n-1}{2}}, \dots, w_{i-1}, w_i', w_{i+1}, \dots, w_{i+\frac{n-1}{2}})$ 。改变后的短语 s' 通常在语义或语法上是不正确的，C&W 模型对 s 和 s' 打分，目标是使正确短语 s 的分数 $score(s)$ 至少比错误短语 s' 的分数 $score(s')$ 高 1。具体说，即最小化式 4.7 所示的目标函数：

$$L = \sum_{s \in \mathbb{D}} \sum_{w \in \mathbb{V}} \max(0, 1 - score(s) + score(s')) \quad (4.7)$$

其中， \mathbb{D} 是训练语料库， \mathbb{V} 是由训练语料库组成的词典。C&W 模型将目标词放入输入层，直接对 n 元短语打分，语料中出现的短语得分较高。而对于随机短语，模型会对其打较低的分。通过这种方式，C&W 模型更直接地学习得到符合分布假说的词向量。

C&W 模型与 NNLM 相比最大的不同点在于，C&W 模型将目标词放入输入层，输出层也由 V 个节点变为一个节点，该节点数值表示模型对当前短语的打分，打分不具有概率特性，无需归一化操作。C&W 模型将 NNLM 模型隐藏层到输出层的 $|V| \times |h|$ 次运算降低到 $|h|$ 次，极大降低模型的时间复杂度。

(4) CBOW 模型和 Skip-Gram 模型

为了更高效地获取词向量，MikolZov 等人在 NNLM 和 C&W 模型的基础上，继续简化，保留其核心部分得到了两种结构简单、训练高效的神经网络语言模型：CBOW(Continuous Bag-of-Words Model)模型和 Skip-Gram(Continuous Skip-Gram Model)模型。Google 在 2013 年提出的目前影响力最大的 Word2vec 的词向量生成模型^{[51][52]}就是根据 CBOW 和 Skip-Gram 模

型建立神经网络进行工作。图 4.4 所示为 CBOW 和 Skip-Gram 模型的结构示意图。

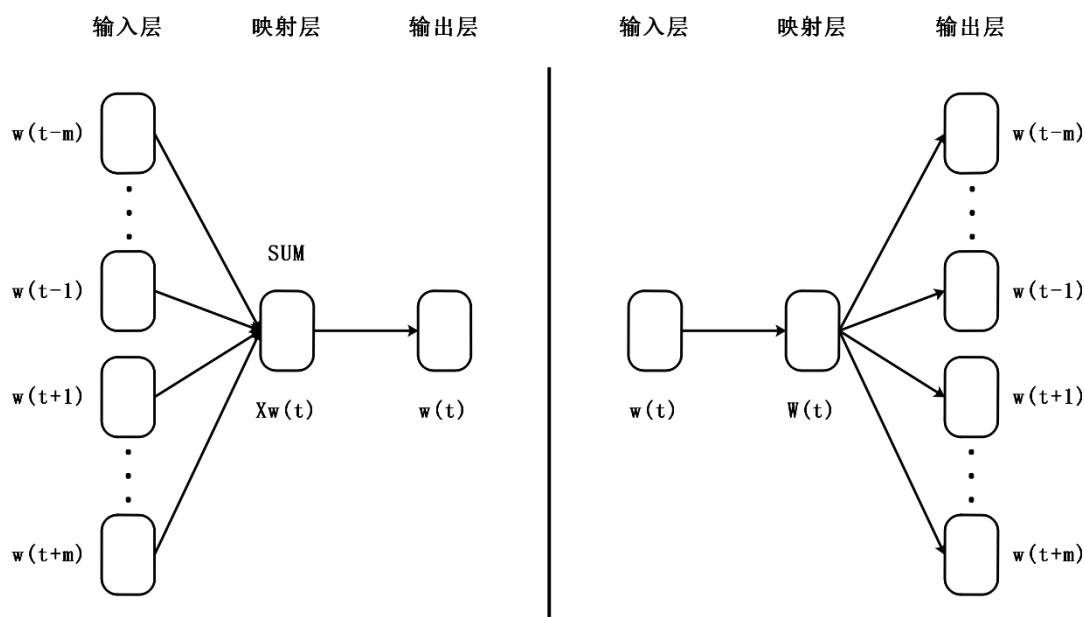


图 4.3 CBOW 和 Skip-Gram 模型的结构示意图

(左图为 CBOW, 右图为 Skip-Gram)

如图 4.3 所示, CBOW 和 Skip-Gram 模型都包括输入层、映射层和输出层三层结构。CBOW 模型参考 C&W 模型的做法, 根据当前词语的上下文 $Context(w_t)$ 预测当前词语 w_t , 而 Skip-Gram 模型则是根据当前词语 w_t 预测其上下文 $Context(w_t)$ 。其中

$$Context(w_t) = \{w_{i-m}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+m}\} \quad (4.8)$$

m 是当模型预测当前词语时使用的上下文窗口大小。

CBOW 模型还以 NNLM 为基础, 做了两个简化: 1) 抛弃了隐藏层, 输入层到映射层直接对输入词向量求和平均得到新的向量 v , 提升了模型的训练速度。其中 $v \in \mathbb{R}^d$, d 是词向量的维度; 2) 降低隐藏层到输出层计算量, 为了避免计算所有词的 SoftMax 概率, 使用 Hierarchical SoftMax 的输出层结构替代原来从隐藏层到输出 SoftMax 层的映射, 进一步简化模型。

Hierarchical SoftMax 是 Word2vec 优化模型的关键技术。CBOW 模型的输出层对应一棵二叉树, 该二叉树是以训练集语料中各词语出现的频次当作权值构建的哈夫曼树。哈夫曼树的内部节点等效于神经网络语言模型中的神经元, 其中, 根节点的词向量等效于隐藏层映射后的词向量, 叶子节点等效于神经网络 SoftMax 输出层的神经元, 叶子节点个数等于词典规模 $|V|$ 。在哈夫曼树中, 隐藏层到输出层的映射是沿着树结构逐步完成的, 因此叫 Hierarchical SoftMax。

在训练过程中, 哈夫曼树中从根节点到词 w 的路径为 p^w , 该路径一定存在, 经过的节点数为 l^w 。路径 p^w 有 $l^w - 1$ 个分支, 每个分支都是二分类问题, 每次分类都产生一个概率, 该路

径上所有分支概率相乘即为对目标词 w 的预测概率 $P(w|Context(w))$ 。

$$P(w|Context(w)) = \prod_{j=2}^{l^w} p(d_j^w | X_w, \theta_{j-1}^w) \quad (4.9)$$

其中, θ_{j-1}^w 是第 $j-1$ 个节点与向量 X_w 相连边的权重向量。 $d_j^w \in \{0,1\}$ 是路径 p^w 上第 j 个节点的编码, $d_2^w, d_3^w, \dots, d_{l^w}^w$ 表示词 w 的哈夫曼编码。

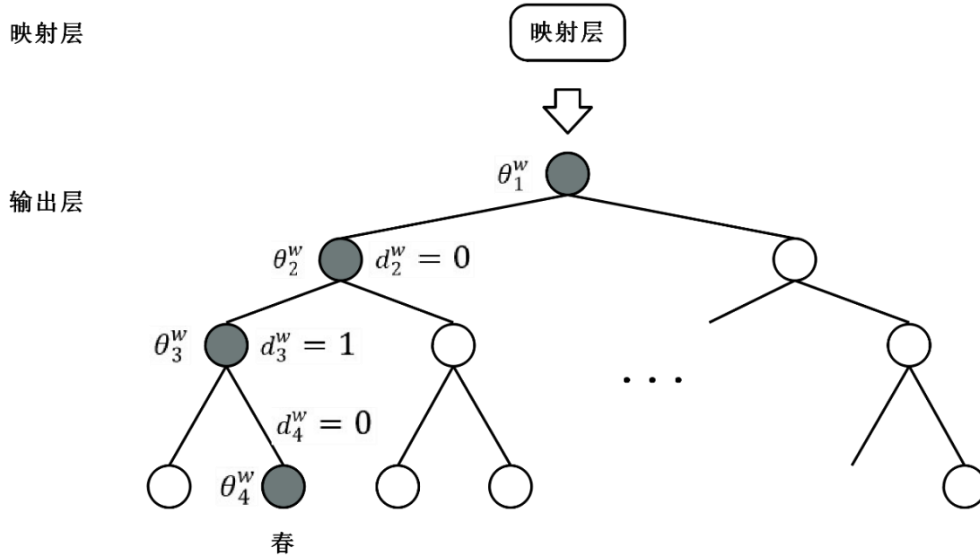


图 4.4 Hierarchical SoftMax 结构示意图

如图 4.4 是当前词语是“春”时, CBOW 模型输出层的 Hierarchical Huffman 结构图。其中灰色的四个节点 $\{p_1^w, p_2^w, p_3^w, p_4^w\}$ 构成路径 p^w , 其长度 $l^w=4$, p_1^w 是根节点, 没有 Huffman 编码, 节点 p_2^w, p_3^w, p_4^w 的编码分别为 0、1、0。因此 $P(\text{春} | Context(\text{春})) = \prod_{j=2}^4 p(d_j^w | X_w, \theta_{j-1}^w)$

对节点 $j-1$ 进行二分类时, 节点 $j-1$ 上使用逻辑回归函数, 得:

$$p(d_j^w | X_w, \theta_{j-1}^w) = \begin{cases} \sigma(X_w^T \theta_{j-1}^w), & d_j^w = 1 \\ 1 - \sigma(X_w^T \theta_{j-1}^w), & d_j^w = 0 \end{cases} \quad (4.10)$$

其中, σ 表示逻辑回归函数, $\sigma(x) = \frac{1}{1+\exp(-x)}$ 。该式表示将节点 $j-1$ 编码为 1 的概率是 $\sigma(X_w^T \theta_{j-1}^w)$, 编码为 0 的概率为 $1 - \sigma(X_w^T \theta_{j-1}^w)$ 。

CBOW 模型的目标是最大化当前目标词 w 的条件概率 $P(w|Context(w))$, 目标函数是整个语料库上的对数似然函数:

$$L = \sum_{w \in \mathbb{D}} \log P(w|Context(w)) = \sum_{w \in \mathbb{D}} \log \prod_{j=2}^{l^w} p(d_j^w | X_w, \theta_{j-1}^w) \quad (4.11)$$

使用随机梯度上升法更新网络参数和词向量, 得到最终结果。

Skip-Gram 模型使用当前词语 w_t 预测其上下文 $Context(w_t)$, 其映射层完成的是恒等映射, 并没有实际意义。Skip-Gram 模型的输出同样对应一个哈夫曼树。其条件概率公式为:

$$P(Context(w)|w) = \prod_{u \in Context(w)} P(u|w) \quad (4.12)$$

参考公式:

$$P(u|w) = \prod_{j=2}^{l^u} p(d_j^u | C(w), \theta_{j-1}^u) \quad (4.13)$$

Skip-Gram 模型的目标函数为:

$$\begin{aligned} L &= \sum_{w \in \mathbb{D}} \log P(Context(w)|w) \\ &= \sum_{w \in \mathbb{D}} \log \prod_{u \in Context(w)} \prod_{j=2}^{l^u} p(d_j^u | C(w), \theta_{j-1}^u) \end{aligned} \quad (4.14)$$

使用随机梯度上升法更新网络参数和词向量, 得到词向量。

CBOW 在小型数据集上的表现较好, Skip-Gram 模型则适用于大型数据集。由于模型的输出层的矩阵运算量很大, 对于这两种模型, 都可以使用层次化 SoftMax 和负采样(Negative Sampling)两种模型优化方法提高分类速度, 减小计算量。本小节主要使用基于层次化 SoftMax 的模型优化方法。

4.1.3 词向量模型对比

使用计算机进行对联的自动生成首先需要将文本语料表示为词向量的形式, 上一小节中介绍了常用的几种词向量生成方法, 如何找到适合本文研究的对联生成系统的词向量表示, 需要对各个词向量生成模型进行比较。

(1) 参数量

表 4.1 词向量模型参数量对比

模型类别	参数量
神经网络语言模型	$(d + h) \times V + (len_{win} - 1)d \times h $
C&W 模型	$d \times V + (len_{win} \times d + 2) h $
CBOW 模型	$(2 V - 1) \times \theta + d V $
Skip-Gram 模型	$(2 V - 1) \times \theta + d V $

如表 4.1 所示为上述各种模型的参数量, d 表示词向量维度, $|V|$ 是词典大小, len_{win} 表示

模型上下文窗口大小, $|h|$ 表示隐藏层的维度。在基于层次 SoftMax 的 CBOW 模型和 Skip-Gram 模型中 $|\theta|$ 表示权重参数向量维度, 实际 $|\theta| = d$, 因此实际参数个数可以表示为 $(3|V| - 1) \times d$ 。

(2) 时间复杂度

表 4.2 词向量模型时间复杂度对比

模型类别	时间复杂度
神经网络语言模型	$O((len_{win} - 1)d \times h) + h \times V $
C&W 模型	$O(len_{win} \times d \times h)$
CBOW 模型	$d \times \log_2(V)$
Skip-Gram 模型	$d \times \log_2(V)$

神经网络语言模型的训练时间主要用在参数运算更新上, 模型的一次计算由两次矩阵运算组成, 即输入层到隐藏层、隐藏层到输出层两次运损; C&W 模型的主要运算在输入层到隐藏层; 而 CBOW 模型和 Skip-Gram 模型, 因为忽略了隐藏层和词序信息, 因此初始复杂度为 $d \times |h|$, 采用层次 SoftMax 方法优化输出层后将 $|V|$ 降至 $\log_2(|V|)$ 。

4.1.4 词向量模型参数选择

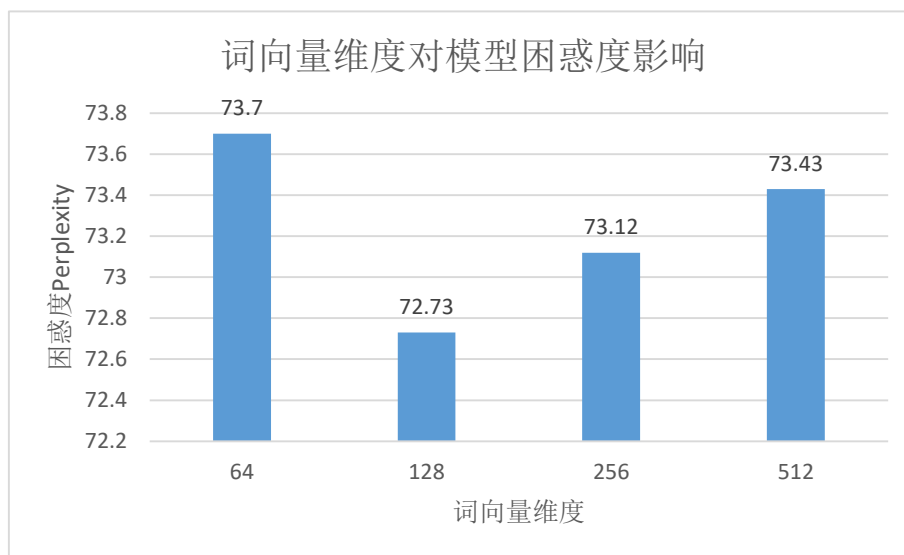


图 4.6 词向量维度对模型困惑度的影响实验结果图

在训练模型获取词向量时, 语料的规模、词向量的维度、采用的算法都会影响生成的词向量的质量。根据上一小节中对各种词向量模型的比较, CBOW 模型和 Skip-Gram 模型在参数量和复杂度两个方面的表现都要优于其他模型, 而且 CBOW 模型相较于 Skip-Gram 模型速度更快, 效率更高, 因此本文选择 CBOW 模型进行词向量的生成。本文使用 Word2vec

模块的 CBOW 模型生成实验所需的词向量, 本节主要研究词向量的维度对 Transformer 模型造成的影响, 确定本文实验中使用的词向量的维度。实验的词向量维度从 64 维取到 512 维, 即取 2 的整数次幂。实验结果如上图 4.6。

从词向量的维度对模型困惑度的影响结果来看, 词向量的维度对模型的性能存在一定但有限的影响。词向量取 128 维时, 模型的困惑度较低, 词向量维度增大后模型的困惑度反而会升高。对此可能的解释是, 本实验采用的语料库规模依然不够庞大, 高维度的词向量得不到充分训练, 效果反而不好。因此, 对于具体的任务, 选择合适规模的词向量依然需要考虑。根据本实验, 本文将在下文的各种模型之中都采用 128 维的词向量进行模型训练。

4.2 融合词性信息的对联生成模型框架

4.2.1 词性序列信息

结合注意力机制的编码-解码框架完成对联生成任务时, 编码器将输入的上联映射为一个背景向量序列, 解码器再根据背景向量序列解码出对应的下联。因此, 背景向量在对联生成模型中扮演了极其重要的桥梁的角色。因此, 将语言学先验知识融入背景向量, 在理论上有助于进一步提升模型的表现。本章在上一章的基于 Transformer 的对联生成模型的基础上, 将对联的词性信息加入到模型当中, 具体的做法是, 将语料的词性向量与其原来的词向量以某种方式相结合, 使用结合词性信息后的词向量训练语言模型, 提高模型对语言学知识的运用。

表 4.3 词性分析结果示例

原文	晚风摇树树还挺, 晨露润花花更红。
词性标注	晚/a 风/n 摇/v 树/n 树/n 还/d 挺/d , /wp 晨/a 露/n 润/v 花/n 花/n 更/d 红/a 。 /wp

词性信息特征是指训练集语料中每个词语的语法范畴、词性或词汇类别的标注信息。词性信息特征的获取是自然语言处理中一项重要的基础研究。要获取对联的词性信息首先需要对其进行句法分析, 指分析句子中各个词语的语法功能。例如, 对于上联“晚风摇树树还挺”, “晚风”是主语, “摇”是谓语, “树”是宾语。句法分析通过构造句法树确定句子的组成结构和句法成分之间的关系。现有的中文句法分析工具主要包括斯坦福大学自然语言处理小组开发的 Stanford CoreNLP, 哈工大开发的语言技术平台 LTP, 清华大学词法分析器 THU 等。本章使用的是哈工大社会计算与信息检索研究中心发布的 LTP(Language Technology Platfor)

中文自然语言处理工具集，词性分析结果示例如上表 4.3 所示。其中，各个词性的含义如下表 4.4 所示：

表 4.4 LTP 中文自然语言处理工具部分词性含义对照表

标志	描述	示例
a	adjective	美丽
d	adverb	很
n	general noun	苹果
v	verb	跑
wp	punctuation	， 。 ！

明显可以看到，示例中的上下联对应位置的词性是一致的，在词向量中加入词性信息能够为对联生成模型提供更多的约束条件，本文期望将词性信息特征加入到对联生成模型中，模型在训练过程中学习到这种上下联词性对仗的关系。

4.2.2 加入词性特征的对联生成模型

在对联生成任务上，和传统的基于编码-解码框架的模型及其改进形式相比，基于 Transformer 的对联生成模型的表现要更好，但是非线性的神经网络结构使得模型难以运用先验知识，考虑到对联的上下联对应位置的词语词性对仗的特性，本节将对联的词性特征融入基于 Transformer 的对联生成模型当中，理论上使模型可以学习到上下联之间的对仗关系，提高模型映射的准确性。

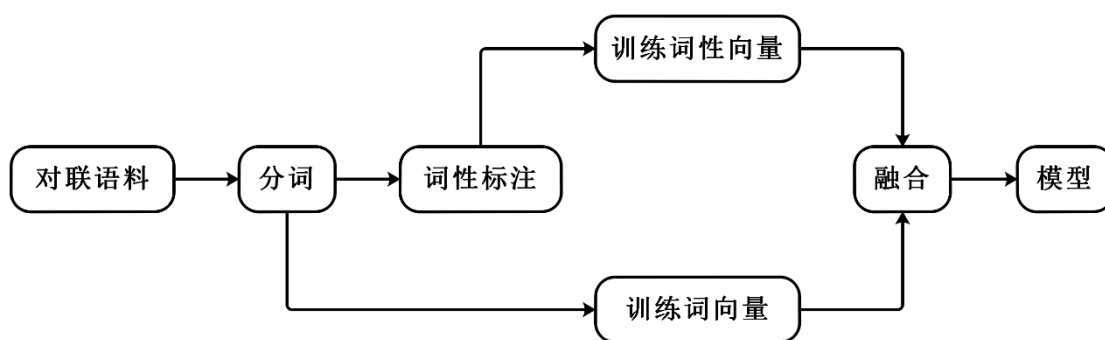


图 4.6 对联生成模型融合词性特征信息的流程图

本节将对联语料的词性信息加入模型中参与训练的具体流程如上图 4.6 所示，具体步骤如下：

1) 将对联语料进行分词，对联语料与普通书面语或口语不同，其包含了很多古汉语词语，使用分词工具的分词效果不理想。因此，本文采用逐字分词的方式，减小分词操作的难度；

- 2) 使用 LTP 词性标注工具对分词后的对联进行词性标注, 将获取到的语料词性信息和原语料分开;
- 3) 将对联语料和提取到的词性信息分别进行词向量训练, 由对联语料训练得到对联的词向量, 由提取到的词性特征信息训练得到其对应的词性向量;
- 4) 将上一步获得的词向量和词性向量以一定的方式相结合, 有多种结合方式, 如直接相加、以一定的比例的前后连接等;
- 5) 最后, 使用结合后的词向量参与模型训练。

将词性信息转化为向量形式, 和原始的词向量相结合, 组成全新的初始词向量, 参与到对联生成模型的训练当中。这种方法理论上可以将语言学的先验知识加入了模型当中, 使词向量所携带的信息更加丰富。

Transformer 模型结合词性信息的具体做法是: 模型的训练过程中, 将词性向量和词向量结合后形成的全新的词向量, 作为模型编码器和解码器底层的输入。使用加入词性信息特征的词向量作为注意力机制的初始词向量, 对任意位置的词语都对两种编码进行结合, 实现了对输入序列更充分的语义表达。

关于词向量和词性向量的结合, 本文主要采用了两种方式: 采用直接相加的方法; 采用向量拼接的方式。直接相加的方式要求词性向量和词向量保持相同的维度。向量拼合的方式时, 需要考虑词向量和词性向量的具体的维度的选择。如果词性向量的维度太低, 会导致加入模型中的词性信息不足, 模型的改善有限; 如果词向量的维度过高, 由于最终词向量的维度恒定, 导致词语原有的信息损失, 模型的性能甚至会下降。如何确定词向量和词性向量合适的维度, 需要在实验中进一步验证。而采取直接相加的方式则不用考虑这些问题。

4.3 引入低频词处理的对联生成模型框

4.3.1 未登录词和低频词问题

中文汉字的数量大约有接近十万个, 但是, 人们日常生活中使用的汉字却十分有限。根据统计, 1000 个常用汉字能覆盖约 92% 的书面资料, 2000 个汉字能覆盖 98% 以上, 3000 字就可以达到 99%, 绝大多数汉字在日常生活中使用频率都很低, 甚至是不会出现。本文实验中使用的数据集包含 70 多万条对联, 在这些对联中, 大部分词语的出现频率也是很低的。如图 4.7 所示为训练集中所有出现的词语的词频统计图, 可以看出, 词语的使用频率存在明显长尾效应。

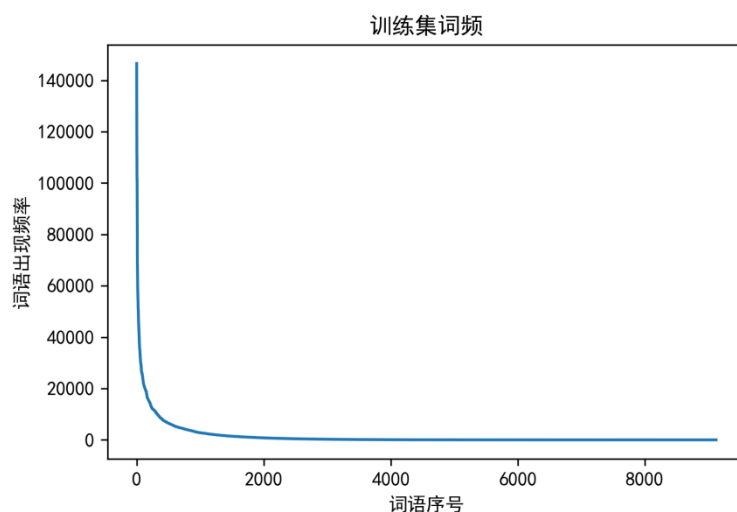


图 4.7 训练集词频统计

本节主要研究对联生成模型中的未登录词和低频词处理的问题。在神经网络语言模型中，出现未登录词的原因主要有两个：1) 词语规模庞大，训练语料覆盖不足；2) 实际的神经网络训练任务中，经常需要限制目标词典的大小，导致某些词语虽然出现在训练语料中，却不被词典所包含。神经网络语言模型中遇到未登录词主要是这两个原因。而实际在语料中出现频次很低的低频词，却显著地增加了系统词典的规模。模型预测目标词语时，需要针对目标词典中的每一个词，计算其成为目标词语的概率，整个目标词典都是解码器的搜索空间，S. Jean 的研究^[53]表明，解码器的训练时间和词典规模保持线性相关甚至是线性以上的相关度，随着词典规模的扩大，无论是训练还是预测的速度都会显著下降。因此，在有限资源的情况下，神经网络语言模型需要将目标词典的规模控制在合适的大小，这样未登录词和低频词如何解决成为每个自然语言处理的研究者要考虑的问题。

在第三章中对联生成模型的训练中，本文都统一使用“UNK”符号表示系统中的未登录词。这种方法会给模型带来负面的影响：将所有未登录词都替换为“UNK”符号，会导致句子丢失部分语义信息；而大量低频词语同样会降低模型的表现：由于低频词的稀疏性，在模型的训练中，低频词语难以学习到较好的语义表示。为了解决对联生成模型中未登录词和低频词对语言模型的消极影响，本节提出了一种针对未登录词和低频词的解决方法。

4.3.2 未登录词和低频词的解决方法

业内相关研究人员为了解决未登录词和低频词对神经网络语言模型造成的影响，提出了几种解决方法，主要可以分为两大类：基于字符的解决方法和基于子词切分的解决方法。

基于字符的表示方法利用大部分语言中字符数量有限的特性，按照字符为单位，将一个

文本进行切分。在一种语言中, 尽管词语的数目庞大且不断更新, 但是单个字符的数目却是有限的。例如, 对于中英机器翻译模型, 所有的英文词语都是由不超过 128 个 (ASCII 码) 英文字母组成, 因此在编码器和解码器只需要记录 128 个字符就可以避免未登录词的出现。

基于子词切分的方法将未登录词和低频词切分为更细粒度的子词。Sennrich R 提出通过无监督学习方法将单词切分为子词, 以此解决低频词问题^[54]。

上述方法都在一定程度上改善了模型处理未登录词和低频词的能力, 但是也会造成一些问题。基于字符的表示方法可以彻底解决未登录词的问题, 但是将文本分割为字符会极大增加文本长度, 这种方法依赖模型解决长距离依赖问题的能力, 本文直接将对联文本逐字分词, 因此这种方法并不适用, 而子词切分算法会破坏句子结构, 同样存在缺点。因此, 对于未登录词和低频词的问题, 仍然需要进一步的研究。

本文利用词向量的相似度计算方法, 对于语料中的低频词和在目标词典之外的未登录词, 计算它们和词典中其他词语的相似度, 选择与之最相近的词语对其进行替换。通过这种方法, 将训练语料中出现的低频词都从词典中去除, 进一步缩小词典规模, 降低神经网络的复杂度和训练时间。同时, 与使用“UNK”替换未登录词的方法相比, 使用近义词替换未登录词的方法, 理论上可以保证整个句子的语义上的完整性。

4.3.3 基于词向量相似度的低频词处理方法

在神经网络语言模型中, 整个目标词典都是模型预测当前词语的搜索空间, 搜索空间的规模大小影响模型的复杂度。理论上, 在计算资源允许的情况下, 词典的规模越大, 语言模型的性能越好。但是在实际情况中, 词典规模过大会导致模型复杂度迅速上升。神经网络语言模型可以通过限制词典规模达到降低模型复杂度的目标, 但是随着词典规模的缩小, 未登录词的问题又会大量出现。

因此, 本文提出基于词向量相似度计算的未登录词和低频词处理方法, 降低未登录词和低频词对模型造成的影响, 同时降低了系统的复杂度。本方法思路起源于基于语义相似度的信息检索方法。基于语义相似度的信息检索方法的具体含义是: 在用户输入查询的关键词之后, 搜索引擎可以得到用户查询的语义向量, 选择一个与用户的查询向量相似度最高的内容作为返回结果; 如果查询不到准确内容, 也会返回一个与用户查询最相关的内容返回。本章提出的未登录词和低频词替换方法, 在系统输入包含未登录词和低频词时, 使用词向量的相似度计算方法, 寻找与之相似度最高的高频词对其进行替换, 这种方法可以将低频词从目标词典中剔除, 减小模型的搜索空间, 同时避免未登录词被替换为“UNK”, 保持句子的语义

完整度。

在本文研究的对联生成系统中，举这样一个对低频词进行替换的例子。对于训练语料中的“鲂”（fǎng）字，根据百度汉语的解释，鲂是一种鱼类，“鲂”使用的频率较低，在第三章的对联生成系统中，如果对联中包含“鲂”，则该字就会被“UNK”替换，造成整个句子的语义上的不完整。而众所周知，在日常生活中，“鱼”字的使用频率是大大高于“鲂”的，实际在对联的生成任务中，可以认为，使用“鲂”或者“鱼”对于整个对联来说，并没有明显的区别，可以假设，和用“UNK”替换“鲂”相比，使用“鱼”对其进行替换在训练模型时，保持了句子的语义完整性，理论上模型会表现更好。使用词向量的相似度计算方法，选择与未登录词和低频词的词向量相似度最高的高频词对其进行替换，这种方法对于对联结构的破坏要明显小于使用 UNK 符号替换。如下表 4.5 所示，在实际的词向量相似度计算中，“鱼”和“鲂”的相似度确实较高。

表 4.5 相似度替换表

相似词	相似度
鳊	0.5730155110359192
鱼	0.5214133858680725
鲑	0.5020157694816589
鲢	0.49255096912384033
鲈	0.482464075088501

基于以上分析，本文提出使用词典中高频词替换未登录词和低频词的方法，减小词典规模，降低对联生成模型的复杂度。在对联包含未登录词时，保持句子的完整性。具体的替换方法如下：

假设 w_l 是未登录词或低频词， w_h 为词典中的高频词， $w_l = (x_{l1}, x_{l2}, \dots, x_{ln})$ ， $w_l = (x_{h1}, x_{h2}, \dots, x_{hn})$ ，则 w_l 和 w_h 的相似度定义为：

$$Similarity(w_l, w_h) = \frac{\sum_{k=1}^n x_{lk} x_{hk}}{\sqrt{\sum_{k=1}^n x_{lk}^2} \sqrt{\sum_{k=1}^n x_{hk}^2}} \quad (4.15)$$

定义词典中词频低于一定数值的词为低频词，该数值的具体取值需要通过实验确定，不在词典中的词为登录词。对于低频词和未登录词 w_l ，使用 Word2vec 模块里的词向量相似度计算方法，取得与之相似度最高的十个词，选择与之相似度最高且属于高频词的一个词 w_h 作为低频词 w_l 的替换。

使用高频词替换未登录词时可以避免对联语义的缺失。使用高频词替换低频词，可以将

低频词从词典中剔除，减小了词典的规模，降低模型的复杂度，但是这种替换方法也会在一定程度上影响原本词典中词语的词义，降低训练语料的质量。

如图 4.8 所示，是使用高频词替换未登录词和低频词的流程图。首先判断输入语料中的词语是否在词典中；如果输入的对联中有词典中的未登录词，则需要对未登录词与词典中的词语进行词向量的相似度计算，取得与其相似度最高的高频词语对其进行替换；如果输入对联中的所有词语都在词典当中，则再判断输入对联中的所有词语是否含有低频词，如果含有低频词，需要使用同样的方式替换掉低频词，再将替换后的对联对应的词向量输入对联生成模型；如果输入对联中的所有词语都在对联当中，且不含低频词，则直接将其对应的词向量输入模型。

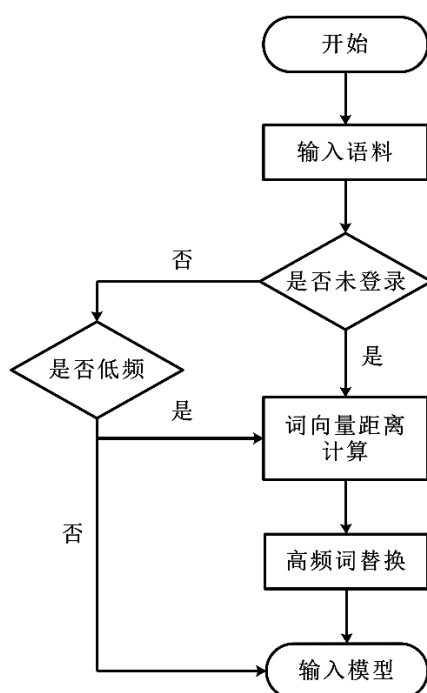


图 4.8 未登录词和低频词替换方法流程图

4.4 实验与结果分析

本章提出的两种改进策略都是在第三章中基于 Transformer 模型的基础上进行的，其训练数据和实验环境与第三章中的三个模型保持一致，按照 4.1 节中的实验，本实验中的词向量的维度选择 128 维。其他实验参数的设置与第三章中的基于 Transformer 的对联生成模型一致，本节主要包括三个部分：1) 对词向量和词性向量的结合方式进行实验；2) 将本章提出的结合词性信息的模型和与基础的 Transformer 的对联生成模型进行实验比较；3) 将进行了未登录词和低频词替换的模型与基础的 Transformer 的对联生成模型进行实验比较。

4.4.1 词性特征实验

本节采用困惑度和 BLEU 两种评测方法对模型的效果进行评价,每个模型进行 5 次实验,采用 5 组实验的均值作为衡量模型的标准。如下表 4.6 所示是不同词性向量和词向量结合方式的困惑度和 BLEU 评分数值。根据表中的实验结果可以发现,在结合了词性信息特征之后,对联生成模型和第三章中 Transformer 基准模型相比,表现有所提升。但是不同的词向量和词性向量的维度对模型也存在不同的影响。

表 4.6 结合词性特征的对联生成模型对比实验

模型名称	结合方法	Perplexity	BLEU
Transformer	/	72.73	0.272
结合词性信息的 Transformer	词性向量 128 维与词向量 128 维	70.71	0.311
	直接相加		
结合词性信息的 Transformer	词性向量 64 维与词向量 64 维拼接	72.17	0.285
结合词性信息的 Transformer	词性向量 32 维与词向量 96 维拼接	70.22*	0.331*
结合词性信息的 Transformer	词性向量 16 维与词向量 112 维拼接	71.12	0.294

从上表 4.6 中结果可以看出,和向量直接相加的方式相比,采取向量拼接的词性特征结合方式对模型的提升更明显。其中当词向量的维度选取 96 维,词性向量的维度选取 32 维时,对比基准模型,模型的 BLEU 评分提高了 0.059,困惑度降低了 2.51。其次是采用直接相加的方式,BLEU 评分提高了 0.039,困惑度降低了 2.02。而词向量维度和词性向量维度分别取 64 维和 64 维,112 维和 16 维时,结合词性信息对模型的改善效果不如采用向量直接相加的方式。原因可能是,当词性向量维度过大时,词向量自身携带的语义可能被压缩,当词性向量的维度过小,引入的词性信息特征又会不足。尽管各种词性结合方式间的差异不大,但是可以发现模型结合词性信息后,整体的表现呈现上升的趋势,符合对联的上下联对应位置的词语词性相同的特征。

4.4.2 低频词处理实验

本节通过实验证明,本章提出的使用高频词替换未登录词和低频词方法的有效性。在缩小了目标词典规模之后,比较了在不同的低频词替换比例下,基准模型和加入低频词处理后模型的性能。低频词替换的比例过高,降低了目标词典的规模,但是会影响高频词的词义,低频词替换比例过低,对模型的改善效果不明显,具体的替换比例需要通过实验验证。模型

的参数与第三章中的基准模型一致，词向量规模为 128 维，模型未引入词性信息。

对比实验结果如下表 4.7 所示，实验结果证明，当仅替换掉频次为 1 的低频词时，模型的 BLEU 评分提升了 0.004，虽然 BLEU 评分的提升有限，但是目标词典减少了 1437 个，词典规模缩小了约 16%。而替换掉词频小于等于 3 和小于等于 5 的低频词时，虽然可以进一步缩小词典规模，但是对联生成模型的性能略有下降，可能的原因是采用高频词替换低频词时，部分高频词替换的频率过高，引入噪声对其原有词义产生消极影响。

上述的未登录词和低频词的替换方法虽然可以缩小解码器目标词典的搜索空间，但是，寻找与未登录词和低频词语义相近的高频词的过程又会增大系统的计算量。寻找相似度高的高频词时，需要将低频词和词典中的每个词语进行语义相似度计算，根据统计，语料中频次为 1 的词语占词语的比例仅为 0.01%，假设输入语料总共包含 $|S|$ 个词语，原词典的规模为 $|V|$ ，则增加的计算量为 $|S||V| \times 10^{-4}$ ，替换掉频次为 1 的低频词后，目标词典规模缩小为 $0.84|V|$ ，词典规模缩小了 16%，降低的计算量为 $0.16|V|^2$ 。与目标词典缩小的规模相比，替换过程中增加的计算量几乎可以忽略不计。

虽然该替换方法对模型的表现提升有限，但是在略微提升模型生成对联质量的情况下，减小了目标词典的规模，降低了系统的复杂度，依然值得研究。

表 4.7 结合低频词处理的对联生成模型性能对比实验结果

模型名称	低频词词频	BLEU
Transformer	/	0.272
进行低频词替换的 Transformer	词频 ≤ 1	0.276*
进行低频词替换的 Transformer	词频 ≤ 3	0.271
进行低频词替换的 Transformer	词频 ≤ 5	0.27

4.5 本章小结

本章提出了两种对基于 Transformer 的对联生成模型的改进策略：1) 将词性特征信息和对联生成模型相结合，使模型可以学习到上下联对应位置词性相同的特征；2) 提出一种对未登录词和低频词替换的方法，在略微提升模型准确度的情况下，降低模型的目标词典规模。首先，本章对词向量的相关基础理论进行详细解释，将多种词向量生成模型进行对比，根据实验结果选择最合适的词向量模型，使用该模型生成词向量作为所有实验的基础。对于对联生成任务，通过实验确定最合适的词向量维度。之后，提取对联语料的词性，将对联的词性信息和原语料分开，分别进行词向量训练，得到词性向量和词向量，通过多次实验，确定了

词性向量和词向量的结合方式，以此将词性信息特征和模型相结合。实验表明，引入词性信息提高了模型的 BLEU 评分，降低了模型的困惑度。最后，本章使用词向量的相似度计算方法，提出了基于相似度计算的未登录词和低频词替换方法，将词典中的未登录词和低频词替换为语义相近的高频词，实验表明，这种替换方法在略微提升模型准确度的情况下，降低了模型的复杂度。

第五章 加入润色机制的中文对联生成模型

5.1 问题描述

对联是中华语言的一种独特的艺术形式,人们常常使用对联表达自己的情感、政治观点,或者在节日的活动场合传达信息。在中国的春节期间,人们一般以书法的形式,将对联书写在红色的纸上来烘托节日氛围。对联甚至可以作为一项竞赛的内容,在对联竞赛中,挑战者提出上联,被挑战者需要对出下联,下联需要与上联长度相等、断句相同,甚至对应位置的词语需要符合特定的语法约束。因此,创作对联需要有深厚的知识底蕴,甚至需要反复思考。

我国古代诗人在创作诗歌时往往会字字斟酌、调动语句,以求准确、妥帖地把形象物化为定型产品,唐代诗人贾岛在创作《题李凝幽居》这首作品时,曾为了一个字的使用反复推敲。对联的创作过程也和诗歌类似,也需要推敲平仄、对仗的问题。

借鉴诗人创作诗歌时反复打磨的过程,本文也为计算机创作对联引入一种润色机制,具体而言,在生成下联之后,模型将已经生成的下联当作“草稿”,再进行一次“润色”的过程。

5.2 润色机制概述

为了提升计算机创作诗歌或对联的准确性,有相关研究者已经对此做出了研究,主要还是通过将隐藏状态向量反复迭代计算来实现这种“润色”机制。

学者 Rui Yan 提出了一种名称为 iPoet 的诗歌生成模型^[25],iPoet 也是基于注意力机制和编码-解码框架相结合的神经网络模型,iPoet 逐句进行诗歌的生成。如图 5.1 所示为其模型框架图。iPoet 进行诗歌的生成主要包括写作意图表示、诗歌生成、诗歌润色三个阶段。

首先,编码器使用循环神经网络或卷积神经网络,将用户提供的写作意图关键词序列 Query 转化为向量表示,之后,iPoet 使用一个池化层将所有的关键词向量转化成一个背景向量,这个背景向量表示用户的写作意图;在诗歌生成阶段,iPoet 使用了全局 RNN 和局部 RNN 两部分组成的层次化 RNN 结构,全局 RNN 负责输出诗歌每一行的背景向量,而局部 RNN 负责输出该行每个字符概率分布;在诗歌润色阶段,iPoet 试图模拟诗人斟酌词句的过程,在诗歌生成系统中加入了润色机制,具体而言,在生成了完整的一首诗歌后,模型会将第一轮生成的诗歌的向量表示作为用户意图向量的附加信息,再次作为模型解码器的输入,进行新一轮的诗歌生成,具体过程和第一轮解码器进行的操作相同,iPoet 希望使用这种反复迭代的

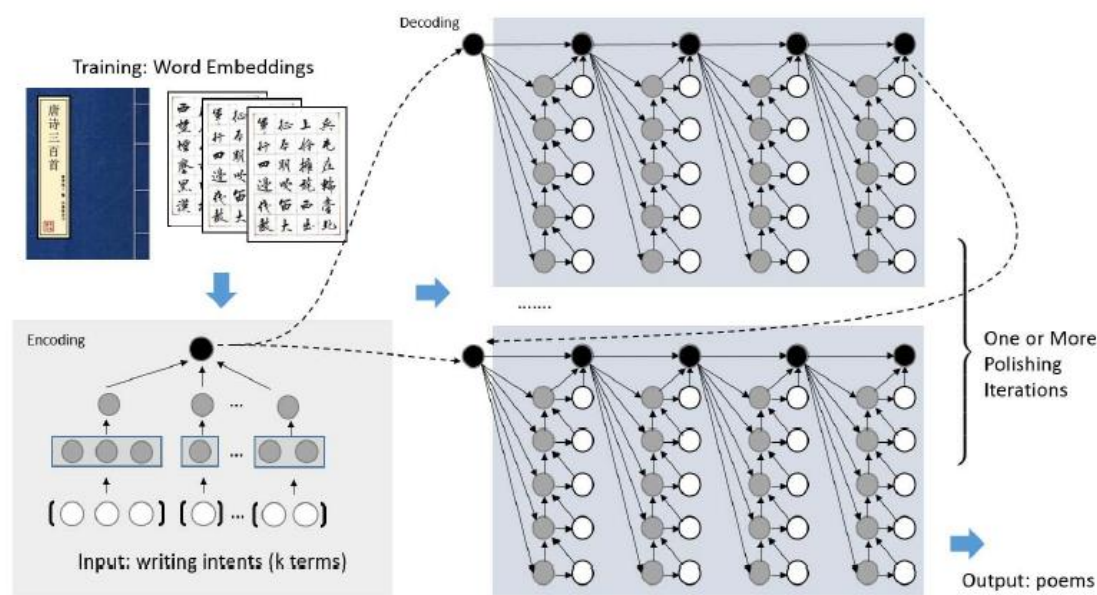


图 5.1 iPoet 模型框架图

Rui Yan 同样也将这种润色机制应用到对联的生成任务中^[55]，模型的主要结构依然是结合注意力机制的编码-解码框架。与 iPoet 使用的润色机制类似，模型引入了一种类似于 CNN 网络的润色机制，在生成下联对应的隐藏状态向量序列之后，将其作为附加信息加入到下一轮迭代当中。其结构如图 5.2 所示，在第一轮计算当中，解码器生成下联对应的隐藏状态向量序列；在第二轮计算中，解码器在产生当前时刻的隐藏状态向量序列时，会同时接收上一轮中对应的相邻位置的解码器隐藏层状态向量作为输入，如此迭代计算，直到产生最终结果。这里采用类似卷积神经网络的结构可以充分提取相邻位置的信息，使模型产生的下联连贯性更好。

总体而言，目前大多数模型采用的润色机制结构都很类似，都是对已经生成的结果再进行若干次迭代计算，本章采用的方法也是采用了这种思路。

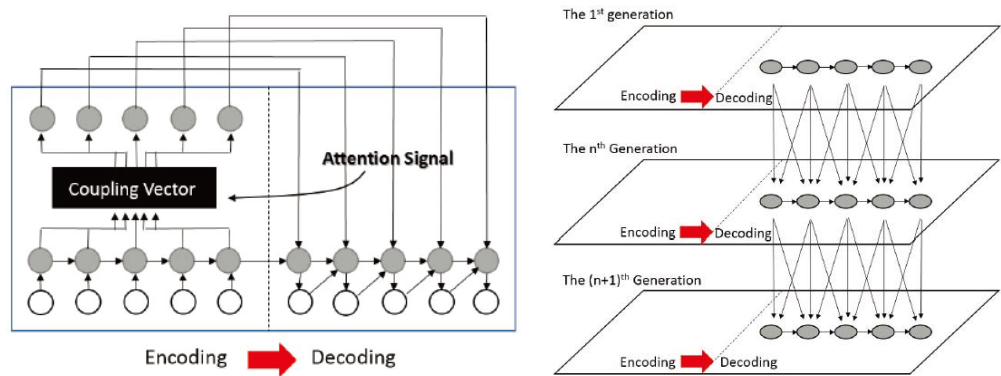


图 5.2 基于卷积神经网络的润色机制结构图

5.3 加入润色机制的对联生成模型

与诗人创作诗歌时的反复修改的过程类似，本章将一种润色机制加入到对联生成模型当中。具体来说，在生成下联之后，对联生成模型将已经生成的下联当作草稿，再次输入模型解码器中进行计算，这样模型可以对下联中的每个词语进行润色。

如图 5.3 所示，基于 Transformer 的对联生成模型的解码器由若干个完全相同的模块堆叠而成，每个模块又包括自注意力机制、上下文注意力机制和一个前馈神经网络三大部分，自注意力机制的作用是获取下联句子内部的各个词语之间的依赖关系，上下文注意力机制的作用是获取下联与上联之间的依赖关系，前馈神经网络的作用是将注意力机制生成的背景向量和当前词语的信息进行整合，输出包含下联全部信息的隐藏状态向量序列。将解码器中最后一个模块的输出向量再次输入解码器的底层模块。具体地讲，将解码器生成的下联对应的向量序列再经过一次自注意力计算和上下文注意力计算。从宏观上看，进行自注意力计算的目的是增强下联内部之间的语意连贯性，进行上下文注意力计算的目的是增强下联和上联之间的联系。采用这种将生成的下联再次输入模型进行迭代的计算方式和诗人创作诗歌的润色方式很相似。本章使用注意力机制对解码器的隐藏状态向量进行润色处理，能够使下联中每个词语的润色计算都同时进行，避免了循环神经网络或卷积神经网络的顺序计算，减小了润色机制对模型训练速度的影响。

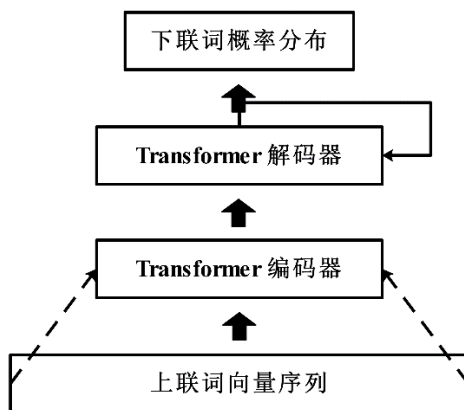


图 5.3 结合润色机制的对联生成模型结构图

模型结构的简单示意如图 5.3 所示。在得到下联对应的隐藏状态向量序列之后，将其再次返回基于 Transformer 的对联生成模型的解码器输入端，再进行一轮自注意力计算和上下文注意力计算，得到最终的下联词概率分布，本文期望使用这种迭代计算的方法可以进一步改善生成的下联，提高模型的准确性。

5.4 实验与结果分析

为了验证本章提出的润色机制在基于 Transformer 的对联生成模型中的有效性，本文首先将结合润色机制的 Transformer 对联生成模型与第四章提出的结合词性信息的模型、加入低频词处理的模型、第三章的传统的基于 Transformer 的模型分别进行实验对比，最后将本文提出的三种改进策略同时应用于 Transformer 对联生成模型，与之前三种改进单独应用的情况作对比，以验证本文提出的方法对模型的综合效果。五种模型的 BLEU 评分和困惑度评价的具体得分如下图 5.4 所示：

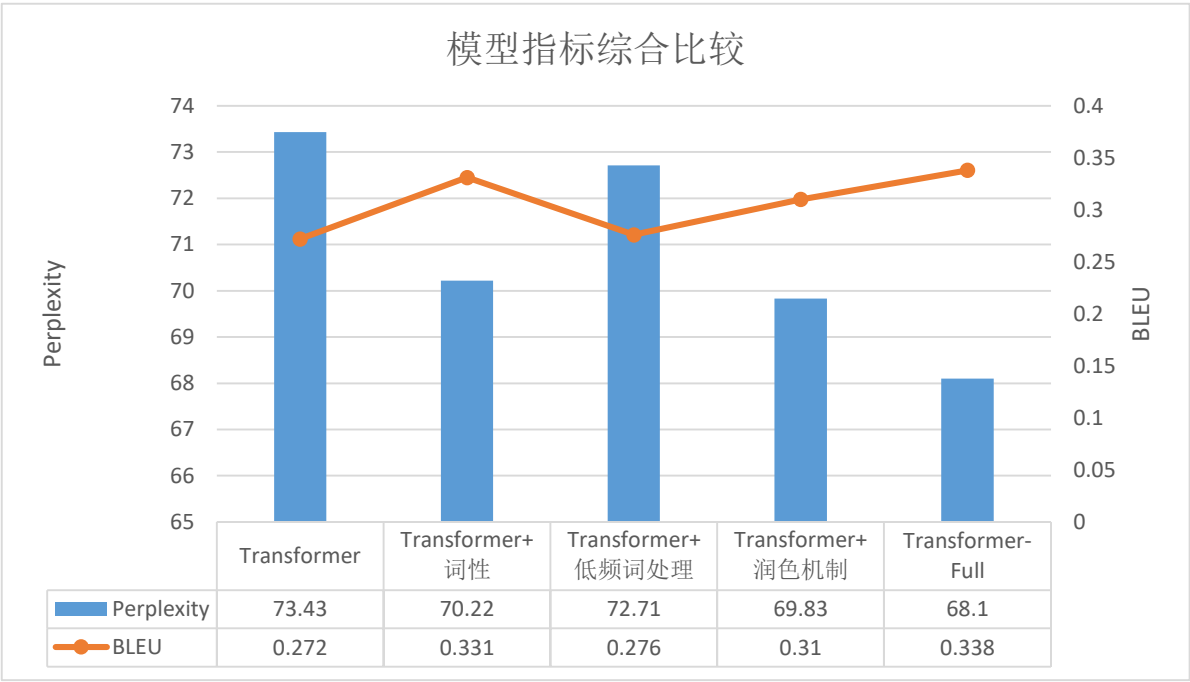


图 5.4 基于 Transformer 的基线对联生成模型及其改进模型综合比较实验结果图

由于人工评价的代价比较大，本章仅将同时应用三种改进机制的综合 Transformer 模型和基线模型进行人工评价对比，两种模型的人工评价得分具体如下表 5.1 所示，评价人员的具体构成和最终得分的计算方式与 3.4.4 节中的描述一致。

表 5.1 综合 Transformer 对联生成模型和基线 Transformer 对联生成模型人工评价表

模型名称	人工评价		
	句法	语法	整体
Transformer	3.67	3.77	3.72
Transformer-Full	3.92	3.97	3.95

本文对基于 Transformer 的对联生成模型做出如下改进：1) 加入词性信息特征；2) 加入未登录词和低频词处理；3) 加入润色机制。实验结果如图 5.4 所示，本文提出的三种改进方

法都对基础的 Transformer 模型的表现有一定的提升作用,结合了润色机制的基于 Transformer 的对联生成模型的 BLEU 得分较基准的 Transformer 模型提升了 0.038,困惑度降低了 3.6。和结合词性信息的 Transformer 模型相比, BLEU 得分低 0.021,困惑度降低了 0.39。和加入低频词处理的 Transformer 模型比较, BLEU 得分提升了 0.034,困惑度降低了 2.88。将三种改进机制综合应用于 Transformer 模型后,和基准的 Transformer 模型相比,综合 Transformer 对联生成模型的 BLEU 得分提升了 0.066,困惑度降低了 5.34,模型的表现提升较为明显。从表 5.1 中的人工评价结果中也可以看出,综合 Transformer 模型的各项得分要高于基准 Transformer 对联生成模型。

但是也可以看到,结合润色机制的 Transformer 对联生成模型的 BLEU 得分低于结合词性信息的 Transformer 模型,困惑度指标上的表现却优于结合词性信息的 Transformer 模型,其原因可能是润色机制的主要作用是提升生成的下联的语意连贯性,提升了对联生成模型预测下联的能力,使模型更好地拟合语料库,因此在困惑度上的表现较好,而结合词性信息的作用在于使模型可以学习到上下联对应位置之间的对仗关系,增强上下联之间的联系,因此 BLEU 评分略高。综合的 Transformer 模型的性能与各种单一的模型比较,各项表现都有所提升。下表 5.2 所示为改进后的综合 Transformer 对联生成模型生成的下联示例。

表 5.2 改进后的综合 Transformer 对联生成模型生成示例

上联	基础模型生成下联	改进模型生成下联	真实下联
晚风摇树树还挺	春雨润花花自香	春雨润花花正红	晨露润花花更红
隔岸春云邀翰墨	临江春水伴诗篇	临江春水醉诗篇	绕城波色动楼台
露浥杏花红欲滴	风吹柳絮绿初来	风吹柳絮绿初融	春辉杨柳绿含烟
医卜精编松隐集	文章妙笔墨书香	文章妙笔墨生花	尚书封赐益阳侯
刊花滋雨开三载	把酒迎春醉九州	把酒吟诗醉九州	喜讯乘风上九天

5.5 本章小结

本章借鉴了诗人创作作品时反复修改的创作方式,引入一种润色机制到 Transformer 模型当中,目的是提升模型生成的对联的语意连贯性。参考了其他研究者对解码器生成结果反复迭代运算的做法后,本章使用基于注意力计算的方法构造润色机制,将传统的基于 Transformer 的对联生成模型解码器的最后一层的输出结果再次输入解码器的底层,进行自注意力计算和上下文注意力计算,以增强生成的下联的语义连贯性。最后通过实验对比,将本文提出的三种改进方法共同应用于基线 Transformer 模型,并与本文提出的其他模型以及已有的研究方案

中的模型比较，证实了结合润色机制对模型的提升作用。

第六章 总结与展望

6.1 总结

对联是中华文化一种独特的艺术形式,其上下联之间讲究对仗工整、平仄协调,这要求对联创作者具备丰富的知识储备和深厚的文学素养,因此创作对联对普通人来说稍显困难。对于计算机来说,在自然语言处理领域,对联的生成也是一项比较困难的任务。近年来,深度学习技术快速发展,在如图像识别、语音识别等机器学习领域表现出色,自然语言处理作为机器学习的重要分支,深度学习技术也推动着自然语言处理技术不断发展。

本文使用 Google 提出的机器翻译领域的 Transformer 模型,在对联的自动生成领域,针对若干具体问题,做出研究和探索,研究的主要工作如下:

对已有的诗歌和对联自动生成的研究工作进行归纳总结,了解了自然语言处理领域相关深度学习技术的基本概念,如基本的神经网络单元结构,循环神经网络、卷积神经网络等,掌握了基于编码-解码框架的神经网络模型、注意力机制模型等算法,明确研究方向,抛弃了传统的基于循环神经网络或卷积神经网络的方法,完全使用注意力机制的神经网络结构进行对联的自动生成。

将完全基于注意力机制的 Transformer 模型和已有的传统的基于编码-解码框架的模型、结合注意力机制的编码-解码框架模型应用于对联的自动生成任务,对三种模型的性能实验比较,证实了注意力机制在对联的自动生成任务上的优越性。

本文做出的主要贡献如下:

1) 由于对联要求上下联之间严格对仗,结合对对联语料的研究发现,上下联对应位置的词语词性是相同的。因此,本文提出了将对联的词性信息引入到模型中。通过提取语料的词性信息,将语料的词性信息和原语料分开,分别进行词向量训练,得到词性向量和词向量,将两种向量以一定的方式结合,将对联的词性信息引入对联生成模型当中,使模型可以学习到对联的这种对仗关系。

2) 对联生成模型的训练中会遇到未登录词和低频词的问题,未登录词会导致对联语义的缺失,而语料中出现频率较低的低频词却在很大程度上增大了词典的规模,降低了模型的计算速度。本文提出了一种对未登录词和低频词进行替换的方法。利用词向量的相似度计算方法,得到与未登录词、低频词语义相似的高频词对其进行替换。通过这种方法缩小了词典规模,降低了对联生成模型的复杂度。

3) 最后, 本文借鉴了诗人创作诗歌时反复修改润色的做法, 引入了一种润色机制到对联生成模型当中, 具体做法为: 在解码器生成完整的下联对应的隐藏层状态向量序列之后, 将其返回解码器的输入端再额外进行若干轮注意力计算, 充分利用注意力机制捕捉词语间依赖关系的能力。

从实验结果来看, 注意力机制在对联生成任务上效果良好, 基于词向量的词性信息的引入和未登录词处理也进一步提升了模型的性能, 尽管润色机制对模型的性能提升有限, 但是这种思路相信对自然语言处理领域的其他研究会有参考价值。

6.2 展望

虽然注意力机制在对联的生成任务上取得了不错的效果, 但是在未来的研究工作中, 仍然有许多方面需要改进。

利用深度学习技术进行自然语言处理领域的研究, 高质量的训练语料十分重要, 本文所使用的语料主要由部分对联、古诗词中对仗较为工整的诗句和一些结构相似的短语组成, 并不全是对仗工整的对联语料, 开发大量的优秀的对联数据对进一步提升对联生成模型的表现有积极的作用。

第四章将词语的词性特征引入模型当中, 一定程度上提升了模型的表现。但是对联还讲究平仄协调, 如何将更多的语言学先验知识, 如词语的读音信息等, 也引入到模型当中, 需要进一步的研究。

使用深度学习方法进行对联的自动生成还缺少科学的评价机制。尽管对联生成类似于机器翻译, 可以借鉴机器翻译领域内常用评价指标, 如 BLEU 的基于精确度的相似性度量和 ROUGE 的基于召回率的相似性度量等评价方法, 还可以使用神经网络语言模型的常用评价指标困惑度等。但是汉语的多样性给予了对联很大的灵活性, 给定的上联可能有多个下联都符合要求, 使用此类评价方法对模型的表现进行评估都缺乏合理性。而人工评价有较强的主观性且代价较大。因此, 如何制定一个合理的对联生成的评价标准, 这个问题值得进一步研究。

此外, 尽管本文提出的润色机制对模型的效果并没有明显提升, 但是这种思路依然值得深入探索。

总而言之, 对联的自动生成是一项很有意义的研究方向, 对推广中华优秀传统文化具有很大帮助, 值得研究者们继续关注和探索。

参考文献

- [1] Manurung H. An evolutionary algorithm approach to poetry generation[J]. 2004.
- [2] Tosa N, Obara H, Minoh M. Hitch haiku: An interactive supporting system for composing haiku poem[C]//International Conference on Entertainment Computing. Springer, Berlin, Heidelberg, 2008: 209-216.
- [3] Netzer Y, Gabay D, Goldberg Y, et al. Gaiku: Generating haiku with word associations norms[C]//Proceedings of the Workshop on Computational Approaches to Linguistic Creativity. 2009: 32-39.
- [4] Oliveira H. Automatic generation of poetry: an overview[J]. Universidade de Coimbra, 2009.
- [5] Gervás P. Wasp: Evaluation of different strategies for the automatic generation of spanish verse[C]//Proceedings of the AISB-00 symposium on creative & cultural aspects of AI. 2000: 93-100.
- [6] Wu X, Tosa N, Nakatsu R. New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system[C]//International Conference on Entertainment Computing. Springer, Berlin, Heidelberg, 2009: 191-196.
- [7] Gervás P. An expert system for the composition of formal spanish poetry[M]//Applications and Innovations in Intelligent Systems VIII. Springer, London, 2001: 19-32.
- [8] Gonçalo Oliveira H R, Cardoso F A, Pereira F C. Tra-la-Lyrics: An approach to generate text based on rhythm[C]//Proceedings of the 4th. International Joint Workshop on Computational Creativity. A. Cardoso and G. Wiggins, 2007.
- [9] Kempe V, Levy R, Graci C. Neural networks as fitness evaluators in genetic algorithms: Simulating human creativity[C]//Proceedings of the Annual Meeting of the Cognitive Science Society. 2001, 23(23).
- [10] 周昌乐, 游维, 丁晓君. 一种宋词自动生成的遗传算法及其机器实现[J]. 软件学报, 2010, 21(3): 427-437.
- [11] Jiang L, Zhou M. Generating Chinese couplets using a statistical MT approach[C]//Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008). 2008: 377-384.
- [12] Zhou M, Jiang L, He J. Generating Chinese couplets and quatrain using a statistical approach[C]//Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1. 2009: 43-52.
- [13] Zhang X, Lapata M. Chinese poetry generation with recurrent neural networks[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 670-680.
- [14] Kalchbrenner N, Blunsom P. Recurrent continuous translation models[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013: 1700-1709.
- [15] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.
- [16] Serban I V, Sordoni A, Lowe R, et al. A hierarchical latent variable encoder-decoder model for generating dialogues[C]//Thirty-First AAAI Conference on Artificial Intelligence. 2017.
- [17] Cho K, Courville A, Bengio Y. Describing multimedia content using attention-based encoder-decoder networks[J]. IEEE Transactions on Multimedia, 2015, 17(11): 1875-1886.
- [18] Yang J, Price B, Cohen S, et al. Object contour detection with a fully convolutional encoder-decoder network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 193-202.
- [19] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation:

- Encoder-decoder approaches[J]. arXiv preprint arXiv:1409.1259, 2014.
- [20] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.
- [21] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.
- [22] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]//Advances in neural information processing systems. 2014: 2204-2212.
- [23] Wang Q, Luo T, Wang D, et al. Chinese song iambics generation with neural attention-based model[J]. arXiv preprint arXiv:1604.06274, 2016.
- [24] Wang Z, He W, Wu H, et al. Chinese poetry generation with planning based neural network[J]. arXiv preprint arXiv:1610.09889, 2016.
- [25] Yan R. i, Poet: Automatic Poetry Composition through Recurrent Neural Networks with Iterative Polishing Schema[C]//IJCAI. 2016: 2238-2244.
- [26] Yu L, Zhang W, Wang J, et al. Seqgan: Sequence generative adversarial nets with policy gradient[C]//Thirty-First AAAI Conference on Artificial Intelligence. 2017.
- [27] Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 1243-1252.
- [28] Kalchbrenner N, Espeholt L, Simonyan K, et al. Neural machine translation in linear time[J]. arXiv preprint arXiv:1610.10099, 2016.
- [29] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal processing magazine, 2012, 29(6): 82-97.
- [30] Dahl G E, Yu D, Deng L, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition[J]. IEEE Transactions on audio, speech, and language processing, 2011, 20(1): 30-42.
- [31] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
- [32] Ciregan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3642-3649.
- [33] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [34] Le Q V. Building high-level features using large scale unsupervised learning[C]//2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 8595-8598.
- [35] Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies[J]. 2001.
- [36] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks[C]//International conference on machine learning. 2013: 1310-1318.
- [37] Koutnik J, Greff K, Gomez F, et al. A clockwork rnn[J]. arXiv preprint arXiv:1402.3511, 2014.
- [38] Larochelle H, Hinton G E. Learning to combine foveal glimpses with a third-order Boltzmann machine[C]//Advances in neural information processing systems. 2010: 1243-1251.
- [39] Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization[J]. arXiv preprint arXiv:1509.00685, 2015.
- [40] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
- [41] Wang F, Tax D M J. Survey on the attention based RNN model and its applications in computer vision[J]. arXiv preprint arXiv:1601.06823, 2016.
- [42] Kuchaiev O, Ginsburg B. Factorization tricks for LSTM networks[J]. arXiv preprint arXiv:1703.10722,

- 2017.
- [43] Shazeer N, Mirhoseini A, Maziarz K, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer[J]. arXiv preprint arXiv:1701.06538, 2017.
- [44] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [45] Inan H, Khosravi K, Socher R. Tying word vectors and word classifiers: A loss framework for language modeling[J]. arXiv preprint arXiv:1611.01462, 2016.
- [46] Press O, Wolf L. Using the output embedding to improve language models[J]. arXiv preprint arXiv:1608.05859, 2016.
- [47] Hinton G E. Learning distributed representations of concepts[C]//Proceedings of the eighth annual conference of the cognitive science society. 1986, 1: 12.
- [48] Xu W, Rudnicky A. Can artificial neural networks learn language models?[C]//Sixth international conference on spoken language processing. 2000.
- [49] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.
- [50] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th international conference on Machine learning. 2008: 160-167.
- [51] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.
- [52] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- [53] Jean S, Cho K, Memisevic R, et al. On using very large target vocabulary for neural machine translation[J]. arXiv preprint arXiv:1412.2007, 2014.
- [54] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units[J]. arXiv preprint arXiv:1508.07909, 2015.
- [55] Yan R, Li C T, Hu X, et al. Chinese couplet generation with neural network structures[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016: 2347-2357.
- [56] 于政. 基于深度学习的文本向量化研究与应用[D]. 上海: 华东师范大学, 2016.

附录 1 攻读硕士学位期间撰写的论文

- [1] Jiang Zhang, Yufeng Wang, Zhiyuan Yuan, Qun Jin. Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation[J]. Tsinghua Science and Technology, 2020, 25(02): 180-191. (SCI 检索, WOS:000517497300002)

附录 2 攻读硕士学位期间申请的专利

[1]王玉峰，张江.一种基于深度神经网络的中文语言处理模型及方法，CN201910378653.6，2019.5.7，未授权

附录 3 攻读硕士学位期间参加的科研项目

（1）江苏省高校 自然科学重大研究计划，基于虚拟化的邻近区域移动社交网络体系结构和关键技术（14KJA510004）；

（2）国家重点实验室开放课题，基于社交网络的移动众包系统关键问题研究（SKLNST-2016-2-01）

致谢

时光飞逝，不知不觉间我已经在南京邮电大学度过七年时光，从本科到研究生，从文苑路到三牌楼，在这里学习，在这里生活，南邮对我来说已经不仅仅是学校，这里承载了我太多的回忆，承载了我太多的情感，在这里我要感谢所有帮助过我的老师同学，感谢各位帮助我成长，感谢各位给予我的美好记忆。

回顾三年的研究生生涯，首先要特别感谢我的导师王玉峰教授。无论是学习还是生活，王老师都给了我很多的帮助。王老师知识渊博，涉猎广泛，在给予我充分的自由寻找自己感兴趣的研究方向的同时，王老师还可以给我充分的指导。本文从选题到实验，从下笔到定稿，都离不开王老师的帮助。师者，传道授业解惑也，王老师帮助我克服了众多困难。桃李不言，下自成蹊。在此，我要向王老师表达衷心的感谢，祝王老师在今后的生活工作顺利，万事顺心。

感谢我的父母，在我整个学生生涯中给了我无微不至的关爱，他们是我坚强的后盾。从小学起，我就在寄宿学校读书，每个周末才可以回家，虽然他们没有陪在我的身边，但是却尽他们所能，给我提供了最好的学习环境。中学时，我也是在离家几十公里的地方读书，有时一个月才可以回家一次，但是每个周末，无论是炎炎酷暑，还是凛凛寒冬，我的父母都会来学校看我，让我安心学业。上了大学，他们也是完全尊重我的选择并全力支持，在专业和工作方面给了我最大的自由，感谢他们二十多年来的付出。父母之恩，水不能溺，火不能灭，祝愿他们身体安康，万事顺遂。

感谢我的全体室友和同门，三年的研究生，实验室里孜孜不倦，刻苦钻研的学习环境，给了我认真学术的场所，寝室里欢声笑语让我能够闲暇之余放松自我。你们的好学和上进激励着我不断进步。长风破浪会有时，直挂云帆济沧海。祝愿你们在未来的生活中大展宏图，都拥有一个精彩的人生。

最后，还有感谢坚守在抗击疫情一线的英雄。没有生而英勇，只有选择无畏，在此，向最平凡也最伟大的英雄们致敬。