电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

学士学位论文

BACHELOR THESIS

论文题目 **知识驱动的聊天机器人系统设计与实现**

学　　院 **计算机科学与工程学院**

专　　业 **计算机科学与技术**

学　　号 **2016070911017**

作者姓名 **张星迪**

指导教师 **卢国明**

摘　要

随着深度学习在自然语言处理领域的发展，学术界和产业界越来越重视人机对话系统，将其视为下一代智能人机交互的新形式。传统的端到端生成式神经网络能够建模语言模型，生成符合语境的回复，进行多轮、单论对话交互，然而这样的系统倾向于生成高频的万能回复，这种回复缺乏有用的信息，言之无物，无法给予用户帮助信息，这个问题严重限制了非任务驱动的人机对话系统的实际应用。

为解决这个问题，本论文设计并实现了一个引入外部结构化知识的神经网络对话模型，在基于动态注意力机制的序列到序列模型的基础上，引入外部知识的嵌入表示、“老师引导”的思想结合先验概率和后验概率指导的知识选择机制，在学习使用知识的同时减少模型在训练和测试时的行为差别，使神经网络可以选择和融合知识，指导回复的生成。此外本文引入对话路径作为输入，训练神经网络主动发起对话、组织对话。

最后，基于前面的工作，本文实现并部署了一个电影领域的主动式的闲聊对话系统。该系统可以主动发起、组织聊天，在聊天过程中根据对话历史和知识生成富含信息的回答。

关键词：自然语言处理，对话系统，生成式模型，深度学习，知识驱动人机对话系统

ABSTRACT

The development of deep learning in the field of natural language processing has facilitated the research in dialogue systems in recent years, bringing increasing attention to dialogue systems from academia and industry. Traditional end-to-end generative neural network can model language, generate contextual responses, and conduct multi-round, single-round dialogues. However, such systems tend to generate high-frequency responses that are correct but useless. This kind of reply lacks useful information and can't give users help information. This problem severely limits the practical application of non-task-driven human-machine dialogue systems.

To solve this problem, this work designs and implements an end-to-end human-machine dialogue model that introduces external structured knowledge. Based on the standard attention-based sequence-to-sequence model, this work introduces the use of knowledge embedding, and the use of a scheme that combine "teacher forcing" with prior probability and posterior probability to guide knowledge selection. This scheme can help model learn how to incorporate external knowledge, while reducing the difference in model's behavior during training and testing phase, enabling neural network to select and fuse knowledge and guide the generation of meaningful replies. In addition, this thesis introduces the dialogue goal as input, and trains the neural network to initiate dialogue and organize dialogue actively.

Finally, based on the previous work, this thesis implements and deploys an proactive dialogue system in the field of movies. The system can initiate and organize dialogues actively, and generate answers substantively based on the conversation history and knowledge during the chat process.

Keywords: natural language process, conversation models, generation, deep learning, knowledge-driven dialogue system

目 录

第一章 绪 论

1.1 研究工作的背景与意义

赋予机器具有与人沟通的能力，是人工智能领域的一项极其重要的子任务，构建智能对话系统既是一项非常具有挑战性的任务，也是走向通用人工智能的必经之路，因此吸引了非常多研究人员的兴趣，在这个领域开展了广泛的研究。但由于人类的语言是意义丰富的符号系统，简简单单的单词、短语文本中蕴含大量语言信息，这给机器理解和处理人类自然语言带来了巨大的困难，也导致很长时间以来，人工对话系统的应用场景一直都非常局限。因此长久以来，拥有一个有着足够智能的聊天伴侣系统或虚拟助手听上去都是虚无缥缈的、都是只存在于科幻电影中的情节。

最近几年，随着机器学习技术、理论的发展，尤其是随着卷积网络（CNN[1]），循环神经网络（RNN、LSTM[2]、GRU[3]），transformer[4]，生成对抗网络（GAN[5]），序列到序列[6]等神经网络技术开始在自然语言处理领域(Natural Language Process, NLP)应用，借助这些强大的神经网络，构建一个智能的人机交互系统，作为我们的个人助理，帮助我们的日常生活，已经不再像以前听起来那么天方夜谭、遥不可及了。一方面，深度学习技术已经被证明在识别抽取数据集中的复杂模式上是有效的，并且已经跳出计算机视觉的范畴，促进了各种研究领域的研究工作，比如基于深度学习的语音识别、表情识别、智能电网(Smart Grid)、信号检测、车载雷达测距等等。大量研究人员也在利用深度学习和海量的数据构建高性能的对话系统。另一方面，伴随机器学习的突破性发展，带来大量的资金、人力物力投入到开源数据集构建中，我们可以很轻松地获取整理好的互联网上的大型对话数据集，从而对于（几乎）任何输入学习应该如何回复。这极大的方便了我们建立完全基于数据驱动的人机对话系统。

产业界也极度看重对话系统作为摆脱键盘、鼠标和屏幕等传统媒介的下一代人机交互的主要形式的潜力，近年来投入大量人力到相关研发工作中。现在已经有一些商用系统进入了批量生产、大规模开放使用的阶段，比如苹果手机搭载的 Siri、微软操作系统搭载的 Cortana 为代表的语音助手，结合对话系统和语音识别等技术，赋予人与设备新的交流形式。以谷歌 Home 和天猫精灵、小米智能音箱小爱等为代表的虚拟助手式智能音箱也已经在陪伴千千万万的消费者了。另外，在开放领域对话系统方面，微软针对不同语种开发了使用用户达到千万级别的聊天机器人 Zo、微软小冰等。可以想象在未来对话系统技术更加成熟的时候，现在

陪伴人们的略显笨拙的 Siri 会成为人们工作生活中最重要的贴心管家和伴侣。

人机对话系统是自然语言处理领域极其重要的一个子任务，包含着丰富的学术意义和应用价值。对话系统与自然语言处理的各个其他领域的技术紧密相关，又常常互相结合在一起进行实际应用。人机对话系统的进步依赖于自然语言处理中各种相关技术，同时一个优秀的人机对话系统也会促进 NLP 中各种相关技术如词嵌入表示、词性分析、语言模型、语义解析、文本和语言生成、对话策略优化等的发展。· ·

1.2 知识驱动对话系统的国内外研究历史与现状

1.2.1 对话系统概述

对话系统的分类有多种方式。一般来说，对话系统可以分为三大类，即问答型系统，闲聊型系统，任务完成型系统。问答型系统是一种单轮会话模型，用户提出问题，系统根据问题从数据库中检索相关的回答。一个简单的问答型系统的示例是常见网络的 FAQ(Frequently Asked Questions)页面，该页面使用硬编码的方式，手写脚本来回答用户常见的问题。问答系统是简单的单轮对话系统，而其他常见的对话系统可能支持多轮对话，对话系统的反馈会由多轮对话历史共同决定。闲聊型系统的目标是生成单轮或多轮对话的人类语言的响应，而无需完成特定任务或回答任何问题。与之对比的则是任务完成型系统，任务完成型系统会有特定的任务需要完成，比如电子商务平台的对话系统将通过单轮或多轮对话尝试引导客户选择商品或填写订单表格，酒店服务的对话系统将引导客户明确预订的酒店时间、地点等信息。

根据构建对话系统的技术，对话系统又可以分为五大类：流水线模型，端到端模型，基于生成模型的方法，基于检索的方法和混合方法。图 1-1 展示了五种对话系统之间的关系。

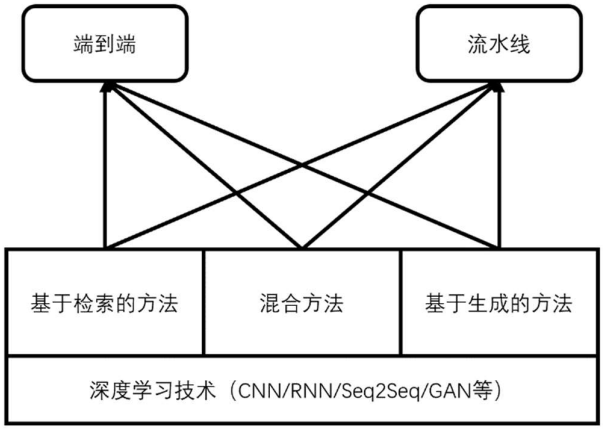

图 1-1 对话系统分类

流水线模型（管道模型）：传统流水线模型将对话任务分解为四个步骤，交付给不同的模块处理。自然语言理解模块（Natural Language Understanding, NLU)用预定义的语义槽(slot)解析输入的用户话语，比如把任意的句子都解析为“主谓宾”三个语义槽。对话状态跟踪器 (Dialog State Track, DST)记录对话历史（语境），输出当前的对话状态，比如记录任务完成型对话系统当前的任务完成进度。对话策略学习模块(Dialogue Policy, DP)根据当前对话状态决定下一步的动作。自然语言生成模块(Natural Language Understanding Generation, NLG)将选择的动作映射到自然语言层面，生成回复。使用深度学习来强化流水线各个步骤的文献非常多，包括使用 RNN 的槽位(slot)语义理解[7][8],使用强化学习技术学习 DP 策略[9],基于 LSTM 实现的语言生成模块[10]，基于神经网络语言模型的语言生成模块[11]等。

端到端模型：所谓端到端即从输入端到输出端，模型接受用户输入的原始话语数据，输出自然语言的结果（回答），中间的神经网络自成一体，加上神经网络缺乏可解释性，整个网络可以视为一个黑匣子。

基于生成的方法：该方法使用对话语料库训练神经网络，使神经网络学习输入与输出之间的对应关系。强大的深度神经网络可以通过大规模数据语料库进行训练，从而了解对话本身的潜在关系，因此在此方法中不需要手动编码对话,节省了大量的人力工作。Sequence-to-Sequence (seq2seq)方法[6]的成功带来了大量的基于生成方法的端到端模型,比如基于 RNN 的模型[12][13]，基于层次 RNN 的模型[14]等。

基于检索的方法：检索方法类似于文件查询，它包含一个会话语料库（索引数据库），通过对用户输入的语言进行语义解析，然后将语义信息作为键值查询会话语料库中相对应的回答。从匹配的待选队列中排序、选择最佳回答。该方法的

关键是让匹配模型克服查询和回复之间的语义差异。基于检索方法的多轮对话模型尝试用神经网络进行语义的匹配查询[15],[16][17]。

混合方法：将检索式和生成式相结合的方法。还有一些系统不限于使用单一方法，而是同时使用搜索和生成方法。结合他们的信息，构建对话系统。

考虑到对话话题范围，所有对话系统又可以分为两类：受限域对话系统和开放域对话系统。受限域系统可以理解人类输入语言，并在诸如电影，音乐，体育，电子商务等有限领域中产生响应。与开放域系统相比，它在构建时更容易实现，因此在现实世界的应用程序中比开放域系统中更常见。顾名思义，开放域系统是一种可以在不受限制的区域内进行人机沟通的对话系统。人类可以根据自己的意愿切换主题，无论主题是什么，系统都能够正确做出反应。相应的，开放域系统的构建难度和完备性要求会远远高于受限域系统。

最后，对话系统还可以分为主动式、被动式、单轮、多轮，这是根据对话的形式划分的。主动式系统不需要人的输入，主动发起对话。被动式则更加普遍，接受人的输入并产生响应。单轮对话系统常见的形式为用户输入一次问题，系统给出一次回答后结束对话。多轮对话系统则包含任意多轮的信息交互。

1.2.2 知识驱动的对话系统

相较于计算机视觉领域，深度学习技术在自然语言处理领域遇到的困难更多，除了因为自然语言处理的数据集的质量往往相对较差以外，也和自然语言本身的特点有关系。

建模语言特征、理解自然语言对于计算机来说是非常困难的，原因有两个方面。第一个方面在于自然语言使用的基本单元是「单词」而图片、视频的最小单位则是「像素」，语言中的「单词」是高智能人类使用的符号系统，每一个「单词」背后关联着一系列的意元，多种意元组织成丰富、并且随着语境动态变化的语义信息。例如，当我们谈论苹果（Apple）时，我们人类就知道这是一家电子设备公司，当然在另一个语境下会是一种水果，但是理解一词多义对于机器来说，却是一种难以想象的困难。因此，建模语言、构建对话系统的难题之一是如何打破「单词」这个屏障，更精确地捕捉词句背后的语义信息。[18]

第二个方面在于，人类间的交流和对话系统之间有一个巨大的横沟：对话是否与现实世界的知识相集成。人际交流中包括各种背景知识，比如当对话谈论到苹果（Apple）时，我们也极有可能在接下来的对话中涉及 Apple 和 Microsoft，IBM 的产品，由于它们都是 IT 公司，因此我们将本能地将他们联系在一起讨论。但是，完全由数据驱动的系统在现实世界中仍然缺乏基础，无法访问外部知识（文本或

结构化知识），这使得此类系统难以产生实质性的响应。

深度学习是一种强大的表示学习方法，它所提供的解决方案是，将词的语义信息表示为词向量(word2vec[19])，通过大量数据训练，得到词之间的相互关系，然后以词嵌入向量形式表示词，这种分布式的表示不能准确刻画一个词的含义但能用词彼此间的关系组合出句子语义，从而支持各种下游自然语言处理任务（如对话系统、文本分类）。这种在大规模数据集上预训练后在特定任务上微调的模式是目前最广泛使用的构建范式。深度学习的表示学习方法主要被诟病的问题是缺乏可解释性，我们只知道某个嵌入的低维实向量表示了每个词，但具体是如何表示的却不能理解。甚至每一次重新训练后，词的分布式表示也完全不同。

词嵌入向量也无法解决第二个问题，如何利用知识？人类的各种行为都涉及背后的知识和潜在的推理，我们完成各种任务时都会有这么一个「世界模型」进行指导。借助这个「世界模型」可以帮助我们做出决策、对任务的决策做出理性的解释[18]。纯粹的数据驱动方法只能从大规模文本中学习与任务相关的某些模式（pattern），并且目前的深度学习方法依然无法确切的理解、利用和改造这些模式。可见在对话中融合知识是构造高级智能系统的必要条件，但也是目前深度学习无法解决的难题。因此构建一种对话系统，可以利用知识在闲聊中产生吸引人的、有意义的或个性化的回答是一个困难但重要的研究方向。

知识驱动的智能系统，最早可以追溯到深度学习技术发展以前的专家系统、产生式系统，利用知识进行推理，但这些系统需要大量的人力工作，并且实际应用能力非常的局限。现代信息技术、深度学习技术为核心的表示学习促进了对于知识库的建模、使用。目前人类已经花费巨大的人力物力构建了 WordNet[20]，FreeBase[21]等大型知识库。关于知识（知识图谱）表示的研究非常多，比如非常成功的翻译模型 TransE[23]和整个 Trans 家族 TransH[24], TransR[25]，TransD[26]，TranSparse [27]。经过知识嵌入表示的知识可以从其表示向量中得到实体间的关系，是计算机理解知识的核心步骤。在本文的第二章第三节将介绍知识表示的相关研究。

近几年来，利用知识构建对话系统的研究非常热门。2016 年，Vougiouklis P, Hare J, Simperl E.等构建了使用知识的闲聊系统[28]。Ghazvininejad M, Brockett C, Chang M W, et al. [29] 和 Zhu W, Mo K, Zhang Y, et al. [30]各自提出了一种端到端的知识驱动对话系统。Lian R, Xie M, Wang F 等[31]提出了一种在网络以注意力计算概率分布的学习选择知识的端到端对话系统。这些研究使用结构化或非结构化的知识文本。而 Zhou H, Young T, Huang M 等[32]提出使用动静态的图注意力机制，将知识图谱表示与对话生成结合。Shen Y, Deng Y, Yang M 等[33]使用双向 LSTM 结合

知识做问答关系的匹配。Zhang Y, Ren P, de Rijke M 等[34]和 Li Z, Niu C, Meng F 等[35]则分别使用循环神经网络和 transformer 结构[4]学习从非结构化的文章、背景知识中生成“阅读文章并回答问题式的对话”。

1.3 本文的主要贡献与创新

本文设计并实现了一个可以利用外部知识的，基于神经网络的端到端生成式、主动式闲聊人机对话系统，主要工作包括以下两个方面：首先，利用深度学习技术构建了一种融合外部结构化知识（事实）的端到端闲聊人机对话模型，在传统的端到端生成模型的基础上，通过知识选取、知识融合的步骤，实现引入外部结构化知识指导模型生成回复的目标。其次，引入对话目标作为网络的输入数据，训练神经网络模型学习如何组织对话并完成话题切换，以实现对话目标，并通过话题、知识之间的相互关联一定程度上提升对话蕴含的信息含量。最后，设计并实现了一个具备良好扩展性的电影领域对话服务系统，并以之为例，对系统的功能进行了深入的验证和分析。本文主要的创新点与贡献如下：

1. 提出了一种使用知识嵌入表示、先后验概率结合 Teacher Forcing 技术实现选择知识的深度学习的端到端的模型，该模型可以引用外部知识实现人机对话。

2. 不同于大部分人机对话系统只能回复人的询问（query），本论文最终构建了一个主动式的人机对话系统，可以主动的发起、组织对话。具体属于基于生成方法的、端到端的、主动式的、多轮的、受限域的闲聊对话系统。

1.4 本论文的结构安排

本文的章节结构安排如下：

第一章为绪论，讲述人机对话系统、知识驱动对话系统相关的研究进展。第二章第一部分将详细讲述论文相关的深度学习理论基础，涉及到多层感知机、循环神经网络、门控循环神经网络、词向量、序列到序列建模、动态注意力机制等。第二部分将简单介绍知识表示的相关理论。第三章介绍本文实现的基于深度学习的端到端的神经网络对话系统，第一节具体描述使用的数据集、定义何为主动式对话系统，第二、三、四节介绍本文的神经网络对话系统如何根据语境选取知识、融合知识、生成对话，第五节为实验，通过主客观指标的性能对比，展示了本文模型的有效性，第六节简述项目的部署方案。第四章为全文总结与展望。

第二章 深度学习理论基础

深度学习首先在计算机视觉领域取得了突破性的进展，这和卷积神经网络（CNN）能非常好的抽取图片局部纹理信息有关。而对于自然语言来说，文本序列的特征抽取非常困难，符号系统背后的语义更是难以概括和理解。正是依靠递归神经网络（RNN），Transformer、Seq2seq 架构等新技术，现代神经网络才可以捕捉到文本的某些特征。这些强大的神经组件已被广泛使用在各种 NLP 问题中，例如对话系统，潜在语义分析，蕴涵性分析，文本分类等。本章将会介绍和本文密切相关的深度学习自然语言处理的基础知识、介绍知识图谱、知识表示。

2.1 神经网络

与本论文关系紧密的神经网络结构包括多层感知机，循环神经网络，门控循环神经网络、长短期记忆循环神经网络，他们是深度学习自然语言处理的核心技术，接下来将本文简述他们的实现原理。

2.1.1 多层感知机

线性层（全连接层）是最基础的人工神经网络结构，本质上就是普通的矩阵乘法。而多层感知机(MLP[36])是由多个线性层（全连接层）和非线性层堆叠组合形成的简单的神经网络。图 2-1 多层感知机结构示意图具体描述了一个多层感知机的结构，展示了一个第一层有四个神经元，第二层有三个神经元，第三层有一个神经元的三层的多层感知机。

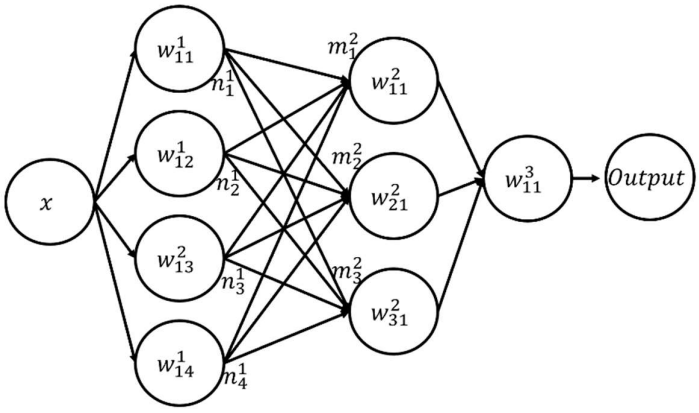

图 2-1 多层感知机结构示意图

多层感知机网络中若干节点构成了输入层（第一层）、输出层（最后一层）、若干隐藏层（中间的若干层）。层与层之间包含非线性的激活函数。简单来说任何神经网络，包括 MLP 就是若干线性运算与非线性运算的组合，线性层中的矩阵作为网络的权重，也是我们训练的目标，可以理解为网络的记忆区。而非线性运算的作用在于划分网络的层次，假如一个多层的神经网络层次间没有非线性运算，多层的线性运算为多个矩阵乘法，可以合并成一个矩阵乘法，换句话说，没有非线性运算，神经网络就永远只有一层。常用非线性激活函数包括 RELU 函数如公式（2-2），双曲正切函数(tanh)如公式（2-3）和 sigmoid 函数如公式（2-4）所示。根据任务，多层感知机网络常常选用不同的非线性运算。比如概率、分类的问题常常选用 sigmoid 函数，因为 sigmoid 函数输出的范围为 0 到 1，恰好可以表征概率值，后文将遵循惯例用σ代表 sigmoid 函数。

多层感知机网络是一种前向（前馈）网络，前向网络通过在网络的每个节点上做出的一系列线性和非线性的操作传递信息，一层一层传递下去，这个过程也被称为前向传播。多层感知机的每一层包含独立的网络权重和非线性函数，多层感知机的多个层间的数学变换组合成一个关于输入的复合函数$\mathrm{F_m}$，多层感知机的输出和输入间的关系如公式（2-1）所示。

$$Y' = F_m(X) \qquad (2-1)$$

$$RELU(x) = max(0, x) \qquad (2-2)$$

$$tan\,h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (2-3)$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \qquad (2-4)$$

多层感知机内部的数学运算过程并不复杂。以第 l 层和第 l+1 层间的运算过程为例。定义 m^l_i 表示第 l 层的第i个神经元的输入，n^l_i 表示第 l 层第i个神经元的输出, w_{ij}^l 表示 m^l_i 连接到下一层 m^{l+1}_j 的权重，b^l_j 表示第 l 层第i个神经元的线性偏置常数(bias)，偏置同样是可以学习的，有的神经网络层也会取缔这个偏置。那么第 l 层输出是其输入与当前层所有神经元（及偏置）的线性组合，如式(2-5)、(2-6）所示。

$$n_i^l = \sum_{j=1}^{N_l} w_{ij}^l m_j^l + b_i l \qquad (2-5)$$

$$m_i^{l+1} = \sigma\left(n_i^l\right) \qquad (2-6)$$

多层感知机网络的具体训练过程如下：首先进行前向传播，输入数据 X 逐层流过多层感知机的每一层后得到输出$\mathrm{Y'} = \mathrm{F_m(X)}$。然后计算$\mathrm{Y'}$关于待拟合函数F的损失,比如计算损失 L 为$\mathrm{Y'}$与$\mathrm{Y} = \mathrm{F(X)}$ 间的均方误差或 KL 散度距离等。接着损失

函数反向传播回神经网络的各层权重，即求损失函数关于各层权重的偏导数（梯度）。接下来按照梯度下降原则更新多层感知机的各层权重，关于梯度下降的参数更新会在 2.1.4 节介绍。根据梯度下降的原理，Y'关于待拟合函数 F 的损失 L 会在每一次迭代更新后变得更小。最终当这个损失足够小时，多层感知机的输出Y'等同于Y，那么多层感知机等同于拟合了函数$F(X) = Y$,就这样虽然我们不能显示的表示出目标拟合的函数$F(X) = Y$，却通过反向传播逼近拟合了它。我们也可以想象越大、结构越复杂的神经网络包含越多的层、参数，可以拟合越复杂的函数。另外，如何寻找适合特定问题的神经网络的结构，即神经网络架构搜索（NAS）也是目前深度学习的研究非常热门的领域。

2.1.2 循环神经网络

前向网络通过逐层间的操作，数据信息从输入到输出单向流动（正向），而对应的梯度则反向流动由损失端传回输入端。而循环神经网络的机制则完全不同，循环网络不仅将当前的输入样本作为网络输入，还将它们之前感知到的信息（自身的隐藏状态）一并作为输入。具体而言，循环神经网络指一种包含隐藏状态的神经网络，并且引入时间步的概念，每次输入一个数据为一个时间步，每个时间步的输出不仅仅由当前的输入决定，还由当前的隐藏状态决定。

图 2-2 描述了循环神经网络的结构，将 RNN 展开后可以看到包含一个不断向后传递的内部隐藏状态，这使得 RNN 具备对序列建模的能力。这种序列建模能力使得 RNN 可以高效的抽取语音序列、信号序列、文本序列等的信息，并且 RNN 按输入单词的顺序逐个处理单词，不仅能够抽取序列中每个单词的语义特征，还天生对输入的单词有“谁在前谁在后”的顺序的认知，这恰好与自然语言的顺序语言模型理论不谋而合。而像 Transformer 这样的并行结构[4]处理整个句子时，为了让神经网络清楚一个句子中词之间的先后顺序还得加入特殊的位置信息。正因如此，近年来，RNN 一直是自然语言处理最常用的核心技术。

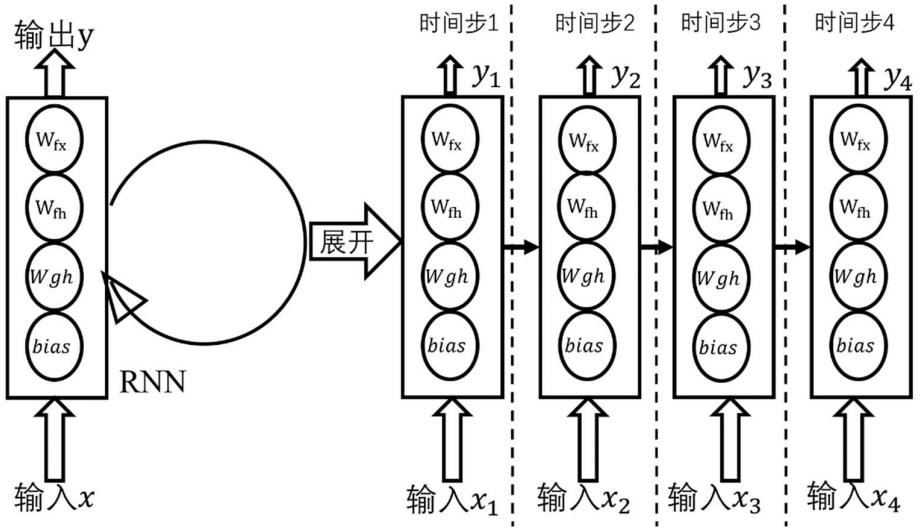

图 2-2 循环神经网络示意图

一个循环神经网络单元可以被视为一个独立的神经元，接受当前输入，根据神经元自身过去的记忆和当前输入进行下一步动作。神经元内部的数学运算过程可以简单的抽象成公式（2-7）、（2-8）。其中W_{fx}、W_{fh}、b_f、b_g、Wgh为网络内部参数，h_{t-1}表示 t-1 时刻的隐藏状态，x_t表示 t 时刻的输入，y_t为 t 时刻的输出。显而易见，针对某个时间步 t，首先根据前一个时间步 t-1 的隐藏状态h_{t-1}更新当前时刻的隐藏状态，然后计算当前时刻的输出。不同于前馈神经网络，在 RNN 的计算过程中，不同时刻网络参数是完全共享的，没有层到层间的信息传播，只有上个时间点神经元自身到这个时间点神经元自身的信息传播。

通过上述的方法，理论上可以编码任意长度的序列。但事实上，如公式（2-9）所示，RNN 的输出与其内部参数W_{fx}、W_{fh}、b_f、b_g、Wgh，呈现指数级的运算关系，随着序列的长度增长，这给反向传播带来了难以想象的计算时间复杂度，最严重的是，产生指数级的梯度爆炸的问题。假如最后一层损失函数关于某个参数的梯度为 5，经过十五个时间步的反向传播，这个梯度就已经达到 5^{15}，接近为无穷大，不仅仅计算机机器的数值表示方法已经难以表示这么大的数，即使数值表示不溢出，神经网络也会一次变化过大、极度不稳定，难以学习。

$$h_t = f\left(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + b_f\right) \qquad (2-7)$$

$$y_t = g\left(Wgh \cdot h_t + b_g\right) \qquad (2-8)$$

$$h_t = f\left(W_{fx} \cdot x_t + W_{fh} \cdot f\left(W_{fx} \cdot x_{t-1} + W_{fh} \cdot h_{t-2} + b_f\right) + b_f\right)$$
$$= \cdots = F\left(W_{fx}^{t-1} \cdot x_1 + W_{fh}^{t-1} \cdot h_1 + R\right) \qquad (2-9)$$

2.1.3 门控循环神经网络、长短期记忆循环神经网络

循环神经网络的梯度爆炸问题是致命的，这导致普通的 RNN 遇到稍长的文本序列时就会无法训练。门控循环神经网络、长短期记忆循环神经网络都是通过更加复杂的神经元操作，引入各种门控操作，减缓梯度爆炸带来的问题。

2.1.3.1 门控循环神经网络（GRU）

本质上门控循环神经网络（GRU[3]）、长短期记忆网络（LSTM[2]）与普通的循环神经网络没有特别重大的区别，只不过 GRU、LSTM 在神经元内部设定了比 RNN 更加复杂的运算。

GRU 在神经元内部主要分为对于记忆选取和记忆更新的两种操作。

记忆选取：GRU 引入了记忆控制门z_t，该门根据当前的时间步输入和自己的记忆给出一个限流值（系数），该限流值用于控制前一时刻的历史信息有多少可以被带入到当前时刻的隐藏状态，成为自己现在的记忆，限流值越大，过去记忆进入当前隐藏状态的就越少。

记忆更新：GRU 引入了记忆重置门r_t，同样根据过去的历史记忆与当前的时间步输入，得出一个重置系数，控制前一状态有多少信息被写入到当前的候选集$\widetilde{h_t}$上，可以理解为选用多少历史记忆的信息用于处理当前输入，重置系数越小，前一状态的信息被写入的越少。最终的记忆更新（状态更新）为记忆控制门z_t选择的过去记忆$(1-z_t)*\mathrm{h_{t-1}}$和根据当前时间步形成的新记忆$z_t*\widetilde{h_t}$的和。

大量的实验表明这样的设定可以减缓梯度爆炸，不仅仅让循环网络可以处理更长的序列，而且可以取得更好的性能。

图 2-3 描述了门控循环神经网络的结构。公式(2-10，11，12，13，14)描述了门控循环神经网络的数学本质。其中W_r、W_z、$W_{\widetilde{h}}$、W_o为网络内部参数，$\mathrm{h_{t-1}}$表示 t-1 时刻的隐藏状态，$\mathrm{x_t}$表示 t 时刻的输入，y_t为 t 时刻的输出。

$$r_t=\sigma(W_r\cdot[\mathrm{h_{t-1}},x_t]) \qquad (2-10)$$

$$z_t=\sigma(W_z\cdot[\mathrm{h_{t-1}},x_t]) \qquad (2-11)$$

$$\widetilde{h_t}=tanh(W_{\widetilde{h}}\cdot[r_t*\mathrm{h_{t-1}},x_t]) \qquad (2-12)$$

$$h_t=(1-z_t)*\mathrm{h_{t-1}}+z_t*\widetilde{h_t} \qquad (2-13)$$

$$y_t=\sigma(W_o\cdot h_t) \qquad (2-14)$$

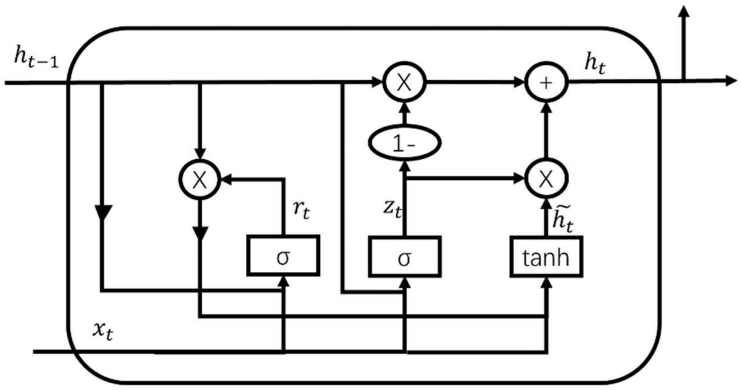

图 2-3 GRU 内部结构示意图

2.1.3.2 长短期记忆循环神经网络

虽然本文最终的网络并没有使用 LSTM，但为了更全面的介绍循环神经网络，LSTM 作为 RNN 最重要的变体之一非常重要。与 GRU 十分类似，长短期记忆（Long short-term memory, LSTM）也在神经元中引入了门控机制，通过门控系数来控制历史记忆、当前状态的更新。LSTM 设定了丢弃门、输入门、输出门三个门来控制历史记忆和状态，较好的缓和了长序列训练过程中的指数级梯度的问题，相比普通的 RNN，LSTM 在处理长序列能力、综合性能上都有更好的表现。

丢弃门：如 GRU 一样，第一步是选择历史信息，选择神经元还需要历史记忆中的多少成分。丢弃门就负责筛除无用的历史记忆信息，该门会读取$\mathrm{h_{t-1}}$和$\mathrm{x_t}$输出丢弃门控系数f_t，表示对过去信息的保留程度。如公式(2-15)所示。

输入门：输入门是决定让多少新的输入信息$\mathrm{x_t}$成为影响神经元状态的有效输入中来。对输入信息的处理分为两步：如公式(2-16)，首先构造输入有效系数，经过输入先选层，计算输入有效系数i_t，这个系数越大输入的有效性越高。接着构造实质的输入信息，如公式(2-17)。最后将输入和历史这两部分联合起来，对神经元的状态进行一个更新。如公式(2-18)。

输出门：输出门负责构建输出信息，这是和神经元自身状态如何更新无关的一个门。主要将神经元的隐藏状态映射成输出信号。这就像神经元在大脑经过信息、记忆的组合进行了复杂的更新操作以后，决定了要说一句话，而输出门则将大脑的（神经元的）内部动作信号翻译成实际的语言。输出门具体使用线性运算和 tanh 函数、σ函数得到输出结果。如公式(2-19,2-20)所示。

$$f_t = \sigma\left(W_f \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}]\right) \qquad (2-15)$$

$$i_t = \sigma(W_i \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + b_i) \qquad (2-16)$$

$$\widetilde{C_t} = tanh(W_C \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + b_C) \quad (2-17)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (2-18)$$

$$O_t = \sigma(W_o \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + b_o) \quad (2-19)$$

$$h_t = O_t * \tanh(C_t) \quad (2-20)$$

2.1.4 神经网络优化算法

2.1.4.1 Adam 优化器

神经网络的训练是建立在梯度下降原理上的。在最简单的梯度下降法中，网络的迭代公式如（2-21）所示，$\mathrm{W_t}$表示第 t 次迭代后的网络参数，lr表示学习率。$\frac{\partial \mathrm{L}}{\partial \mathrm{W_{t-1}}}$表示损失函数L关于当前网络参数$\mathrm{W_{t-1}}$的偏导数。

$$W_t = W_{t-1} - \frac{\partial L}{\partial W_{t-1}} \times lr \quad (2-21)$$

一般而言，理论上我们在数据集 D 训练模型则应该计算 D 上所有数据点的损失和梯度进行优化，这样才是全局有效的学习，但是实际中我们工作的数据集 D 非常大，受限于计算设备的内存资源，我们几乎不可能一次性计算得出 D 上所有数据点的损失函数L，并进行全局的梯度下降优化。因此按照深度学习的惯例，会把数据集 D 划分为 n 个小组（batch），每个小组（batch）中包含批大小（batchsize）个数据点。每一次迭代计算一个小组（batch）上批大小（batchsize）个数据点的输出，然后计算输出对应的损失函数L，最后进行梯度下降优化。每次抽样的小组是随机的，因此这种优化方法也被称为“分批随机梯度下降法”（Mini-Batch SGD）。每一步训练只使用一个小组进行训练被称为一个训练步（step）每次遍历完全部的小组，等于遍历完一次数据集 D，被称为一个训练周期（epoch）。当然，批大小（batchsize）越大，训练的过程越接近全局最优化的训练模式，训练的损失函数也会更加稳定，为此也有很多深度学习的开发人员在使用多步损失累加然后在反向传播的“梯度累积”方法。

在随机梯度下降法的基础上，RMSpropt[37], Adam[38]等优化方法则考虑到网络参数的维度大小、如何避免“陷入”局部最优点等问题对随机梯度下降法进行改进。本文最终也选用 Adam 优化器，因此重点介绍 Adam 优化器的原理。

Adam 优化器由 Kingma D P, Ba J 等人于 2014 年提出，也被为带“惯性”的自

适应学习方法或带“动量”的自适应学习方法。其核心思想在于携带“动量”和维度自适应，并且不仅仅计算一阶矩、还计算目标函数的二阶矩指导梯度下降的方向。携带“动量”是指 Adam 优化器会累积之前的梯度作为惯性，每一次迭代时的学习方向由当前梯度方向和之前累积的梯度（惯性）方向共同决定，指导网络的学习方向。从直觉上说，梯度下降如同从山顶往山下滚石头，带“动量”的学习方法如同利用石头滚下山时的惯性，带动石头跳过局部的小坑，即避免陷入局部最优点。

其数学公式为：

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (2-22)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (2-23)$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \quad (2-24)$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad (2-25)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \varepsilon} \widehat{m_t} \quad (2-26)$$

其中m_t和v_t，可以理解为积累的惯性，m_t为一阶矩的估计（均值），初始化为 0 向量；v_t为二阶矩的估计（方差），初始化为 0 向量。Adam 的作者发现当衰减率（decay rate）很小时（β_1和β_2接近于 1），m_t和v_t由于初始化为零，整个训练过程中都趋于零。因而 Adam 算法又提出对一阶矩m_t和二阶矩v_t进行数值校正，得到$\widehat{m_t}$，$\widehat{v_t}$。θ_t代表网络参数，β_1和β_2的上标 t 代表 t 次方，其余的下标 t 代表是时间步。默认情况下，β_1=0.9，β_2=0.99，ε=10^{-8}，η=0.001。

2.1.4.2 Warmup Optimizer 学习策略

论文《Attention is all you need》[4]提出了 Warmup 学习策略，Warmup 是指根据训练周期(epoch)或训练步（step）对于学习率lr的动态调节。在《Attention is all you need》[4]之前，也有多种动态调节学习率lr的动态策略，比如公式（2-27）展示了一个常用的阶段性调节学习率的方法。这种动态调节学习率的模式广泛的在深度学习的各个领域中使用。根据论文[4]描述，使用 Warmup 策略根本问题在于，Adam 自适应学习率优化器的方差太大，尤其是在训练的早期阶段，容易基于有限的训练数据进行过度的“跳跃”，波动较大，Warmup 策略克服了这种波动，体现出了良好的适应性。而与其他的动态策略相比，Warmup 策略需要更少的人为设定的超参数，并且在多种网络和任务上都有较稳定的表现。

Warmup 的原理如公式（2-28）所示,d_{model}为网络的中间隐向量大小，原文中

为 512，$steps$为当前迭代步数，$warmup_step$为 Warmup 步数，是预先设定的超参数，原文中为 4000。图 2-4 展示了不同 Warmup 步数下的学习率随训练步数的变化情况，黑色圆形、方块、三角形的三条曲线分别为设定的 warmup 步数为 2000、4000、8000 时的学习率曲线（d_{model}=1）。

$$lr = \begin{cases} 1, 0 < epoch \leq 5 \\ \frac{1}{2}, 5 < epoch \leq 10 \\ \frac{1}{4}, 10 < epoch \leq 15 \end{cases} \quad (2-27)$$

$$lr = {d_{model}}^{-0.5} \cdot min(steps^{-0.5}, steps \cdot warmup\_step^{-1.5}) \quad (2-28)$$

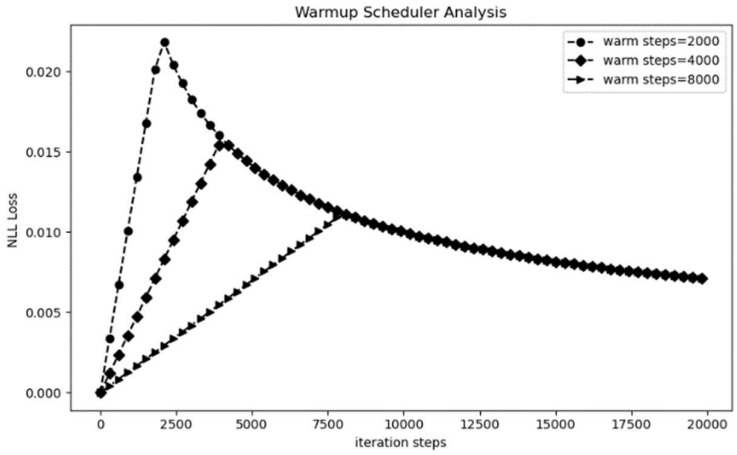

图 2-4 Warmup 策略的学习率

2.2 深度学习自然语言处理

2.2.1 词向量

如在 1.1.2 中讲述的，自然语言中的「词」是高智能人类使用的符号系统，每一个词语背后关联着丰富的、多样的、多变的语义信息。一个苹果（Apple）既可以是一家电子设备公司也可以是一种水果。不仅仅文本本身存在一词多义，而且常常还有引申含义等复杂的语义蕴涵。

最直接的词的编码为独热编码（One-Hot），比如现在有，杭州、上海、宁波、北京四个词，按照独热编码的方式他们被编码为：杭州 [1,0,0,0]，上海 [0,1,0,0]，宁波 [0,0,1,0]，北京 [0,0,0,1]。独热编码非常简单，没有任何计算代价，但同样问题也非常严重，比如词汇库非常大时，独热编码出的向量维度也非常大，最严重

的是独热编码下，词语间的语义关系无法表示，任意两个词的独热编码都是正交的，无法计算相似度。

词嵌入（word2vec[19]）正是为了解决这种问题而产生的，通过神经网络将词从独热编码转换成低维的稠密向量，等同于将词从高维空间投影到低维空间，并尽可能保持原有的位置关系不变。图 2-5 词嵌入示意图展示了三维空间向二维平面的映射。

把单词转成向量的过程也叫词嵌入（embedding）。关于词嵌入向量的尝试包括 CBOW[39]、Skip-Gram[39]、ELMO[40]等。Transformer[4]中提出了独特的可以给非循环神经网络学习的词嵌入方式，具体包含两个部分，位置编码[4]和语义 embedding 向量。基于 Transformer[4]、位置嵌入[4]、warmup[4]等技术，后续的研究 BERT[41]、ALBERT[42]、GPT-2[43]等在多个自然语言任务上取得了重大的突破，超过了当时的最佳方案（State of Art），从此用 BERT、ALBERT 等大型网络在大规模数据集上预训练，以无监督方式学习词向量，然后在特定任务上微调训练(fine-tune)的模式成工业界的通用方式。

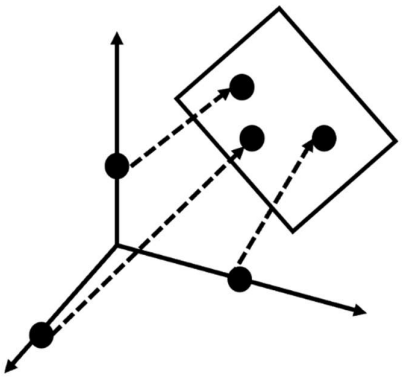

图 2-5 词嵌入示意图

2.2.2 序列到序列模型

2.2.2.1 序列到序列建模

序列到序列，即 Sequence-to-Sequence (seq2seq)方法[6]由 Sutskever, Vinyals, and Le 等人在 2014 年提出并被广泛应用于自然语言处理上。具体地说，序列到序列模型是一种编码器-解码器结构，编码器和解码器可以用各种神经网络来搭建、甚至编码器和解码器使用的模型可各不相同。常见的处理文本序列的模型，都是用循环神经网络搭建的，当然，也可以用 CNN, LSTM 等搭建。图 2-6 是这种结构的说明示意图。

编码器通过循环神经网络(或别的结构)进行语义解析，一个词一个词地将输入文本“欢/迎/你/来/成/都”转换成了 RNN 的隐藏状态，即语义隐向量 C，语义隐向量蕴含了表征这句话含义的信息，相当于从“欢/迎/你/来/成/都”这几个词中提炼出来的大致意思。

解码器用语义隐藏向量初始化自身，然后每一时刻的输入为解码器前一时刻解码的输出 Y_{t-1}，生成对应的序列新的一个词 Y_t。第一个时刻 Y_0 为开始符“SOS”，然后逐步解码下去，直到解码解出终止符“EOS”，标志着解码的结束。

seq2seq 方法利用编解码器的循环神经网络有效的建模了输入序列和输出序列，采集序列的特征，经过训练，高效的学习到了输入序列和输出序列的统计学对应关系，在神经网络翻译、问答、对话系统等方面获得了广泛的应用。

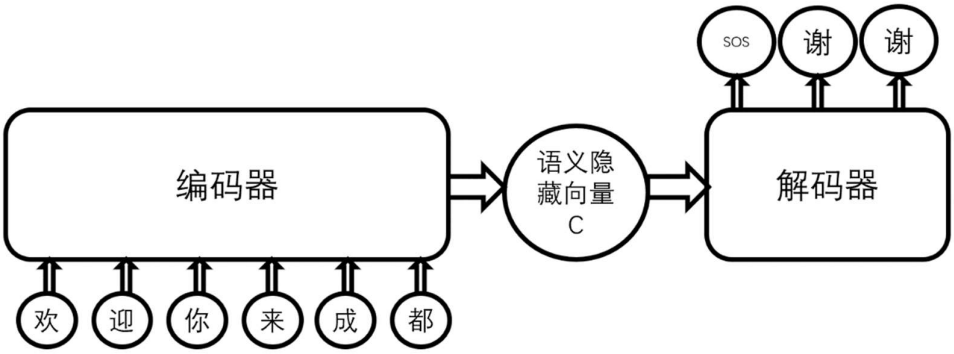

图 2-6 seq2seq 模型示意图

2.2.2.2 注意力机制

简单的序列到序列模型的信息交流模式是编码器编码语义信息（语境），解码器直接使用编码器的最终状态来初始化自身的初始隐藏神经元状态，这也是唯一一次编解码器信息交换。这种方式导致信息传递十分有限，并且解码器伴随不断预测出的新词发生变化，在自身的初始隐藏神经元状态发生多次更新以后，初始化时的语义信息被稀释、损失甚至遗忘。Bahdanau D, Cho K, Bengio Y.提出了注意力机制[12]，来让网络学习如何更好的使用编码器编码语义信息。它在每一个解码时刻，将解码器 RNN 的前一个时刻的隐藏神经元状态向量h_{t-1}与编码器在编码输入序列过程中的每一个时间步的隐藏状态向量h_s链接在一起计算相似度，得出编码输入的每一个单词和当前解码器隐藏状态的相关性，作为当前解码的注意力权重，如图 2-7。

动态注意力相似度的计算有多种方式，在这了介绍 Bahdanau D, Cho K, Bengio Y.[12]所提出的范式和本文最后使用的点乘方式（$scale\ dot\ version$）。首先需要计

算隐藏单元的相似度如公式(2-28)，其次计算相似度的归一化后的权重如公式(2-29)。

$$\text{score}(h_t, h_s) = \begin{cases} h_s \cdot h_t^T, scale\ dot\ version \\ v^T \tanh(w_1 h_t + w_2 h_s), Bahdanau's\ version \end{cases} \quad (2-28)$$

$$\alpha_{ts} = \frac{\exp\left(score(h_t, h_s)\right)}{\sum_{s=1}^{S} \exp\left(score(h_t, h_s)\right)} \quad (2-29)$$

从某种意义上来说，注意力机制是通过当前解码器状态查找相关的编码器状态，可以理解为一种回溯策略，使用当前解码器状态得出当前状态下，如何分配注意力权重给源信息的不同部分。这种注意力机制就像人脑在看一幅画时，会把自己的注意力投放在目光所看到的某个局部地方。对对话系统而言，这种注意力可以让网络学习如何更好的使用编码器编码语义信息，让系统在回答“我喜欢北京，住在这里很方便，就业发展的机会也很多，你喜欢哪座城市呢?”时把注意力放在关键的“你喜欢哪座城市”，而尽量忽视掉无关的信息。

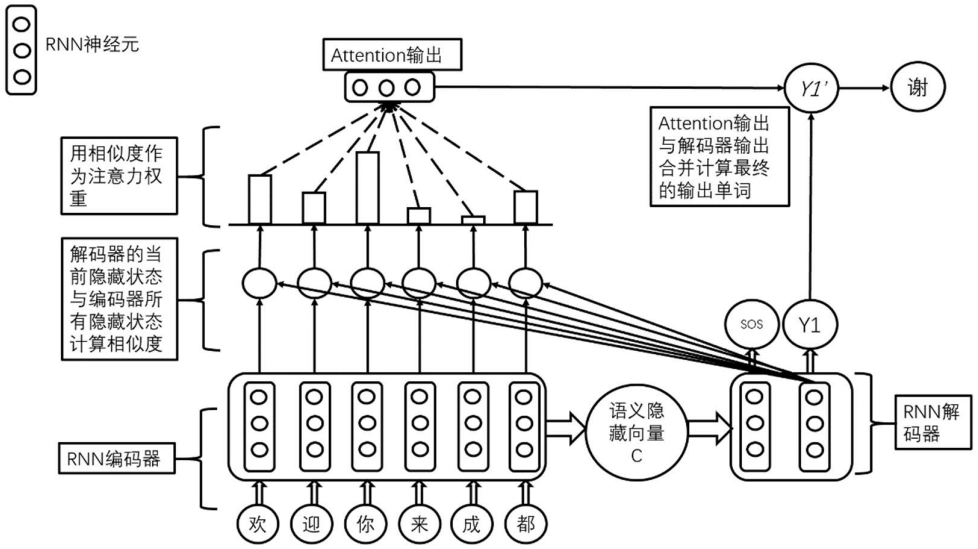

图 2-7 动态注意力机制

2.3 外部知识

人们的日常行为都和人类学习到的知识有关，这些知识可以是客观事实规律也可以只是两个事物间的联系，指导着人的一举一动。人脑内通过大量的神经元间的激活、抑制存储了这些知识，并在我们需要的时候取出来使用。如何类似人脑一般建立数字化的知识库，表征人类的知识供电子设备理解、处理是引领智能计算设备发展的核心问题。目前人类已经花费巨大的人力物力构建了 WordNet[20]，FreeBase[21]等大型知识库。

知识可以是结构化的和非结构化的。非结构化的知识指的是没有固定的组织关系的知识。比如一篇文章就是一些非结构化的知识的集合，文章包含了作者的价值判断、包含了一些客观事实，但这些知识信息都是隐式蕴含在文本中的。结构化的知识库以网络的形式组织知识库中的知识，网络中每个节点代表一个实体（真实世界客观存在的独立个体，如人名、公司、概念等），实体间的关系用网络中的边表示。在每一条边上，分别有两个实体，这样的由（实体 A,关系，实体 B）组成的三元组就是知识的通用表示方式。万维网联盟（W3C）发布的资源描述框架（RDF）[22]技术标准就是以三元组表示知识的。

2.3.1 知识图谱

知识图谱（Knowledge Graph）的概念是 2012 年 Google 提出的，本质上，知识图谱是一种实体对象、关系的连接表示，以 SPO(Subject-Predicate-Object)结构化三元组表示知识。在这里必须指出知识图谱和图的区别，一个知识图谱系统就是由一条条 SPO 三元组知识所组成，这些三原组在理论上可以构成一张大的有向图，比如图 2-8 展示了由（成都，北京，大熊猫，中国，竹子，武侯祠）六个实体的一些三元组构成的语义网络或者叫有向图。但是知识图谱也可以并没有图结构，而仅仅是一条一条零散的 SPO 三元组。比如我们可以从图 2-8 展示的语义网络中抽取出（武侯祠，属于，成都），（成都，属于，中国），（中国，包含，北京），（大熊猫，来自，中国），（大熊猫，来自，成都），（大熊猫，喜欢吃，竹子），（北京，是…的首都，中国）七条知识 SPO 三元组，抽取出来的这些没有图结构的零散的一条条结构化三元组数据也可以被称为知识图谱。这就是图结构的知识图谱和非图结构的知识图谱间的关系。从图结构到零散三元组的过程中，其实还有很多隐含的知识关系存在但不容易被识别出。假如我们只有一条一条零散的 SPO 三元组，反过来试图恢复有向图的空间结构，难度也会随着实体、关系的数量不断上升。

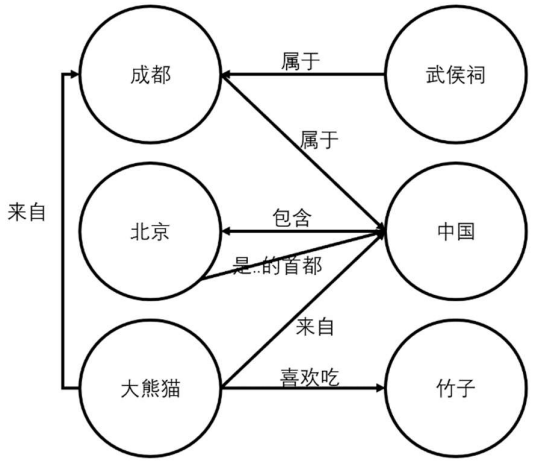

图 2-8 知识图谱示意图

知识图谱的构建是一种工作量庞大的大型的系统工程，需要数据抓取、数据存储、数据挖掘、实体链接等多种处理技术、大量资金的支持。简单来说，构建知识图谱需要经过知识抽取（实体、属性等的识别抽取）、融合（多种知识点相互验证排除错误）等步骤。在深度学习应用火热的当下，也有文献研究基于深度学习的实体识别等。本文并不涉及知识图谱的构建，因此不再赘述。

2.3.2 知识表示

和词汇需要嵌入成词向量一样，知识也需要合适的表示方式才能被计算设备有效的理解和使用。大量的研究人员在探索知识的表示学习，这是一个独立于对话系统、自然语言处理的研究领域。

“表示学习的目标是，通过机器学习将研究对象的语义信息表示为稠密低纬实值向量。以知识库中的实体e和关系r为例，我们将表示学习得到的向量表示为l_e和l_r.在该向量空间中，我们可以通过欧氏距离或余弦距离等方式，计算任意 2 个对象之间的语义相似度”。[44]

对于知识表示学习的主要方法有，结构距离模型、语义匹配能量模型、翻译模型等。距离模型的结构表示（structured embedding, SE[45]）是较早的知识表示技术，将实体投影到 d 维空间，然后在 d 维空间计算两个实体在关系 r 下的语义相关度。能量模型即语义匹配能量模型（semantic matching energy, SME[46]），提出以复杂的矩阵运算关系来刻画实体间的关系，以线性或双线性的评分函数衡量语义匹配能量。

基于语句翻译的知识表示模型的最初的启发来源于 Mikolov 等人于 2013 年提出的 word2vec[19]词嵌入表示学习模型。word2vec[19]中指出词嵌入向量能得到不同

单词之间的类比关系，例如「公牛-男性=母牛-女性」。虽然我们不能得到每个词确切的表示，每次重新训练后，表示某个词的向量都是不一样的，但这种分布式的表示捕捉到了词之间的相互关联。受此启发，Bordes A, Usunier N, Garcia-Duran A 等人提出了 TransE[23]模型，将知识库库中的实体看作某种语义单元，关系 r 看作实体间的某种语义关联运算。如图 2-9 所示，对于每个三元组(h, r, t),首先将三个元素映射成对应的向量用关系l_h,l_r, l_t, TransE 认为l_r可以看成l_h到l_t的翻译，希望l_h,l_r, l_t满足：$l_h + l_r \approx l_t$。所有 TransE 定义了如下的损失函数：

$$f_r(h,t) = |l_h + l_r - l_t|_{L_1/L_2} \qquad (2-30)$$

即$l_h + l_r$和l_t的L_1或L_2距离。

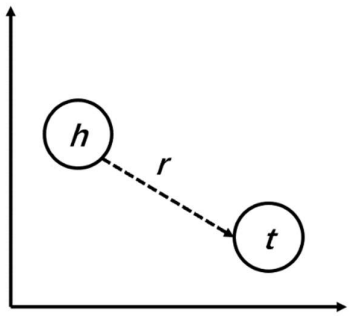

图 2-9 TransE 模型的假设示意图

TransE 的具体训练算法如图 2-10 所示。本次论文项目中使用的 TransE 代码来自 Yankai Lin, Zhiyuan Liu, Maosong Sun 等[47]提供的开源代码。

input: Training set $S = \{(\text{sub,rel,obj})\}$, margin γ, learning rate λ
initialize $\mathbf{r} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each rel
    $\mathbf{r} \leftarrow \ell / \|\ell\|$ for each ℓ
    $\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{j}}, \frac{6}{\sqrt{k}})$ for each entity ent(sub or obj)
loop
    $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity ent
    $S_{batch} \leftarrow sample(S, b)$ //sample minibatch of size b
    $T_{batch} \leftarrow \emptyset$ //initialize set of pairs
    for $(\text{sub,rel,obj}) \in S_{batch}$ **do**
        $(\text{sub}',\text{rel},\text{obj}') \leftarrow \text{sample}(S'(\text{sub,rel,obj}))$ //sample negative triplet
        $T_{batch} \leftarrow T_{batch} \cup \{((\text{sub,rel,obj}), (\text{sub}',\text{rel},\text{obj}'))\}$
    end for
    Update embeddings w.r.t. $\sum_{T_{batch}} \nabla\left[\gamma + \|\mathbf{s} + \mathbf{r} - \mathbf{o}\|_2^2 - \|\mathbf{s}' + \mathbf{r} - \mathbf{o}'\|_2^2\right]_+$
end loop

图 2-10 TransE 训练算法。图片来自 Advances in neural information processing systems[23]

2.4 本章小结

本章介绍了本论文后续将会涉及的各种深度学习自然语言处理的基础理论知识。第一节讲述了神经网络的内容，包括常用的循环神经网络结构、序列到序列建模、注意力机制、梯度下降和 Adam 优化方法、warmup 学习策略。第二节讲述了知识图谱和知识表示等内容，重点描述了 TransE 知识表示的训练算法。

第三章 知识驱动对话系统的研究与设计

3.1 系统的总体描述

3.1.1 数据集与任务描述

本课题研究的是知识驱动的对话系统。该任务问题的抽象描述为：构建系统 L，L 接受对话历史输入 H(History)和知识库 KGL(Knowledge Graph Library),并生成回答 R(Response),R⊙C(⊙表示满足，C 表示语境)。这个任务记为问题 A。

我们使用百度的 DuConv 数据集[48]，该数据集合的一条记录包含对话(Conversation),对话路径(Goal),对话相关的知识(Knowledge)。表 3-1 DuConv 数据格式表展示了一条完整的 DuConv 训练数据集的记录样本。其中对话路径(goal)表示对话涉及的两个话题 A，B 和它们之间的关系。对话（conversation）包含 4 到 8 轮的多次对话。知识(knowledge)包含于话题相关的若干条知识 SPO 三元组。

表 3-1 DuConv 数据格式表

Goal	["START","星球大战：最后的绝地武士","莱恩·约翰逊"], ["星球大战: 最后的绝地武士","导演","莱恩·约翰逊"]
Knowledge	Conversation
"星球大战 ： 最后 的 绝地武士", "时光网评分", "7.4"	"你喜欢系列电影吗？", "喜欢，你有推荐吗？", "昨天看了一部系列电影，星球大战：最后的绝地武士。", "谁是主演 呢？", "约翰·波耶加。", "那我去看看。", "这部电影的导演是莱恩·约翰逊，非常有才华的人哦！", "对他不是很了解。", "网友们评论说他能说故事又能写故事，是一个大有前途的导演呢！"
"星球大战: 最后的绝地武士","口碑","口碑一般"	
"星球大战: 最后 的 绝地武士", "领域","电影"	
"星球大战：最后的绝地武士","导演","莱恩·约翰逊"	
"莱恩·约翰逊","评论","能说故事又能写故事，大有前途的导演。"	
"莱恩·约翰逊", "描述","知名导演"	
"莱恩·约翰逊", "职业","导演"	
"莱恩·约翰逊","领域","明星"	
"星球大战: 最后的绝地武士""主演","约翰·波耶加"	

DuConv 数据集共计包含约三万组对话，三十万句对话，具体的统计结果如表 3-2。

表 3-2 DuConv 数据集的统计结果表

统计项目	统计结果
对话数	29858
训练集对话数	19858
测试集样本数	5000
验证集对话数	2000
语句数	270399
每组多轮对话的平均语句数	9.1
每句话平均单词数	10.6
每组多轮对话的平均关联知识数	17.1

DuConv 数据集提出了一种新的任务挑战：主动发起对话和组织对话。这种新的任务的抽象描述为：构建系统 L，L 接受对话历史输入 H(History)和知识库 KGL(Knowledge Graph Library)以及对话目标路径(goal),并生成回答 R(Response), R⊙C, R⊙G (⊙表示满足，C 语境, G 表示对话目标路径)，这个任务记为问题 B。

主动发起对话和组织对话这一目标要求系统在输入的对话历史为空时，根据对话路径(goal)生成第一句文本语言，接着根据对话路径(goal)中指出的两个话题 A，B，和它们之间的关系，以及用户的输入信息、知识，不断生成新的语言，并且最终实现引导对话从一开始谈论 A 到谈论 B。表 3- 7 多轮对话实例就展示了对话系统如何从“海豚湾”电影切换到“伊莎贝尔·卢卡斯”演员，主动发起对话并实现对话的话题切换目标。

本文在 DuConv 数据集实现的模型也紧随 DuConv 数据集提出了的任务挑战，即具体的任务描述为问题 B。

3.1.2 系统的总体框架

图 3-1 展示了系统的实现总体过程的框架。在模型训练阶段，包括数据预处理、知识和词表示、模型训练三个部分。训练结束后再部署为线下系统。接下来本章的各个部分将详细描述总体框架图中的各个步骤。

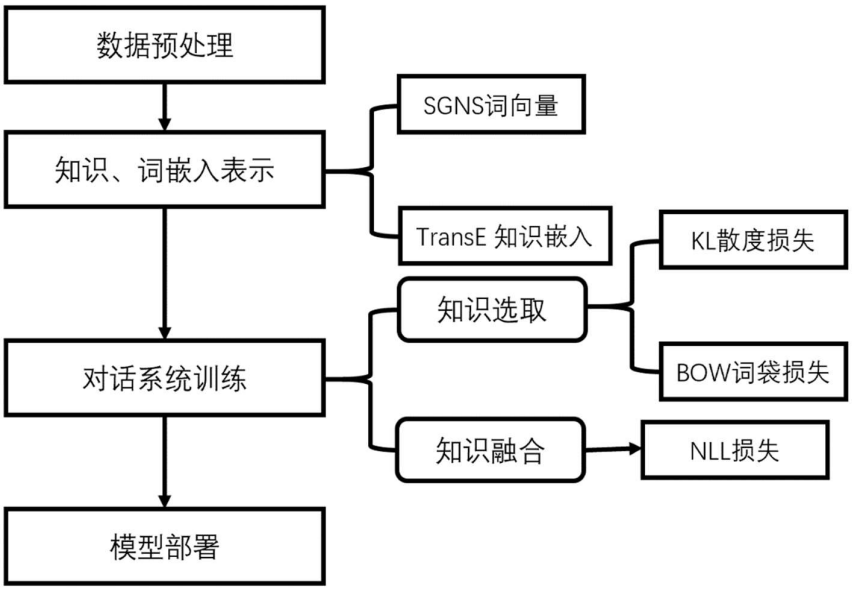
数据预处理
知识、词嵌入表示
SGNS词向量
TransE 知识嵌入
对话系统训练
知识选取
KL散度损失
BOW词袋损失
知识融合
NLL损失
模型部署

图 3-1 总体框架图

3.2 知识和词的表示

表 3-3 对话拆解示意表

拆解前	组 1	组 2	组 3
"你喜欢系列电影吗？", "喜欢，你有推荐吗？", "昨天看了一部系列电影 ， 星球大战：最后的绝地武士。", "谁是主演呢 ？", "约翰·波耶加。", "那我去看看。", "这部电影的导演是莱恩·约翰逊，非常有才华的人哦！", "对他不是很了解。", "网友们评论说他能说故事又能写故事，是一个大有前途的导演呢！",	空	"你喜欢系列电影吗？", "喜欢，你有推荐吗？"	"你喜欢系列电影吗？", "喜欢，你有推荐吗？", "昨天看一部系列电影，星球大战：最后的绝地武士。", "谁是主演呢？",
	"你喜欢系列电影吗？"	"昨天看了一部系列电影，星球大战：最后的绝地武士。"	"约翰·波耶加。"

在词的表示方面，数据预处理阶段需要进行：

1.将一组对话（conversation）拆解为多组的“问答关系”即拆解为多组(history,response)的二元组，按照 DuConv 数据集挑战的要求，为构造主动式的对话机器人，拆解的方法为：对话历史 history 为对话的开头到偶数句（包括第零句，即对话历史为空句），response 为对应的奇数句。表 3-3 展示了拆解前后的数据，篇幅有限，仅展示了拆解后的前三组，每一组表格的上半部分为对话历史，下半部分为回复。

2.为了对词语进行嵌入表示，在数据预处理阶段，需要对拆解以后的对话历史和回答进行长度筛选。

3.筛选以后，在整个数据集（训练集、测试集、验证集）上建立词的统计计数，

同时根据词的频率，过滤掉低频词。给每个词分配一个索引计数(token)，即用一个整数代表一个词。

4.使用新建立的词库查找 SGNS_wikipidia 预训练的词向量[49]中对应的词的嵌入值，构造词嵌入矩阵。SGNS_wikipidia 是使用 Skip-Gram[39]算法在中文维基百科上预训练的中文词向量。在预训练词向量中没有查到的词的嵌入值则采用随机初始化。

为了实现知识的表示，数据预处理阶段需要进行：1.将每组单条记录的知识抽取后合并成三元组结构化的知识库。2.对知识库里的实体和关系进行统计计数，同时根据实体文本的长度，过滤掉文本过长的实体、关系。3.给每个实体和关系分配一个索引计数，即用一个整数代表一个实体或关系。4.使用 TransE 算法对知识库进行嵌入表示，得到每个实体或关系的索引计数(token)和对应的嵌入表示(embedding)。5.最终的知识表示为每个实体或关系的文本单词的词嵌入表示和该实体或关系单词关联的实体的 TransE 嵌入表示的和。

如公式(3-1),其中$E(wi)$代表该实体或关系的第 i 个词的词嵌入向量，$Etrans$表示该实体或关系自己的 TransE 嵌入向量。

$$E(kg) = E(w1) + Etrans(kg), E(w2) + Etrans(kg), \ldots. \quad (3-1)$$

3.3 知识选取

3.3.1 先验、后验分布

参照 Lian R, Xie M, Wang F 等的工作[31]，本文对于知识的选取是基于 BOW 词袋损失和先验和后验分布的。现在规定一下符号，下文将用 k 表示知识，kc 表示经过选择后的知识，x 表示对话历史，y 表示标准回复也被称为目标句子，Target, ground truth。

对于知识的选取，使用 BOW 词袋损失使得知识选取的后验分布 p(k|x,y)可以选择合适的知识。BOW 词袋损失的本质是用目标句子和生成的句子中的每一个词算一次交叉熵损失。该知识选择模型[31]的核心在于在网络中用隐向量表示出知识选取的后验分布 p(k|x,y)和先验分布 p(k|x)，然后使用 KL 散度衡量两个概率分布的分布距离，网络在学习优化 KL 散度时等同于在学习用对话历史估计知识的选择。KL 散度损失、$p(k_i|x)$、$p(k_i|x,y)$的计算公式如下：

$$p(k_i|x,y) = \frac{\exp\left(k_i \cdot \mathrm{MLP}([x:y])\right)}{\sum_{j=1}^{N} \exp\left(k_j \cdot \mathrm{MLp}([x:y])\right)} \quad (3-2)$$

$$p(k_i|x) = \frac{\exp(k_i \cdot x)}{\sum_{j=1}^{N} \exp(k_j \cdot x)} \quad (3-3)$$

$$L_{KL}(\theta) = \frac{1}{N}\sum_{1}^{N} p(k_i|x,y) log \frac{p(k_i|x,y)}{p(k_i|x)} \quad (3-4)$$

在网络训练时，具体的知识选择流程为：首先通过对话历史隐藏向量和知识隐藏向量进行点乘注意力计算（$x \cdot kg^T$）得到先验的知识选择概率分布 p(k|x)，然后使用多层感知机合并对话历史隐藏向量和目标句子隐藏向量，合并后送入后验知识选择模块，用点乘注意力计算得到后验的知识选择概率分布 p(k|x,y)，使用 BOW 损失使得后验的知识选择概率分布 p(k|x,y)能选择适合目标文本生成的知识，使用 KL 散度损失让先验概率分布 p(k|x)逼近后验概率分布 p(k|x,y)实现对知识点的选取。由于在测试时没有标准回复，测试时，网络将不计算后验分布 p(k|x,y)、BOW 词袋损失、KL 散度损失，示意图如图 3-2。

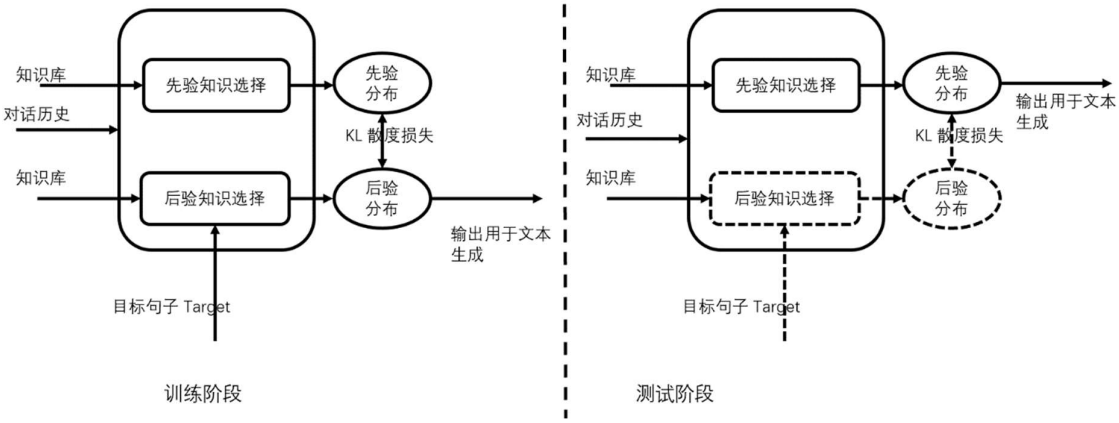

图 3-2 无 Teacher Forcing 时的训练测试阶段的知识选取示意图

3.3.2 老师引导

如图 3-2 在知识选择模型[31]中，网络在训练时使用后验知识概率用于文本生成，在测试时使用先验知识概率用于文本生成，虽然通过 KL 散度使得两个分布相逼近，但毕竟两个分布还是有本质区别的。借鉴 RNN 中的 Teacher Forcing 的思想，本文提出以 Teacher Forcing 的方式决定训练时使用后验知识 p(k|x,y)还是先验知识概率 p(k|x)。

在 RNN 的训练迭代过程中，解码器每一步的输入为（编码器输出的语义隐藏向量，解码器上一步的输出）。由于训练早期的 RNN 参数来自随机初始化，预测能力非常弱，几乎不能给出好的生成结果，这导致解码器上一步的输出就极可能

是错误的，这样当前步骤解码器的输入就极可能是错误的。可见，在 RNN 训练中，前期的误差将会不断积累，最终导致 RNN 的学习效率非常差。最初“老师引导”（Teacher Forcing）的提出就是为了解决误差累积这个问题。Teacher Forcing 具体的实现是让解码器每一步的输入可以为（编码器输出的语义隐藏向量，解码器上一步的输出）或（编码器输出的语义隐藏向量，标准答案的上一个词）。并且在训练过程中，逐步提高选用（编码器输出的语义隐藏向量，解码器上一步的输出）的概率，最终摆脱标准回答的指导。

不同于 RNN 中以 Teacher Forcing 避免误差累积，在本文的深度模型训练阶段，设定选取后验知识的概率 teacher forcing ratio(tfr)，每一次迭代，按照$\mathrm{p} = tfr$的概率选择后验知识分布，以$\mathrm{p} = 1 - tfr$的概率选择先验知识分布用于下一步的文本生成，这是为了让网络在训练时和测试的行为更加接近。在原方案[31]中最终的 NLL 损失只会训练后验知识的选取而不直接指导先验知识的选取，而现在，用最终的 NLL 损失和散度损失一起来指导先验知识的选取。示意图如图 3-3。本文设定的tfr的计算公式为公式(3-5)。

$$tfr = \frac{pre}{epoch} \qquad (3-5)$$

其中pre指的是 3.4 节中将介绍的两阶段训练中第一阶段的周期数，是由人为设定的一个超参数。Epoch是指当前的周期数。随着训练周期的增加，选用 teacher forcing 的概率会越来越小，模型在测试时的表现将会和训练时更加接近。

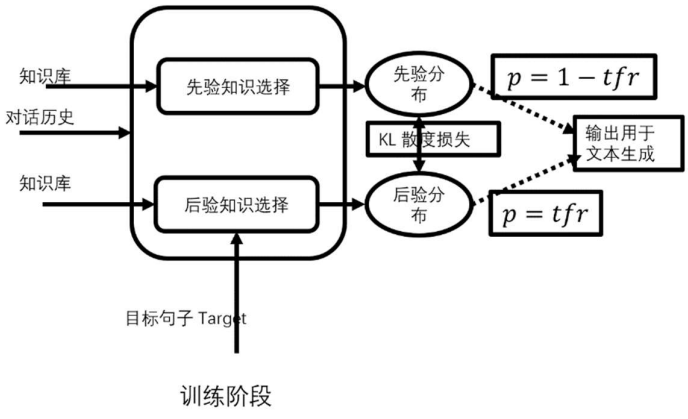

图 3-3 Teacher Forcing 下的知识选取示意图

3.4 对话生成

如图 3-4, 本论文的神经网络包含四个核心的部分，分别是：

嵌入表示层，使用预先初始化的 SGNS 词嵌入矩阵，和 TransE 知识嵌入算法

得到的知识嵌入矩阵表示知识和词语。

编码层，包含两个单层的双向动态注意力的门控循环神经网络（Bi-direction GRU），双向就是指输入文本会被正向和逆向的交给循环神经网络两次。

知识管理层，通过对话历史隐藏向量和知识隐藏向量进行点乘注意力计算（$x \cdot kg^T$）得到先验的知识选择概率分布 p(k|x)，然后使用多层感知机合并对话历史隐藏向量和目标句子隐藏向量，合并后送入后验知识选择模块，用点乘注意力计算得到后验的知识选择概率分布 p(k|x,y)，使用 BOW 损失使得后验的知识选择概率分布 p(k|x,y)能选择适合目标文本生成的知识，使用 KL 散度损失让先验概率分布 p(k|x)逼近后验概率分布 p(k|x,y)实现对知识点选取。

解码层，利用两个单向的门控循环神经网络分别处理选择后的知识和对话历史，各自输出一个词表的概率分布隐藏向量，再经过四层的 MLP 得到两个隐藏向量的融和表示，然后将融合表示的隐藏向量用全连接网络映射成一个词在单词表上的概率分布，该分布中概率最大的那个词就成为新生成的词语。生成的单词表上的概率分布和目标句子对应的词计算 NLL 损失（Negative Likelihood Loss），指导网络生成的单词和目标句子一致。

任意两个概率分布P、Q的 NLL 损失（Negative Likelihood Loss）公式如下：

$$\begin{aligned}\text{NLL Loss} &= E_{x\sim p}\big(-logQ(x)\big) \\ &= E_{x\sim p}\big(logP(x) - logQ(x)\big) - E_{x\sim p}(\log P(x)) \quad (3-6)\end{aligned}$$

本质上 NLL 损失计算的是两个分布的交叉熵$E_{x\sim p}\big(-logQ(x)\big)$，但是因为标准句子的每一个单词的概率分布$P(x)$都呈现独热型，因此标准句子的每一个单词在词表上的概率分布的香农熵$E_{x\sim p}(\log\ P(x))$总是为零，此时预测单词分布$Q(x)$和标准句子单词分布$P(x)$的交叉熵$E_{x\sim p}\big(-logQ(x)\big)$等同于$P$、$Q$的 KL 散度距离$E_{x\sim p}\big(logP(x) - logQ(x)\big)$。优化交叉熵和词袋损失一样，等同于优化目标词分布$P$、生成词分布$Q$的分布距离，迫使预测词语的概率分布逼近标准句子的单词概率分布。

本文核心的网络结构如图 3-4 所示。参照知识选择模型[31]，网络的训练分为两个阶段，阶段一只训练词袋损失使得后验分布能选取合适的知识，阶段二同时训练三个损失函数。我们还在训练过程中使用了 warmup 策略，设定好 warmup step 为 4000。

训练的阶段一：

1.输入对话历史和对话目标合并成的语境作为 X，输入相关知识作为 KG，输入目标回答作为 Y。X 经过预训练词向量嵌入表示、GRU 语境编码器编码为隐藏向量x, KG 经过知识嵌入、GRU 知识编码器编码为隐藏向量$\mathrm{k_i}$，Y 经过预训练词向量嵌入、GRU 知识编码器编码为隐藏向量y。

2.隐藏向量x和隐藏向量y经过四层 MLP 合并为 z 隐藏向量,然后 z 与隐藏向量k_i进行公式(3-2)描述的点乘注意力计算，得到后验分布 p(k|x,y)。隐藏向量x和与隐藏向量k_i进行公式(3-1)描述的点乘注意力计算，得到先验分布 p(k|x)。

3.p(k|x,y)乘以k_i后得到 kc,经过 MLP 一次性输出目标句子长度个预测单词，并计算输出每一个词与目标句子每一个词的交叉熵损失，作为词袋损失，如公式(3-6)。

4.反向传播词袋损失，然后重新进入步骤 1。

训练的阶段二：

5.完成 1、2、3 步以后，计算先验分布 p(k|x)与后验分布 p(k|x,y)的 KL 散度损失。然后根据 tfr 概率选择先验分布 p(k|x)或后验分布 p(k|x,y)乘以知识隐藏向量k_i得到 kc,然后输入到解码器的知识解码部分。

6.语境编码器每一个编码输入的隐藏状态输入到解码器以计算动态注意力，解码器的知识解码部分利用语境编码器的隐藏状态、kc、Y_{t-1}预测单词。语境解码部分利用语境编码器的隐藏状态、x、Y_{t-1}预测单词。两个模块的预测结果经过一个 MLP 合并。输出的词与Y_t计算 NLL 损失。

7.反向传播词袋损失、KL 散度损失、NLL 损失，重新进入步骤 1。

测试、验证阶段：

1.输入对话历史和目标合并成的语境作为 X，输入相关知识作为 KG，输入目标回答作为 Y。X 经过预训练词向量嵌入表示、GRU 语境编码器编码为隐藏向量x, KG 经过知识嵌入、GRU 知识编码器编码为隐藏向量k_i。

2.隐藏向量x和与隐藏向量k_i进行公式(3-1)描述的点乘注意力计算，得到先验分布 p(k|x)。

3.用先验分布 p(k|x)乘以知识隐藏向量得到 kc,输入到解码器的知识解码部分。

4.语境编码器每一个编码输入的隐藏状态输入到解码器以计算动态注意力，解码器的知识解码部分利用语境编码器的隐藏状态、kc、T_{t-1}预测单词。同样的，语境解码部分利用语境编码器的隐藏状态、x、T_{t-1}预测单词。两个模块的预测经过一个 MLP 合并，最终输出T_t。T_{t-1}代表解码器上一次的输出单词，在第一步时为“SOS”开始符。

5.重新进入步骤 1，生成下一个单词，直至生成的单词为“EOS”结束符或达到人为设定的最大长度。

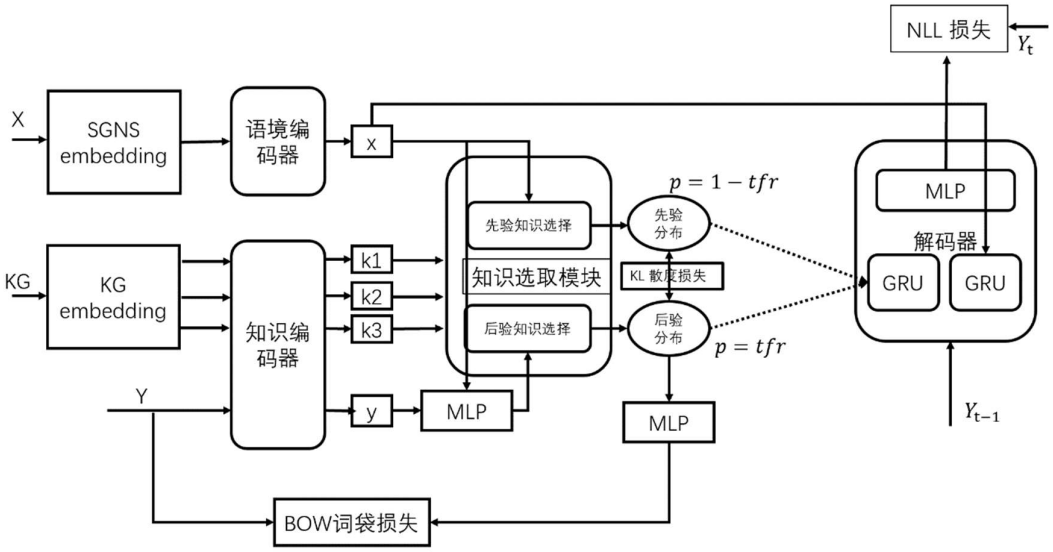

图 3-4 网络模型结构图

3.5 性能对比

3.5.1 评价指标介绍

3.5.1.1 F1 指标

在介绍 F1 指标之前，必须介绍机器学习分类问题中四种重要分类结果。

真阳性 TP（True Positive）：预测为正，实际也为正。

假阳性 FP（False Positive）：预测为正，实际为负。

假阴性 FN（False Negative）：预测与负、实际为正。

真阴性 TN（True Negative）：预测为负、实际也为负。

举个例子，假如某人去医院看病，则 TP 表示医生预测他有病，事实上他真的有疾病；FP 表示医生预测他有病，事实上他没有疾病；FN 表示医生预测他没有病，事实上他真的有疾病；TN 表示医生预测他没有病，事实上他真的没有疾病。

本文使用的 F1 指标与 TP、FP、FN、TN 紧密相关，计算公式如下：

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \qquad (3-7)$$

F1 指标将作为评价对话系统最关键的性能评价指标。

3.5.1.2 BELU 指标

BELU[50]最初的应用是为了评价神经网络翻译的结果好坏，可以衡量翻译输出句子和一个或多个翻译参考答案的综合近似程度。BELU 的计分假设为：如果输出

句子中的任意一个单词属于任何一个参考句，就给它打 1 分，如果没有出现，打 0 分。为了使计数标准化，我们可以将输出译文中单词的得分总数除以输出句子的单词总数，使分数始终介于 0 和 1 之间。在翻译中考虑到语言顺序、单个词多次重复等问题，BELU 的计分假设修改为：如果输出句子中的任意 N 个连续的单词(n-gram)组成的词组属于任何一个参考句，就给它打 1 分，如果没有出现，打 0 分。求出来的标准化得分记作 P_n，P_1，P_2，P_3，P_4，…… P_n,的平均值乘以 BP 就是最终的 BELU 指标。N 可以因为任务而不同，选定 N 后的最终得到指标被称为 BELU（N），在本次论文中，选用的是 BELU1，BELU2，BELU4。BP 则是为了避免译文过短额外引入的惩罚项。值得注意的是 BELU 并不能衡量语义，要从语义上衡量输出结果，机器指标是难以做到的，目前更多的还是依赖人工评判。

他们的具体计算公式如下,其中c和r表示生成的句子的长度和参考句子的长度。

$$BELU1 = BP * P_1 \qquad (3-8)$$

$$BELU2 = BP * avg(P_1, P_2) \qquad (3-9)$$

$$BELU4 = BP * avg(P_1, P_2, P_3, P_4) \qquad (3-10)$$

$$BP = \begin{cases} 1, c > r \\ e^{1-c/r}, c \le r \end{cases} \qquad (3-11)$$

3.5.1.3 DSTINCT 指标

DSTINCT 和 BELU 指标的形式非常类似，不同的是 DSTINCT 计算的是回复的多样性，他不需要和参考句进行比对，而是对生成的句子中的任意一个单词如果是独特的新单词，就给它打 1 分，如果是已经出现过的单词，打 0 分。举个简单的例子，句子“你您你您好”在此时获得的得分为三，因为有三个独特的词语“你”,“您”,“好”。

同样 DSTINCT 也有 DSTINCT1(n-gram=1)，DSTINCT2 (n-gram=2)等形式。比如 DSTINCT2 计算的就是生成的句子中每出现一个由连续两个单词组成的词组，如果是独特的新词组，就给它打 1 分，如果是已经出现过的单词，打 0 分。最后标准化到 0 到 1。

本文选用 DSTINCT1，DSTINCT2 两个指标衡量回答的多样性。

3.5.1.4 人类裁判

自动评价指标不能作为性能判定的唯一标准，因为这些自动化指标都无法像人类一样理解语义。在这里邀请了三位人类裁判对测试集合的 100 组对话进行打分，每一个模型生成 100 组回复，人类裁判如果认为回复符合语境（语意连贯）

或富含信息（句中提及输入的知识）则会给对应模型的成功回复次数、成功包含知识次数加一。每个模型最后会有两个得分，即成功回复次数、成功包含知识次数。

3.5.2 对比模型介绍

为了验证本文提出的模型的有效性，现在引入了三个相似的端到端神经网络模型进行性能对比实验：其一，为了验证加入外部知识能够更好指导对话的生成，选择了基于 GRU 的无知识 Seq2seq[6]。其二，为了和现有其它的基于知识驱动的闲聊系统进行比较，本文选择了 knowledge-grounded[29]模型作为第二个对比模型。由于 knowledge-grounded 模型引入的外部知识是无结构的纯文本知识，因此需要对 knowledge-grounded 中的事实编码器进行适应性修改，使之能够处理结构化的知识。具体的数据结构修改为：将原始模型中输入的纯文本知识替换为对话历史相关的 SPO 知识三元组，例如对话历史提到电影“银河补习班”，该实体出现在了 DuConv 的知识库，则将相关的导演、演员等实体都一并从知识库中检出，构建成一条条知识三元组，用和 DuConv 一条记录样本的知识数据域一样的格式保存。然后经词嵌入转换为低维连续向量送入原始模型中的事实编码器中，编码后和对话历史的语义隐向量一起做解码器的输入。 ·

第三个对比模型是 Lian R, Xie M, Wang F 等提出的模型[31]，也是在 DuConv 数据集上的 State of Art,记作 KGS(knowledge select)。本论文的模型记作 TFKS(teacher forcing knowledge select)。

对比的四个模型全部使用在数据预处理阶段提到的主动式对话的数据格式进行训练，批大小(batch size)为 20，以 GRU 作为编码器和解码器，隐藏层的层数为 1，隐藏单元数（hidden size）为 800，词和知识的嵌入维度 300，随机失活（dropout）的比率为 0.1，其它参数大小也都设置成同样的值，TransE 代码除了嵌入维度为 300，其余参数均与原论文保持一致。三个对比模型训练过程中使用 Adam 优化算法以及默认的 Adam 超参数配置。训练的默认配置为：学习速率初始化为 0.01，当模型在验证集上的预测准确率出现增长的情况时，将学习速率缩小到原来的十分之一，然后继续训练直到学习速率小于 0.000001 停止训练。本论文模型的训练则使用的时 warmup 策略控制学习速率。在剥离实验中，去除 warmup 策略时本论文模型的训练学习率变化则和其他三个模型一致。

3.5.3 性能对比

表 3-4 中列出了四个模型在测试集上的多个自动机器评价指标，在表 3-5 列出

了 100 组对话中人类裁判的打分情况，表中每个单元格列出的数据为 100 组对话中三位裁判认为该模型的回答“语句上连贯的次数/信息含量饱满的次数”。另在表 3-6 中列出了 TFKS 的实例结果展示，为了方便展示，相关的知识被省略没有列在表中，关于知识的格式可以参考表 3-1。表 3- 7 多轮对话实例展示了 TFKS 的组织的一次多轮对话交互。

首先可以看到，引入知识带来的性能提升。引入知识的模型普遍都比没有引入知识的模型 Seq2seq 表现要好，我认为外部知识带来了额外的信息，限定了模型生成对话的范围，使网络更有目标性和方向性，这对于主动式对话来说非常关键，毕竟没有知识的 Seq2seq 在对话历史为空的时候仅仅依靠空对话历史（空句）和对话路径中的少量信息生成对话，仅仅知道会谈论什么然后发起对话，这样的行为及其具有一定的随机性，网络难以学习。

在使用知识的三个模型中，KGS 和 TFKS 在多个指标中比较接近, 均超过了 knowledge-grounded，这和 KGS 和 TFKS 的网络模型更加复杂，包含更多参数密切相关。大型的网络可以拟合更加复杂的函数，这使得 KGS 和 TFKS 在学习过程中能记忆更多的信息。

KGS 甚至在 BELU2、BELU4 指标上超过了 TFKS 的，直觉上分析认为是和 TFKS 更加复杂的网络逻辑有关，Teacher Forcing、TransE 技术使得 TFKS 在生成时做出了更多关于知识的选择，选择的知识文本毕竟不同于目标句子文本，这使得网络使用的信息包含了知识串与目标句子的偏差，进而导致监督学习的结果 BELU1，BELU2 指标有一定程度的下降，但正因为选用的知识，使得 TFKS 生成的回答独特性高，重复率低，DSTINCT1，DSTINCT2 都超过了 KGS。

虽然受限于人力有限，仅有三位裁判在 100 组数据上打分，在人类裁判打分的结果上 TFKS 也体现了优于别的模型的得分，这也是本论文模型有效性的强力证明。

从表 3-6、表 3- 7 多轮对话实例中从不难看出，对话系统可以学习到了对话的模式（pattern）特点，并可以组织、发起对话。不足之处在于，受限于数据中对话形式较为单调，用其作为监督数据训练的模型，其生成的对话形式还是比较单调的，目的性极强的在仿照 DuConv 数据集合进行对话组织。

表 3-4 性能对比表

模型	F1	BELU1	BELU2	BELU4	DSTINCT1	DSTINCT2
Seq2seq	31.7823	0.2871	0.1964	0.0621	0.0649	0.1356
KGS	48.5346	0.4436	0.3072	0.1143	0.1078	0.231
knowledge-grounded	36.85	0.3235	0.1628	0.0641	0.0545	0.1312
TFKS（ours）	49.0335	0.4474	0.2859	0.0991	0.1328	0.2994

表 3-5 人类裁判评分表

模型	裁判 1	裁判 2	裁判 3	平均
Seq2seq	23/6	25/5	27/5	28/5.333
KGS	40/15	46/18	43/20	43/17.667
knowledge-grounded	33/15	39/13	38/14	36.667/14
TFKS（ours）	45/19	44/17	42/22	43.667/19.333

表 3-6 生成实例表

对话历史(//分句)	标准回答	对话路径	机器回复
空	剧情类的电影喜欢吗？	START-> 安伯森情史->阿方索•阿雷奥 [安伯森情史,导演,阿方索·阿雷奥]	我最近看了一部电影，你喜欢不？

你好像很无聊没什么 事做唉。// 是啊无聊到发慌。// 看电影磨时间啰。// 那你有什么推荐吗？	开麦拉狂想曲看过吗？	START-> 开 麦 拉 狂 想 曲 -> 德 蒙 特 • 莫罗尼 [德蒙特 • 莫罗尼, 代表作,开麦拉狂想曲]	我看了一部电影，开麦拉狂想曲，很好看。
平时看不看话剧？// 恩，还可以。	辛柏青的代表作人活一张脸，还不错。	START-> 辛 柏 青 ->宋笠娜 [辛柏青, 代表作, 人活一张脸],[宋笠娜, 代表作, 人活一张脸]	我最喜欢看辛柏青演的角色，我都喜欢的。

表 3- 7 多轮对话实例

对话路路径	START->海豚湾->伊莎贝尔 • 卢卡斯
知识	对话
"伊莎贝尔 • 卢卡斯","获奖","变形金刚 2_提名_(2010 第 19 届)_MTV 电影奖_MTV 电影奖-最惊人场面"	robot:你喜看国外的电影吗 ？ human:还行
"海豚湾","导演","路易 • 西霍尤斯"	robot:给你推荐一部《海豚湾》，评分 9.1 的历史纪录片，你有看过吗？
"海豚湾","时光网短评","残酷的人类"	human:没有
"海豚湾","时光网评分","9.1"	robot:我很喜欢这部电影里面的 一个演员，伊莎贝尔 • 卢卡斯，她和你 是 老乡呢。
"海豚湾","类型","犯罪"	
"海豚湾","领域","电影"	
"伊莎贝尔 • 卢卡斯","评论","漂亮的霸天虎"	human:我不是美国的

"伊莎贝尔·卢卡斯","出生日期","1985-1-29" "伊莎贝尔·卢卡斯","领域","明星" "海豚湾","国家","美国" "海豚湾", "时光网短评","震撼、愤怒。"	robot:这部电影的主演是一位非常 优秀的人，她还获得了人民选择奖 最受欢迎女演员。 human:嗯，不错，有时间看看 robot:非常有才华的演员，在奥斯卡金像奖中提名了呢

3.5.4 剥离实验

在性能对比实验中，本文已经分析了三个知识驱动人机对话模型和 seq2seq 的差距，证明了引入知识带来的收益和本文模型的有效性。为了更加有力的证明本文在 KGS 上的改进（warmup、Teacher Forcing、TransE Embedding）是有效的，现在进行剥离实验。分别去除 warmup、Teacher Forcing、TransE Embedding 中一项技术，记为 No Warmup, No Teacher Forcing, No TransE Embedding,什么也没有去除的设定记为 Origin。

在去除 warmup 的设定中，学习率的调节为：学习速率初始化为 0.01，当模型的验证集损失函数出现增大的情况时，将学习速率缩小到原来的十分之一，然后继续训练直到学习速率小于 0.000001 停止训练。在去除 TransE Embedding 的设定中，知识三元组里的文本只使用 SGNS 词向量进行嵌入。在去除 Teacher Forcing 的设定中，训练时使用后验知识分布，测试时使用先验知识分布用于对话生成。因为训练分为两个阶段，第一个阶段每个设定都是预先训练 2 个 epoch，但为了方便比较，第一个阶段的训练周期数没有加入以下的图表中。即本小结以下所有图表中的第一个 epoch 指的是第二个训练阶段的第一个 epoch。

图 3-5 为四种设定的训练损失对比，展示了他们的训练损失下降趋势，可以看到加入 Teacher Forcing 实际上给网络的训练制造了更大的困难，让损失略微的上升，但也更加贴近测试的场景，在表 3-8 中也的确体现了更好的测试集表现。去除 Warmup 的损失函数有更大的振荡情况，这表明使用 Warmup 有避免学习早期每一次迭代的偏差较大的问题，一定程度上使得学习更加稳定。去除 TransE embedding 则带来了较大的性能损失，不论是训练的损失函数、还是在测试集上的表现，去除 TransE 都让网络的表现出现了比其他设定更加突出的损失函数上升、评价指标下降。

以上的剥离实验进一步证明本文使用的策略和模型是有效的，Teacher Forcing

训练策略带来了更好的测试集表现，Warmup 使得学习更加稳定，TransE embedding 带来了较为显著的性能提升。

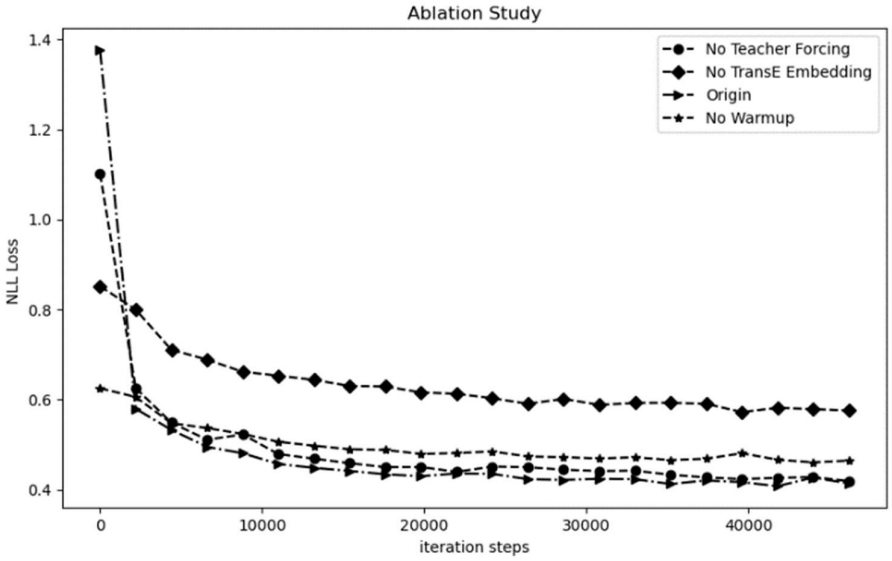

图 3-5 剥离实验的训练损失对比

表 3-8 剥离实验性能对比表

设定	F1 on 10 epoch	F1 on 15 epoch
No Warmup	38. 1755	45.2436
No Teacher Forcing	36.0217	39.8632
No TransE Embedding	32.1237	36.2903
Origin	40.4735	46.8955

3.6 系统部署

3.6.1 模型部署

在结束性能测试以后，为了更好的泛化应用，对模型进行第二次微调训练。第二次微调训练的主要内容是数据增强，通过让网络见到一些残缺、乱序的文本，促进网络在遇到未知的输入时仍然能产生合理的回答。

数据增强（data augment）：在原训练集中，以各三分之一的概率对对话历史的每个输入句子进行截断后半句、抹去随机一个词、逆序。构造的新数据集和原始训练集合并为一个集合，使模型在新的数据集上微调训练。本文最终模型是在第一次训练十六个周期（epoch）后（阶段一训练两个周期，阶段二训练十四个周期），在第二次微调训练三个周期（epoch）。

在结束第二次微调训练以后，将模型压缩存储为二进制文件。然后实现对话软件系统。实现的对话软件系统的主要模块包括：载入模块、数据转换模块、系统逻辑层、GUI 界面。总体的模块架构如图 3-6 系统模块划分图。

载入模块：实现深度学习模型和知识库的载入，本文使用百度飞桨（paddlepaddle 1.0）深度学习框架，模型的导出、导入都已经由框架 API 实现。在系统中本模块主要负责调用导出、导入 API，读取和转换数据集中的知识记录、对话路径记录。

数据转换模块：实现将用户输入信息转变为 DuConv 训练集格式的数据流，将对话历史和从库中检索的相关知识、对话目标路径组合成 python 字典对象。

系统逻辑层：负责系统启动、退出、运行、和用户交互的逻辑，接受用户输入和指令，转发数据给数据转换模块。实现系统的运行逻辑，如图 3-7 所示。

GUI 界面：可视化 UI 交互，将在下一节具体描述和展示。

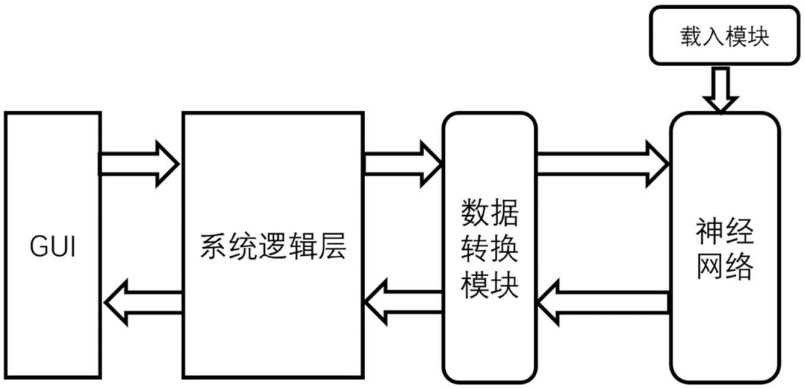

图 3-6 系统模块划分图

由于本系统属于主动式的对话系统，用户需要先选择话题，底层将话题对应的 goal、knowledge、空对话历史组织成 DuConv 格式的数据流送给网络，网络生成回答返回给前端。接下来底层会将继续话题对应的 goal、knowledge、人和机器的对话历史组织成 DuConv 格式的数据流送给网络、生成回答，如此往复进行多轮对话。任何时刻用户都可以选择重新开始，更换话题。

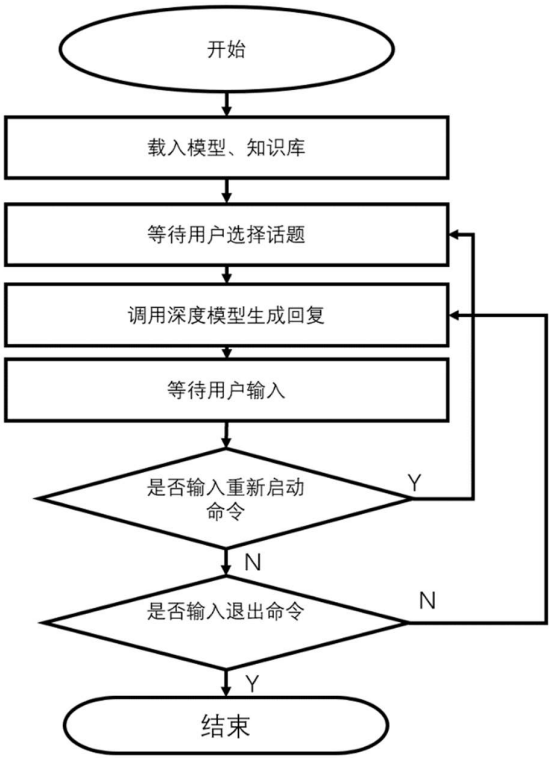

图 3-7 对话系统流程图

3.6.2 图形界面开发

本文的训练测试、部署是基于 python2.7 环境和百度飞桨（paddlepaddle 1.0）深度学习框架，在图形界面框架上选用 tkinter 和 ttk 两个 python 图形界面库的混合编程。主要使用 label、button、combobox、messagebox、scolltext、notebook 等可视化组件搭建可视化界面。除了可视化的呈现图 3-7 对话系统流程图的运行逻辑以外，在图形化界面中还额外实现了帮助、参考链接、消息提醒、对话历史保存文件等功能。图形化的用户界面交互情况展示如图 3-8。

其中分图(a)为系统的初始界面，左上的“文件”下拉菜单可以选择导出对话记录到文件，和选择退出系统。“相关”下拉菜单可以给出帮助信息、和相关论文的参考链接。分图（b）展示了开始使用系统的第一步，用户从下拉选单中选择话题，底层系统会组织对应的数据流给网络生成第一次机器的语言，发起对话。分图（c）展示了对话系统主动发起对话。分图(d)展示了多轮对话的过程，多轮对话的记录在下方的可滑窗文本框中记录，并可以导出为文件。

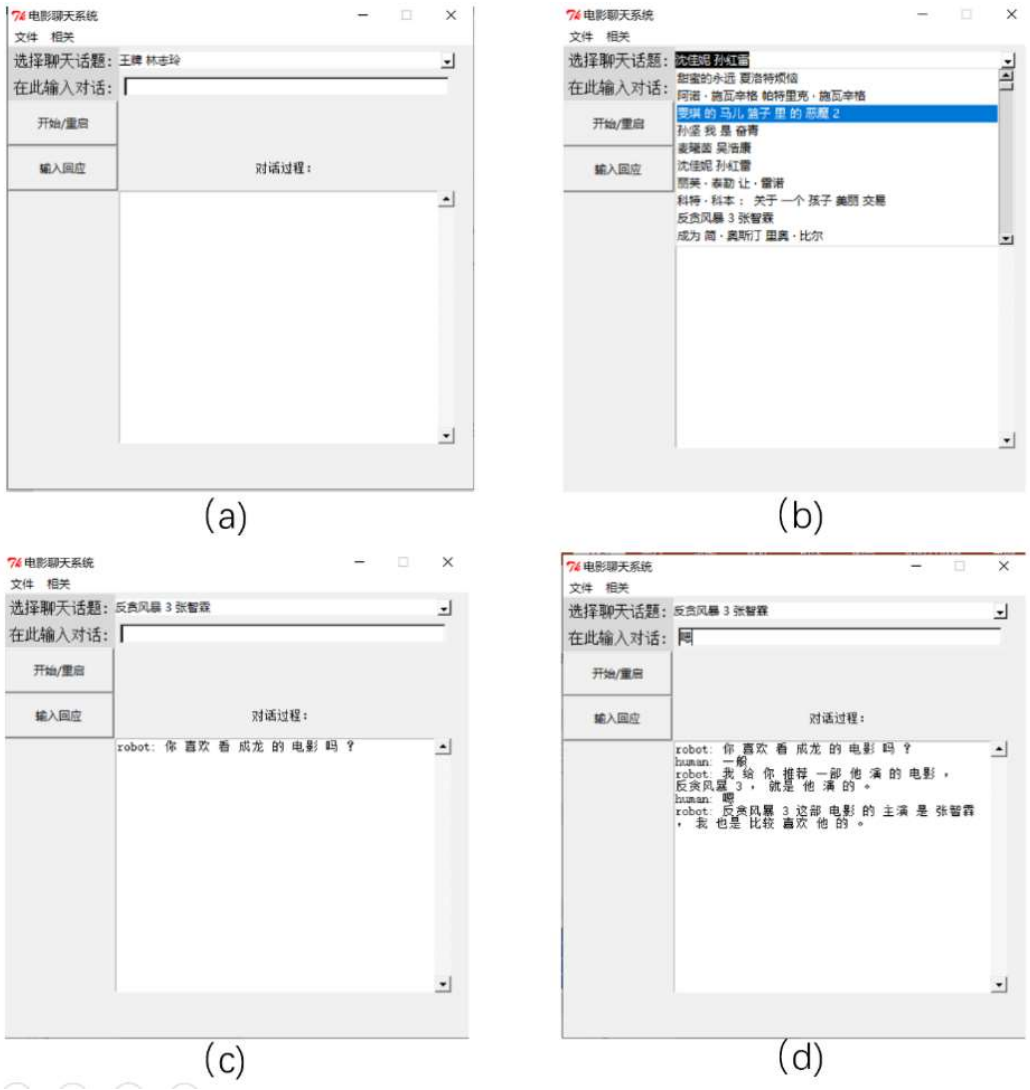

图 3-8 系统展示图 (a)为系统的初始界面，（b）展示了选择话题，（c）展示了对话系统发起对话，(d)展示了多轮对话的记录

3.7 本章小结

本章第一节首先介绍了本次系统设计使用的数据集、任务形式、总体框架，然后第二、三、四节根据总体框架，详细描述了神经网络方面的数据预处理、知识表示、选择、融合的模型设计。然后第五节介绍了性能评价指标，进行了横向对比实验和剥离实验，从多个层面的实验详细分析和证明了本论文所提出模型和策略的有效性。最后本章的第六节简述了该神经网络的部署过程，展示了可视化界面的实现和工作过程。

第四章 全文总结与展望

4.1 全文总结

得益于人工智能技术、高性能计算设备的飞速发展，尤其是高度并行的 GPU 计算设备、神经网络技术的进步，以往看似遥不可及的智能对话系统已经成为可能。目前基于端到端的非任务驱动型对话系统，往往生成高频通用回复，应答中缺乏实际有用的信息，回答符合语境却“言之无物”，比如总是生成“嗯”这样的回答。这样的回答不能给用户提供任何有意义的信息和帮助，这导致闲聊对话系统在电子商务、酒店咨询服务等各种应用领域的应用远远不及任务驱动对话系统。另外，目前绝大多数端到端的对话模型生成应答的过程可以简单描述为接受用户询问然后生成相关回应。这种完全被动的对话生成，会使用户逐渐失去对话的兴趣。如何让对话系统主动发起对话，这是一种全新的挑战，也是最终走向通用人工智能关键的一步。

针对上述这两个问题，本文在基于 seq2seq[6]的生成式端到端模型中，引入外部 SPO 三元组的结构化知识数据，赋予模型表示、选择、利用知识，生成包含丰富信息语句的能力。另外本文通过输入对话路径和知识实体之间的相互关联，赋予对话系统主动性，使网络不仅仅能建模多轮对话，还能主动的发起对话、组织多轮对话，甚至实现多轮对话的话题切换。本文最后在电影领域的对话数据上进行了实验验证，从多方面验证了本文模型、策略的有效性。另外本文的深度学习模型、知识驱动方案也可以适应大部分类型的知识对话系统，包括面向任务的对话系统和非任务驱动的被动式闲聊系统。

最后本文通过 python 语言，将前面训练的知识驱动对话神经网络模型部署成本地服务，并实现了一个可视化图形界面，形成一个可以和用户就电影知识展开多轮聊天的人机对话系统。

4.2 后续工作展望

随着智能对话系统的发展，人们对于对话形式有了各种要求。引入知识（事实）的闲聊人机对话系统在最近几年越来越受到学术界和工业界的重视，本文改进了基于注意力机制的序列到序列模型，使得模型可以融合外部的结构化知识数据，使得系统能够主动生成包含丰富信息的多轮聊天，但是仍然存在一些问题，值得在将来的工作中深入探讨：· ·

1.如何充分利用外部结构化知识数据。在传统的专家系统、产生式系统中，系统的推理都是可以解释的，而数据驱动的解决方案无法解释系统内部究竟是如何表达和使用知识的，如何强制网络用知识进行推理、关联，这个问题对进一步研究知识融合有重大意义。

2.在自然语言处理领域，最近一两年里基于 Transformer[4]结构的文本特征提取器在各个领域都取得了巨大的突破，比如 BERT[41]，GPT-2[43], Transformer 被认为是高效率的可并行的语言特征抽取结构。本文虽然有做用 Transformer 结构替代 GRU 结构的尝试，发现 Transformer 的确能更好的建模对话，但却没有找到合适的融合知识的方案，如何利用 Transformer 架构在网络内部实现知识选择，融合知识和对话，这是未来自己可以重点关注的一个方向。

致　谢

本论文的工作是在我的导师卢国明教授悉心指导下完成的，卢国明教授帮助我选择前沿的相关文献、为我提供计算资源，在多次遇到瓶颈和困难的时候给予我指导和帮助。本次毕业设计除了倾注我个人的心血和汗水更多的是导师的无私指导和谆谆教诲。因此在这里我首先对卢国明教授致以最真诚的感谢。其次，我需要感谢实验室鲁辰喜学长，多次为我解疑答惑，指点方向。

在电子科技大学四年的学习生涯中，感谢英才实验学院、计算机学院的领导和老师的帮助和提供的学术资源。尤其感谢我的辅导员刘今杰老师、郭培老师等，为我提供了很多关照。我也想感谢我校的刘帅成教授，从大二开始指导我研究计算机视觉，引领我真正的爱上了学术道路。

当然我也要感谢我的母亲还有我的女朋友高高，感谢她们始终给予我的鼓励、关心和支持。

最后，由衷地感谢在百忙之中抽出时间参与本次答辩的各位评委老师！

参考文献

[1] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences[C]. Advances in neural information processing systems. 2014: 2042-2050.

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

[3] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches[EB/OL]. arXiv preprint arXiv:1409.1259, 2014, https://arxiv.org/abs/1409.1259

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

[5] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C].Advances in neural information processing systems. 2014: 2672-2680.

[6] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]. Advances in neural information processing systems. 2014: 3104-3112.

[7] Mesnil G, He X, Deng L, et al. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding[C].Interspeech. 2013: 3771-3775.

[8] Yao K, Zweig G, Hwang M Y, et al. Recurrent neural networks for language understanding[C]. Interspeech. 2013: 2524-2528.

[9] Cuayáhuitl H, Keizer S, Lemon O. Strategic dialogue management via deep reinforcement learning[EB/OL]. arXiv preprint arXiv:1511.08099, 2015, https://arxiv.org/abs/1511.08099

[10] Zhou H, Huang M, Zhu X. Context-aware natural language generation for spoken dialogue systems[C]. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016: 2032-2041.

[11] Wen T H, Gasic M, Kim D, et al. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking[EB/OL]. arXiv preprint arXiv:1508.01755, 2015, https://arxiv.org/abs/1508.01755

[12] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB/OL]. arXiv preprint arXiv:1409.0473, 2014, https://arxiv.org/abs/1409.0473

[13] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]. Eleventh annual conference of the international speech communication association. 2010.

[14] Serban I V, Sordoni A, Bengio Y, et al. Building end-to-end dialogue systems using generative

hierarchical neural network models[C]. Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[15] Lowe R, Pow N, Serban I, et al. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems[EB/OL]. arXiv preprint arXiv:1506.08909, 2015, https://arxiv.org/abs/1506.08909

[16] Inaba M, Takahashi K. Neural utterance ranking model for conversational dialogue systems[C]. Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. 2016: 393-403.

[17] Zhou X, Dong D, Wu H, et al. Multi-view response selection for human-computer conversation[C]. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016: 372-381.

[18] 对话清华 NLP 实验室刘知远：NLP 搞事情少不了知识库与图神经网络[EB/OL]. https://tech.sina.com.cn/roll/2019-02-07/doc-ihrfqzka4023061.shtml

[19] 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.

[20] Fellbaum C. WordNet[J]. The encyclopedia of applied linguistics, 2012.

[21] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]. Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008: 1247-1250.

[22] Candan K S, Liu H, Suvarna R. Resource description framework[J]. ACM SIGKDD Explorations Newsletter, 2001, 3(1).

[23] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[C]. Advances in neural information processing systems. 2013: 2787-2795.

[24] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]. Twenty-Eighth AAAI conference on artificial intelligence. 2014.

[25] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]. Twenty-ninth AAAI conference on artificial intelligence. 2015.

[26] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 687-696.

[27] Ji G, Liu K, He S, et al. Knowledge graph completion with adaptive sparse transfer

matrix[C].Thirtieth AAAI conference on artificial intelligence. 2016.

[28] Vougiouklis P, Hare J, Simperl E. A neural network approach for knowledge-driven response generation[C]. 2016. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics:Technical Papers, pages 3370-3380.

[29] Ghazvininejad M, Brockett C, Chang M W, et al. A knowledge-grounded neural conversation model[C]. Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[30] Zhu W, Mo K, Zhang Y, et al. Flexible end-to-end dialogue system for knowledge grounded conversation[EB/OL]. arXiv preprint arXiv:1709.04264, 2017, https://arxiv.org/abs/1709.04264

[31] Lian R, Xie M, Wang F, et al. Learning to select knowledge for response generation in dialog systems[EB/OL]. arXiv preprint arXiv:1902.04911, 2019, https://arxiv.org/abs/1902.04911

[32] Zhou H, Young T, Huang M, et al. Commonsense Knowledge Aware Conversation Generation with Graph Attention[C]. IJCAI. 2018: 4623-4629.

[33] Shen Y, Deng Y, Yang M, et al. Knowledge-aware attentive neural network for ranking question answer pairs[C]. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 901-904.

[34] Zhang Y, Ren P, de Rijke M. Improving background based conversation with context-aware knowledge pre-selection[EB/OL]. arXiv preprint arXiv:1906.06685, 2019, https://arxiv.org/abs/1906.06685

[35] Li Z, Niu C, Meng F, et al. Incremental transformer with deliberation decoder for document grounded conversations[EB/OL]. arXiv preprint arXiv:1907.08854, 2019, https://arxiv.org/abs/1907.08854

[36] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural networks, 1989, 2(5): 359-366.

[37] Geoff Hinton Neural Networks for Machine Learning[EB/OL].lecture slides from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[38] Kingma D P, Ba J. Adam: A method for stochastic optimization[EB/OL]. arXiv preprint arXiv:1412.6980, 2014, https://arxiv.org/abs/1412.6980

[39] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[EB/OL]. arXiv preprint arXiv:1301.3781, 2013, https://arxiv.org/abs/1301.3781

[40] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[EB/OL]. arXiv preprint arXiv:1802.05365, 2018, https://arxiv.org/abs/1802.05365

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

https://arxiv.org/abs/1810.04805

[42] Lan Z, Chen M, Goodman S, et al. Albert: A lite bert for self-supervised learning of language representations[EB/OL]. arXiv preprint arXiv:1909.11942, 2019, https://arxiv.org/abs/1909.11942

[43] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI Blog, 2019, 1(8): 9.

[44] 刘知远，孙茂松，林衍凯,等. 知识表示学习研究进展[J]. 计算机研究与发展，2016, 53(2):247-261.

[45] Bordes A, Weston J, Collobert R, et al. Learning structured embeddings of knowledge bases[C].Twenty-Fifth AAAI Conference on Artificial Intelligence. 2011.

[46] Bordes A, Glorot X, Weston J, et al. A semantic matching energy function for learning with multi-relational data[J]. Machine Learning, 2014, 94(2): 233-259.

[47] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion[C]. The 29th AAAI Conference on Artificial Intelligence (AAAI'15).

[48] Wu W, Guo Z, Zhou X, et al. Proactive human-machine conversation with explicit conversation goals[EB/OL]. arXiv preprint arXiv:1906.05572, 2019, https://arxiv.org/abs/1906.05572

[49] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, Xiaoyong Du. Analogical Reasoning on Chinese Morphological and Semantic Relations[C]. Accepted by ACL 2018.

[50] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002: 311-318.

外文资料原文

The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

A Knowledge-Grounded Neural Conversation Model

**Marjan Ghazvininejad,[1]* Chris Brockett,[2] Ming-Wei Chang,[2]†
Bill Dolan,[2] Jianfeng Gao,[2] Wen-tau Yih,[2]‡ Michel Galley[2]**
[1]Information Sciences Institute, USC
[2]Microsoft
ghazvini@isi.edu, mgalley@microsoft.com

Abstract

Neural network models are capable of generating extremely natural sounding conversational interactions. However, these models have been mostly applied to casual scenarios (e.g., as "chatbots") and have yet to demonstrate they can serve in more useful conversational applications. This paper presents a novel, *fully data-driven*, and knowledge-grounded neural conversation model aimed at producing more contentful responses. We generalize the widely-used Sequence-to-Sequence (SEQ2SEQ) approach by conditioning responses on both conversation history and external "facts", allowing the model to be versatile and applicable in an open-domain setting. Our approach yields significant improvements over a competitive SEQ2SEQ baseline. Human judges found that our outputs are significantly more informative.

Introduction

Recent work has shown that conversational chatbot models can be trained in an end-to-end and completely data-driven fashion, without hand-coding (Ritter, Cherry, and Dolan 2011; Sordoni et al. 2015; Shang, Lu, and Li 2015; Vinyals and Le 2015; Serban et al. 2016, *inter alia*). However, fully data-driven systems still lack grounding in the real world and do not have access to external knowledge (textual or structured), which makes it challenging for such systems to respond substantively. Fig. 1 illustrates the difficulty: while an ideal response would directly reflect on the entities mentioned in the query (user input), neural models produce responses that, while conversationally appropriate, seldom include factual content. This contrasts with traditional dialog systems, which can readily inject entities and facts into responses, but often at the cost of significant hand-coding. Slot-filler dialog systems are hard put to come up with a natural sounding utterance like the second response in Fig. 1 in a manner that is generalizable and scalable.

The goal of this work is to benefit from the versatility and scalability of fully data-driven models, while simultaneously seeking to produce models that are usefully grounded in external knowledge, permitting them to be deployed in, for example, recommendation systems (e.g., for restaurants), and to adapt quickly and easily to new domains. The objective here is not task completion as in traditional dialog systems, but the ability to engage a user in a relevant and informative conversation. The tie to external data is critical, as the requisite knowledge is often not stored in conversational corpora. Much of this information is not found in structured databases either, but is textual, and can be mined from online resources such as Wikipedia, book reviews on Goodreads, and restaurant reviews on Foursquare.

Figure 1: Responses of fully data-driven conversation models are often appropriate, but generally lack content characteristic of human responses.

This paper presents a novel, *fully data-driven*, knowledge-grounded neural conversation model aimed at producing contentful responses. Our framework generalizes the Sequence-to-Sequence (SEQ2SEQ) approach (Hochreiter and Schmidhuber 1997; Sutskever, Vinyals, and Le 2014) of previous neural conversation models, as it naturally combines conversational and non-conversational data via techniques such as multi-task learning (Caruana 1997; Liu et al. 2015). The key idea is that we can condition responses not only based on conversation history (Sordoni et al. 2015), but also on external "facts" that are relevant to the current context (for example, Foursquare entries as in Fig. 1). Our approach only requires a way to infuse external information

* This work was conducted at Microsoft.
† Now at Google Research.
‡ Now at Allen Institute for Artificial Intelligence.

外文资料译文

本文主要引用的外文资料翻译如下：

1.一种基于知识的神经网络对话模型

摘要：神经网络模型能够产生听起来极其自然的对话回答。但是，这些模型主要应用于休闲场景（例如，作为“聊天机器人”），并且没有证据证明它们可以在更实际的对话应用中进行应用。本文提出了一种新颖的，完全是由数据驱动的，以知识为聊天基础的神经对话模型，旨在产生更有意义的回答。通过结合对话历史的信息和外部“事实”信息进行综合条件响应，我们改进了广泛使用的“序列到序列（SEQ2SEQ）”的建模方法，从而使无外部知识驱动的基线模型具有了通用性，并可以适用于开放域的应用场景设置。我们的方法在 SEQ2SEQ 基线模型之上产生了重大的改进。人类的主观裁判会发现我们的输出信息含量比基线模型大得多。

介绍：最近几年的研究工作表明，对话式聊天机器人模型可以完全以端到端的方式，由数据驱动进行训练学习，而无需进行人类手工编码（Ritter，Cherry 和 Dolan 2011；Sordoni 等人 2015； Shang，Lu 和 Li 2015；2012 年。Vinyals 和 Le，2015 年；Serban 等，2016 年）。但是，这种完全由数据驱动的系统在现实世界中仍然缺乏现实的基础概念，没有现实依据，无法访问外部的知识（文本或结构化知识），这使得这一类系统难以做出有实质性意义的响应。图 1 说明了这一种困难：理想的人类响应将直接反映查询中提到的实体（用户输入）的某些具体信息，给用户提供帮助，而神经模型产生的响应虽然在对话上符合语境，语意连贯，但很少包含具体的实际信息。这与传统的对话系统形成对比，传统的对话系统可以轻松地将实体和事实强制注入响应中，但通常会花费大量的手工编码,去人为的构造对话模板等。这种基于插槽填充的对话系统和生成式系统完全相反，能产生有信息的回答，却很难以可扩展的方式，像端到端模型一样生成如图 1 中的第二个响应那样自然、连贯的回答。

本文这项工作的目的是从完全由数据驱动的模型的多功能性和可扩展性中受益，同时力求产生有用的，以外部知识为基础的对话，以允许将对话模型部署在推荐系统（例如，餐馆）中，并能够快速，轻松地适应新的应用领域。这里的对话目标不是像传统任务驱动型对话系统那样完成具体的任务，而是提供用户包含相关信息且内容丰富的对话。如何实现这一目标，与外部知识数据的联系非常关

键，因为对话中必不可少的知识通常却不会存储在会话的语料库中。这些信息中的大部分也不是结构化的数据，而是以文本的形式，存在于诸如 Wikipedia，Goodreads 上的书评和 Foursquare 上的餐厅评论网站等在线资源中。本文提出了一种新颖的，完全由数据驱动的，有利用外部基础知识的神经网络对话模型，旨在让模型产生更加有意义的回答。我们的框架对以前的神经对话模型的序列到序列（SEQ2SEQ）方法（Hochreiter 和 Schmidhuber 1997;Sutskever，Vinyals 和 Le 2014）进行了泛化改造，因为它可以通过多任务等技术自然地将对话数据和非对话数据结合在一起学习（Caruana 1997; Liu et al.2015）。我们的关键思想是我们不仅可以根据对话历史记录来调节响应（Sordoni 等，2015），而且还能使用当前相关的外部“事实”，融合知识以产生回复。

我们的方法仅仅需要一种基于对话上下文的注入外部信息的方法（例如，通过简单的实体名称匹配），这使得本模型具有高度的通用性并适用于开放域场景设置。使用此模型框架，我们使用 Twitter 的 2300 万个通用域对话和 1.1M Foursquare 小贴士数据集对系统进行了大规模训练，与大规模 SEQ2SEQ 模型基线进行对比，其在信息性（人类裁判评估）方面显示出了巨大的改进。据我们所知，这是第一个大规模的，完全数据驱动的神经对话模型，并能有效地利用外部知识进行对话生成。