吉首大学

JISHOU UNIVERSITY

本科生毕业论文

题　　目：	基于深度学习的聊天机器人实现
作　　者：	伍逸凡
学　　号：	20154042039
所属学院：	信息科学与工程学院
专业年级：	计算机科学与技术，2015 级
指导教师：	石俊萍　　**职 称：** 副教授
完成时间：	2019 年 4 月 12 日

吉首大学教务处制

目　　录

基于深度学习的聊天机器人实现

摘　要

生成式对话策略不再依赖于人工编制的匹配规则，转变为设计神经网络模型对语料库中的抽象对话关系进行学习，并通过梯度下降法与误差反向传播调整模型的参数。

通过基于循环神经网络（Recurrent Neural Network，RNN）的序列到序列模型（Sequence to Sequence，Seq2Seq）对语料数据进行建模，实现单轮的对话生成。首先利用词嵌入技术对输入序列进行词向量转换，然后送入编码器中进行特征提取，编码器选用双向的门控循环单元（Gated Recurrent Unit，GRU）。而后解码器通过单向 GRU 对编码器的编码信息进行解码，并加入了注意力机制（Attention）。通过对话数据对不同模型结构和超参数（包括编码器双向 GRU 层数、解码器单向 GRU 层数、注意力计算方式等）的测试，最终确定为 5 层双向 GRU 的编码器、3 层单向 GRU 的解码器以及 Luong Attention（concat）的 Seq2Seq 结构。在 Decoder 生成答案的搜索策略上，分析了贪心策略和集束搜索策略的不足，对集束搜索策略进行了简单改进，实验结果表明，改进后的集束搜索策略不仅丰富了回复的多样性，还保证了回复的质量。

最后，利用青云语料集对模型进行了训练与测试。结果表明，模型的对话生成效果良好，改进的集束搜索算法为模型提供了丰富的优质回答。

关键词：聊天机器人；序列到序列模型；门循环控制单元；注意力机制；集束搜索；

Implementation of Deep Learning Based Chat Robot

Abstract

The generative dialogue strategy no longer relies on artificially compiled matching rules, but transforms into designing a neural network model to learn the abstract dialogue relationships in the corpus, and adjust the parameters of the model through gradient descent and error back propagation.

The corpus data is modeled by the Recurrent Neural Network(RNN) based sequence-to-sequence model(Seq2Seq) to implement the one-round dialogue generation. Firstly, word embedding technology is used to convert the input sequence into word vector sequence, and then sent to the encoder for feature extraction. The bidirectional Gated Recurrent Unit(GRU) is used in encoder. And then the decoder uses GRU decodes the hidden state from the encoder. And also the attention mechanism is added to the model. The dialogue data is used to test different model structure and hyperparameters(including the number of bidirectional GRU layer in encoder, the number of GRU layer in decoder, the computational type of attention, etc), and finally the Seq2Seq structure with 5 layers of bidirectional GRU encoder, 3 layer one-way GRU decoder and Luong Attention(general) is used. In search strategy of generating solution in decoder, the deficiencies of greedy search and beam search are analyzed, and the modified beam search is proposed. The experimental results show that the modified beam search strategy not only enriches the diversity of answers, but also ensures the qualities.

Finally, the model was trained and tested using the Qingyun corpus. The results show that the model's dialogue generation is well, and the modified beam search algorithm provides rich and good answers for the model.

Key words: chatbot; sequence-to-sequence model; gated recurrent unit; attention mechanism; beam search;

第 1 章　绪 论

1.1　课题研究的背景和目的

自 2006 年 Geoff Hinton 等人发表论文《A fast learning algorithm for deep belief nets》和《Reducing the Dimensionality of Data with Neural Networks》后，深度学习的概念首次进入了我们的视野，文中提出的基于深度置信网络的非监督贪心逐层训练算法[1]和多层自动编码器深层结构[2]大大提高了深度网络结构训练的稳定性。此后，网络模型的结构越来越深，效果越来越好。

在以卷积神经网络为主的图象处理方面，2012 年，由 Hinton 和他的学生设计的 Alex Net 发表在论文《ImageNet Classification with Deep Convolutional Neural Networks》上，该网络在同年 ImageNet 竞赛中获得冠军[3]，进一步掀起了学术界对深度学习的研究热潮。VGG、Google Net 等优秀的深度卷积神经网络相继出现，图像处理技术在深度学习的帮助下，得到巨大发展。

在以循环神经网络为主的自然语言处理方面，自 LSTM、GRU 提出后，大大改善了序列间的长期依赖问题。2013 年，Tomas Mikolov 等人在论文《Efficient Estimation of Word Representations in Vector Space》提出 word2vector 算法[4]，加强了单词的表达能力，推动了自然语言技术的发展。

在自然语言方面，代表应用之一便是聊天机器人技术，即通过自然语言技术模拟人类进行对话。它最初源于 Alan Mathison Turing 于 1950 年在《Mind》上发表的文章《Computing Machinery and Intelligence》，文章中提出了机器能否思考的疑问，并创造了著名的“Turing Test”去验证，“Turing Test”也被称为是人工智能的终极目标[5]。

在当今时代，让计算机能够正确地理解并处理人类的自然语言，已成为一个十分具有挑战性和研究意义的课题。智能医疗诊断、智能客服助理等都是聊天机器人在现实中的应用代表。这些研究，将大大促进社会的智能化发展，方便人类的生活。

1.2　国内外的研究现状

最早的聊天机器人系统是 1966 年由麻省理工学院的 Joseph Weizenbaum 开发的聊天机器人 ELIZA，用于在临床治疗中模仿心理医生，该机器人主要由关键词匹配及人工编写的回复规则实现[6]；而后，1988 年加州大学伯克利分校的 Robert Wilensky 等人开发了名为 UNIX Consultant 的聊天机器人系统，旨在指导用户学习如何实用 UNIX 操作系统[7]；1995 年，Richard S. Wallace 受 ELIZA 的启发开发了著名的 ALICE 系统，其基于启发式模板匹配的对话策略，被认为是同类型聊天机

器人中性能最好的系统之一。以上被认为是聊天机器人技术在发展中所经历的三个重要历史时期。如今，聊天机器人随着人工智能技术的兴起又开始迅速发展，诸如身边的微软小冰[8]、阿里小蜜[9]等都是十分优秀的聊天机器人代表。

一般而言，聊天机器人的对话生成策略可以分为以下两种：

1）基于信息检索和模式匹配技术：即通过信息检索技术，结合问题在语料库中进行检索，提取相关特征然后进行模式匹配或关键字匹配等，最后选用相关算法对候选答案进行排序，输出最优答案。

2）基于自然语言理解的生成式对话策略：即通过机器学习算法构建深层语义模型，结合句词分析等统计规律提取特征，让模型从大量的已有对话中学习对话规则，最终利用训练好的模型预测结果。

两种方式前者生成的答案质量较高，但需要大量的人工操作来构建匹配规则，且当遇到语料库中没出现过的问题时无法回答；后者与前者几乎相反，虽然答案的质量普遍没有前者高，但不需要大量的人工操作去构建规则，且生成的答案具有一定随机性，对于语料库中没有的问题能够有一定的联想与分析能力得出不太差的答案。后者较符合人脑的思维方式，是未来聊天机器人技术发展的核心技术，也是本论文研究的重点。以下着重介绍国内外在生成式对话策略中的研究现状。

生成式对话策略，即对对话间的高维抽象关系进行建模，自动生成回复。一般思路为先将句子分解为单词序列，而后对单词序列进行编码，转换为一个序列问题并建模解决。研究者们在早期就提出了单词编码和序列模型来解决这一类问题，近些年又提出了 Attention 机制大大改善了模型效果。

1.2.1 词嵌入模型

最早提出的单词编码方式是词袋模型（Bag of Words，BOW），其通过稀疏的独热编码向量（one-hot）对不同单词进行编码，这种方式简单有效，是早期用得较多的一种方式；1986 年 Hinton 在论文《Learning Distributed Representations of Concepts》中提出了单词的分布式表示，也即现在所说的词向量或词嵌入[10]；2000 年，百度的徐伟提出了用神经网络构建二元语言模型的方法[11]；2001 年，Yoshua Bengio 通过三层神经网络构建的 n-gram 模型对词向量进行训练[12]；2013 年，Google 团队的 Tomas Mikolov 等人基于 CBOW 和 Skip-Gram 算法提出了 word2vec 词向量的训练方法，类似哈夫曼编码，利用 Hierarchical Softmax 对语言模型进行训练[4]；2018 年，AllenNLP 的 ELMo[13]、Google 的 BERT[14]也相继问世，实现了上下文相关的动态词向量建模。

1.2.2 序列模型

基于深度学习技术构建的序列模型即循环神经网络，其最早可追溯到上世纪七八十年代，研究者们为模拟循环反馈系统而建立的各类数学模型为循环神经网络的发展奠定了基础。Michael I. Jordan 和 Jeffrey Elman 在 1986 年和 1990 年提出的

Jordan 网络[15]和 Elman 网络[16]，是最早出现的面向序列的循环神经网络（Recurrent Neural Network，RNN）；1991 年，Sepp Hochreiter 等人发现了循环神经网络在序列的长期依赖上存在问题，即在处理长难序列时会出现梯度消失和梯度爆炸现象，RNN 难以捕获长时间跨度的非线性关系[17]；1997 年，Sepp Hochreiter 在论文《Long Short-Term Memory》中提出了长短记忆循环神经单元（Long Short-Term Memory，LSTM），通过设置多种门单元的形式实现对历史信息的长期记忆，改善了梯度消失和梯度爆炸的问题，却也增加了网络训练的代价[18]；2014 年，Kyunghyun Cho 等人在论文《Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation》中提出了 LSTM 的简化版本门控循环单元（Gated Recurrent Unit，GRU），其结构更简单，对长期依赖问题的解决也较好[19]。LSTM 和 GRU 是目前循环神经网络在序列数据处理上用得最多的两种循环神经单元。此外，Kyunghyun Cho 等人在论文中也提出了一种新的循环神经网络架构——Seq2Seq（Sequence to Sequence），Ilya Sutskever 等人也几乎在同一时间在论文《Sequence to Sequence Learning with Neural Networks》[20]中提出了相似思路。自此，Seq2Seq 诞生，它主要被广泛运用于机器翻译系统中，也是聊天机器人系统的生成式对话策略的主要思路之一。

1.2.3 注意力机制

注意力机制即 Attention，其最早出现于图像领域，是一种模拟人脑对图像感兴趣部分的局部聚焦机制，提出于上世纪末。Attention 通过一个简单的浅层神经网络，对隐层状态或相关特征进行非线性变换，得到关注度权重，实现特征的局部重视。2014 年，Google Mind 团队的论文《Recurrent Models of Visual Attention》[21]使得 Attention 重新得到了研究者们的重视，文中使用 RNN+Attention 的模型对图像进行分类，取得了较好的成果。

Attention 分为 Soft Attention 和 Hard Attention，前者只关注局部，后者是为每一个部分设置一个权重，以权重决定关注度[22]。2014 年，Dzmitry Bahdanau 等人首次将 Attention 应用到了自然语言处理领域的神经网络机器翻译上[23]；其后的 2015 年，Minh-Thang Luong 又在 Bahdanau 的基础上，对 RNN 中的 Attention 进行了扩展，提出了全局注意力机制（Global Attention）和局部注意力机制（Local Attention），还提出了三种计算注意力权重的方法：dot、cat、general，且论文中的实验表明，dot 作为最简单的 Attention 计算方式，取得了最好的成绩，研究结果发表在论文《Effective Approaches to Attention-based Neural Machine Translation》[24]上。

Attention 的加入让许多模型提高了几个百分点，但也带来了更多的运算成本，为了解决 Attention 的并行训练问题，2017 年，Facebook AI Lab 提出用 CNN（卷积神经网络）代替 RNN[25]；同年，Google 的团队提出 Self Attention，完全摒弃 RNN 单元，以做到并行训练[26]。

1.3 本论文的研究内容和设计结构

本论文的研究重点是如何基于 Seq2Seq 模型加上注意力机制设计一个优秀的聊天机器人对话生成策略。首先介绍了课题的研究背景和目的，并对聊天机器人的发展概述与生成式对话策略的研究现状作了详细说明。之后，首先将对用到的模型或算法技术作一个详细介绍与梳理，然后将对本论文所使用的聊天机器人设计思路做详细的分析与介绍，接着通过语料数据集对模型进行训练与测试，根据结果分析不足之处，并提出相应的改进思路。

本论文共分为六章，各章节的内容编排如下：

1）第 1 章介绍了深度学习和聊天机器人系统的研究背景以及意义，介绍了相关算法模型在国内外的研究历程和现状。最后简述了本论文的研究内容与思路，并给出了本论文的层次结构。

2）第 2 章对于需要用到的基础理论与算法模型做了一个详细的阐述与介绍，包括 Embedding、GRU、Seq2Seq、Attention、BLEU、Beam Search 等。

3）第 3 章对于聊天机器人的对话生成策略做了一个详细的设计。

4）第 4 章使用青云对话语料对候选模型分别进行了训练与测试，记录了实验过程，并比较了不同模型的效果，选取最好的候选模型作为最终的模型。

5）第 5 章根据最终模型的测试结果分析存在的不足之处，并讨论了相应的改进思路。

6）第 6 章为总结与展望，主要对本论文工作进行了总结，并对未来聊天机器人的技术方向发展进行了展望。

第 2 章　相关技术与理论基础

2.1　词嵌入相关

词嵌入（Word Embedding），即通过某种方式，将不同的单词嵌入到一个低维稠密的向量空间中，也被称为词向量。低维和稠密，是相对于传统词袋模型（BOW）的独热编码（One-hot）而言的。

2.1.1　独热编码

在 One-hot 编码中，每一个词都被唯一编码为了一个 01 向量，该向量仅有一个维度是 1，其余维度都是 0，向量长度为不同单词的总数量，如图 1 所示。

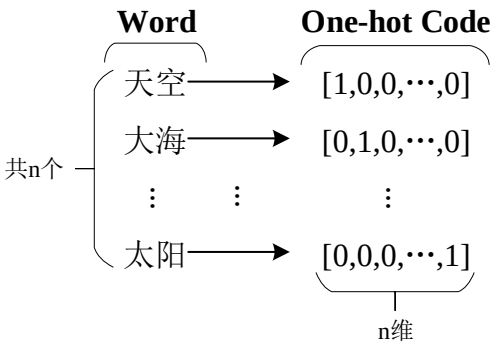

图 1　one-hot 编码

以n表示$Word$表的大小；x_i表示第i个单词的编码；oh表示记录单词 One-hot 编码的$n \times n$二维矩阵。编码公式可表示为：

$$x_i = oh_i \tag{2.1}$$

其中，oh_i表示矩阵oh的第i行。

这种编码方式优点是十分简单，且能与 Softmax 函数直接对应。缺点是词与词之间是相互独立的，没有任何联系，且需要花费巨大的空间代价存储高维稀疏矩阵oh，空间代价为$O(n^2)$。词嵌入很好地解决了这些问题。

2.1.2　词嵌入模型

Word Embedding 将单词嵌入到低维稠密的向量空间，通过语料数据的学习，调整单词向量的各特征权重，以实现单词的向量表示。词向量使得单词间不再相互独立，词向量间的余弦相似度可被用来估计单词间的语义相关度，或是根据词向量对之差判断单词间的相互对应关系，如图 2 所示。

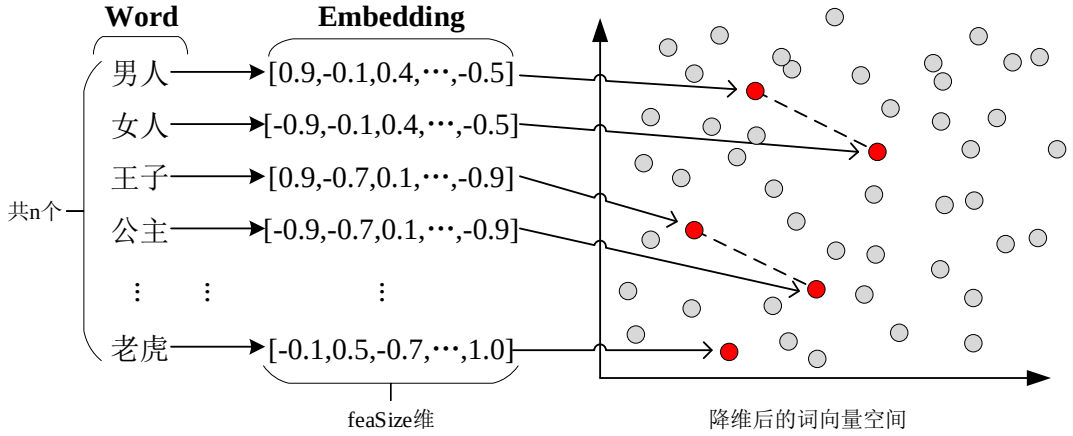

图 2　词向量编码

多数情况下，词向量是模型训练的附加产物，通过梯度下降和目标函数的误差反向传播，词向量将被学习和调整。

以 CBOW 算法为例，它通过以上下文预测中心词为目标，训练模型。设Emb表示$n \times feaSize$的词向量矩阵；x_i表示第i个词的 One-hot 编码，是一个$1 \times n$的 01 矩阵；W_{out}表示一个$feaSize \times n$的输出层全连接矩阵；b_{out}为输出层的偏置项；c表示上下文窗口的大小。算法步骤如下：

1）随机初始化Emb和W_{out}。

2）计算中心词x_i窗口内上下文词向量的平均词向量v_{ic}：

$$v_{ic} = \frac{1}{c}\left(\sum_{j=i-\frac{c}{2},j\neq i}^{i+\frac{c}{2}} x_i\right) \cdot Emb \tag{2.2}$$

3）利用W_{out}全连接矩阵进行线性映射，得到在中心词x_i处每个单词的得分序列s_i：

$$s_i = v_{ic} \cdot W_{out} + b_{out} \tag{2.3}$$

4）通过一个 Softmax 函数计算中心词的概率分布p。$s_i^{(j)}$表示中心词x_i处第j个单词的得分：

$$p\left(x_i \middle| x_{i-\frac{c}{2}}, x_{i-\frac{c}{2}+1}, \dots, x_i, \dots, x_{i+\frac{c}{2}}\right) = softmax(s_i) = \frac{exp(s_i^{(j)})}{\sum_{k=0}^{n} exp(s_i^{(k)})} \tag{2.4}$$

5）以交叉熵损失函数L作为目标函数，使用梯度下降法与误差反向传播调整参数Emb、W_{out}。p_j表示中心词为x_j的概率；y_j表示中心词是否是x_j，$y_j = \begin{cases} 1, if\ x_j = x_i \\ 0, if\ x_j \neq x_i \end{cases}$。

$$L = -\sum_{j=1}^{n} y_j \log(p_j) \tag{2.5}$$

最终训练得出的Emb矩阵便是词向量矩阵。

2.2　人工神经网络相关

人工神经网络（ANN），如图 3，是一种从信息处理的角度对人脑神经元进行抽象的网络模型。一般通过对输入数据进行多次非线性变换，并根据最终得到的预测输出与真实值建立目标函数，通过梯度下降法与误差反向传播调整各层网络参数，以实现模型的学习。

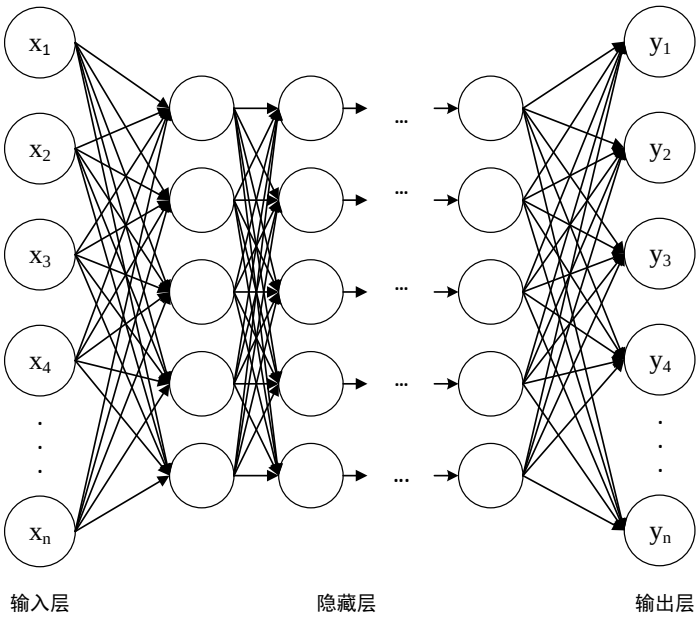

图 3　ANN 基本结构

2.2.1　BP 神经网络模型

以 BP 神经网络[27]为例。令x_i表示第i条样本；每条样本包含m个特征；W_{in}和b_{in}表示输入层的全连接矩阵和偏置项。则输入层的运算可表示为：

$$a_1 = \delta(x_i \cdot W_{in} + b_{in}) \tag{2.6}$$

其中a_1表示输入层的输出，也即隐藏层的输入；δ表示非线性激活函数，一般可选为$sigmoid(x) = \frac{1}{1+e^{-x}}$、$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$、$ReLU(x) = \begin{cases} x, if\ x > 0 \\ 0, if\ x \leq 0 \end{cases}$等。

隐藏层可包含多层运算，是神经网络的潜力所在。设隐藏层第j层的输入为a_j；W_j和b_j表示隐藏层第j层的全连接矩阵和偏置项。则该层的运算可表示为：

$$a_{j+1} = \delta(x_j \cdot W_j + b_j) \tag{2.7}$$

输出层和输入层类似，只包含单层。令a_k表示隐藏层末层的输出；W_{out}和b_{out}表示输出层的全连接矩阵和偏置项。则有：

$$\hat{y}_i = f(a_k \cdot W_{out} + b_{out}) \tag{2.8}$$

其中$\hat{y}_i$表示最终的预测输出；f可为线性函数$f(x) = x$，或者 Sigmoid、Softmax 函数。前者对应回归问题，后者对应分类问题。

目标函数一般选为均方误差函数（Mean Square Error，MSE）或交叉熵损失函数（Cross Entropy Loss）。前者对应回归问题，后者对应分类问题。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (2.9)$$

$$CrossEntropyLoss = \sum_{i=1}^{n}\sum_{j=1}^{c} y_i^{(j)} log(\hat{y}_i^{(j)}) \quad (2.10)$$

最后利用训练数据对神经网络进行训练，调整各层参数。BP 神经网络也是神经网络的最基本形式。

2.2.2 循环神经网络模型

RNN 是通过将神经元节点以链式连接的方式构成的一种能够处理序列数据的神经网络模型。通过将神经运算单元在时序上展开，循环地接受序列的输入，并用一个隐层状态变量记录历史信息，以学习序列数据在时序上的前后逻辑关系，如图 4。

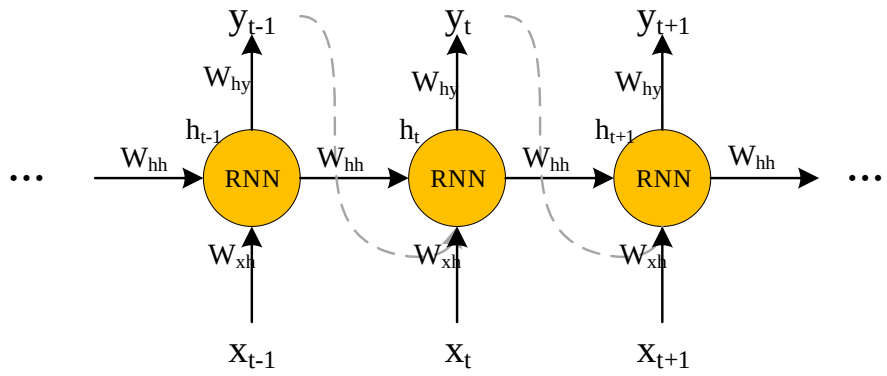

图 4　RNN 结构图

设t时刻的输入为x_t；输入层的全连接矩阵为W_{xh}；隐藏层的全连接矩阵为W_{hh}；偏置项为b_h；$t-1$时刻的隐层状态为h_{t-1}。那么t时刻的隐层状态h_t可由如下公式给出：

$$h_t = \delta(h_{t-1} \cdot W_{hh} + x_t \cdot W_{xh} + b_h) \quad (2.11)$$

t时刻的预测输出$\hat{y}_t$由输出层的全连接矩阵W_{hy}对隐层状态h_t线性变换得到，b_y为偏置项：

$$\hat{y}_t = f(h_t \cdot W_{hy} + b_y) \quad (2.12)$$

每一个时刻t的W_{hh}, W_{xh}, W_{hy}都是权重共享的；δ、f与前文 BP 神经网络类似，标函数与训练方式也同理。

为了解决长期依赖问题，又出现了以 LSTM 和 GRU 为运算单元的神经网络，以 GRU 为例。

GRU 通过引入更新门单元和重置门单元来控制输入值、记忆值和输出值，如图 5 所示。

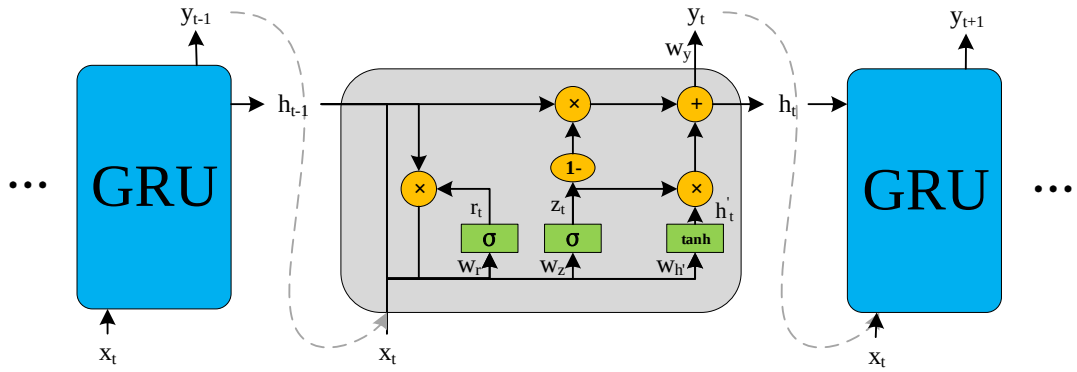

图 5　GRU 结构图

首先计算t时刻的重置门变量r_t，其决定了前一时刻的隐层状态h_{t-1}的重置程度。W_r和b_r表示重置门的全连接矩阵和偏置项，则有：

$$r_t = \sigma([h_{t-1}, x_t] \cdot W_r + b_r) \tag{2.13}$$

$W_{h'}$和$b_{h'}$表示对计算隐层状态h'_t的全连接矩阵和偏置项：

$$h'_t = tanh([r_t * h_{t-1}, x_t] \cdot W_{h'} + b_{h'}) \tag{2.14}$$

接着计算更新门变量z_t，其决定了t时刻隐层状态h_t对h'_t和h_{t-1}的取舍程度。W_z和b_z表示更新门的全连接矩阵和偏置项，则有：

$$z_t = \sigma([h_{t-1}, x_t] \cdot W_z + b_z) \tag{2.15}$$

然后即可计算出t时刻的隐层状态h_t：

$$h_t = (1 - z_t) * h_{t-1} + z_t * h'_t \tag{2.16}$$

最后计算t时刻的预测输出y_t。w_y和b_y分别表示输出层的全连接矩阵和偏置项：

$$\hat{y}_t = f(h_t \cdot W_y + b_y) \tag{2.17}$$

σ表示 Sigmoid 函数，f和之前一样。目标函数与训练方式也同理。

2.2.3　序列到序列模型与注意力机制

Seq2Seq 模型通过训练两个 RNN，一个 RNN 作为编码器（Encoder）对输入序列进行编码，另一个 RNN 作为解码器（Decoder）将编码器末时刻的隐层状态解码，旨在解码出对应的输出序列，如图 6 所示。

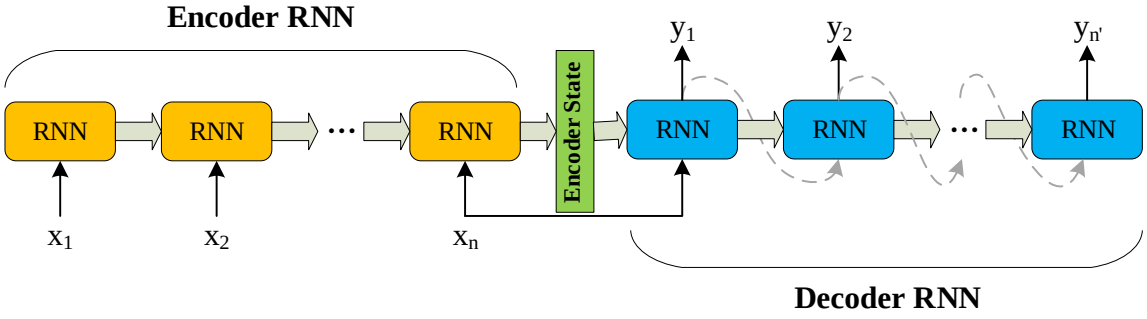

图 6　Seq2Seq 模型

设输入序列$X=[x_1,x_2,\ldots,x_n]$；h_0表示 Encoder 初始时刻的隐层状态输入，一般初始化为0向量。则末时刻隐层编码输出即 Decoder 的初始时刻的隐层状态输入h'_0可表示为：

$$h'_0 = RNN_{encoder}(X, h_0) \tag{2.18}$$

而后t时刻解码器的隐层状态h'_t可表示为：

$$h'_t = RNN_{decoder}(\hat{y}_{t-1}, h'_{t-1}) \tag{2.19}$$

$\hat{y}_t$表示t时刻的预测输出。计算方式和 RNN 一样，目标函数与训练方式也同理。

基本的 Seq2Seq 函数在 Encoder 部分仅仅利用了最后的隐层状态，虽然最终输出的隐层状态h'_0包含了所有隐藏层的状态信息，但由公式（2.11）、（2.16）可知，越靠近最后时刻，将有越大的概率被保留。而靠近初始时刻的隐层状态信息，将被较少地保留在h'_0中。因此，Encoder 中引入了 Attention 机制，单向 RNN 也被替换成了双向 RNN（Bi-RNN）。

Attention 通过一个浅层神经网络，学习对 Encoder 所有隐层状态的关注度权重，求出每一时刻 Decoder 加入了关注变量之后的隐层状态，以下以 Bahdanau 提出的 Attention 计算方式为例，结构如图 7。

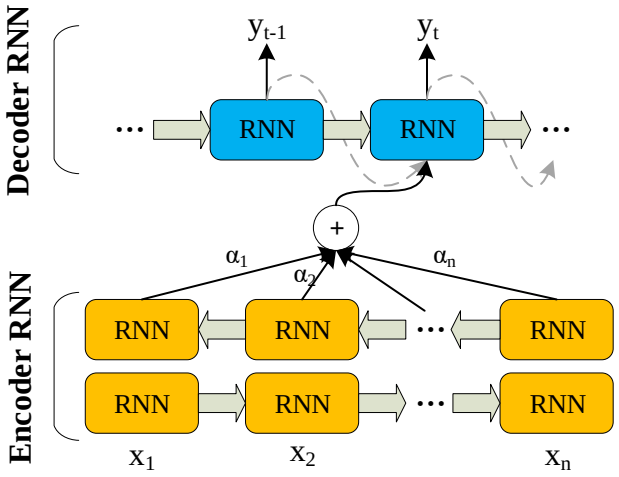

图 7　Bahdanau Attention 结构

对于 Bi-RNN 的 Encoder 而言，总共存在$2n$个隐层输出，包括n个正向的隐层状态和n个反向的隐层状态，分别设为$\vec{h}$和$\overleftarrow{h}$。首先求得 Encoder 最终输出的隐层状态h：

$$h = \vec{h} + \overleftarrow{h} \tag{2.20}$$

在t时刻，设h'_{t-1}表示前一时刻 Decoder 的隐层状态，则该时刻对于 Encoder 第j个隐层状态h_j的关注度$\alpha_t^{(j)}$可表示为：

$$\alpha_t^{(j)} = \frac{exp(e_t^{(j)})}{\sum_{k=1}^{n} exp(e_t^{(k)})} \tag{2.21}$$

其中，$e_t^{(j)}$表示t时刻对h_j关注度得分，由如下公式给出：

$$e_t^{(j)} = a(h'_{t-1}, h_j) \tag{2.22}$$

a表示线性变换。t时刻的 Attention 语义变量c_t便可表示为关注度权重α_t与 Encoder 隐层状态h的加权求和：

$$c_t = \sum_{j=1}^{n} \alpha_t^{(j)} * h_j \tag{2.23}$$

最后，便可得出t时刻输入 Decoder 的最终输入信息$attnInput_t$：

$$attnInput_t = \delta(a(\hat{y}_{t-1}, c_t)) \tag{2.24}$$

其中，δ与a的含义与前文相同。

2.3 其它自然语言技术相关

2.3.1 词频-逆文档频率

词频-逆文档频率（Term Frequency-Inverse Document Frequency，TF-IDF），是一个在文本挖掘领域常用的统计指标，可以反应一个词在文中的重要程度。它分为两部分组成：词频（Term Frequency，TF）和逆文档频率（Inverse Document Frequency，IDF）。

设文档i中单词w_j的出现次数记为$c_{w_j}^{(i)}$，则w_j在该文档中的词频$TF_{w_j}^{(i)}$为：

$$TF_{w_j}^{(i)} = \frac{c_{w_j}^{(i)}}{\sum_k c_{w_k}^{(i)}} \tag{2.25}$$

设语料库的文档总数记为n，包含w_j的文档数记为nc_{w_j}，则w_j的逆文档频率IDF_{w_j}可表示为：

$$IDF_{w_j} = log(\frac{n}{nc_{w_j}+1}) \tag{2.26}$$

最终得出w_j在文档i中的词频-逆文档频率$TF\text{-}IDF_{w_j}^{i}$：

$$TF\text{-}IDF_{w_j}^{i} = TF_{w_j}^{(i)} \times IDF_{w_j} \tag{2.27}$$

2.3.2 BLEU 得分

BLEU（Bilingual Evaluation Understudy）是来源于机器翻译的一种文本评估算法，可反应输出序列与参考序列的匹配程度，一般被用来衡量 Seq2Seq 模型的效果。

设真实输出序列为Y，模型的预测输出序列为$\hat{Y}$；对于在$\hat{Y}$中出现的每个单词w，我们分别统计它在$\hat{Y}$和Y中的出现次数，记为$\hat{c}_w$和c_w；$n_{\hat{Y}}$表示$\hat{Y}$中的单词数。则预测输出的精度p可表示为：

$$p = \frac{\sum_{w \in \{\hat{Y}\}} min(\hat{c}_w, c_w)}{n_{\hat{Y}}} \tag{2.28}$$

考虑到存在预测序列$\hat{Y}$比真实序列Y短而造成精度p较高的情况，引入惩罚因子

BP，设n_Y表示Y中的单词数：

$$BP=\begin{cases}1, if\ n_{\hat{Y}}>n_Y\\ exp(1-\frac{n_Y}{n_{\hat{Y}}}), if\ n_{\hat{Y}}\le n_Y\end{cases} \tag{2.29}$$

最终的 BLEU 得分计算如下：

$$BLEU=BP\times exp(log(p)) \tag{2.30}$$

以上是 1-gram 的形式，还可推广至 n-gram 的形式，各部分加权求和即可。随着 n-gram 中 n 的增大，总体的得分会呈指数下降，一般 n 最多取到 4。

2.3.3 句向量相似度

句向量相似度也是衡量文本相似度的方式之一，它通过对词向量进行加权平均计算句向量，根据句向量间的余弦相似度进行评价。

设语句s_i由长度为n的单词序列组成$[w_1,w_2,\dots,w_n]$，其对应的词向量序列为$[v_{w_1},v_{w_2},\dots,v_{w_n}]$，那么其句向量$v_{s_i}$可以表示为：

$$v_{s_i}=\frac{1}{n}\sum_{j=1}^{n}v_{w_j} \tag{2.31}$$

语句s_i和语句s_j间的相似度表示为：

$$\text{Similarity}(s_i,s_j)=\frac{v_{s_i}\cdot v_{s_j}}{\|v_{s_i}\|\|v_{s_j}\|} \tag{2.32}$$

其中$\|v\|$表示向量v的欧式距离。

2.3.4 集束搜索算法

集束搜索（Beam Search）是一种启发式图搜索算法。不同于深度优先搜索和广度优先搜索，Beam Search 每次仅将一定数量的局部最优选择作为候选路径，大大节省了搜索成本。其本质仍是贪心搜索，扩大了解的搜索范围，提高了找到全局最优解的概率。在 Seq2Seq 模型中，Decoder 输出部分其实就是一种图搜索，传统方式为每一个时间步t找最优的一个$\hat{y}_t$。而对于 Beam Search 而言，定义搜索束宽度为$beamWidth$，设上一个时间步$t-1$的$beamWidth$个最优解序列为$\hat{s}_{t-1}^{(1)},\hat{s}_{t-1}^{(2)},\dots,\hat{s}_{t-1}^{(beamWidth)}$，$\hat{y}'^{(i)}_j$表示$\hat{s}_{t-1}^{(i)}$中的第$j$个单词输出。

$$\hat{s}_{t-1}^{(i)}=\left[\hat{y}'^{(i)}_1,\hat{y}'^{(i)}_2,\dots,\hat{y}'^{(i)}_{t-1}\right] \tag{2.33}$$

其对应的似然概率为$p_{t-1}^{(1)},p_{t-1}^{(2)},\dots,p_{t-1}^{(beamWidth)}$：

$$p_{t-1}^{(i)}=P\left(\hat{y}'^{(i)}_{t-1},\hat{y}'^{(i)}_{t-2},\dots,\hat{y}'^{(i)}_1\middle|x\right)=\prod_{j=1}^{t-1}P(\hat{y}'^{(i)}_j|x,\hat{y}'^{(i)}_1,\hat{y}'^{(i)}_2,\dots\hat{y}'^{(i)}_{j-1}) \tag{2.34}$$

其中x表示 Decoder 的输入；$P\left(\hat{y}'^{(i)}_{t-1},\hat{y}'^{(i)}_{t-2},\dots,\hat{y}'^{(i)}_1\middle|x\right)$表示给定$x$的情况下，出现$\hat{y}'^{(i)}_{t-1},\hat{y}'^{(i)}_{t-2},\dots,\hat{y}'^{(i)}_1$的条件概率；$P(\hat{y}'^{(i)}_j|x,\hat{y}'^{(i)}_1,\hat{y}'^{(i)}_2,\dots\hat{y}'^{(i)}_{j-1})$同理。

为了防止下溢等情况的发生，一般对似然概率取对数得到$log(p_{t-1}^{(i)})$。

对于每一个$\hat{s}_{t-1}^{(i)}$,计算所有可能的的$\hat{s'}_t^{(i)}$。然后对于所有$\hat{s'}_t^{(i)}$，根据其对应的$log(p_t^{(i)})$，保留最优的$beamWidth$个$\hat{s'}_t^{(i)}$，作为t时刻的$beamWidth$个最优解序列$\hat{s}_t^{(i)}$。

最终，对于末时刻$t_{n'}$，对于所有保留的最优解序列，按照其对数似然概率值与序列长度的比值作为评价指标：

$$score\left(\hat{s}_{t_{n'}}^{(i)}\right) = \frac{1}{{t_{n'}}^{\alpha}} log(p_{t_{n'}}^{(i)}) \tag{2.35}$$

其中α表示柔化系数，用来对结果在完全按长度归一化与完全不按长度归一化之间作一个权衡。取该指标最大的序列作为最终的答案。

2.4 本章小结

本章主要介绍了聊天机器人关于生成式对话策略的相关技术与算法。第一部分主要介绍了词嵌入技术，one-hot 编码实现简单但无词表征能力且浪费资源，word embedding 实现了词编码的降维并大大提高了词表征能力，最后给出了 CBOW 算法的过程；第二部分主要介绍了人工神经网络的相关技术，BP 神经网络给出了对于非线性关系的学习过程，RNN 解决了序列数据上的处理，GRU 解决了 RNN 存在的长期依赖问题，Seq2Seq 模型为机器翻译、聊天问答等问题提供了较好的解决方案，Attention 的引入优化了 Seq2Seq 的模型效果；第三部分介绍了本论文用到的自然语言的相关技术，TF-IDF 反应了词语在文中的重要程度，BLEU 和句向量相似度序列间的匹配度提供了一个参考，Beam Search 扩大了 Decoder 在解空间中的搜索范围。

第 3 章　聊天机器人生成式对话策略的设计

聊天机器人的对话生成策略，即模型能针对问题给出相应的答复。问题和答复都可看作是一个单词序列，因此可以使用 Seq2Seq 模型进行建模，并利用训练数据对模型进行训练，以学习问句特征序列到答句特征序列之间的非线性关系。本论文的总体模型结构如图 8：

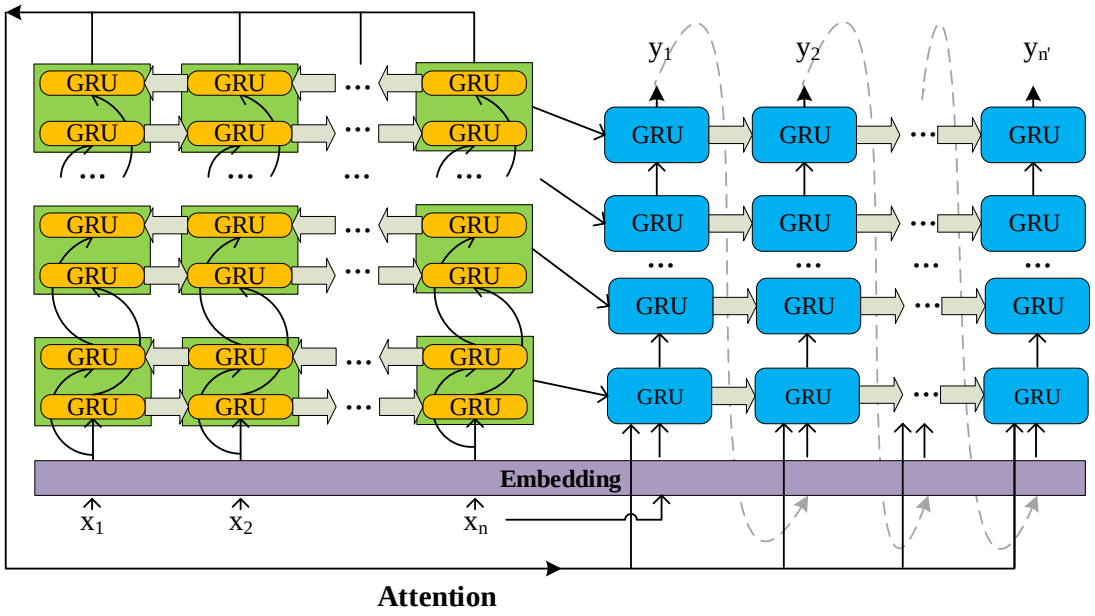

图 8　Seq2Seq+Attention 整体结构

3.1　数据预处理

第一步是对语料库进行清洗，由于语料中存在很多中英文符号，而它们的意义相同，需要做统一化处理。例如将全角句号'。'替换为半角句号'.'等。

然后将语料库中的所有句子转换为单词序列，英文语料可以直接按空格字符分割，而中文语料需要通过算法进行分词。即对于语句s，需被分割成以单词w_i组成的序列，并在末尾添加结束标记< EOS >。$wordSplit$表示分词算法：

$$wordSplit(s) = [w_1, w_2, \dots, w_n, \text{<EOS>}] \quad (3.1)$$

最后考虑到计算性能问题，有些句子的词序列过长，将导致序列对齐时造成巨大的空间浪费以及训练时的时间浪费。故设置一个序列的最大长度，若问句或答句超过该长度，将过滤掉该样本。

3.2　编码器解码器神经网络

该层主要通过一个 Encoder-Decoder 神经网络作深度学习的模型搭建。

3.2.1　词嵌入层

Encoder 和 Decoder 首先经过的都是 Embedding 层，该层的参数权重共享。该层通过一个浅层的神经网络对单词进行向量嵌入，并通过误差反向传播在训练过程中调整词嵌入矩阵Emb。设单词w_i的词向量为v_i；oh表示求单词的独热编码：

$$v_i = oh(w_i) \cdot Emb \tag{3.2}$$

3.2.2 编码器层

该层通过一个 RNN 对输入序列进行编码，以学习输入序列间各单词的前后逻辑关系，计算所有隐层状态。考虑效率问题，本论文选用 LSTM 的简化版本 GRU 作为循环神经网络的计算单元；设置为双向的 GRU，以对问句作前后向分别编码，按照公式（20）计算最终隐层状态。

$$[h_1, h_2, \dots, h_n], h'_0 = Bi\text{-}GRU_{encoder}(X, h_0) \tag{3.3}$$

3.2.3 注意力层

该层通过 Attention 为 Decoder 的输入或隐层状态加入注意力信息，以实现对$[h_1, h_2, \dots, h_n]$的合理关注。

选用 Bahdanau 的注意力模型（见公式（2.21）-（2.24））和 Luong 论文中的 Global Attention 的 dot、general、concat 注意力模型作备选。Luong Attention 的三种 Attention 权重计算如下：

$$\text{score}\left(h_{[:]}, h'_t\right) = \begin{cases} h_{[:]} * h'_t, & dot \\ h_{[:]} * W_a(h'_t), & general \\ v_a * \tanh\left(W_a\left([h_{[:]}, h'_t]\right)\right), & concat \end{cases} \tag{3.4}$$

$h_{[:]}$表示 Encoder 的所有隐层输出，即$h_{[:]} = [h_1, h_2, \dots h_n]$；其中$W_a$表示单层神经网络；$v$表示一个可学习参数。

3.2.4 解码器层

该层通过一个单向 RNN 的隐层信息进行解码，同样选取 GRU 作为循环神经网络网络的计算单元，类似公式（2.19）。

然后将输出的h'_t输入 Softmax 作分类，以得到当前时刻t的单词预测输出。

$$\hat{y}_t = softmax(h'_t) \tag{3.5}$$

同时本论文使用 Teacher Forcing 机制[28]，即设置一个概率，使得$t+1$时刻解码器的输入有一定概率使用真实输出y_t，以加速 RNN 的收敛。

3.3 目标函数与优化器

目标函数选用交叉熵损失函数，公式见（2.10）。通过梯度下降法以及误差反向传播对各参数进行调整与优化，优化器选用较快的自适应矩估计（Adaptive Moment Estimation，Adam）[29]。Adam 是 Momentum、AdaGrad[30]、RMSProp 的集大成者，公式如下：

$$g_t = \Delta_\theta J(\theta_{t-1}) \tag{3.6}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (3.7)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (3.8)$$

$$\hat{m}_t = m_t / (1-\beta_1^t) \quad (3.9)$$

$$\hat{v}_t = v_t / (1-\beta_2^t) \quad (3.10)$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon) \quad (3.11)$$

θ_t表示 t时间步的各层参数；g_t表示 t时间步的梯度；m_t和v_t表示 t时间步的一阶矩估计和二阶矩估计；β_1和β_2为一二阶矩估计的指数衰减率；$\hat{m}_t$和$\hat{v}_t$表示对一二阶矩估计作偏差修正后的结果；α表示默认学习率；ε防止除数为 0。

3.4 评价指标

选用 BLEU 得分和句向量相似度作为评价指标，见公式（2.28）-（2.30）和公式（2.31）-（2.32）。

BLEU 得分一般被用来评价机器翻译模型，此处使用 BLEU 作为评价指标之一，其可以在一定程度上反应回答结果与真实答案的匹配度。但生活中的对话其实是十分随意且灵活的，一个语句的正确回复可以有很多很多种，因此加上句向量相似度作为另一评价指标，其能一定程度上反应的答案多样性。

3.5 改进的解空间搜索策略

在 Decoder 生成答案时，若使用传统的贪心搜索策略会导致生产答案唯一，且极其容易陷入局部最优解。对于聊天机器人，若相同问题每次都产生相同答案，将大大降低用户体验。而集束搜索（见公式（2.33-2.35））虽然提高了找到全局最优解的概率，同时末时刻会有$beamWidth$个解输出，通过概率抽样即可实现答案的多样化。但事实上，在对话生成的 Seq2Seq 模型中，其实现的答案多样性是十分有限的。

集束搜索算法有效的重要原因之一便是解空间的客观性和准确性。对于普通的图论问题，图结构是固定的，即解空间是客观且准确的；而对于 Seq2Seq 中 Decoder 而言，其解空间是由模型根据训练集训练得到的，即模型的泛化性能直接影响解空间的准确性，进而影响集束搜索算法搜索的结果。对于机器翻译问题来说，其源语言与目标语言间的单词对应关系是比较固定的，因而过拟合问题不明显；而对于对话生成而言，其问句与答句间的对应关系是十分模棱两可的，这也直接导致了过拟合问题在基于 Seq2Seq 的对话生成上十分明显，从而导致其解空间不够准确，不够准确的解空间将导致不准确的序列似然概率估计（公式（2.34））, 最终集束搜索得到的解也将不够好。且序列越长，其似然概率估计的误差越大。

基于以上观点，对集束搜索算法做出改进。将公式（2.34）改为如下：

$$p_{t-1}^{(i)} = P\left(\hat{y}'^{(i)}_{t-1} \middle| x\right) \quad (3.12)$$

即仅考虑序列$\hat{s}_{t-1}^{(i)}$最末时刻的$\hat{y}'^{(i)}_{t-1}$在给定输入x下的条件概率，并以其对数作为下一时刻t最优解序列的择优标准。

3.6 本章小结

本章主要介绍了本论文对生成式对话策略的总体设计框架。第一部分，介绍了数据预处理的方式；第二部分介绍了总体算法模型架构，由词嵌入层、编码器层、注意力层和解码器层组成；第三部分介绍了训练过程中对参数的调整思路，使用Adam优化器；第四部分介绍了对模型效果所使用的评价指标为BLEU得分和句向量相似度；第五部分介绍了Decoder解空间的搜索策略，设计了一种改进的集束搜索算法。

第 4 章　聊天机器人生成式对话策略的实现

以下所有实验均在 GPU：GTX1080 Ti；CPU：i9-9900k es 8 核；RAM：32G 的 Linux 系统下进行。程序语言为 Python，环境为 pytorch 1.0.1+cuda9+cudnn7。

4.1　数据集选取

经过对多份中文对话语料库的简单筛选，兼顾对话质量与训练成本，最终选取青云语料作为本论文的训练数据。

青云语料是来自一个聊天机器人交流群的对话语料库，共包含 105914 条对话，部分数据如下表：

表 1　青云语料部分数据

Question	***Answer***
推荐个好电影给我看看	看什么看，您就别凑热闹了
考研成绩什么时候出来	我哪知道什么时候
鸟语花香	等等哈，正吃着切糕呢
敢不敢说句话	为什么不敢，胆小鬼
笨葡萄的紫葡萄啊	啊嫩啊嫩绿得刚发芽
…	…

如表 1 可知该语料偏生活化，噪音较少。

4.2　模型调优

4.2.1　确定最大序列长度

对数据集中的所有数据分词之后，对所有所得序列的长度做统计，绘制出小提琴图如图 9：

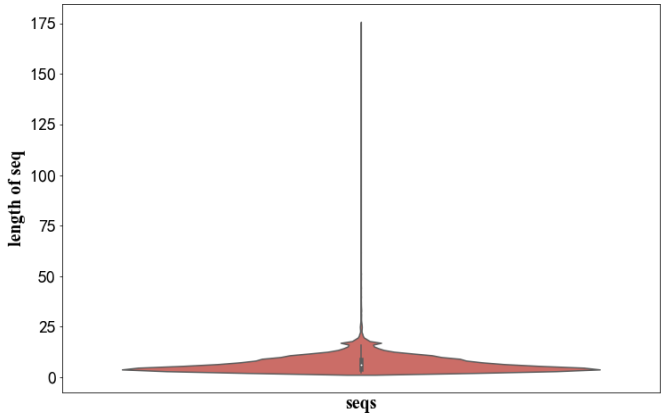

图 9　序列长度统计图

可知大部分序列的单词个数都在 25 以内，故设置序列最大长度为 25，问句或答句长度超过 25 将过滤掉该数据。过滤后，剩余对话 105148 条。

4.2.2 确定 RNN 结构

测试集占比设为 0.1；词嵌入大小设为 300；GRU 隐藏层大小设为 256；多层 GRU 间的 dropout 比例设为 0.1；Teacher Forcing 的概率设为 0.5；Adam 优化器选用默认参数：$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.99$。

接着使用网格搜索法确定 Encoder RNN 和 Decoder RNN 的层数。

设 Encoder 和 Decoder 的 RNN 层数分别为n_E、n_D；候选值为 1、2、3、4、5。且保证$n_E > n_D$，共 15 种参数组合。考虑时间成本问题，选用一个小的对话数据进行测试，该数据集仅包含 560 条对话，使用 Bahdanau 的 Attention 结构测试，每次训练 1500 轮。实验结果如表 2，其中 Sent Sim 表示句向量相似度：

表 2　不同 RNN 层数实验结果

n_E	n_D	*Train*		*Test*	
		BLEU	***Sent Sim***	***BLEU***	***Sent Sim***
1	***1***	0.766	0.728	0.201	0.256
2	***1***	0.752	0.725	0.206	0.266
2	***2***	0.759	0.726	0.200	0.259
3	***1***	0.776	0.732	0.199	0.275
3	***2***	0.741	0.724	0.220	0.296
3	***3***	0.763	0.718	0.160	0.237
4	***1***	0.756	0.725	0.198	0.267
4	***2***	0.743	0.719	0.179	0.243
4	***3***	0.751	0.736	0.175	0.248
4	***4***	0.74	0.711	0.178	0.237
5	***1***	0.743	0.734	0.181	0.259
5	***2***	0.755	0.717	0.199	0.236
5	***3***	0.754	0.730	0.210	0.267
5	***4***	0.761	0.726	0.157	0.219
5	***5***	0.743	0.714	0.163	0.201

由该结果可以看出，测试集得分远低于训练集，这其实是意料之中的。因为对话数据不像其它数据，其随机性很大，且答案多样化，BLEU 和 Sent Sim 仅仅只能在一定程度上作为参考标准，在测试集上具有高 BLEU 和高 Sent Sim 是一个好对话生成模型的充分不必要条件。

根据实验结果，最终选取$n_E = 5$、$n_D = 3$。

4.2.3 确定 Attention 结构

分别按照 Bahdanau 的 Attention 与 Luong 的 dot、general、concat 共计四种 Attention 结构在青云数据集上进行测试。每次训练 350 轮；批量数据训练，批大小*batchSize*设置为 1024。结果如表 3。

表 3 不同 Attention 实验结果

Attention	*Train*		*Test*	
Method	*BLEU*	*Sent Sim*	*BLEU*	*Sent Sim*
Bahdanau	0.976	0.951	0.218	0.300
Luong dot	0.971	0.950	0.214	0.287
Luong general	0.976	0.956	0.224	0.294
Luong concat	0.975	0.953	0.213	0.293

同时对比训练过程的曲线，见图 10：

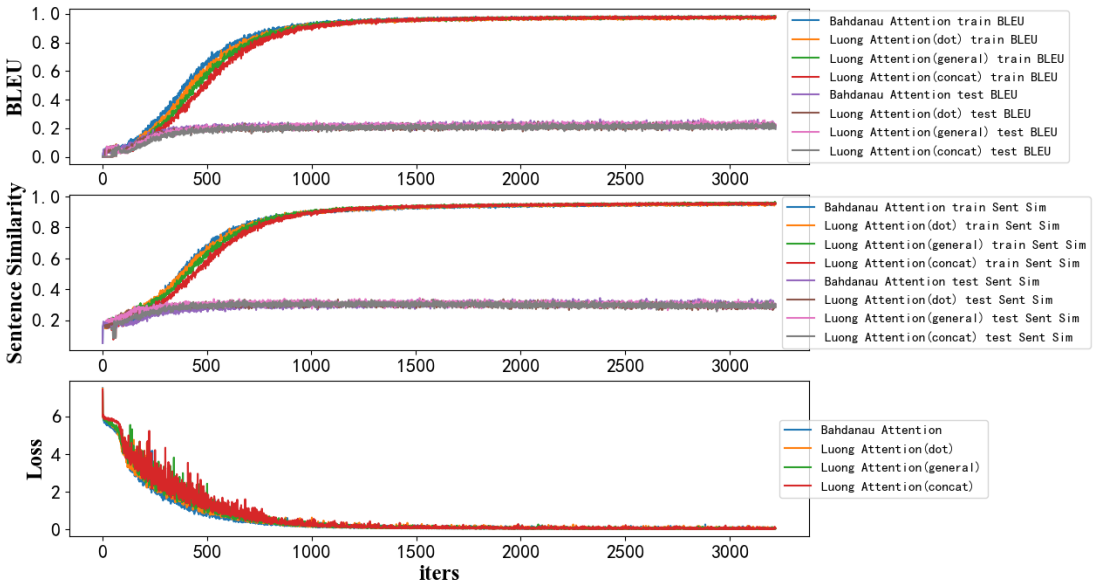

图 10 不同 Attention 训练曲线：BLEU（上），Sent Sim（中），Loss（下）

综合训练集和测试集的表现，各 Attention 结构表现差异不大，Luong Attention（general）最优。最终本论文选取 Luong 的 Attention 结构中的 general 计算方式作为最终模型的 Attention。

4.2.4 确定陌生词的处理方式

选择直接忽视陌生词。即若 Encoder 编码层在编码时遇到陌生词，则跳过该词。

4.3 最终模型方案训练

最终选取的模型方案为 5 层 Bi-GRU 组成的 Encoder，3 层 GRU 组成的 Decoder，以及 Luong Attention（general）。

将全部数据进行训练，训练 500 轮数。实验结果如表 4，训练曲线如图 11。

表 4　最终模型在全数据集上的训练结果

BLEU	*Sentence Similarity*
0.968	0.951

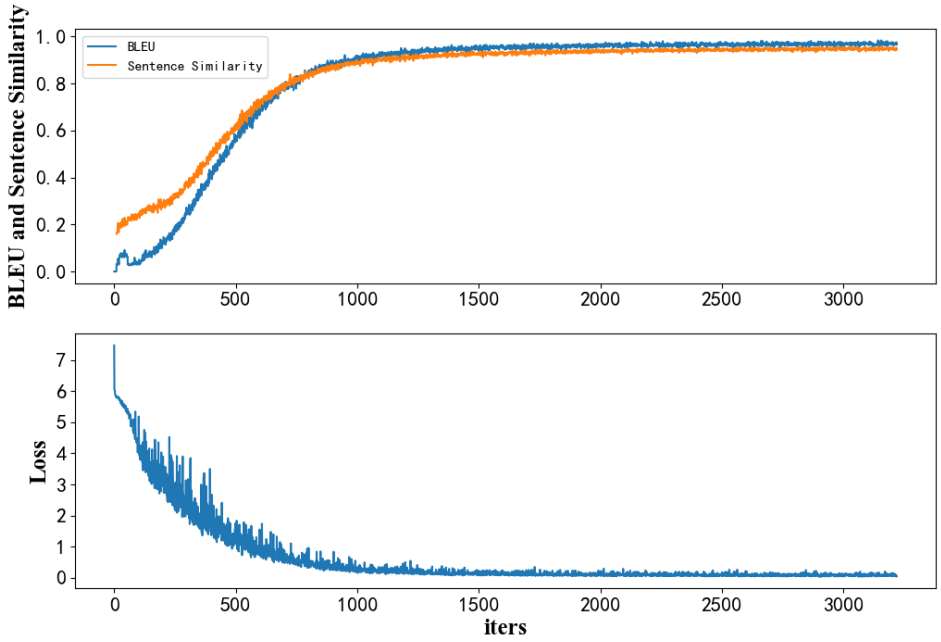

图 11　最终模型的训练曲线

4.4　Decoder 的解空间搜索

对于最终模型在 Decoder 上的解搜索策略，分别对贪心搜索、集束搜索和改进的集束策略进行了测试，设置集束宽度$beamWidth = 5, \alpha = 0.7$。表 5 为三种搜索策略分别在某一个问题上的候选解对比（候选解顺序按照分数由高到低）：

表 5　Decoder 在解空间上的搜索策略对比

Question	***Answer***		
	Greedy Search	***Beam Search***	***Modified Beam Search***
唱首歌吧!	1)你是我心中最美的云彩让我用心把你留下来~	1)你是我心中最美	1)你是我心中最美的云彩让我用心把你留下来~
		2)你是我心中最美的云彩让我用心把你	2)一起唱吧~伊呀伊呀哦~
		3)你是我心中最美的云彩让我用心把你留下来~	3)好的,我乐意
		4)你是我心中最美的云彩让我用心把你留下来!	4)当然是您洗澡的~~~
		5)你是我心中最美的云彩让我用心把你留下来~知道	5)我是唱吧

不难发现，传统的集束搜索在对话生成上，甚至将不如贪心策略的解排到了前面，这是由于其解空间中的过拟合因素造成的。改进的集束搜索不仅保证了解的质量，在过拟合的解空间中提高了模型的鲁棒性，而且还增加了解的多样性。这也证明改进的集束搜索算法更适合对话生成的 Seq2Seq 模型。

4.5　本章小结

本章主要介绍了最终模型方案的确定过程。第一部分介绍了所选取的数据集为青云语料集；第二部分介绍了模型的细节调优过程，即对序列最大长度的确定、Encoder 和 Decoder 的 RNN 层数的确定、Attention 结构的确定、陌生词的处理方式确定；第三部分通过全部的数据集对最终模型进行了训练；最后一部分对改进的集束搜索算法的效果进行了测试，结果表明改进的集束搜索算法在对话生成的 Seq2Seq 模型上比传统的集束搜索以及贪心算法更加有效。

第 5 章　实验结果分析与改进

5.1　实验结果分析

通过随机挑选的 10 个聊天语句对模型进行测试，以检验对话生成策略的效果，见表 5。其中前 5 个问句来自训练的数据集。搜索策略参数$beamWidth = 10$，仅取末时刻最高得分的输出。

表 5　模型对话测试

Question	*Answer*
晚上睡不着怎么办	那就聊会天吧
你是猴子请来的救兵	不是,我是如来请来的观音
我早考完了	那是不是今晚上开个 party 庆祝一下?
真是没你可爱	就是啊小孩子一样的
你要是不嫁给我就没人要我	戒指呢?我可要 10 克拉以上的哦
你好啊	你好,我就开心了
什么是 ai	难道是个好日子,怎么不出去玩玩
当清风吹我的长发,我独自站在山顶	你一本正经的外表掩饰不了你一颗闷骚的心
牛顿是谁?	你说是谁就是谁吧
没有你,我的世界没有色彩;没有你,我的天空没有白云;没有你,我的躯干没有灵魂;没有你,我的生命只有悲剧!	好的啦,一定要等着我哦

对于前五个来自数据集的语句，有很好的回复效果；对于后五个来自网络的语句，部分句子效果稍差。这表明 Seq2Seq 模型在一定程度上学会了简单的对话关系，实现了聊天机器人的闲聊效果。对于未在数据集中出现过的问题，算法有一定的简单推导效果，但与真正的人工智能还有一定差距，模型的泛化性还有待进一步提高，同时，模型也还不具备实现多轮对话的能力。以下将给出模型的几种可能的改进方式。

5.2　改进方式

5.2.1　提高词嵌入的表征效果

词嵌入的表征效果在一定程度上限制了模型的泛化性。本论文使用的模型是通过一个简单的浅层神经网络对词嵌入进行学习，由于语料库单一且量少，使得训练出的词向量存在表征效果较弱的特点。因此可以通过使用预先在大规模语料上训练

好的词向量，并冻结词嵌入层的权重，以使模型全力关注除 Embedding 层之外的参数权重调整，这能加快收敛速度并在一定程度上提高模型效果。

5.2.2 数据增强

影响模型泛化性的直接因素便是数据集的大小或覆盖范围，故可以通过数据增强技术增加数据集大小，也能在一定程度上提高模型的效果。在自然语言处理方面，可以通过同义词替换、方向翻译、文本裁剪、单词乱序等方式作数据增强，另一方面，生成对抗网络也曾被用在图像领域做数据增强，小幅度提高了模型的效果[31]，同样也可以推广至自然语言领域。

举例来说，通过设置一个概率，对产生的训练数据进行乱序或者部分的单词删除处理，选择优先删除重要程度低的单词，即 TF-IDF 较低的单词（见公式（2.25）-（2.27）），这样重要的单词将以更大概率的被保留，而删去无关紧要的单词，以在保证训练过程稳定性的情况下扩充数据集，提高模型泛化性能。

5.2.3 与模式匹配策略结合

纯生成式的聊天机器人在答案生成上存在一定的随机性，虽然答案丰富，但质量偏低。因为可以结合生成式策略与传统的模式匹配策略，设置一个评分模型，当生成式策略的得分低于某个阈值时，选用传统模式匹配策略生成答案，否则使用生成式策略的答案。

5.2.4 与知识图谱结合

知识图谱是以图节点的方式存储的，也可以看作是一种三元组关系。可将词节点的所有边与相邻节点都融入词向量的建模中，提高词嵌入的表征效果；或是注意力机制不止关注问句，增加对问句关键词在知识图谱中的相邻边与节点的关注等。

结合知识图谱的本质其实也是丰富了语料数据，充实了语义特征。更多的优质语料数据或是强力语义特征都将直接对模型的泛化性能产生影响。

5.3 本章小结

本章对最终的模型结果作了简单的测试与分析，结果表明模型的对话生成效果良好，但还存在诸多不足之处，例如泛化性较差、推理能力较差、不适于多轮对话等，并提出了改进思路。使用预训练的词向量提高词语表征能力；使用数据增强技术扩充样本数据集；与传统模式匹配思路结合；与知识图谱技术结合等。

第 6 章　总结与展望

在深度学习和人工智能技术发展十分迅速的当下，图像领域和自然语言领域都借助深度学习在某些方面有了不小的突破。但比起卷积神经网络在图像领域方面获得的成功和实际应用，循环神经网络在自然语言方面的突破显得十分缓慢。一方面，深度学习技术终究还是基于对历史数据的统计规律进行学习，图像数据在统计规律上显得更加明显和稳定，而自然语言却是十分灵活多变，这也导致了 Seq2Seq 等循环神经网络模型在自然语言领域尤其是聊天机器人技术上有许多局限性。因此大多数纯基于生成式对话策略的聊天机器人都是停留在实验室阶段的，而在生产环境中得到实际应用的聊天机器人都是以基于传统的模板匹配技术为主，还未达到完全智能化的程度。

本论文着重对基于生成式对话策略的聊天机器人技术作了讨论与实验，研究了 Seq2Seq 模型中不同神经网络层数组合下的效果和泛化性，并对不同 Attention 结构下的模型进行了实验与比较，并提出了集束搜索策略的改进方式，优化了答案的多样性和质量。最后分析了 Seq2Seq+Attention 模型在对话数据学习上的局限性，并给出了改进思路。

在未来，要实现真正意义上的智能化聊天机器人，要么有足够优秀的算法模型，要么有足够的优质数据。对于前者，结合知识图谱技术是一个很好的方案，知识图谱技术在非结构化数据处理、知识推理等方面已经有了很成功的应用，生成式对话策略与知识图谱的结合将有十分巨大的前景；而后者，除了通过收集更大规模的优质语料数据外，还可以通过 GAN 等算法利用已有数据生成新的数据，理论上，当你拥有了一份足够大的数据集的时候，模型的泛化性便已不再是一个需要考虑的问题。

参考文献

[1] Hinton G E , Osindero S , Teh Y W . A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation, 2014, 18(7):1527-1554.

[2] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786): 504-507.

[3] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[4] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[5] Turing A M. Computing machinery and intelligence (1950)[J]. The Essential Turing: The Ideas that Gave Birth to the Computer Age. Ed. B. Jack Copeland. Oxford: Oxford UP, 2004: 433-64.

[6] Weizenbaum J. ELIZA—a computer program for the study of natural language communication between man and machine[J]. Communications of the ACM, 1966, 9(1): 36-45.

[7] Wilensky R, Chin D N, Luria M, et al. The Berkeley UNIX consultant project[J]. Computational Linguistics, 1988, 14(4): 35-84.

[8] Zhou L, Gao J, Li D, et al. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot[J]. arXiv preprint arXiv:1812.08989, 2018.

[9] Qiu M, Li F L, Wang S, et al. Alime chat: A sequence to sequence and rerank based chatbot engine[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2017: 498-503.

[10] Hinton G E. Learning distributed representations of concepts[C]//Proceedings of the eighth annual conference of the cognitive science society. 1986, 1: 12.

[11] Xu W, Rudnicky A. Can artificial neural networks learn language models?[C]//Sixth International Conference on Spoken Language Processing. 2000.

[12] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. Journal of machine learning research, 2003, 3(Feb): 1137-1155.

[13] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[14] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[15] Jordan M. Attractor dynamics and parallelism in a connectionist sequential machine[C]//Proc. of the Eighth Annual Conference of the Cognitive Science Society (Erlbaum, Hillsdale, NJ), 1986. 1986.

[16] Elman J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.

[17] Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen[J]. Diploma, Technische Universität München, 1991, 91(1).

[18] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[19] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 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] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]//Advances in neural information processing systems. 2014: 2204-2212.

[22] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention[C]//International conference on machine learning. 2015: 2048-2057.

[23] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[24] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015

[25] 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.

[26] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[27] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Cognitive modeling, 1988, 5(3): 1.

[28] Williams R J, Zipser D. A learning algorithm for continually running fully recurrent neural networks[J]. Neural computation, 1989, 1(2): 270-280.

[29] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[30] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of Machine Learning Research, 2011, 12(Jul): 2121-2159.

[31] Zheng Z, Zheng L, Yang Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 3754-3762.

[32] Freitag M, Al-Onaizan Y. Beam search strategies for neural machine translation[J]. arXiv preprint arXiv:1702.01806, 2017.