单位代码 10006
学　　号 15061084
分 类 号 TP312

北京航空航天大學
BEIHANG UNIVERSITY

毕业设计 (论文)

面向生成式对话的多样性评价指标分析

学 院 名 称　计算机学院
专 业 名 称　计算机科学与技术专业
学 生 姓 名　冯聪
指 导 教 师　荣文戈

2019 年 6 月

北京航空航天大学

本科生毕业设计（论文）任务书

I、毕业设计（论文）题目：

面向生成式对话的多样性评价指标分析

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

由三部分组成：学术论文、数据集和项目工程。其中学术论文主要为发表在人工智能和计算语言学等领域的国际顶级期刊的论文。数据集主要为训练对话系统的结构化或者非结构化的对话预料库。项目工程则是实现了某一模型的，用于定量实验的程序。设计要求为理解对话系统中各种评价指标的原理、优点和局限性。

III、毕业设计（论文）工作内容：

本课题力求对现有的面向生成的对话系统的评价指标做一个尽可能完备的文献综述。在对话系统领域，由于人类对话的多样性和歧义性，评价系统输出的响应是一个比较困难的问题，也是一个开放的学术问题。我们的工作是把目前所有的评价指标都整理出来，对它们作逐一的考察，在考察现在的评价指标的基础上，我们将分析不同评价指标在不同的数据集上的特点，总结出好的指标应该具有的优点，促进对话系统评估的自动化。

IV、主要参考资料：

How NOT to Evaluate Your Dialogue System: An Empirical Study of Unsupervised Metrics for Response Generation (Liu et al. 2016)

Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Model (Serban et al. 2016)

A Survey of Available Corpora for Building Data-Driven Dialogue Systems (Serban et al.)

BLEU: A Method for Automatic Evalutation of Machine Translation (Kishore Papineni et al. 2002)

计算机 学院 计算机科学与技术 专业类 150613 班

学生 冯聪

毕业设计 (论文) 时间： 2018 年 10 月 23 日至 2019 年 6 月 5 日

答辩时间： 2019 年 5 月 27 日

成　　绩：

指导教师：

兼职教师或答疑教师（并指出所负责部分）：

系（教研室）主任（签字）：

注：任务书应该附在已完成的毕业设计（论文）的首页。

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：冯聪

签字：

时间：2019 年 6 月

面向生成式对话的多样性评价指标分析

学　　生：冯聪
指导教师：荣文戈

摘　要

尽管基于序列到序列的生成式对话系统已经能够生成自然而流畅的响应，这类模型普遍存在着生成单调响应（Generic Response）的倾向。对话系统的目标是生成多样的，有意义的，能引起人们兴趣的对话。为了实现这一目标，学者们提出了各种模型，从不同角度解决单调响应的问题。但是由于缺乏好的自动化评价指标，模型的评估高度依赖于人类评价，导致评估系统的代价高昂，规模难以扩大。为了了解不同的评价指标的优缺点，本文在三个公开数据集上训练了三个生成式对话模型，测定不同指标的系统层面得分和句子层面得分。我们发现，不同数据集上的模型在不同指标上的得分没有完全的一致性。总体来说，数据集对得分的影响大于模型对得分的影响，使用相同特征的指标具有相似的句子层面得分分布。本文从经验上总结了上述现象的原因，包括合理的响应空间过于庞大，给评价增加了难度；模型在不同数据集上的泛化能力有待加强；指标的分布各异，给评价造成了混乱。本文为解决上述问题提出了若干方向，包括使用新的模型体系结构，研究开放式领域对话数据集的统计规律和发展特定数据集上的指标等等。

关键词：自然语言处理，响应生成，聊天机器人，评价指标

An Diversity-Oriented Analysis on Evaluating the Generative Dialogue Models

Author: Cong Feng
Tutor: Wenge Rong

Abstract

Although the Seq2Seq-based generative dialogue systems are able to generate natural and fluent responses, they have been widely known for preferring to generate simple and repeated responses. Towards the goal of generating diverse, meaningful and engaging dialogues, many researchers proposed various methods to address the problem of low-quality responses. However, it is known that the lack of good automatic evaluation metrics has led field to rely heavily on human evaluation, which is both expensive and unscalable. To better understand the pros and cons of various evaluation metrics, we trained three generative models on three public-available datasets and measured their performances with various metrics on system level as well as utterance level. We found that there is no universal consistency among the scores of all models on all datasets with all metrics. However, the datasets generally pose a heavier impact on the scores than the models do. In addition, the distributions of utterance-level scores are similar if the metrics use the same set of features. We made empirical explanation on the observation that the enormous space for reasonable responses makes the evaluation harder to tackle. Meanwhile, the ability of the models to generalize across various datasets remain highly enhanceable. Worse still, the highly diversed distributions of scores of various metrics makes the results somehow confusing. Based on the empirical results, we pointed out several directions for future work, including using new model architectures, investigating the statistical nature of the open-domain dialogue datasets and developing dataset-specific metrics.

Key words: Natural Language Processing, Response Generation, Chatbot, Evaluation Metrics

目 录

1 绪论

1.1 课题研究背景

早期的对话系统的主要用途是帮助用户用自然语言完成某项任务，比如技术支持（Technical Support），预订机票、预订餐馆的座位、查询航班等。这类系统又被称为面向任务的系统（Task-Oriented Dialogue System），其实现技术包括关键词匹配、规则和模板以及对话状态追踪（Dialogue State Tracking）等等，往往需要大量人工标注的数据。这些系统只能处理特定领域内的对话，不能回答开放性问题，用途局限于特定领域[1–3]。

随着在线聊天的流行，社交媒体和互联网论坛积累了大量的聊天语料数据，具有代表性的社交媒体和论坛有 Twitter，Reddit 和微博。大量的数据使人们可以构建数据驱动的（Data-Driven），开放领域（Open-Domain）的对话系统[4]。这种系统能根据对话的上下文和用户的提问产生语义相关的回答，用途有娱乐、语言学习工具和陪伴[5] 等等。本领域主要考察二人对话，两人聊天的历史记录称为上下文（Context），记为 c；当前说话的人说出的话语称为消息（Message），记为 m；另外一个人对该消息的回复称为响应（Response），记为 r。c, m, r 三者的关系如图 1.1 所示。系统的输入是 c, m，输出是 r，也就是输入对话的上下文和消息，输出响应，这个问题被称为对话响应生成（Dialogue Response Generation）。

图 1.1 上下文，消息和响应的关系[6]

对话系统又可以分为生成式对话系统（Generative System）和检索式对话系统（Retrieval-based System）[1, 7]。如果一个系统能生成训练集里没有的句子，就把它称为生成式系统[8]，反之称为检索式系统。生成式系统建立了给定输入句子，所有输出句子的条件概率，并从这个分布中生成句子。在解码（Decode）时，由于可能的句子空间过于庞大，在实际中

通常采用某种启发式搜索方法，如集束搜索（Beam Search），贪婪搜索（Greedy Search）和随机取样（Random Sample）等等，产生一个次优解。设 X 为输入句子，Y 为输出句子，U 是全部句子的集合，生成式模型的一般表示为：

$$Y = \underset{Y \in U}{\operatorname{argmax}}\, p(Y|X) \tag{1.1}$$

实际中 argmax 会被某种搜索算法代替。

检索式系统根据输入句子从一个文库 C 中检索输出句子。C 通常由人类撰写的语句组成，并且足够大，使得输出句子不容易重复。系统通过某种打分机制，如词频-逆文档频率（Term Frequency-Inverse Document Frequency，TF-IDF）或语义相似度（Semantic Similarity），对数据库中的候选句子进行打分，并把得分较高的候选句子作为输出：

$$Y = \underset{Y \in C}{\operatorname{argmax}}\, Score(Y, X) \tag{1.2}$$

可见，生成式系统和检索式系统的根本区别在于获得输出句子的机制不同。这两种方法各有优劣：检索式模型的输出没有语法错误并且可以筛选输出的内容[7]，但是不能生成新句子；生成式的模型能端对端的训练而且可以生成新的句子[3]，但是容易生成过短的句子[9]。在实际环境中，通常将它们作为模型联合体（Model Ensemble）使用[10]，而检索式系统也经常作为生成式系统的基线系统[3, 6]。

生成式模型的流行得益于自然语言处理领域发展的一系列基础技术，包括为单词提供平滑特征的词嵌入（Word Embedding）[11–13]；能对变长序列建模的循环神经网络语言模型（Recurrent Neural Networks Language Model，RNNLM）[14]；易于训练，能避免梯度消失问题[15] 的循环门单元，如长短期记忆单元（Long Short-Term Memory，LSTM）[16] 和门循环单元（Gated Recurrent Unit，GRU）[17]；以及序列到序列框架（Sequence to Sequence Framework，Seq2Seq）[2, 17]。

Seq2Seq 框架在自然语言处理的多项任务上都超过了之前的最佳水平，因此被广泛应用到对话生成领域。在国外，最早把 Seq2Seq 用到对话生成领域的是 Vinyals 等人[18]，他们在 OpenSubtitles[19] 上训练的模型能回答简单的常识问题，并且比基于规则的系统 CleverBot[1] 获得了更高的人类评价得分。

Li 等人提出了一系列基于 Seq2Seq 框架的对话系统，包括利用最大互信息（Maximum Mutual Information，MMI）增加输出多样性的目标函数[20]；在解码器端（Decoder）加

[1] http://www.cleverbot.com/

入说话人身份信息（Speaker ID），促进输出的人格一致性（Personality Coherence）[21]；利用对抗生成网络（Generative Adversarial Networks，GAN）[22] 使系统输出和人类输出难以分辨[23] 等等。

Serban 等人把 Sordoni 等人提出的用于查询建议（Query Suggestion）的多层编解码器（Hierarchical Recurrent Encoder-Decoder，HRED）[24] 应用到对话生成领域，提出了能捕捉对话的层级结构的 HRED 模型[5]。基于 HRED，Serban 等人又提出了利用随机潜变量（Stochastic Latent Variable）增加对话多样性的 Variational Hierarchical Recurrent Encoder-Decoder，即 VHRED[25]；以及加入了高层次抽象信息的多精度循环网络（Multiresolution Recurrent Neural Networks，MultiRNN）[26]。

在国内，Shang 等人研究了微博数据集上的短文本对话生成问题（Short-Text Conversation），提出了以 GRU 为门单元的编解码器模型（Neural Response Machine，NRM）[3]，并在人类评价上取得了比检索式系统和翻译式系统[4] 更好的成绩。Chen Xing 等人加入了从预训练 LDA 模型中获取的主题词（Topic Words），提出了 Topic-Aware Seq2Seq [27]，由于篇幅有限，不能一一介绍。

1.2 课题研究意义

本领域目前有两个难题：一是评价系统生成的响应的困难性[3, 8, 28–30]；二是如何避免生成单调的响应 [4–6, 20, 31]。对第一个难题的一个普遍解释是，和机器翻译相比，对话生成的输出具有更高的熵（Entropy）。如表 1.1 所示，给定一个消息，往往存在大量合理的响应，这些响应的语义多种多样，和参考响应（Ground Truth）的语义可能完全不同。

表 1.1 合理的响应不唯一[32]

Query:	为什么不领养一个?
	Why not adopt one?
Groundtruth:	我也想领养，但是不好找
	I hope so, but it's hard to find one
Candidate 1:	恩, 我已经在朋友中找了
	I'm looking for one from my friends
Candidate 2:	巴比伦是稀有品种
	Pappilon is a rare breed
Candidate 3:	可以哈, 谢谢你的建议
	OK, thank you for your advice

这种固有的多样性决定了评价响应生成系统的困难。Liu 等人在[8] 中研究了两类指标，分别是基于词重叠的指标（Word Overlap Based）和基于词嵌入的指标（Embedding Based），并且发现这些指标在非技术性的 Twitter 数据集上和人类评价只有弱相关性，在技术性的 Ubuntu Dialogue Corpus[33] 没有相关性。朝着 Liu 等人提出的新方向，许多学者提出了新的对话指标。

本课题延续 Liu 等人的工作，对自动评价指标进行深入研究。Liu 等人已经研究了评价指标与人类评价的相关性，并且发现了它们的弱点。本课题进一步研究了评价指标之间的相关性，以及模型的性能在不同数据集上的一致性。虽然，目前大多数评价指标尚不能和人类评价一样准确的衡量系统的性能，但是对它们性质的研究将有助于理解现有指标的弱点，进而有助于自动化指标的发展。

1.3 课题研究内容

本课题以[25] 的实验中使用的三个模型 LSTM，HRED，VHRED 为基线，扩展了[8] 中考察的两类指标，并在三个具有代表性的公开数据集：Ubuntu Dialogue Corpus，OpenSubtitles 和 LSDSCC[34] 上进行了实验。尽管没有进行人类评价，但是我们对指标之间和模型之间的一致性分析还是取得了有意义的结论。

1.4 论文组织结构

本文的组织结构如下：第 2 章相关工作介绍了本领域的模型，指标和数据集的基本情况。第 3 章研究方法介绍了我们的实验框架，模型超参数，数据集预处理和指标参数的选择等等。第 4 章实验结果与讨论详细展示了实验数据和结论。最后一章结论总结了本课题的研究成果，提出了若干研究方向。

2 相关工作

2.1 生成式模型

2.1.1 定义

一个生成式模型定义了给定输入序列 $X = x_1, x_2, \ldots, x_n$，任意输出序列 $Y = y_1, y_2, \ldots, y_m$ 的条件概率：

$$p(Y|X) = P(y_1, y_2, \ldots, y_m | x_1, x_2, \ldots, x_n) \tag{2.1}$$

模型的训练目标就是在数据集 S 上最大化给定 X，Y 的对数概率（Log Probability）：

$$L = \frac{1}{|S|} \sum_{(Y,X)\in S} \log p(Y|X) \tag{2.2}$$

从定义上看，语言模型[11, 14]（Language Model）和编解码器（Encoder-Decoder）都属于生成式模型，因为它们都定义了条件概率 $p(Y|X)$。

生成式模型需要把一个长度可变的序列 X 映射到另一个长度可变的序列 Y，且 X 和 Y 的长度可以不相等。循环神经网络（RNN）为这个问题提供了自然的解决方案。RNN 的基本思想是：序列由有序的元素组成，每一个时刻（Time Step）输入一个元素，同时更新内部的隐藏状态（Hidden State），然后输出一个元素。如图 2.1[1] 所示，在时间轴上展开的 RNN 和一般的前馈神经网络（Feed Forward Neural Networks）很像，不过每一个时刻的权重矩阵都是共享的。这个共享的权重矩阵 A 又被称为循环矩阵（Recurrent Matrix），它的作用是对输入序列进行某种保持顺序信息的编码。

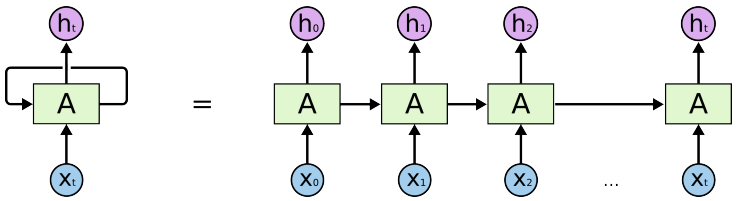


图 2.1 在时间轴上展开的 RNN

根据是否使用了某种门单元，RNN 可分为普通 RNN[14]，LSTM[16] 和 GRU[17]。根

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

据是否对反向序列（Reversed Sequence）编码，RNN 可分为单向 RNN（Unidirectional RNN）和双向 RNN（Bidirectional RNN）[35]。由于普通 RNN 受到梯度消失的影响，目前学界普遍采用 LSTM 或者 GRU；尽管后者受到梯度爆炸的影响，但是可以通过梯度剪裁（Gradient Clipping）[18] 解决。此外，多层 RNN 组成的深度循环神经网络比单层 RNN 有更好的性能[18]。

2.1.2 RNN 语言模型

RNN 语言模型（RNNLM）通过循环神经网络估计给定序列 $X = x_1, x_2, \ldots, x_n$ 的概率分布：

$$p(X) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) \tag{2.3}$$

$$p(x_i = w | x_1, x_2, \ldots, x_{i-1}) = \frac{\exp o_{tw}}{\sum_{v=1}^{V} \exp o_{tv}} \tag{2.4}$$

o_t 是 RNN 的在 t 时刻的输出向量，V 是词汇表的大小，RNN 语言模型[2]通过如下公式计算出 o_t：

$$o_i = h_i^T W_{out} \tag{2.5}$$

$$h_i = \sigma\left(x_i^T W_{in} + h_{i-1}^T W_{hh}\right) \tag{2.6}$$

W_{in} 是输入矩阵，W_{out} 是输出矩阵，W_{hh} 是循环矩阵，σ 是激活函数。RNN 语言模型在训练时最大化训练集上的句子的对数概率：

$$L(X) = \sum_{i=1}^{n} \log p(x_i | x_1, x_2, \ldots, x_{i-1}) \tag{2.7}$$

2.1.3 Seq2Seq 框架

如图 2.2 所示，Seq2Seq 框架使用两个有着独立参数的 RNN 分别作为编码器和解码器。首先，编码器把输入序列 X 编码成一个定长向量 v。该向量又称为思考向量（Though Vector），是编码器的最后一个隐藏状态（Last Hidden State）。接着，解码器以 v 为初始隐藏状态（Initial Hidden State）生成输出序列。

[2]为了简洁起见，我们描述了不使用任何门单元的 RNN。LSTM 和 GRU 有着更复杂的数学表达式。

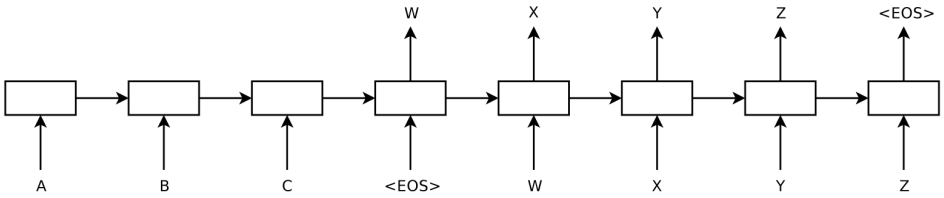


图 2.2 Seq2Seq 框架图[2]

通过先把输入序列 X 变换成某种压缩编码 v，再把 v 还原为另一个序列 Y，Seq2Seq 把公式 2.1 作了如下转化：

$$h_i = f(x_i, h_{i-1}) \tag{2.8}$$

$$v = h_n \tag{2.9}$$

$$p(y_1, y_2, \ldots, y_m | x_1, x_2, \ldots, x_n) = \prod_{i=1}^{m} p(y_i | v, y_1, y_2, \ldots, y_{i-1}) \tag{2.10}$$

h_n 是编码器的最后一个隐藏状态，f 是编码器采用的门单元函数。编码器和解码器以同一个目标函数同时训练。

为了更好的处理长序列，Seq2Seq 一般引入注意力机制（Attention Mechanism）[36, 37]，使输入序列的信息不必全部通过固定长度的向量 v 传递。该机制使解码器能自动关注和当前输出最相关的输入部分，实现输入序列与输出序列的对自动对齐（Automatic Alignment）。

2.1.4 解码算法

生成式模型仅仅定义了条件概率 $p(Y|X)$，在解码阶段，需要采用某种启发式搜索算法从概率分布中生成输出 Y。最简单的搜索算法是贪心搜索（Greedy Search）：在每一时刻都输出条件概率最大的单词：

$$y_i = \underset{w \in V}{\operatorname{argmax}}\, p(w | y_1, y_2, \ldots, y_{i-1}, X) \tag{2.11}$$

因为各个 y_i 的概率都不是独立的，而是受之前输出的单词的影响，贪心搜索不能保证得到概率最大的输出序列。

随机取样（Random Sampling）每生一步都从模型估计的全体词汇的离散概率分布

中随机选取一个单词：

$$p(y_i = w) = p(w|y_1, y_2, \ldots, y_{i-1}, X), w \in V \tag{2.12}$$

Serban 等人发现随机取样能避免单调响应的问题，并且能产生多样化的，话题相关的输出[5]，但是 Li 等人指出随机取样会导致输出中出现语法错误[38]。

最为常用的方法是集束搜索（Beam Search），它在生成整个句子的过程中维护一个大小为 B 的列表，称为集束（Beam）。算法开始时，集束初始化为模型生成的 B 个概率最高的单词。在每一个时刻开始时，集束中都有 B 个部分生成的句子，它们称为候选 Y_c。在每一个时刻，算法对集束中的每一个候选都生成 B 个概率最大的下一个单词 $w_{i+1,c}$，从而形成 $B \times B$ 个部分生成的句子，称为扩展的候选集。从扩展的候选集中，只保留前 B 个概率最大的句子。算法不断迭代直到某些候选中产生了句子结束符号（End-of-Sentence，EOS），算法将这些结束了的句子作为输出。本质上，集束搜索是一种队列长度有限的宽度优先搜索（Breath First Search）。

为了增加模型输出的多样性，学者们提出了许多改进的解码算法。Li 等人提出了 Diverse Beam 算法，在标准集束搜索中对来自相同父节点的候选加以惩罚，即鼓励来自不同父节点的候选[38]。Li 等人还提出了以最大互信息（MMI）为目标函数的解码算法[20]。本质上，他们训练了一个根据输入预测输出的正向模型（Forward Model）和一个根据输出预测输入的反向模型（Backward Model），再用反向模型的反向概率 $p(Y|X)$ 对正向模型生成的候选集进行重新排序。据此，他们提出了两种解码算法：MMI_antiLM 和 MMI_bidi，分别如公式 2.14和公式 2.15 所示：

$$MMI(S,T) = \log \frac{p(S,T)}{p(S)p(T)} \tag{2.13}$$

$$MMI\_antiLM(S,T) = \log p(T|S) - \lambda \log p(T) \tag{2.14}$$

$$MMI\_bidi(S,T) = (1-\lambda) \log p(T|S) + \lambda \log p(S|T) \tag{2.15}$$

Li 等人还提出了随机贪心取样（Stochastic Greedy Sampling）算法 [31]，以求在随机取样和贪心搜索之间找到一个平衡点。该算法只在条件概率最高的前 K 个候选单词中取样，参数 K 控制了随机取样和贪心搜索之间的比例：K 越大，算法越接近随机取样，K 越小，算法越接近贪心搜索。这些改进的解码算法在不同程度上提高了响应的多样性。

2.2 自动化评价指标

2.2.1 评价指标简介

机器翻译领域已有大量和人类评价相关性较高的指标，例如 BLEU[39]，NIST[40]，METEOR[41]，BEER[42]，CHRF[43]，TER[44] 等等。然而，适用于开放领域的，面向闲聊的对话系统的指标要少得多；在考察本领域对自动指标的使用情况之前，先对各种指标作一个简要介绍。

BLEU，Bilingual Evaluation Understudy[39] 是 Papineni 在 2002 年提出的，用于机器翻译的自动评价指标。它是一个系统层面的评价指标，即评价一个系统在整个测试集上的性能。BLEU 指标只有一个参数 N，表示要计算的各阶 n-gram 准确率的最大值；例如 $N=4$ 表示要计算 1-gram 到 4-gram 的准确率。准确率指的是系统输出和参考输出之间的 n-gram 重叠数占系统输出总的 n-gram 的比例。BLEU 由在整个数据集上计算的各阶 n-gram 准确率的几何平均值（Geometric Mean）和简短惩罚系数（Brevity Penalty）相乘得到。引入简短惩罚系数的原因是，较短的系统输出句子的准确率较高，需要矫正。n-gram 准确率的计算公式为：

$$p_n = \frac{\sum_{C\in\{Candidates\}}\sum_{n\text{-}gram\in C} Count_{clip}(n\text{-}gram)}{\sum_{C'\in\{Candidates\}}\sum_{n\text{-}gram'\in C'} Count(n\text{-}gram')} \tag{2.16}$$

Candidates 为系统输出的句子集合，$Count_{clip}(n\text{-}gram)$ 为截断的 n-gram 共现数，$Count(n\text{-}gram')$ 是 Candidates 中的总 n-gram 数。简短惩罚系数 BP 的计算公式为

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \le r \end{cases} \tag{2.17}$$

其中 c 是模型输出句子的长度，r 是参考输出句子的长度。BLEU 的最终公式为：

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.18}$$

实际使用中一般取 $N=4$，$w_n = 1/N$。由于原始的 BLEU 容易在句子层面给出 0 分，人们提出了各种平滑处理[45]。本文在使用 BLEU 时也采用了一种平滑处理。

ROUGE，Recall-Oriented Understudy for Gisting Evaluation[46] 是一种基于召回率的自动摘要领域的指标。它有多个变形：ROUGE-N，ROUGE-L，ROUGE-W，ROUGE-S，

以及 ROUGE-SU，分别使用了不同的计数单元（Counting Unit），如 n-gram 共现数、最长公共子序列（Longest Common Subsequence，LCS）和二元跳词（Skip-Bigram）等等。这些指标的基础是信息检索领域常用的 F-measure，即准确率和召回率的加权调和平均值：

$$F\text{-}measure = \frac{(1+\beta^2)RP}{R+\beta^2 P} \tag{2.19}$$

β 控制准确率和召回率的相对重要性。以下无特殊说明时，当指标是句子层面的时候，n 是系统句子的长度，m 是参考句子的长度；当指标是摘要层面的时候，n 是系统摘要的总单词数，m 是参考摘要的总单词数。

ROUGE-N 利用了 n-gram 共现数，其公式为：

$$ROUGE\text{-}N = \frac{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count_{matched}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count(gram_n)} \tag{2.20}$$

摘要层面的 ROUGE-N 具有相同的形式。

句子层面（Sentence Level）的 ROUGE-L 的公式为：

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{2.21}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{2.22}$$

$$ROUGE\text{-}L = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{2.23}$$

LCS 是计算两个序列的最长公共子序列的长度的函数。摘要层面的 ROUGE-L 的公式为：

$$R_{lcs} = \frac{\sum_{i=1}^{\mu} LCS_{\cup}(r_i,C)}{m} \tag{2.24}$$

$$P_{lcs} = \frac{\sum_{i=1}^{\mu} LCS_{\cup}(r_i,C)}{n} \tag{2.25}$$

$$ROUGE\text{-}L = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{2.26}$$

μ 是系统输出的摘要句子的数量，$LCS_{\cup}(r_i,C)$ 计算了参考句子 r_i 和候选摘要 C（由多个句子组成）的 LCS 的并集。

句子层面的 ROUGE-W 的公式为：

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \tag{2.27}$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \tag{2.28}$$

$$ROUGE\text{-}W = \frac{(1+\beta^2)R_{wlcs}P_{wlcs}}{R_{wlcs}+\beta^2 P_{wlcs}} \tag{2.29}$$

WLCS 是一个计算两个序列的加权 LCS 的算法，它奖励较长的连续的 LCS。摘要层面的 ROUGE-W 与摘要层面的 ROUGE-L 类似。

二元跳词是句子中顺序不变的一对单词，两个单词之间可以有任意数量的其他单词。基于二元跳词的句子层面 ROUGE-S 定义为：

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \tag{2.30}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \tag{2.31}$$

$$ROUGE\text{-}S = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \tag{2.32}$$

$C(\cdot,\cdot)$ 为组合数。摘要层面的 ROUGE-S 相当于把摘要看作首尾相连的句子来计算。ROUGE-SU 是 ROUGE-S 加入了 Unigram 的扩展。

METEOR, Metric for Evaluation of Translation with Explicit ORdering [41] 是针对 BLEU 的一些弱点提出的机器翻译的指标。与 BLEU 相比，METEOR 在句子水平上与人类评价有更好的相关性。METEOR 首先计算系统输出和参考输出之间的 Unigram 匹配，这些匹配由多个可配置的模块组成，包括 Exact，Porter-stem，WordNet-synonymy，分别表示严格匹配，Porter 词根匹配和 WordNet 同义词匹配。接着，METEOR 在 Unigram 匹配上计算一个对齐，并得到基于 Unigram 匹配的准确率和召回率，进而得到二者的加权调和平均值：

$$Fmean = \frac{10PR}{R+9P} \tag{2.33}$$

METEOR 还加入了对较短的 n-gram 匹配的惩罚系数：

$$Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams\_matched}\right) \tag{2.34}$$

#unigrams_matched 是所有匹配的 Unigram 的数量；一个 Unigram 匹配越短，*#chunks* 就越大。METEOR 的最终公式为：

$$METEOR = Fmean * (1 - Penalty) \tag{2.35}$$

困惑度（Perplexity，PPL）是一种衡量统计语言模型性能的指标。一个模型的困惑度为 P 可以形象的表述为：该模型在预测一个词的时候，平均需要从 P 个词中等可能的选出一个。因此，困惑度越低，语言模型在选择一个词时就越不“困惑”。困惑度的计算公式为：

$$PPL = \exp(-\frac{1}{N}\sum_{i=1}^{N}\log p(x_i)) \tag{2.36}$$

N 是测试集的样本数，x_i 是一个样本，在语言模型中它是一个句子，在生成式模型中它是一对输入输出序列 (X, Y)。$p(x_i)$ 是模型估计的概率。一个好的模型应该给测试集的样本估计较高的概率，所以好的模型 PPL 较低。实际在自然语言处理中使用的 PPL 还要除以文本中的总单词数，得到平均每个词的困惑度（Perplexity per-word）：

$$PPL\text{-}w = \frac{1}{\#words}\exp(-\frac{1}{N}\sum_{i=1}^{N}\log p(x_i)) \tag{2.37}$$

词嵌入（Word Embedding）指标是一类建立在分布式假设 [47, 48]（Distributed Hypothesis）上，用分布式语义（Distributed Semantic）来衡量两个句子的相似程度的指标，常用于句子文本相似性（Sentence Textual Similarity）和学生输入自动打分[49] 等任务中。这类指标一般用某种组合方式从单词的向量表示得到句子的向量表示，再用余弦相似度（Cosine Similarity）测量两个句子向量的相似程度[50]：

$$\cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \tag{2.38}$$

$$\tag{2.39}$$

最常见组合方式是对单词向量取平均值，类似于词袋表示（Bag-of-Words），对应的指标

就是向量平均值（Vector Average）：

$$\bar{e}_r = \frac{\sum_{w \in r} e_w}{|\sum_{w' \in r} e_{w'}|} \tag{2.40}$$

$$\textit{Vector-Average} = \cos(\bar{e}_r, \bar{e}_{\hat{r}}) \tag{2.41}$$

r 是参考输出，$\hat{r}$ 是系统输出，w 是句子中的一个单词。

另外一种组合方式是向量极值（Vector Extrema），它把单词向量每个维度上的极端值作为句子向量在该维度上的值[51]：

$$\text{extrema}(d_i) = \begin{cases} \max d_i & \text{if } \max d_i \geq |\min d_i| \\ \min d_i & \text{otherwise} \end{cases} \tag{2.42}$$

$$e_r^{ex} = [\text{extrema}(d_1), \ldots, \text{extrema}(d_n)] \tag{2.43}$$

$$\textit{Vector-Extrema} = \cos(e_r^{ex}, e_{\hat{r}}^{ex}) \tag{2.44}$$

$[\cdot,\cdot]$ 表示连接多个标量形成一个向量，e_r^{ex} 是参考输出的向量极值表示，$e_{\hat{r}}^{ex}$ 是系统输出的向量极值表示。

贪心匹配（Greedy Matching）得名于边加权的二部图（Weighted Bipartite Graph）的最大匹配问题[49]：把两个句子的单词看做二部图的节点，任意两个节点之间有一条边，边的权重定义为两个单词的余弦相似度，一个匹配定义为一对点，问如何构造一个匹配的集合，使其权重之和最大。贪心匹配给出了一种贪心算法：

$$G(r, \hat{r}) = \frac{\sum_{w \in r} \max_{\hat{w} \in \hat{r}} \cos(e_w, e_{\hat{w}})}{|r|} \tag{2.45}$$

$$\textit{Greedy-Matching} = \frac{G(r, \hat{r}) + G(\hat{r}, r)}{2} \tag{2.46}$$

除了上述三种组合方法外，还有很多其他方法[50]。

Distinct-N 是 Li 等人提出的衡量句子层面的 n-gram 多样性的指标[20]：

$$\textit{Distinct-N} = \frac{\#\textit{unique-ngrams}}{\#\textit{words}} \tag{2.47}$$

Lowe 等人从自动图灵测试（Automatic Turing Test）得到灵感，提出了自动化对话评价模型（Automatic Dialogue Evaluation Model，ADEM）[30]。如图 2.3 所示，ADEM 采

用 VHRED 的多层编码器（Hierarchical Encoder）分别对上下文 c，参考 r 和响应 $\hat{r}$ 进行编码，然后简单的将三者的编码线性组合到一起，并经过归一化得到区间在 $[1,5]$ 的得分。模型通过最小化和人类评价的均方误差（Mean Squared Error，MSE），可以端对端的训练。

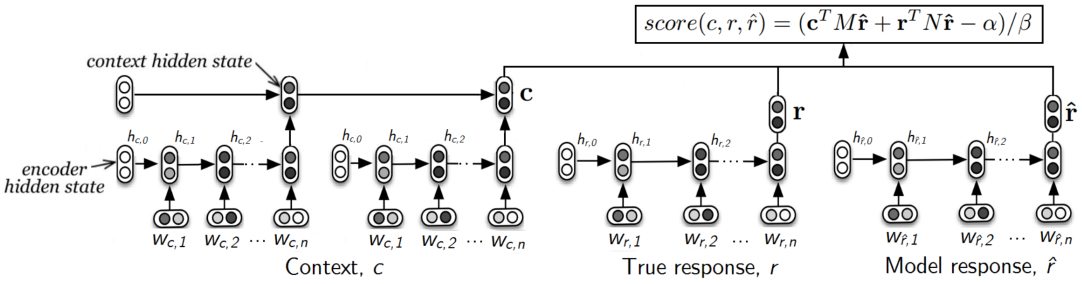


图 2.3 ADEM 模型结构图

公式 2.48 是 ADEM 的打分公式，M,N 是可学习的参数矩阵，α,β 是缩放常量，使得分数落在 $[1,5]$ 区间。模型训练的目标函数如公式 2.49 所示，它是一个带 L2 正则化的均方误差。

$$score(c,r,\hat{r}) = (c^T M\hat{r} + r^T N\hat{r} - \alpha)/\beta \tag{2.48}$$

$$\mathcal{L} = \sum_{i=1:K} [score(c_i,r_i,\hat{r}_i) - human_i]^2 + \gamma\|\theta\|_2 \tag{2.49}$$

Lowe 等人在一个带人类评价的 Twitter 数据集上训练并且评估了 ADEM 模型，发现它和人类评价的相关性在句子水平和系统水平都达到了很高水平。

Tao 等人提出了 Referenced Metric and Unreferenced Metric Blended Evaluation Routine[32]，简称 RUBER。如图 2.4a 所示，RUBER 由带参考的指标（Referenced Score）和无参考的指标（Unreferenced Score）组成。带参考的指标使用了基于词嵌入的最大-最小池化（Max-Min Pooling）作为句子向量，衡量了消息和响应的相似度：

$$v_{max}[i] = \max\{w_1[i], w_2[i], \ldots, w_n[i]\} \tag{2.50}$$

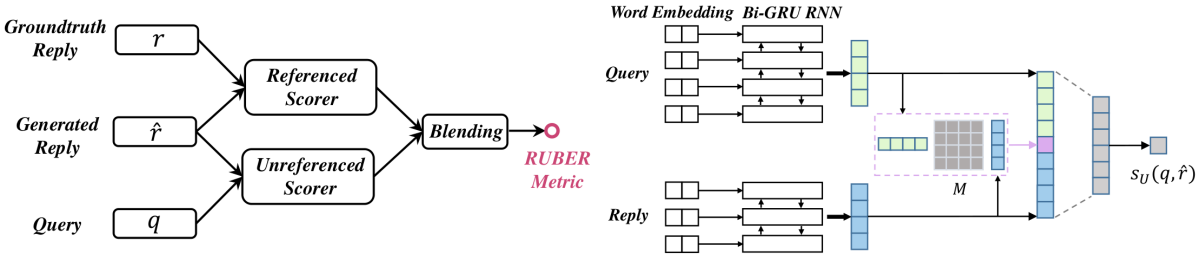
$$v_{min}[i] = \min\{w_1[i], w_2[i], \ldots, w_n[i]\} \tag{2.51}$$

$$v = [v_{max}; v_{min}] \tag{2.52}$$

$$s_R(r,\hat{r}) = \cos(v_r, v_{\hat{r}}) = \frac{v_r^T v_{\hat{r}}}{\|v_r\| \cdot \|v_{\hat{r}}\|} \tag{2.53}$$

无参考的指标通过如图 2.4b 所示的神经网络来估计，使用无监督的负采样（Negative

Sampling）来训练。该指标在中文数据集豆瓣网[3]上取得了很高的人类评价相关性。



(a) 分数组成　　(b) 无参考指标的模型结构

图 2.4　RUBER 指标

Kannan 等人初步尝试基于对抗生成网络的对抗式评价（Adversarial Evaluation）[9]，他们训练了一个鉴别器（Discriminator）来区分一个响应是来自系统还是来自人类，并且发现鉴别器能捕捉到基于 Seq2Seq 的模型倾向于生成短句子和通用句子的缺点。Li 等人在[23] 中构建了完整的对抗生成网络，提出了训练和对抗式评价的一体框架。由于篇幅有限，不再描述。

2.2.2　评价指标使用情况

生成式对话系统还没有公认的，与人类评价相关度较高的评价指标，对话的研究者们很早就意识到了这一点[3–5, 18]，所以他们常常使用人类评价作为核心的评判标准，有时用一些指标来衡量模型某一方面的性质，但很少把指标作为评价系统的主要手段。此外，展示和对比不同模型的响应输出也是这个领域常见的分析手段，通常称为定性分析（Qualitative Analysis），比如表 2.1 展示了模型 GoogleChatbot 对哲学问题的响应。

Ritter 等人首次尝试了数据驱动的，面向闲聊的对话响应生成[4]。因为不清楚面向任务的指标能不能用于评价生成式模型，他们就用了基于 Amazon Mechanical Turk 的人类评价。他们利用人类评价的数据考察了 BLEU 在这方面的适用性，并发现系统的 BLEU 得分非常低，和人类评价的相关性也不是很高，因此他们认为 BLEU 不能直接应用到本领域。

Shang 等人在评价他们的 NRM 模型时分析了几种指标在本领域的适用性[3]。他们认为 BLEU 并不适用，因为合理的响应的范围实在是太大了，参考响应不可能完全覆盖到；而常用于语言模型的指标困惑度（Perplexity，PPL）也不适用，因为它不能测量响应的自然程度及其对消息的相关程度。最后他们选择了人类评价。

[3]http://www.douban.com

表 2.1 定性分析的例子[18]

Human:	what is the purpose of life ?
Machine:	to serve the greater good .
Human:	what is the purpose of living ?
Machine:	to live forever .
Human:	what is the purpose of existence ?
Machine:	to find out what happens when we get to the planet earth .
Human:	where are you now ?
Machine:	i' m in the middle of nowhere .
Human:	what is the purpose of dying ?
Machine:	to have a life .
Human:	what is the purpose of being intelligent ?
Machine:	to find out what it is .
Human:	what is the purpose of emotions ?
Machine:	i don' t know .

Sordoni 在 DCGM 的评价时采用了 BLEU 和 METEOR 两种自动指标[6]。为了处理庞大而且多样的响应空间，他们用信息检索（Information Retrieval，IR）的方法从数据集中挖掘潜在的合理响应，并让人类评估员对其合适度（Appropriateness）打分，从而构造了一个多重参考评测集（Multiple-Responses Benchmark Dataset）。在这样的评测集上，他们发现 BLEU 对系统的排名和人类评价非常一致。这种构建多重响应测评集的方法也见于 LSDSCC[34] 的测评集的构造方法，以及 DeltaBLEU[29] 指标的设计思路。

Serban 等人在 HRED 模型的评价中使用了困惑度和单词分类错误（Word Classification Error，WCE）[5]。低的困惑度表示模型对数据的概率分布的拟合程度好。Serban 等人认为困惑度是适用的，因为在存在多个合理输出的情况下，困惑度总是衡量生成单一参考输出的概率，这意味着它能可靠的衡量模型的质量。WCE 计算模型的输出中正确预测的单词占整个数据集单词数量的比例，这里的“正确预测”要求单词在句子中的顺序也要正确，所以它是一个比困惑度更严苛的指标。尽管使用了自动化指标，Serban 等人指出：这些指标与他们想要衡量的语法正确度（Grammatical Correctness）和语义连贯性（Semantic Coherence）之间的相关性并不明朗。值得注意的是，他们并没有使用人类

评价。

Serban 等人在 VHRED 模型的评价中主要使用了基于词嵌入的指标[25] 和人类评价。Serban 等人认为，虽然这些基于词嵌入的指标和人类评价的相关度不高，但是它们可以用于测量话题相似性（Topic Similarity），也就是模型的输出和参考输出的语义内容是否相近。模型的输出和参考输出可能没有 n-gram 重叠，但分布式语义可以给出非 0 的相似性。

Vinyals 等人对他们提出的基于 Seq2Seq 的对话系统 [18] 的评估使用了困惑度，定性分析和人类评价。但是，他们认为这些测评方法都存在明显的弊端，而设计出能快速评价对话模型的高质量指标仍有待学界研究。

2.3 生成式对话的数据集

本领域所使用的数据集一般称为开放领域的对话数据集（Open-Domain Conversation Corpus），它和传统的面向任务的数据集相比，话题开放，语法形式多样，而且常带有非自然语言符号（表情符号，URL 等等）[1]。表 2.2 列举了一些常见数据集，大规模是指对话数量在 1M 以上的，中等规模是指对话数量在 50K–1M 之间的[4]。Sina Weibo Corpus 是一个中文数据集，其他都是英文数据集。Twitter 相关的数据集由于隐私保护政策的原因，不能公开原始数据。数据集[52–58] 都具有元信息，如时间戳，对话双方身份信息等等。数据集[6, 34, 58] 有人类标注信息，如情感、话题和方面（Aspect）等。元信息和人类标注信息提供了额外的表征，对构建更智能的对话系统很有帮助。本文把注意力集中在三个数据集上：Ubuntu Dialogue Corpus，OpenSubtitles，LSDSCC，它们代表了三个常见的对话领域，分别是：技术支持，电影字幕和在线论坛讨论。

Ubuntu Dialogue Corpus 是 Lowe 等人从 Freenode IRC 网络的 Ubuntu 板块的聊天日志[5] 中获取的技术性两人多轮对话。这个数据集中包含了大量技术符号，比如路径、命令、URL，还有笔误（Typo），缩写（Abbreviation）和俚语（Slang）。它的对话数量多达 930,000（接近 1M），庞大的数据量为数据驱动的模型提供了极佳的试验场。它的多轮特性为模型提供了更长的上下文，有助于能利用多轮对话的模型生成更有意义的响应。

OpenSubtitles 是 The Open Parallel Corpus，OPUS 项目的一部分。它是从一个人们可以自由上传和下载电影字幕的网站[6]的数据库中获取的，庞大而充满噪音的开放领域数据集。它的对话数量达到了 80M。由于 OPUS 的目的是收集机器翻译所用到的双语文

[4] 1K = 1024，1M = 1024 K，1G = 1024 M。

[5] https://irclogs.ubuntu.com/

[6] http://www.opensubtitles.org

表 2.2 数据集一览

名称	规模	领域	是否公开
Ubuntu Dialogue Corpus[33]	大	技术支持	是
Twitter Corpus[4]	大	短文本多领域闲聊	否
Twitter Triple Corpus[6]	大	Twitter Corpus 的扩展版本	否
OpenSubtitles[19, 59]	大	电影字幕（Subtitles）	是
LSDSCC[34]	中	Reddit 论坛电影板块	是
Supreme Court Corpus[52]	中	美国高等法院辩论	是
Wikipedia Talk Pages Corpus[53]	中	维基百科编辑者在线讨论	是
Tennis Corpus[54]	中	网球比赛赛后新闻发布会	是
Parliament Corpus[55]	中	国会讨论	是
Conversations Gone Awry Corpus[56]	中	维基百科讨论页面吵架集锦	是
Movie Dialogs Corpus[57]	中	电影对白	是
Movie-DiC[60]	中	IMSDB 电影对白	否
MovieTriples[5]	中	Movie-DiC 的扩展版本	否
SubTle[61]	中	电影字幕	否
DailyDialog[58]	中	日常对话	是
Sina Weibo Corpus[62]	中	新浪微博短文本闲聊	是

库（Parallel Corpus），OpenSubtitles 也是一个双语文库，它里面并没有对话数据集所需要的轮换信息（Turn-Taking），而且也没有区分旁白、独白和对白。Sordoni 等人在[18] 中把 OpenSubtitles 用到对话领域，他们把相邻的两个句子视为消息和响应，并且每一个句子既是消息又是响应。Li 等人在[20, 21, 23, 28, 31, 38, 63] 中也采用了类似的办法。尽管充满噪音，OpenSubtitles 仍然是目前最大规模的电影字幕数据集。

LSDSCC，A Large Scale Domain-Specific Conversational Corpus，是一个单一领域的单轮对话数据集[34]。有研究认为，单一领域和专注的话题有助于模型规避单调的响应。它有 738,095 个对话，词汇表比较大，达到了 50K。作者从 Reddit 的电影板块[7]中获取了原始数据，设计了一套尽可能保留语料信息的清理程序，还使用人类评价员对消息和响应的相关度进行评价，最终得到一个高质量的信息-响应对（Query-Response Pair）文库。在构建测试集方面，作者用类似[6] 的 IR 方法构建了一个多重参考评测集。他们还进一步让人类评价员对一个消息的所有响应进行分类，并基于这个测试集提出了三个面向多

[7]https://www.reddit.com/r/datasets

样性的评价指标，不过我们并未使用。

2.4 本章小结

本章从模型、指标和数据集三个方面回顾了本领域的研究成果和现存问题。在模型方面，基于 Seq2Seq 的对话系统能够直接生成响应，能端对端的训练，减少写死的规则（Hard Written Rules），能从大量语料数据中自动学习语言规律，能利用话题和情感等多种表征，具有巨大的优越性。然而，这类系统也有弊端，比如无法准确控制生成的内容，偏向于生成单调的响应，没有人格一致性等等。

在指标方面，对话系统的指标还不能像机器翻译的指标那样准确的评价对话系统。可能最大的问题就是，如何提高指标和人类评价的相关性。导致这个困境的核心原因可能是对话的响应具有固有的语义多样性，这个事实可能导致两个结果：1. 用机器翻译的表征，比如单词层面的 n-gram，字符层面的 n-gram，对齐等等，无法有效捕捉语义层面的信息。2. 单纯使用衡量相似性的指标无法捕捉多样性的维度，从而与人类评价不符。

在数据集方面，本领域已经积累了各种领域（Domain）的数据，从技术性问答、社交媒体闲聊，到电影对白和字幕等等。因为数据集的增多，一些数据集在领域上出现了重合，比如 Movie Dialogs Corpus，MovieTriples 和 Movie-DiC 都是电影对白领域；OpenSubtitles 和 SubTle 都是电影字幕领域，Twitter Corpus 和 Sina Weibo Corpus 都是短文本闲聊等等。尽管人类标注是对话文本的重要补充，但是由于其代价的高昂，所得的数据量往往比较少；另一方面，元信息是一类在数据采集过程中伴随对话文本同时获得的描述性信息，比如对话者的性别（Gender）和角色（Character）等等。相比与人类标注，元信息更容易获取，规模不受限制。并且，人类标注和元信息都有助于模型生成更加多样和连贯的响应 [21, 27, 64]。因此，数据集的收集过程应该充分保留原始数据中的元信息，对话系统应该充分利用元信息。同时，还应该发展从对话文本中无监督的提取元信息的方法。

graphicx

3 研究方法

3.1 实验框架

目前，学界普遍认为自动评价指标和人类评价没有很强的相关性[8]，但是不同指标之间的一致性似乎还没有得到充分的研究。另一方面，生成式对话系统在不同数据集上的迁移能力（Transferability）似乎是一个研究的比较少的问题。开放领域的对话数据集所具有的大量噪音，多样的话题和较弱的语法正确度对模型质量的影响似乎还没有引起学者们的重视。通过实验，我们试图对上述问题进行初步的考察，具体来说，我们试图回答以下问题：

1、 模型的性能在不同数据集之间能否保持而没有大幅度下降？

2、 不同的指标在评价同一个数据集上的模型时有没有一致性？

3、 模型和数据集，哪一个对得分的影响较大？

我们的实验涉及在多个数据集上训练多个模型，然后用多种指标测量句子层面得分和系统层面得分。表 3.1 展示了实验所使用的模型，数据集和指标。记模型的集合为 M，数据集的集合为 D，指标的集合为 S，为了避免术语上的模糊，我们指明：一个模型 $m \in M$ 指的是一种生成式模型的体系结构，而不是指一个训练好的模型实例。一个数据集 $d \in D$ 是指某个领域的全部对话的一个子集，它本身又被分为训练集，验证集和测试集三个子集，一个指标 $s \in S$ 是指一种能把上下文 c，参考 r 和响应 $\hat{r}$ 映射为一个实数的函数 $f_s(c, r, \hat{r})$。在不引起歧义时，在数据集 d 上训练指的是在 d 的训练集上训练，在数据集 d 上测试指的是在 d 的测试集上测试。把模型 m 在数据集 d 上训练的实例记为 (m, d)。

如图 3.1 所示，我们的实验首先让每一个模型 m 和数据集 d 的组合 $(m, d) \in M \times D$ 在训练集上进行训练，并让训练好的模型实例 (m, d) 在测试集上解码产生响应 r，然后再用每一个指标 $s \in S$ 给 r 在句子层面打分，给 (m, d) 在系统层面打分，分别得到 λ_u 和 λ_s。一组实验的最终结果是一个 5 元组 $(m, d, s, \lambda_u, \lambda_s)$，表示在数据集 d 上训练的模型 m 在指标 s 的评价下所得的句子层面得分 λ_u 和系统层面得分 λ_s。

我们用 pandas[1]承载实验数据，用 seaborn[2]进行数据可视化。系统层面得分的数据由三个类别变量（Categorical Variable）：模型，数据集和指标和一个数值变量（Numerical

[1]https://pandas.pydata.org/

[2]http://seaborn.pydata.org/

Variable)：得分组成，因此我们采用面向类别变量的柱形图（Bar Plot）和箱体图（Box Plot）作为分析方法。柱形图最大程度保留了所有数据的细节，展示了不同模型在不同数据集和指标上的系统得分，是最为精确的表达，我们在正文对它进行了详细分析。但是，某些模型在某些数据集上的得分过于接近，柱形图难以区分各个模型得分的高低。于是我们分别从数据集和模型两个维度绘制了箱体图，牺牲了一些精确性，但是加强了得分在某个维度上的区分程度。这些箱体图作为辅助分析方法，放在附录 A。

句子层面的得分的数据主要由一个数值变量：得分组成，因此我们采用面向数值变量的分布图（Distribution Plot）进行分析。一组待分析的句子层面得分是一个 n 维实值向量：$U_{(m,d,s)} = x_1, x_2, \ldots, x_n$，$n$ 是测试集的样本数，(m, d, s) 是模型，数据集和指标组成的三元组，x_i 是某个样本的得分。这个实值向量可以看做是从一个总体的取样，由于总体服从的分布的数学形式是未知的，我们使用无参估计法之一的核密度估计（Kernel Density Estimation，KDE）来估计总体的分布。由于句子层面得分比系统层面得分的粒度更细，数据也更多了，为了分析的效率，我们选择了案例分析，固定 $m = m', d = d'$，而分析了所有指标在 (m', d') 上的情况。我们在附录 B 呈现了完整的数据。

表 3.1　实验对象一览

模型	HRED，LSTM，VHRED
数据集	Ubuntu，OpenSubtitles，LSDSCC
指标	BLEU，ROUGE，METEOR，Vector-Average，Vector-Extrema，Greedy-Matching，ADEM，PPL，Distinct-N

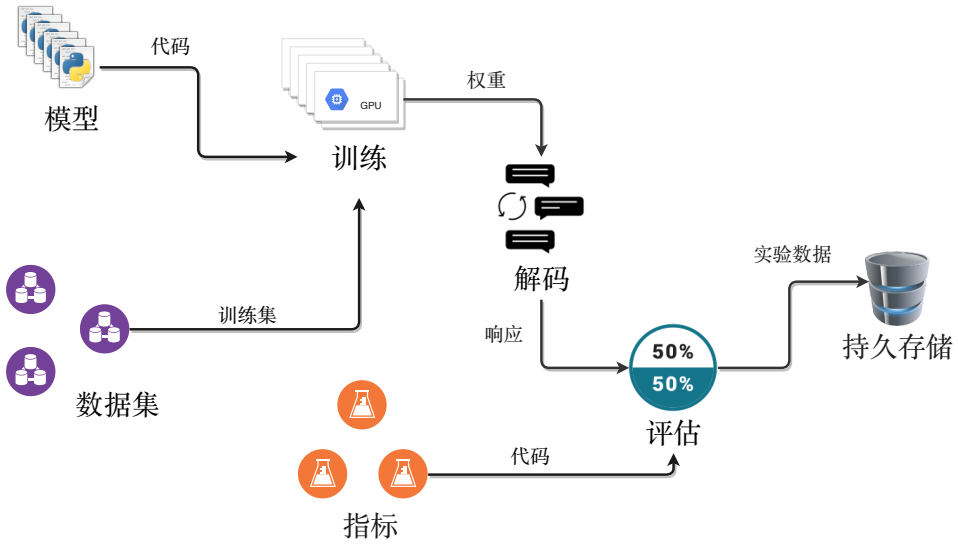


图 3.1　实验框架

3.2 模型选取

第 2 章已经介绍了生成式模型的基本概念和数据集、指标的基本情况。和相对有限的数据集和指标相比，生成式模型的数量众多，光是我们了解到的不同的模型就有不下十几个。由于时间有限，我们把考察的范围限定在基于 Seq2Seq 框架的生成式模型。Serban 等人在[5, 25, 26, 65] 中提出或者使用了 LSTM，HRED，VHRED 和 MrRNN 等模型，它们大多是 Seq2Seq 框架在体系结构上的扩展。除了 LSTM 是作为基线的 RNN 语言模型外，这些模型都各具特色：HRED 能利用长期对话历史，VHRED 能捕捉对话中的不确定性（Uncertainty）和歧义性（Ambiguity），MrRNN 能生成带有高级组合结构（Compositional Structure）的响应。而且，HRED 和 VHRED 都在 Ubuntu Dialogue Corpus 和 Twitter Triple Corpus 两个数据集上取得了不错的成绩，说明它们在迁移能力方面是比较强的基线。是否基于 Seq2Seq 框架和迁移能力是我们选择模型的两大依据，HRED 和 VHRED 很好的符合了我们的需求。

我们没有选取普遍的 Seq2Seq 模型作为基线，而是选取了 LSTM 语言模型，因为它足够简单而且是一个非 Seq2Seq 结构，作为基线模型已经足够好了。我们把加入普通的 Seq2Seq 模型作为以后的工作。下面将会详细介绍 LSTM，HRED 和 VHRED 三个模型。

3.2.1 LSTM 语言模型

LSTM 是一种改进的循环神经网络，用于解决长时间依赖问题[16]。LSTM 用多种“门”代替了普通 RNN 中单一的隐藏矩阵 W_{hh}，这些门实际上是一个个单层神经网络。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t \tag{3.4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t \times \tanh(C_t) \tag{3.6}$$

在公式 3.6 中，x_t 是 t 时刻的输入，h_{t-1} 是 $t-1$ 时刻的隐藏状态，f_t 是遗忘门（Forget Gate）的输出，i_t 是输入门（Input Gate）的输出，$\hat{C}_t$ 是候选单元门（Candidate Cell Gate）的输出，o_t 是输出门（Output Gate）的输出；W_f, W_i, W_C, W_o 是四个门的权重矩阵，b_f, b_i, b_C, b_o

是四个门的偏置向量，h_t 是通过一系列运算后得出的 t 时刻的隐藏状态。图 3.2 直观的展现了 LSTM 内部各个门的连接方式和数据的流动，LSTM 正是通过这些复杂的门运算实现对较长的序列的记忆。

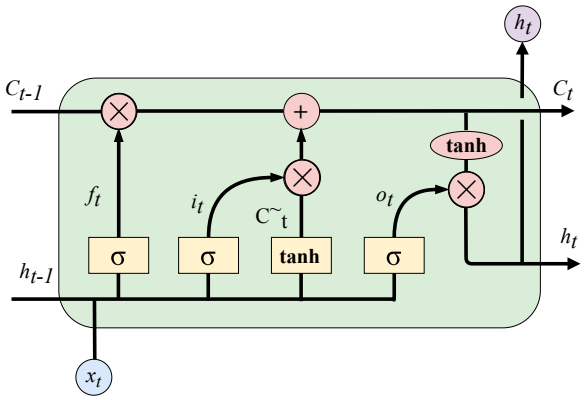
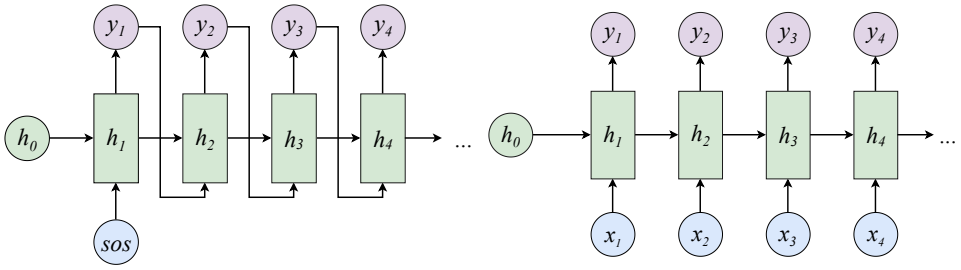


图 3.2 LSTM 内部结构图

如图 3.3 所示，作为语言模型的 LSTM 与作为生成式模型的 LSTM 在数据的输出输出方式上有所不同。作为语言模型时，输入数据为一个序列 X，模型需要重建概率 $p(X)$，在 t_0 时刻，模型输入一个特殊的句子开始符号（Start-of-Sentence），在 t 时刻，模型的输入 x_t 是 $t-1$ 时刻的输出 y_{t-1}，模型的输出 y_t 将和 X 序列的第 t 个元素 X_t 对比，并进行梯度下降。作为生成式模型时，输入数据是一对序列 (X, Y)，模型需要重建条件概率 $p(Y|X)$，在 t 时刻，模型的输入是 X 序列的第 t 个元素，模型的输出 y_t 将和 Y 序列的第 t 个元素 Y_t 对比，并进行梯度下降。我们的实验使用的是作为生成式模型的 LSTM。



(a) 语言模型 (b) 生成式模型

图 3.3 LSTM 模型的两种输入输出方式

3.2.2 HRED 模型

多层编解码器（Hierarchical Recurrent Encoder-Decoder，HRED）[24] 是一种能利用多轮对话结构的生成式模型。它把一个对话看做一个两层序列结构，一个对话 D 由 M 个句子组成：$D = \{U_1, \ldots, U_M\}$，每个句子由 N_m 个单词组成：$U_m = \{w_{m,1}, \ldots, w_{m,N_m}\}$。

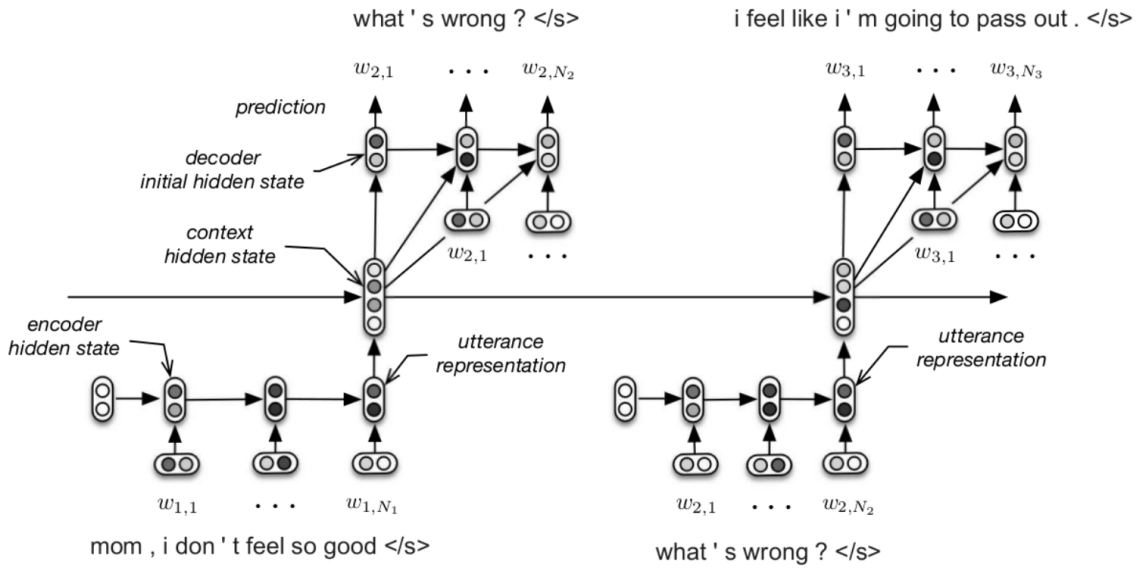


图 3.4 HRED 的计算图[5]

如图 3.4 所示，HRED 由三个部件组成：句子编码器（Utterance Encoder）负责把每一个句子编码成一个固定长度的句子向量（Utterance Vector）e_u，上下文编码器（Context Encoder）负责把 m 个句子向量 $e_{u,1},\ldots,e_{u,m}$ 编码为一个对话向量（Dialogue Vector）e_d，最后，句子解码器（Utterance Decoder）以对话向量 e_d 为输入，生成对话的下一个句子 U_{m+1}。本质上，通过把对话分解为句子的序列，把句子分解为单词的序列，HRED 估计了一个对话 D 的概率 $P_\theta(D)$：

$$P_\theta(D) = P_\theta(U_1,\ldots,U_M) = \prod_{m=1}^{M} P_\theta(U_m|U_{<m}) \tag{3.7}$$

$$= \prod_{m=1}^{M}\prod_{n=1}^{N_m} P_\theta(w_{m,n}|w_{m,<n},U_{<m}) \tag{3.8}$$

θ 是 HRED 模型的参数，$U_{<m}$ 表示 U_m 之前的句子序列，$w_{m,<n}$ 表示第 m 个句子的第 n 个单词之前的单词序列。

3.2.3 VHRED 模型

VHRED（Latent Variable Hierarchical Recurrent Encoder-Decoder）是 HRED 的扩展，它在句子解码器中加入了潜随机变量。

$$P_\theta(z_n|w_1,\ldots,w_{n-1}) = \mathcal{N}(\mu_{\text{prior}}(w_1,\ldots,w_{n-1}),\Sigma_{\text{prior}}(w_1,\ldots,w_{n-1})) \tag{3.9}$$

$$P_\theta(w_n|z_n,w_1,\ldots,w_{n-1}) = \prod_{m=1}^{M_n} P_\theta(w_{n,m}|z_n,w_1,\ldots,w_{n-1},w_{n,1},\ldots,w_{n,m-1}) \tag{3.10}$$

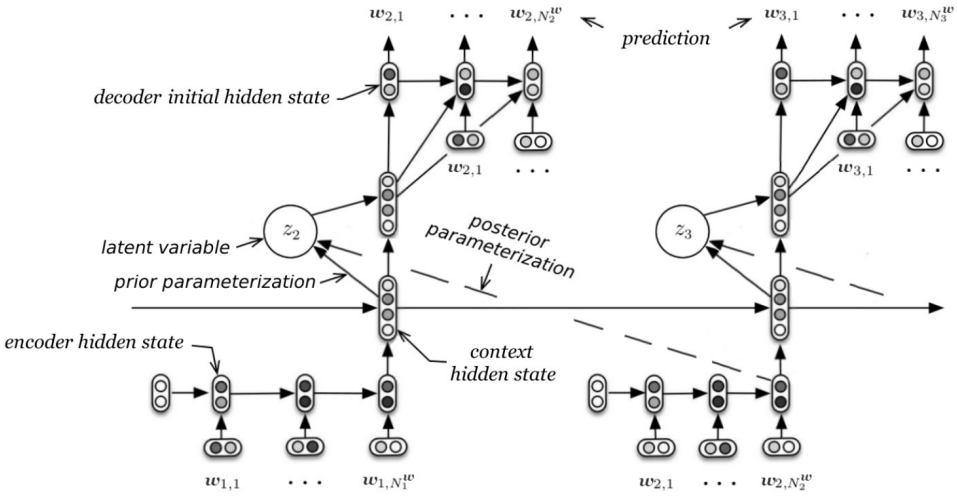


图 3.5 VHRED 的计算图[25]

3.3 模型超参数设置

在模型超参数方面，我们的设置大致和[25] 相同。我们用 Adam[66] 来优化所有模型，一个小批（Mini-batch）处理 20 个样本。我们在 Ubuntu 上使用维度是 300 的词嵌入，在 OpenSubtitles 和 LSDSCC 上使用维度是 400 的词嵌入。HRED 和 VHRED 的句子编码器（Utterance Encoder）在 Ubuntu 上使用 500 个隐藏单元，在 OpenSubtitles 和 LSDSCC 上使用 1000 个隐藏单元[3]。HRED 和 VHRED 的上下文编码器（Context Encoder）在所有数据集上都使用 1000 个隐藏单元。不同模型的句子解码器（Utterance Decoder）在不同数据集上的隐藏单元数量如表 3.2b 所示。一般根据数据集的特点来设置不同 RNN 的隐藏单元数量，在样本数多或者词汇表大的数据集上训练时，隐藏单元数量会相应的增多。

[3]LSTM 模型是一个语言模型，只有句子解码器。

Ubuntu 上的模型的句子编码器都使用了单向 RNN，而 OpenSubtitles 和 LSDSCC 上的模型的句子编码器则使用双向 RNN。所有模型的句子编码器和上下文编码器（如果有）都使用 GRU 作为门单元。句子解码器的门单元类型如表 3.2a 所示 [4]。

表 3.2 句子解码器的配置情况

(a) 门单元类型

	HRED	LSTM	VHRED
LSDSCC	LSTM	GRU	GRU
OpenSubtitles	LSTM	GRU	GRU
Ubuntu	LSTM	LSTM	LSTM

(b) 隐藏状态单元数量

HRED	LSTM	VHRED
1000	2000	1000
1000	2000	1000
500	2000	500

所有的模型都在一台 GeForce GTX TITAN X 上训练了至少 1 周。模型收敛时的困惑度如表 3.3 所示。与[25] 不同的是，我们没有用预训练的 HRED 的参数来初始化对应的 VHRED。所有的模型都使用了梯度剪裁，阈值为 1。所有模型在 Ubuntu 上的学习率为 0.0002，在 OpenSubtitles 和 LSDSCC 上的学习率为 0.0001。在解码时，我们使用随机取样。

表 3.3 模型收敛时的困惑度

	HRED	LSTM	VHRED
LSDSCC	32.9229	32.5599	37.7149
OpenSubtitles	41.6392	34.2724	33.6867
Ubuntu	39.1623	46.4055	40.2486

3.4 数据集预处理

我们使用的 Ubuntu Dialogue Corpus 直接来自 Serban 等人的项目 [5]，没有经过任何额外处理。我们从 Li 等人的项目 [6] 中获取了经过预处理的 OpenSubtitles 数据，并作了如下处理：

1、 将其从整数的下标形式还原为字符串的单词形式；

[4]OpenSubtitles 和 LSDSCC 上的 LSTM 模型的解码器都没有使用 LSTM 门单元，不过它们仍然是语言模型。

[5]https://github.com/julianser/hed-dlg-truncated.git

[6]https://github.com/jiweil/Neural-Dialogue-Generation.git

2、 将其词汇表文件 movie_25000 转化为下标从 0 开始的 pickle[7]格式。

3、 用 Serban 等人的 convert_text2dict.py 将训练集，测试集和开发集均转换为 pickle 格式；

4、 选用 OpenSubtitles 中的轮数为 3，句子最短长度为 6 的 dialogue3_6 格式作为实际使用的数据集；

5、 从测试集随机抽取 1% 的样本作为正式使用的数据集。

LSDSCC 也是一个经过预处理的数据集[8]。由于它的词汇数量多达 50K，为了使内存不至于溢出，我们将其词汇表裁剪至 35000 个最常见的单词。此外，由于我们只使用单个参考的指标，因此对 LSDSCC 的测试集中的多个参考，我们只取第一个参考。剩下的处理过程类似 OpenSubtitles 的处理。

表 3.4a 是三个数据集的训练集的一些统计数据。从表中可以看到，OpenSubtitles 的样本数量要远远超过其他两个数据集，是 Ubuntu 的 26 倍，是 LSDSCC 的 16 倍。但是，从单词数量来看，OpenSubtitles 超过另外两个数据集的倍数却小得多，这是因为 Ubuntu 的每一个样本包含了多个句子，其长度要远远超过 OpenSubtitles 的样本。这也导致了 Ubuntu 的样本数量少于 LSDSCC，但是单词数量却超过了 LSDSCC。测试集的统计数据如表 3.4b 所示。对于 LSDSCC 的测试集，我们其实可以从其训练集分一部分出来作为测试集，这样可以使样本数量更多。但是，为了是训练集尽可能大，我们选择了从 LSDSCC 的多重参考测评集中构造测试集，导致测试集的样本只有不到 300 个。

表 3.4 数据集的统计数据

(a) 训练集

数据集	词汇数量	样本数量	单词数量	轮数
Ubuntu	20000	448833	45697699	多轮
OpenSubtitles	23876	11771393	379346841	多轮
LSDSCC	35008	738095	32355628	单轮

(b) 测试集

样本数量	单词数量
18920	2045082
14714	474074
299	10914

3.5 指标配置

我们尽可能使用业界公认的指标实现和推荐的指标参数。和[8] 一样，我们只考虑单轮对话，并且每个系统输出响应都只有单个参考响应。我们使用 NLTK[9]提供的 BLEU

[7]pickle 是一个 Python 特有的序列化格式

[8]https://drive.google.com/file/d/1nbpbnhwNP14xAc4SAc1NN5lvEr01dQb/view?usp=sharing

[9]https://www.nltk.org/

实现，并且加入了平滑处理。我们实现了的 ROUGE 的一个版本，并将准确率和召回率的比例设置为 1:9，因为在机器翻译领域，召回率和人类评价的相关性比准确率高[41]。ROUGE-W 的权重设置为 1.2 。我们使用了 METEOR 的官方实现 meteor-1.5.jar，并且按照官方文档 [10]的说明，加入对特定语言（英语）的正则化处理。我们使用 Serban 等人的 evaluate.py 脚本测量模型的困惑度，由于该程序对测试样本进行了随机取样，我们无法得知某个样本的确切得分，所以只测得了系统层面的得分。关于词嵌入的指标，我们改写了 Serban 等人的 embedding_metrics.py，使之能测量句子层面的得分；和他们一样，我们使用在谷歌新闻文库（Google News Corpus）上预训练的 word2vec 词嵌入[11]。关于 ADEM 指标，我们使用作者提供的代码库 [12] 和预训练模型[13]。我们实现了比较简单的 Distinct-N 指标。我们还测量了响应的句子长度 *#words*，以观察它对不同指标的得分的影响。

3.6 本章小结

本章详细介绍了实验的各项设置，包括模型的超参数和训练过程，数据集的预处理操作和一些统计数据，以及指标的实现和参数选择。在下一章中，我们将展示实验的结果并进行讨论。

[10]http://www.cs.cmu.edu/~alavie/METEOR/README.html

[11]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM

[12]https://github.com/mike-n-7/ADEM.git

[13]https://drive.google.com/file/d/0B-nb1w_dNuMLY0Fad3N1YU9ZOU0/view?usp=sharing

4 实验结果与讨论

本章从系统层面得分和句子层面得分两个方面展示了实验的数据与结论。在众多数据中，我们发现以下主要的结论：

1、 同一个数据集上的模型的得分差异较小，不同数据集上的模型得分差异较大。
2、 各个指标的分布情况呈现集群现象，同一集群内的指标分布相似，不同集群之间的指标分布迥异。
3、 尽管在某些指标或数据集上一个模型超过另一个模型，但是在全部指标和数据集上，没有哪个模型一致的超过另一个模型。

我们把上述结论归因于以下几个方面：

1、 合理的响应的空间巨大，响应具有很高的熵，给评价增加了难度。
2、 模型在不同数据集上的泛化能力有待加强。
3、 对话数据集的质量参差不齐，而模型的质量和数据集的质量紧密相关。
4、 指标捕捉了错误的表征，不能正确反映模型的性能。

4.1 系统层面得分

表 4.1 展示了各个模型在不同数据集上测定的各项指标的系统层面得分，加粗的得分是三个系统中的最优者。句子平均长度（#words）没有加粗，因为它是一个参考数据。从表中的数据来看，在 LSDSCC 上 HRED 取得了除 ROUGE-4 外所有指标的最优，在 OpenSubtitles 上，VHRED 取得了除了 ADEM 和 Distinct-N 之外所有指标的最优，在 Ubuntu 上情况比较复杂：HRED 取得 9 个指标的最优，LSTM 取得了 6 个指标的最优，VHRED 取得了 3 个指标的最优。如果我们把在一个数据集上取得最优指标最多的模型称为在该数据集上的最优模型的话，从总体上看，HRED 是所有数据集上的最优模型，因为它在 LSDSCC 和 OpenSubtitles 都是最优模型。但是观察在每一项指标上取得最优的模型时我们发现，不同数据集上的取得最优的模型往往不相同。为了了解不同模型在不同数据集上的表现，我们对每一个指标的系统层面得分绘制了柱形图。

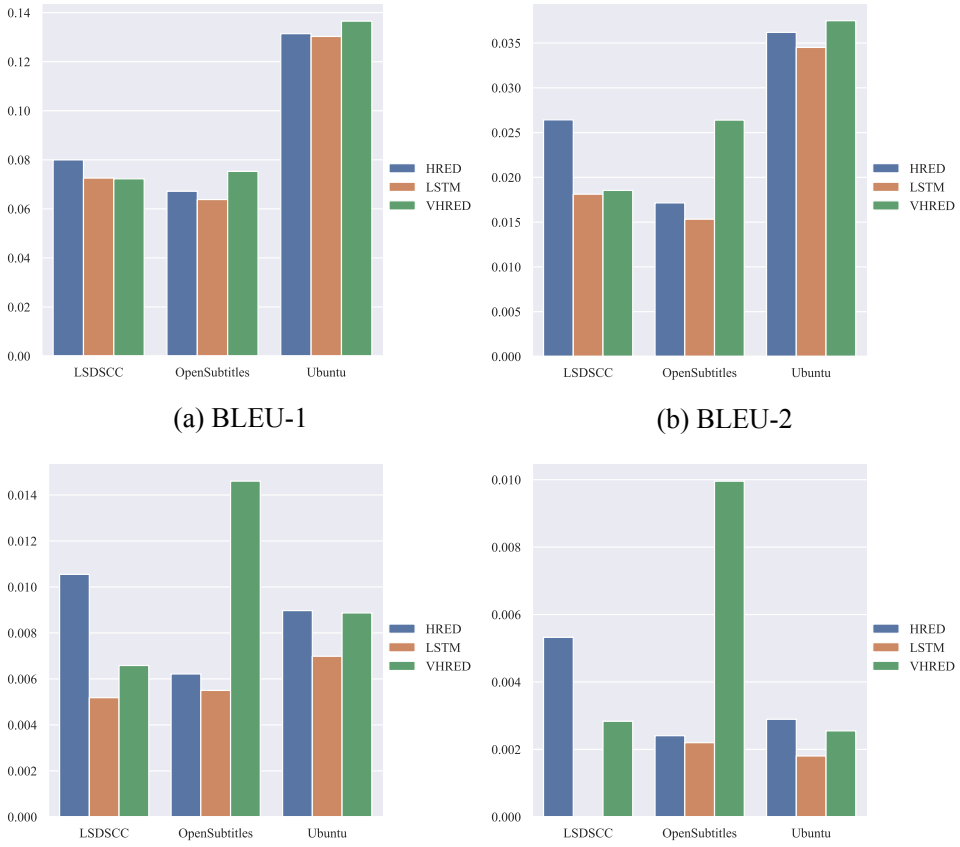
表 4.1 不同数据集上的模型的各种指标得分

	LSDSCC			OpenSubtitles			Ubuntu		
	HRED	LSTM	VHRED	HRED	LSTM	VHRED	HRED	LSTM	VHRED
ADEM	**2.6178**	2.6127	2.6163	**2.6228**	2.6224	2.6219	2.6353	**2.6381**	2.635
BLEU-1	**0.08**	0.0726	0.0722	0.0672	0.0638	**0.0753**	0.1314	0.1303	**0.1365**
BLEU-2	**0.0264**	0.0181	0.0185	0.0171	0.0153	**0.0264**	0.0362	0.0345	**0.0375**
BLEU-3	**0.0105**	0.0052	0.0066	0.0062	0.0055	**0.0146**	**0.009**	0.007	0.0089
BLEU-4	**0.0053**	0.0	0.0028	0.0024	0.0022	**0.01**	**0.0029**	0.0018	0.0025
Distinct-1	**0.9577**	0.9441	0.9558	**0.973**	0.9714	0.9714	0.9074	**0.9257**	0.9113
Distinct-2	**0.8541**	0.8511	0.8497	**0.8669**	0.8594	0.8665	**0.9013**	0.8603	0.8968
Greedy	**0.3303**	0.3292	0.3267	0.3102	0.2998	**0.3145**	**0.2775**	0.2364	0.273
Average	**0.5532**	0.5467	0.5483	0.5453	0.5295	**0.5485**	**0.574**	0.5205	0.5655
Extrema	**0.2841**	0.2835	0.2814	0.3009	0.2929	**0.3061**	**0.29**	0.2663	0.2875
METEOR	**0.0296**	0.0258	0.0281	0.0248	0.0233	**0.0271**	0.1657	0.1635	**0.166**
ROUGE-1	**0.108**	0.083	0.0978	0.0784	0.075	**0.0872**	0.1644	**0.1836**	0.1683
ROUGE-2	**0.0226**	0.0049	0.0081	0.0053	0.0043	**0.0107**	0.0128	**0.0143**	0.0128
ROUGE-3	**0.0057**	0.0002	0.0035	0.0011	0.0009	**0.0053**	**0.0007**	0.0003	0.0005
ROUGE-4	0.0011	0.0	**0.003**	0.0002	0.0002	**0.0038**	**0.0002**	0.0	0.0001
ROUGE-L	**0.0956**	0.0681	0.0846	0.0742	0.0707	**0.0826**	0.1493	**0.1722**	0.1535
ROUGE-W	**0.0792**	0.0537	0.07	0.066	0.0629	**0.0734**	0.1205	**0.1391**	0.1236
PPL	**32.5599**	32.9229	37.7149	41.6392	34.2724	**33.6867**	**39.178**	46.4061	40.2641
#words	13.1605	14.0067	12.3612	8.807	8.6394	8.7798	23.0646	16.4905	21.2449

图 4.1 是 BLEU 的系统得分柱形图。BLEU-1 和 BLEU-2 的情况比较类似：从总体上看，模型在 Ubuntu 的得分普遍高于在另外两个数据集上的得分，而 HRED 和 VHRED 在所有数据集上都比 LSTM 表现好，在 Ubuntu 上，VHRED 超过了 HRED，但在另外两个数据集上，HRED 都一致的超过了 VHRED。模型的 BLEU-1 得分在虽然在不同数据集之间相差较大，但在同一个数据集内却相差不大。然而，模型的 BLEU-2，BLEU-3，BLEU-4 得分即便在同一个数据集内也相差很大。

BLEU-3 和 BLEU-4 的情况比较类似：从总体上看，没有哪一个数据集的得分要普遍高于其他数据集上的得分。在所有数据集上，HRED 和 VHRED 的得分仍然普遍比 LSTM 高，虽然有时候优势不太明显。VHRED 在 OpenSubtitles 上大幅度的超过了 HRED，但在另外两个数据集上，HRED 都超过了 VHRED。从绝对数量的角度来看，BLEU 的得分

随着 N 的增大而减小，这可能是因为两个句子的 n-gram 共现数随着 *N* 的增大而减少。



(a) BLEU-1　　(b) BLEU-2

(c) BLEU-3　　(d) BLEU-4

图 4.1　BLEU 的系统得分

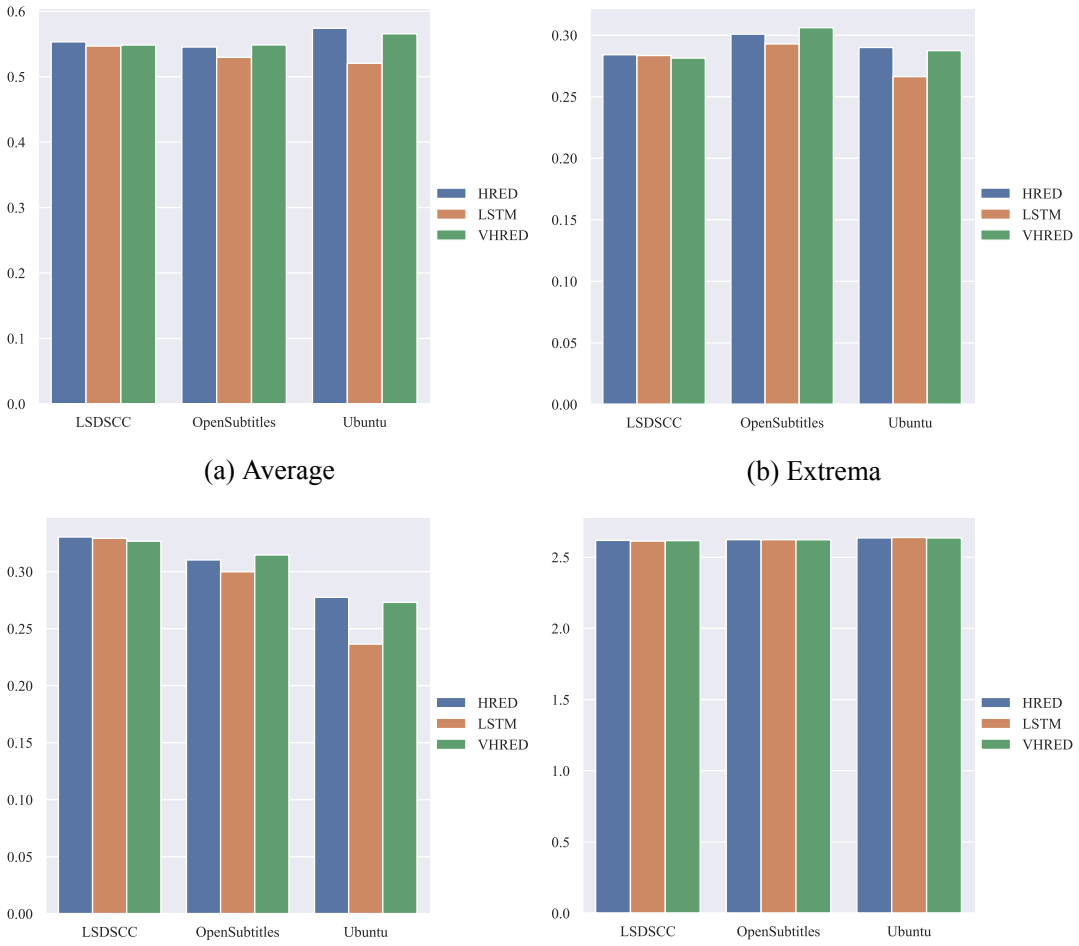
图 4.2 是基于词嵌入的指标和 ADEM 的系统得分柱形图。有趣的是，ADEM 对所有数据集上的所有模型的打分都非常接近 2.6。从表 4.2 看出，当数据集不同时，ADEM 的分数在百分位变化，当数据集相同而模型不同时，ADEM 的分数在千分位或之后变化，这表明数据集对 ADEM 得分的影响要大于模型的影响。从数据集的角度，所有模型在

表 4.2　ADEM 的系统层面打分

	HRED	LSTM	VHRED
LSDSCC	2.6178	2.6127	2.6163
OpenSubtitles	2.6228	2.6224	2.6219
Ubuntu	2.6353	2.6381	2.635

Ubuntu 上的得分好于在 OpenSubtitles 的得分，而后者要好于在 LSDSCC 上的得分。从

模型的角度，在 LSDSCC 上，HRED 的得分好于 VHRED 的得分，后者要好于 LSTM 的得分。在 OpenSubtitles 上，HRED 的得分好于 LSTM 的得分，后者要好于 VHRED 的得分。在 Ubuntu 上，最好的模型是 LSTM，其次是 HRED，最后是 VHRED。从上述分析来看，HRED 在两个数据集上得分最优，VHRED 在两个数据集上得分最差，LSTM 在三个数据集上排名各不相同。尽管得分的差异很小（最大值和最小值之间只相差 0.019），而且各个数据集上模型的排名都不相同，但是还是可以看出，HRED 在 ADEM 指标上表现较好，而 VHRED 则较差。这可能是因为我们没有用预训练的 HRED 初始化 VHRED。



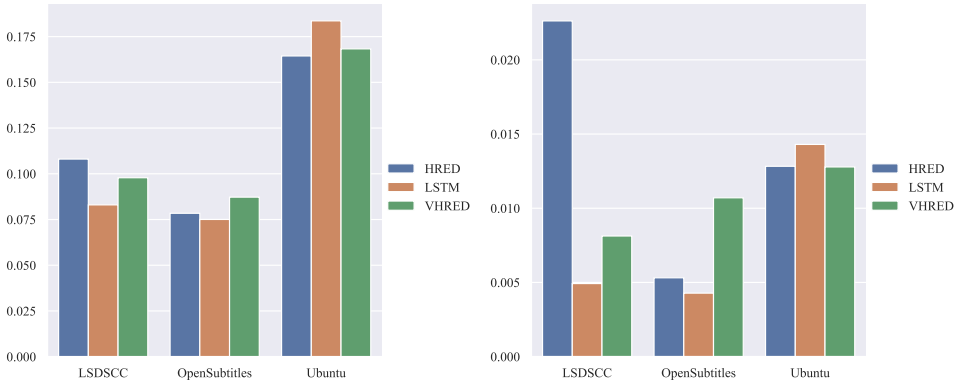
(a) Average　　(b) Extrema

(c) Greedy　　(d) ADEM

图 4.2　基于词嵌入的指标和 ADEM 的系统得分

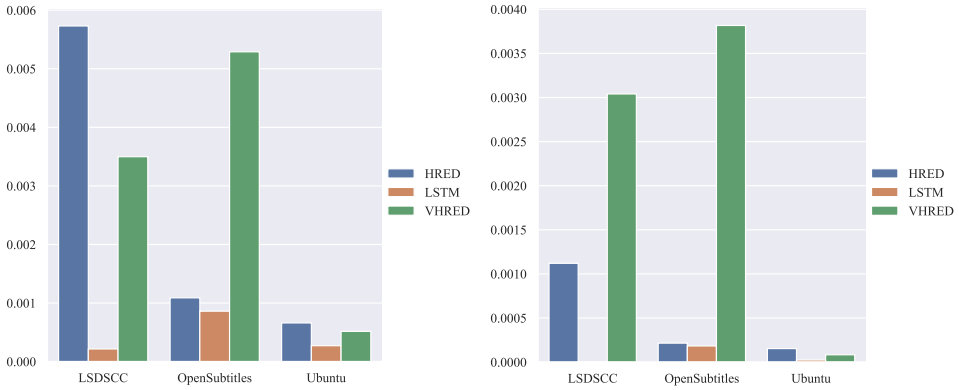
和 ADEM 相比，基于词嵌入的指标表现的更具有一致性。Average，Greedy 和 Extrema 三种指标虽然在绝对数值上有些差异，但是图像的模式却非常相似。从总体来看，在所有数据集和指标上，HRED 和 VHRED 都比 LSTM 表现的好。我们发现在 LSDSCC 上，这三个指标很难将模型区分开来，事实上，所有模型在 LSDSCC 上的某个指标都倾向

于得到相同的分数。我们猜测这是因为 LSDSCC 的测试集样本太少（300 个），没有足够多的样本令模型之间的差异展现出来。在测试集样本数比较多的 OpenSubtitles 上，三个模型的区别比较明显，但是由于 OpenSubtitles 的噪音比较大，话题比较分散，模型相对比较容易产生话题相关的响应，导致三个模型的区分不是特别明显。在 Ubuntu 上，三个模型的区分最明显，HRED 和 VHRED 超过了 LSTM，拉开了较大距离。我们猜测这是因为 Ubuntu 是一个技术领域的数据集，含有大量技术词汇，如 apt-get，java 等等。模型必须能捕捉到消息中的技术相关的语义并生成相关的句子才能的高分，这要求模型对消息的主题有很强的捕捉能力。HRED 和 VHRED 比 LSTM 多了编码器结构，可以肯定它们捕捉消息主题的能力更强。此外，模型在三个数据集上的区分度的不同也和数据集的对话轮数有关，LSDSCC 是单轮对话，OpenSubtitles 是 3 轮对话，而 Ubuntu 是多轮对话。多轮对话数据集更有利于能够利用它们的 HRED 和 VHRED。



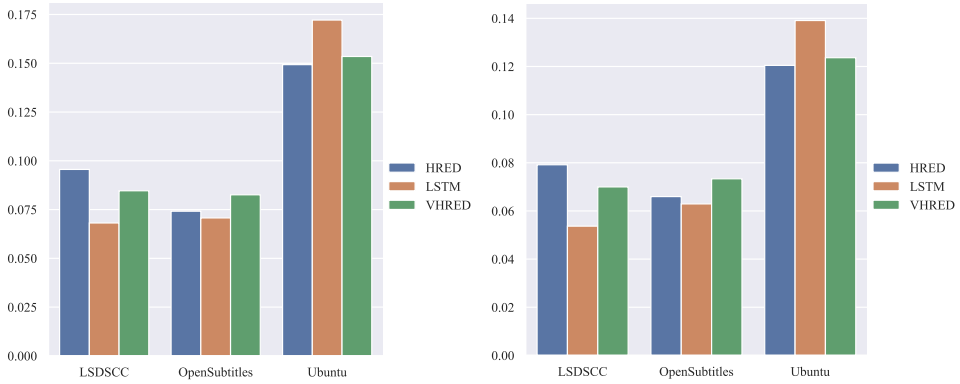
(a) ROUGE-1

(b) ROUGE-2



(c) ROUGE-3

(d) ROUGE-4



(a) ROUGE-L　　(b) ROUGE-W

图 4.4　ROUGE 的系统得分

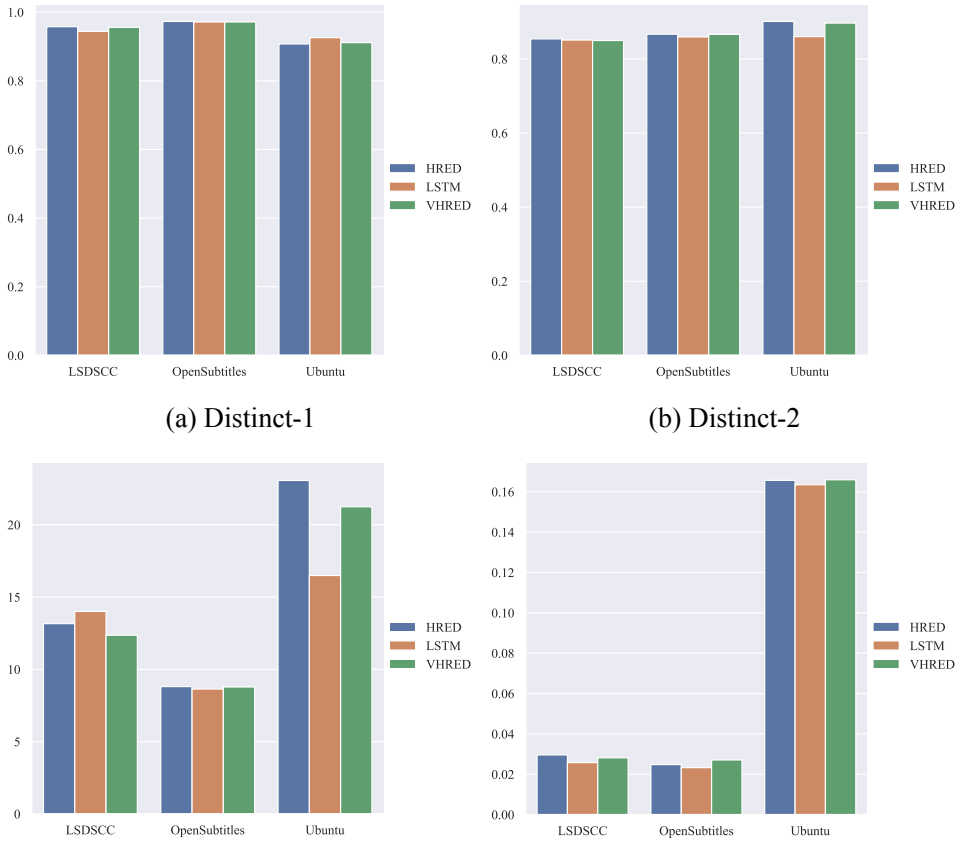
图 4.4 是 ROUGE 的系统得分柱形图。从总体上看，不同数据集上的模型在 ROUGE-1，ROUGE-2，ROUGE-L 和 ROUGE-W 的得分都不为 0，但在 ROUGE-3 和 ROUGE-4 上，多个模型的得分接近 0，这是正常的，因为一般来说，响应和参考之间的高阶 n-gram 重叠非常少。ROUGE-1，ROUGE-L 和 ROUGE-W 的图像非常相似。在这三个指标上，在 LSDSCC 和 OpenSubtitles 上，HRED 和 VHRED 超过了 LSTM，但在 Ubuntu 上，LSTM 却超过了 HRED 和 VHRED；除此之外，各个数据集内三个模型的排名也基本一致，但是模型之间的差距有时很大，比如 LSDSCC 上的 ROUGE-2，有时很小，比如 OpenSubtitles 上的 ROUGE-1。

在 ROUGE-3 和 ROUGE-4 上，LSTM 的得分在所有数据集上普遍很低，而 HRED 和 VHRED 在某些数据集上得分也很低，在 Ubuntu 上，所有模型的得分都很低。令人注目的是，VHRED 在 OpenSubtitles 和 LSDSCC 上都取得了不错的成绩，而 HRED 的表现在 LSDSCC 上较好，在 OpenSubtitles 上较差。

图 4.5 是 Distinct-N，METEOR 和句子长度的系统得分图。在 Ubuntu 上的模型的句子长度明显大于在 LSDSCC 上的模型，而后者大于在 OpenSubtitles 上的模型。从第 3.4 节我们知道，训练集的平均句子长度有 Ubuntu 大于 LSDSCC 大于 OpenSubtitles。模型生成的响应的长度和训练集的平均句子长度具有一致性，这是因为训练的目标函数是最大化训练集样本的对数概率，所以模型倾向于模仿训练集的统计特征。

Distinct-N 对模型的区分度不高，在同一个数据集上训练的模型，它们的 Distinct-N 都比较接近。从数据集的角度，OpenSubtitles 上的模型的 Distinct-1 较高，而 Ubuntu 上的模型的 Distinct-2 较高。METEOR 似乎受句子长度的影响较大，在句子长度较大的 Ubuntu 上，模型的 METEOR 得分要比其他数据集上的得分高得多。而 LSDSCC 上的

METEOR 也比 OpenSubtitles 上的得分略高。METEOR 对不同模型的区分度不是很大，但从总体上看，HRED 和 VHRED 比 LSTM 的表现要好。



(a) Distinct-1 (b) Distinct-2

(c) #words (d) METEOR

图 4.5 Distinct-N，#words 和 METEOR 的系统得分

本节利用柱状图分析了模型，数据集和指标的系统层面得分。在一些图中，各个模型或者数据集的得分差异不十分明显，为了突显得分的差异，我们把观察的维度分成数据集和模型两个维度，并分别绘制了箱体图。通过箱体图得出的结论是相同的，但是由于图像的数量较多，所以放在了附录。附录 A 从数据集的维度和模型的维度展示了不同指标的系统层面得分。

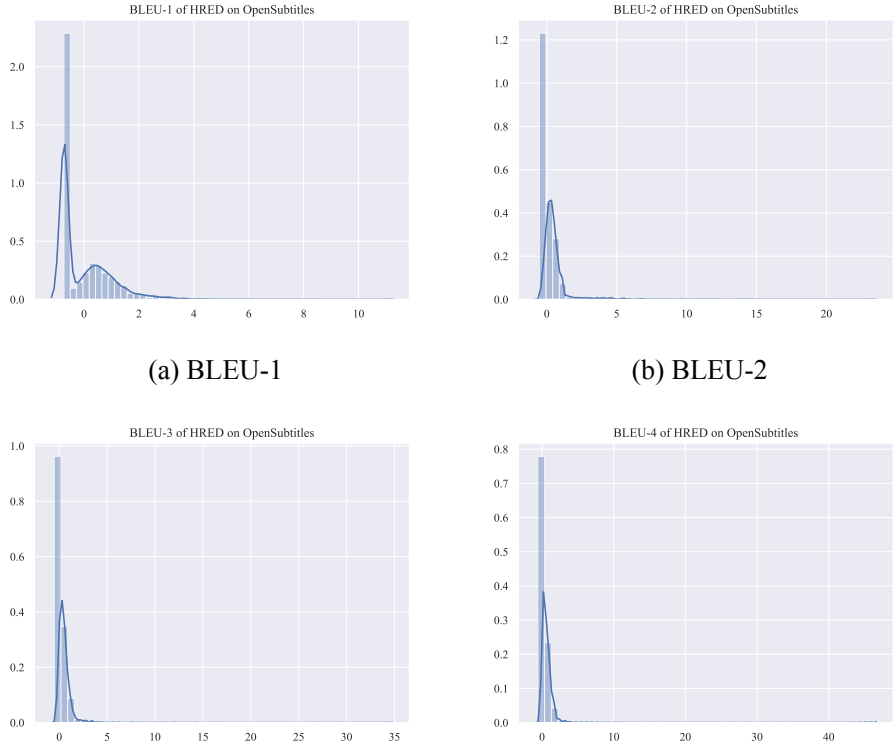
4.2 句子层面得分

我们还从句子层面得分方面进行了分析。因为模型生成的句子 u 可以看做一个句子空间 U 上的随机变量，而指标是一个确定的函数 f_s，所以可以把句子层面的得分看做一个随机变量 $\lambda_u = f_s(u)$。于是我们便可以用描述统计学（Descriptive Statistics）和统计推断断（Statistical Inference）的方法来分析指标在句子水平的情况。

我们绘制了各种指标的句子层面得分的单变量分布（Univariate Distribution）。为了便于从图像上比较各种指标的分布特征，我们对数值作了归一化处理，使数据的平均值为 0，标准差为 1：

$$x' = \frac{x - \mu}{\sigma} \tag{4.1}$$

μ 是 x 的平均值，σ 是 x 的标准差。为了使读者能快速检阅所有指标的分布情况，我们在正文展示了模型为 HRED，数据集为 OpenSubtitles 的各项指标的分布，附录 B 展示了所有的组合上的指标分布。



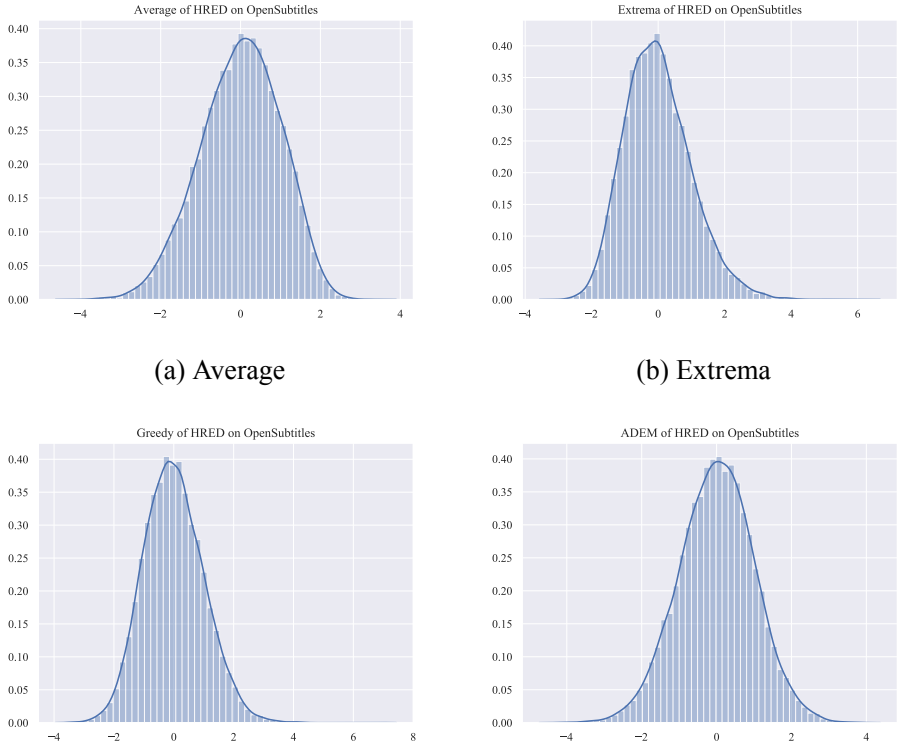
(a) BLEU-1 (b) BLEU-2

(c) BLEU-3 (d) BLEU-4

图 4.6 BLEU 的概率分布

图 4.6 是 BLEU 的 N 取 1 到 4 时的分布图。由于加入了平滑处理，所以大部分得分都有非零数值，在均值附近集中了大量的句子，这可能是因为大部分句子的得分非常低[8]。一元词匹配相对高阶 n-gram 匹配更容易，所以 BLEU-1 在均值的右边有一个低矮的峰值，并且峰值对应的横坐标偏离了均值 0，这说明有相当一部分句子的得分高于均值。而低于均值的部分出现了一个高峰，说明也有大量句子得分低于均值。BLEU-2 至

4 的分布都非常相似，只是峰值的横坐标比 BLEU-1 更接近均值。从总体上来说，BLEU 的分布形状十分尖锐，大量句子集中在均值附近，图像不对称，峰值比高斯分布的峰值 $1/\sqrt{2\pi} \approx 0.4$ 高。

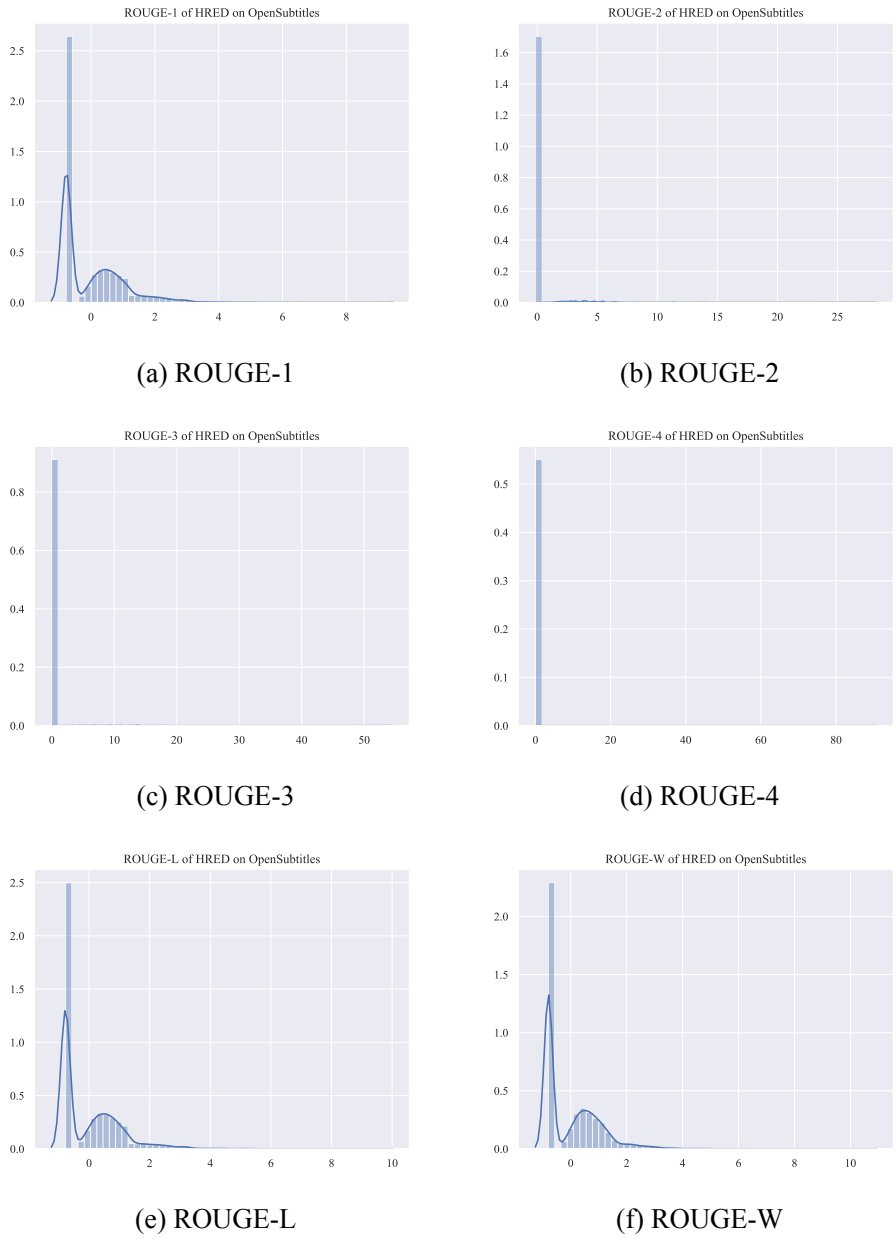


(a) Average (b) Extrema

(c) Greedy (d) ADEM

图 4.7 基于词嵌入的指标和 ADEM 的概率分布

图 4.7 是基于词嵌入的指标和 ADEM 指标的分布图。因为 ADEM 可以看做是一种更高级的词嵌入指标，所以把它和其他词嵌入指标并列。四个图像均呈钟型曲线，峰值非常接近 0.4，峰值的横坐标则非常接近 0。ADEM 曲线显示出几乎完美的对称，Greedy 曲线的对称度次之，Average 曲线向右边偏移，Extrema 曲线向左边偏移。ADEM 和词嵌入指标有相似之处。从句子向量的组合方式来看，Average 的组合方法是平均值，Extrema 的组合方法是极端值，而 ADEM 的则使用了预训练的 VHRED 的编码器。与其他词嵌入指标不同的是，ADEM 考虑了上下文的影响，用神经网络对上下文，响应和参考三者加权，并且显式的优化了和人类评价的相关性，可见 ADEM 采用的机制更为复杂。Lowe 在[30] 中指出，ADEM 倾向于保守打分，即人类评价给高分的句子，ADEM 倾向于给稍低的分数。从 ADEM 的曲线可以从某种意义上验证这一点，得分高于平均值的句子与得分低于平均值的句子几乎一样多，说明它不倾向于打高分或者低分。



(a) ROUGE-1 (b) ROUGE-2

(c) ROUGE-3 (d) ROUGE-4

(e) ROUGE-L (f) ROUGE-W

图 4.8 ROUGE 的概率分布

图 4.8 是 ROUGE 系列指标的分布图。从曲线的形状来看，大致可分为两类：ROUGE-1，ROUGE-L，ROUGE-W 是一类，ROUGE-2，ROUGE-3，ROUGE-4 是一类。ROUGE-2 这类图像的特点是几乎所有句子的得分都集中在均值，除此之外没有其他得分。我们猜想因为没有对 ROUGE 指标做任何平滑处理，除了极个别句子外，其他句子都得了零分。这表明 ROUGE-N 指标当 $N > 1$ 时捕捉不到任何 n-gram 重叠。

另一方面，ROUGE-1 类指标更像是 BLEU-1 指标的噪化版本，它们都是双峰曲线，而且第一个峰值对应的点 (x_1, y_1) 和第二个峰值对应的点 (x_2, y_2) 都有相似的坐标。我们

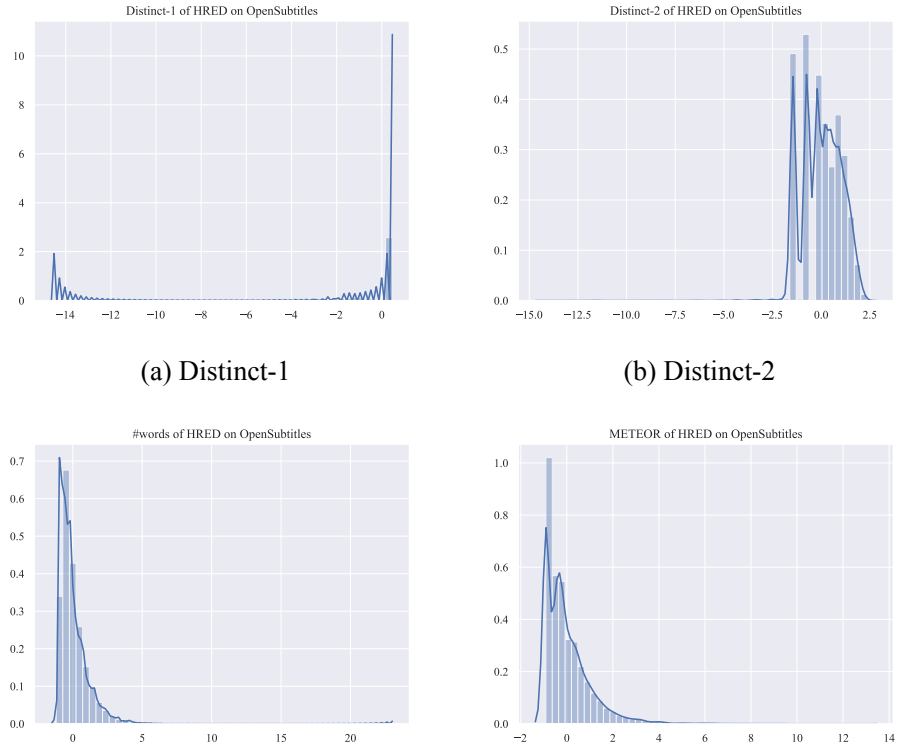
猜想上述现象的原因是所有的响应和参考基本上没有 $N > 1$ 的 n-gram 重叠，因此基于最长公共子序列的匹配退化成了一元词匹配。

图 4.9 是 Distinct-N，METEOR 和 #words 的分布图。我们发现 Distinct-1 的分布呈现两极分化，在均值附近和均值右边较远处都聚集了大量句子。Distinct-N 的分母是句子的长度（#words），分子是句子中各异的 n-gram 数量。从4.9c来看，句子的长度集中在均值附近，长于均值的句子比短于均值的句子多，这表明 Distinct-N 的值主要受分子影响。Distinct-1 的曲线表明，极大部分句子的各异的单词数量都接近平均值，但也有少部分句子各异的单词数量远远小于平均值。Distinct-2 的图像没有出现两极分化，可能是因为二元词的空间比一元词的空间更大，一个句子各异的二元词通常多余各异的一元词。图像的特点是：

1、 大部分句子集中在高于均值的一片区域。

2、 个别低于均值的区间聚集了大量句子。

这可以理解为模型在应对不同的消息时，生成的句子质量不一，对一部分消息产生了多样的响应，对另一部分消息却产生了单调的响应。



(a) Distinct-1　　(b) Distinct-2

(c) #words　　(d) METEOR

图 4.9　Distinct-N，#words 和 METEOR 的概率分布

尽管通常被分类成基于词重叠的指标，METEOR 的图像表现出了和其他基于词重叠的指标（BLEU，ROUGE）相当不同的性质。它似乎能区分不同响应的质量，并且和句子长度有着某种联系。图像在均值的右侧表现出指数衰减，在左侧则是一个尖锐的单峰。本质上，METEOR 是基于一元词匹配的，但是我们尚不清楚 METEOR 的多种匹配模块，对齐算法和惩罚系数如何使它的分布变得与 BLEU-1 等指标的分布如此不同。

中心极限定理（Central Limit Theorems）表明：大量独立随机变量的叠加近似于高斯分布。我们假设当句子足够多时，句子层面的人类评价近似于高斯分布。据此我们认为，句子层面得分的分布接近高斯分布是一个指标和人类评价具有较高相关性的前提条件之一。从上述各个指标的分布来看，ADEM 和词嵌入指标具有很大的潜力。

本节的分析结果是从 OpenSubtitles 和 HRED 的组合中得出的，并不能完全代表所有组合下的指标分布情况。观察附录 B可发现，一般来说，指标的分布当模型不同而数据集相同时比较相似，当模型相同而数据集不同时则有较大差异，因此句子层面的指标分布受数据集的影响较大。

4.3 结果与讨论

通过系统层面的分析，我们发现数据集对指标的影响普遍大于模型的影响，表现为同一个数据集上的模型得分比较接近，同一个模型在不同数据集上的得分相差较远。不排除某些极端情况，但是一般来说，数据集的影响是决定性的。我们还发现有些指标能对同一个数据集上的模型产生差别很大的分数，另外一些指标则倾向于给不同模型相近的分数。

通过句子层面的分析，我们发现不同指标的分布形态各异，某些指标的分布有着相似的形态，它们和其他指标的分布形成了鲜明对比。比如基于词嵌入的指标以及 ADEM 的分布非常相似，呈现类似高斯分布的钟型曲线。和单元词匹配有关的 BLEU-1，ROUGE-1，ROUGE-L 以及 ROUGE-W 的分布也很相似，图像是不对称的双峰曲线。指标分布的不一致性在一定程度上解释了模型在系统层面性能的不一致性。

上述经验性的结论反映了几个问题。从模型的角度来看，基于 Seq2Seq 框架的模型普遍采用的目标函数是最大化训练样本的对数概率，所以本质上，模型是数据集的条件概率分布的拟合器。这意味着如果数据集的噪声很大，那么模型的输出质量会直接受到影响。使用其他模型框架很可能规避此问题，例如 Li 等人尝试用对抗生产网络[23] 以及强化学习[28] 生成对话，取得了不错的效果。

从数据集的角度来看，我们采用的三个数据集都存在大量的错别字，不规范的语法

以及非自然语言符号，这也许是开放领域数据集的通病。当这些错误样本的比例超过了正确样本的比例时，模型就很难学到正确的语法或者拼写。这些数据集大多数来自互联网论坛或者社交媒体，来自不同文化背景和有着不同生活阅历的人在这些平台上的对话可能具有很高的熵。这些数据集的样本在长度，情感态度和思想等方面都具有很高的多样性，其潜在的概率分布可能非常复杂。这对模型的表达能力提出了挑战。

从指标的角度来看，基于不同表征的指标可能有着完全不同的分布，这导致不同评价指标在评价一系列模型时可能出现不一致性。使用这些不一致的指标来评价模型，不但使结果难以解释，还阻碍人们对系统性能的正确认识。我们注意到，生成式对话的响应可能没有一概而论的“好坏”之分。在机器翻译领域，系统的翻译有好坏之分，本质上，无论是从语义上还是从表面形式（Surface Form）上越接近参考翻译就越好。因此适用于机器翻译的指标只需要捕捉两个句子的相似程度即可。生成式对话则不同。系统生成的响应难以用单一的标准去衡量。学者们曾用过话题相关度[25]和 n-gram 多样性[20]去衡量响应的质量，然而这些度量都不能和人类评价产生很好的相关性。摆在我们面前的问题也许不是如何提高指标和人类评价的相关性，而是：什么样的对话才是好的对话?

结论

4.4 总结

本文对生成式对话领域的模型，数据集和指标进行了一次深入考察。我们首先介绍了生成式对话领域的兴起过程中一些重要的模型，比如 Ritter 等人的 SMT 模型[4]，Sordoni 等人的 DCGM 模型[6] 和 Vinyals 等人的 NCM 模型[18]。接着，在相关工作中，我们介绍了生成式模型的定义，模型的核心组件 RNN，以及几种常见的生成式模型，包括最简单的 RNN 语言模型，广为流行的 Seq2Seq 框架和多层编解码器 HRED。我们还介绍了本领域用到过的一些指标，包括基于词重叠的 BLEU，ROUGE 和 METEOR，基于词嵌入的 Average，Extrema 和 Greedy，衡量概率语言模型性能的困惑度，以及专门为生成式对话设计的 ADEM 和 RUBER。最后，我们介绍了生成式对话文献中常见的数据集，包括 Twitter Dialogue Corpus，Ubuntu Dialogue Corpus 和 OpenSubtitles 等等。

在众多模型，数据集和指标中，我们选择了 Serban 等人在[25] 中使用的三个模型 HRED，LSTM 和 VHRED。这三个模型有着成熟的实现，易于其他人复现我们的实验。在数据集方面，我们选择了公开的，代表了不同领域的三个数据集，分别是 Ubuntu Dialogue Corpus，OpenSubtitles 和 LSDSCC。在指标方面，我们尽可能涵盖了对话领域使用过或者提出的指标，在配置方面与[8] 大致对齐。实验的主要工作是：

1、 在多个数据集上训练多个模型。

2、 在训练结果上运行多个评价指标。

3、 对指标进行多种数据分析。

我们的实验数据是多个模型和数据集的组合在不同指标上的得分，以及这些模型输出的响应。由于时间关系，我们只分析了得分的数据。

我们探索了系统层面和句子层面的得分。系统层面得分是是对一个模型在一个数据集上的表现的粗粒度考察，它可能掩盖了一些事实，但是方便我们综合考察模型、数据集和指标三者的整体关系。我们参考[8]，将指标的数据进行分组，并以此组织所有的分析。我们把 BLEU 的 N 取不同值的指标分为一组，把 ROUGE 的所有变形分为一组，把词嵌入的指标和 ADEM 分为一组，把剩下的 METEOR，Distinct-N 和句子长度归到一组。

各个模型的系统层面得分随着数据集的变化和指标的变化有较大差异，比较稳定的

是同一个数据集和指标上各个模型的排名，一般 HRED 和 VHRED 比 LSTM 好，有时优势不太明显，有时会出现 LSTM 反超的情况。模型在 Ubuntu Dialogue Corpus 上的各项指标通常更好，但是有时候某些指标在另外两个数据集上的得分会反超。从附录 A 可以发现，数据集和模型对得分都有影响，从大体上看，HRED 和 VHRED 优于 LSTM，Ubuntu Dialogue Corpus 的得分高于其他数据集，但是这不是绝对的，实验数据中存在许多反例。我们认为，除了实验过程本身带来的噪音外，主要的原因是数据集的特征变化太大，模型无法完全将一个数据集上的性能迁移到另一个数据集上。而且，实验中的指标形成了集群现象，不同集群之间一致性很低，给评价造成了混乱。

我们在句子层面的分析主要采取单变量概率分布的形式，从不同指标的分布图像中验证了指标之间的集群现象。指标的集群反映了提取相同特征的指标很可能有相似的行为，不管它们如何使用这些特征。在这些指标的集群中，我们发现图像大致有两类：

1、 基于词嵌入的指标的图像接近高斯分布。

2、 基于词重叠的指标的图像是不对称的双峰曲线，大量句子集中在均值附近很小的范围。

我们假设在大量句子上的人类评价将接近高斯分布，从而认为分布接近高斯分布的指标是更好的选择。我们发现了指标在大量句子上的统计规律，从统计的角度指出词嵌入指标更适合本领域。

我们的实验有些不完善的地方。所有数据集上的 LSTM 模型的门单元应该统一使用 LSTM；解码时我们使用了随机取样，结果发现响应中有很多语法错误，使用集束搜索可以产生语法错误较少的响应。我们没有针对特定数据集做参数的调优，导致模型在有些数据集上的表现没有达到最优。我们也没有像[25] 那样用预训练的 HRED 初始化 VHRED 的参数，结果 VHRED 的性能没有达到最优。我们将汲取经验教训，尽量减少实验程序带来的噪音。

4.5 展望

在实验结论的基础上，我们对生成式对话领域的模型，数据集以及指标提出几点展望。在模型方面，我们提出两个可能的思路是：

1、 使用新的模型体系结构。

2、 加入更多特征。

尽管 Seq2Seq 框架在机器翻译领域取得成功，但是在更加困难的对话生成领域，它的表达力可能遇到了瓶颈。Seq2Seq 框架本质上是一种带注意力的基于 RNN 的编解码

器结构，我们可以尝试其他编解码器结构，比如 Transformer[67]。我们还可以尝试对抗生产网络和强化学习，正如 Li 等人在[23, 28] 所做的那样。另一方面，我们可以加入感情色彩[64]，主题词[27] 以及对话者身份信息[21] 等特征，使模型的输出带有上述特征，从而更加符合人类评价。

在数据集方面，我们认为通过互联网收集的数据集容易引入多方面的噪音。事实上，这些数据集的特点和人类在互联网这种匿名平台的表现有密切联系。由于在这种平台上人们的言论没有什么限制，对话的话题非常多样，而且语言风格，语法习惯和感情色彩和具体的用户与具体的对话有关，这就使得数据集具有很高的熵。从概率的角度，数据集的样本分布可以看作是非常多个随机变量的叠加，如果这些随机变量都是独立同分布（Independently Identical Distributed，IID）的话，整个数据集的样本分布就趋向于高斯分布。

在实验中，我们发现基于词嵌入的指标在大量模型响应上的分布和高斯分布很接近，我们当时给出的解释是，接近高斯分布是一个指标和人类评价具有相关性的前提条件。但是，因为模型的响应的分布在一定程度上反映了数据集的样本分布，假设基于词嵌入的指标能反映两个句子的话题相关性，那么对它们的分布接近高斯分布的另一种解释就是，数据集的话题分布接近高斯分布。高斯分布是 $(-\infty, +\infty)$ 上方差已知的连续分布中熵最大的分布，面对这样复杂的数据集，我们应该用概率统计学的工具详细分析它在多个方面的分布情况，例如情感分布，对话轮数分布等等，这样有助于我们理解在这个数据集上训练模型的难度。

在指标方面，我们的实验大体上验证了[8] 的结论。我们认为生成式对话的评价指标首先要解决一个重要的问题，即什么样的对话才是好的对话。这个问题之所以重要，是因为它不仅能指导指标的构建，还决定了模型优化的方向。我们认为这个问题的答案不是“语义相关性高”或者“n-gram 多样性高”这些片面的特征，尽管它们可能是答案的一部分。这个问题可能难以回答，甚至没有普适性的答案，因为人类的对话是在自然环境和社会环境中演化出来的一种语言现象，它根据不同场合和参与者的变化而变化的，非常复杂。

从实用的角度，我们可以通过实验发现在某些数据集上好的对话应有的特征。直观的来说，在 Ubuntu Dialogue Corpus 上，好的对话应该和具体的技术话题有较高的相关性，因为这个数据集上的对话以“提问-解决问题”为主，好的对话应该能帮助人们解决问题；而在 Twitter Dialogue Corpus 上，好的对话应该考虑情感因素，关注主题的同时又具有一定的多样性，因为人们在 Twitter 上主要发布个人的状态信息，经常带有感

情色彩，而且期望从响应中看到相同的话题或者新奇的事物；在 LSDSCC 上，好的对话应该能对特定的电影发表中肯的评价，因为在 Reddit 的电影板块上，人们主要发表对电影的点评，希望和看过相同或者类似电影的人一起讨论，人们虽然有时候会发表极端的评价，但是不希望总是看到极端的评价，所以模型的评价应该中肯，最好带有一点个性化看法。虽然“什么样的对话才是好的对话”这个问题很难回答，但是我们不妨对它加上“在某个数据集上”的限定词，从某个数据集中发现最受欢迎的对话的模式，然后设计出能捕获这些模式的指标，并用它一致的评价在该数据集上训练的模型的表现。这个思路或许能为生成式对话领域带来一个新的范式：

1、 首先构建一个开放式领域对话数据集，并发现“好的对话”在这个数据集上应该具有什么性质。
2、 接着，设计出一系列能够衡量所谓的“好的性质”的指标，并确保它们和人类评价具有一致性。
3、 把数据集和与之匹配的指标称为一个“问题”（Problem），让不同的模型去解决这个问题。

进一步，我们希望把“设计出和人类评价相关度高的指标”这一个任务分解为若干个小任务：

1、 把问题限制在某个数据集上。
2、 找出这个数据集上人类评价高的对话具有的特征。
3、 设计出能准确捕获这些特征的指标。
4、 用人类评价验证指标在对应数据集上的有效性。

必须指出的是，第二步和第三步具有很大的难度。第二步一般需要人类评价员对数据集的样本打分，这导致带有人类评价的样本数量非常受限，这对指标的泛化能力提出了很高的要求[30]。第三步涉及的特征比较抽象，对指标的建模能力提出了很高的要求。

最后，我们来谈谈生成式对话和人的关系，以及这关系背后的意义。面向任务的对话系统的功能是用对话的形式帮助人完成任务。而生成式对话系统的功能是娱乐、语言学习和陪伴，这些功能其实不乏替代品：现代人从来不缺少娱乐，语言学习也有成熟的产业链，而亲人和爱人的陪伴是公认的最好的陪伴。生成式系统要如何在竞争中受到人们的青睐呢？首先，我们要认清生成式系统的功能的本质，就是要让人类在和机器的对话中有所收获，比如让人们感受到聊天的快乐，在人们失落时送去慰藉，在讨论中激发人们的灵感等等，而不仅仅是打发无聊的时间。

虽然这个目标对于现在的系统来说有些遥远，但是为了实现它，我们需要把人类的

需求考虑进来。人们在对话方面的需求是多种多样的，有的人通过对话获取信息，有的人通过对话排解情绪，有的人只是想寻找快乐。虽然说需求分析比较接近应用而偏离理论，但是应用和理论相结合能增加研究成果的实用性，使研究本身从应用所获取的反馈中受益。因此我们建议把最大化人类需求的满足程度作为模型的目标函数之一，并设计相关的指标加以衡量。我们相信考虑了人类需求的模型将更受人类评价的青睐。

致谢

我要感谢我的毕业设计指导老师荣文戈副教授。在毕设一开始，我因为考研、重修课程等事情无法立刻投入工作，荣老师对此表示了极大的理解，对此我深表感谢。在中期报告前的一个月，荣老师给了我关键的支持：他不但给了我两台学院的服务器，而且还在紧缺的实验室工位中给我安排了一个。在此后的毕设工作中，他不断根据我的工作汇报的反馈为我的毕设提供切实有效的指导。没有荣老师的帮助和指导，我的毕设工作可能完全走不上正轨。荣老师温文尔雅的风度和关爱学生的作风是我十分赞赏的。能在他的指导下完成毕设是我的荣幸。

我要感谢我们计算机学院对毕设工作的高度重视和大力支持。学院早在去年 10 月就召开了毕设动员大会，不就之后就组织了全学院的开题报告。据我所知，全校范围内恐怕没有比我们学院更早进行本科生毕设开题的了。得益于此，我们比别的学院多出几乎半年时间准备毕业设计，虽然惭愧的是，我在蹉跎中耗费了大段光阴。我非常感谢高小鹏院长等学院领导对我们毕设事宜的重视。高院长在毕设动员大会上指出了毕设过程中可能会遇到的种种挫折，告诫我们要对自己负责，要多和导师沟通等等，是我在毕设过程中一直受用的。开题报告、中期报告和毕设答辩等环节有赖于学院的全体教职工的辛勤付出。感谢他们不惜牺牲宝贵的科研时间来为我们这些初出茅庐的本科生提意见。

我还想感谢 G951 的各位学长学姐，感谢他们在五一劳动节和我一起看电影，这是一段美好的回忆。感谢学生三公寓的楼管阿姨，她总是把我们当成自己的孩子看待，热心帮助我们，关心我们。感谢密码学课程的郭华老师，她在我毕业这年教会我信息安全的重要性。感谢我校的校车司机，他们一天出车十几趟，将满载的师生安全送达。感谢在某个暴雨的深夜，一位不知名的博士学长用他的伞送我回宿舍，陌生人也能让人感到温暖。

最后，我要感谢我的父母：你们的无私和博大的关爱，我永远无法偿还。

参 考 文 献

[1] Serban I. V., Lowe R., Henderson P., et al. A Survey of Available Corpora for Building Data-Driven Dialogue Systems[J]. CoRR, 2015, abs/1512.05742.

[2] Sutskever I., Vinyals O., Le Q. V. Sequence to Sequence Learning with Neural Networks[A]. Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014[C]., 2014:3104–3112.

[3] Shang L., Lu Z., Li H. Neural Responding Machine for Short-Text Conversation[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing[C]., 2015:1577–1586.

[4] Ritter A., Cherry C., Dolan W. B. Data-driven Response Generation in Social Media[A]. Proceedings of the Conference on Empirical Methods in Natural Language Processing[C]., 2011:583–593.

[5] Serban I. V., Sordoni A., Bengio Y., et al. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models[A]. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence[C]., 2016:3776–3784.

[6] Sordoni A., Galley M., Auli M., et al. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses[J]. CoRR, 2015, abs/1506.06714.

[7] Lowe R., Serban I. V., Noseworthy M., et al. On the Evaluation of Dialogue Systems with Next Utterance Classification[A]. Proceedings of the SIGDIAL 2016 Conference[C]., 2016:264–269.

[8] Liu C., Lowe R., Serban I., et al. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation[A]. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing[C]., 2016:2122–2132.

[9] Kannan A., Vinyals O. Adversarial Evaluation of Dialogue Models[J]. CoRR, 2017, abs/1701.08198.

[10] Song Y., Yan R., Li X., et al. Two are Better than One: An Ensemble of Retrieval- and Generation-Based Dialog Systems[J]. CoRR, 2016, abs/1610.07149.

[11] Bengio Y., Ducharme R., Vincent P., et al. A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research, 2003, 3:1137–1155.

[12] Mikolov T., Chen K., Corrado G., et al. Efficient Estimation of Word Representations in Vector Space[A]. 1st International Conference on Learning Representations[C]., 2013.

[13] Pennington J., Socher R., Manning C. D. Glove: Global Vectors for Word Representation[A]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing[C]., 2014:1532–1543.

[14] Mikolov T., Karafiát M., Burget L., et al. Recurrent neural network based language model[A]. INTERSPEECH 2010[C]., 2010:1045–1048.

[15] Hochreiter S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998, 6(2):107–116.

[16] Hochreiter S., Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735–1780.

[17] Cho K., Merrienboer B., Gülçehre Ç., et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[A]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing[C]., 2014:1724–1734.

[18] Vinyals O., Le Q. V. A Neural Conversational Model[J]. CoRR, 2015, abs/1506.05869.

[19] Lison P., Tiedemann J. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles[A]. Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016[C]., 2016.

[20] Li J., Galley M., Brockett C., et al. A Diversity-Promoting Objective Function for Neural Conversation Models[A]. NAACL HLT 2016[C]., 2016:110–119.

[21] Li J., Galley M., Brockett C., et al. A Persona-Based Neural Conversation Model[J]. CoRR, 2016, abs/1603.06155.

[22] Goodfellow I. J., Pouget-Abadie J., Mirza M., et al. Generative Adversarial Networks[J]. CoRR, 2014, abs/1406.2661.

[23] Li J., Monroe W., Shi T., et al. Adversarial Learning for Neural Dialogue Generation[A]. EMNLP[C]., 2017:2157–2169.

[24] Sordoni A., Bengio Y., Vahabi H., et al. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion[A]. Proceedings of the 24th ACM Interna-

tional Conference on Information and Knowledge Management[C]., 2015:553–562.

[25] Serban I. V., Sordoni A., Lowe R., et al. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues[J]. CoRR, 2016, abs/1605.06069.

[26] Serban I. V., Klinger T., Tesauro G., et al. Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation[A]. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence[C]., 2017:3288–3294.

[27] Xing C., Wu W., Wu Y., et al. Topic Aware Neural Response Generation[A]. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence[C]., 2017:3351–3357.

[28] Li J., Monroe W., Ritter A., et al. Deep Reinforcement Learning for Dialogue Generation[A]. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing[C]., 2016:1192–1202.

[29] Galley M., Brockett C., Sordoni A., et al. deltaBLEU: A Discriminative Metric for Generation Tasks with Intrinsically Diverse Targets[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing[C]., 2015:445–450.

[30] Lowe R., Noseworthy M., Serban I. V., et al. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses[A]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics[C]., 2017:1116–1126.

[31] Li J., Monroe W., Jurafsky D. Data Distillation for Controlling Specificity in Dialogue Generation[J]. CoRR, 2017, abs/1702.06703.

[32] Tao C., Mou L., Zhao D., et al. RUBER: An Unsupervised Method for Automatic Evaluation of Open-Domain Dialog Systems[A]. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence[C]., 2018:722–729.

[33] Lowe R., Pow N., Serban I., et al. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems[A]. Proceedings of the SIGDIAL 2015 Conference[C]., 2015:285–294.

[34] Xu Z., Jiang N., Liu B., et al. LSDSCC: a Large Scale Domain-Specific Conversational Corpus for Response Generation with Diversity Oriented Evaluation Metrics[A]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies[C]., 2018:2070–2080.

[35] Schuster M., Paliwal K. K. Bidirectional recurrent neural networks[J]. IEEE Trans. Signal Processing, 1997, 45(11):2673–2681.

[36] Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[A]. 3rd International Conference on Learning Representations[C]., 2015.

[37] Luong T., Pham H., Manning C. D. Effective Approaches to Attention-based Neural Machine Translation[A]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing[C]., 2015:1412–1421.

[38] Li J., Monroe W., Jurafsky D. A Simple, Fast Diverse Decoding Algorithm for Neural Generation[J]. CoRR, 2016, abs/1611.08562.

[39] Papineni K., Roukos S., Ward T., et al. Bleu: a Method for Automatic Evaluation of Machine Translation[A]. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics[C]., 2002:311–318.

[40] Doddington G. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics[A]. Proceedings of the Second International Conference on Human Language Technology Research[C]., 2002:138–145.

[41] Banerjee S., Lavie A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments[A]. Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005[C]., 2005:65–72.

[42] Stanojevic M., Sima'an K. BEER: BEtter Evaluation as Ranking[A]. Proceedings of the Ninth Workshop on Statistical Machine Translation[C]., 2014:414–419.

[43] Popovic M. chrF: character n-gram F-score for automatic MT evaluation[A]. Proceedings of the Tenth Workshop on Statistical Machine Translation[C]., 2015:392–395.

[44] Snover M., Dorr B., Schwartz R., et al. A Study of Translation Edit Rate with Targeted Human Annotation[A]. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA[C]., 2006:223–231.

[45] Chen B., Cherry C. A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU[A]. Proceedings of the Ninth Workshop on Statistical Machine Translation[C]., 2014:362–367.

[46] Lin C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries[A]. Text Summarization Branches Out[C]., 2004.

[47] Harris Z. Distributional structure[J]. Word, 1954, 10(23):146–162.

[48] Harris Z. S. Mathematical structures of language[M]. Vol. 21, 1968.

[49] Rus V., Lintean M. C. A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics[A]. Proceedings of the Seventh Workshop on Building Educational Applications Using NLP[C]., 2012:157–162.

[50] Mitchell J., Lapata M. Vector-based Models of Semantic Composition[A]. ACL 2008[C]., 2008:236–244.

[51] Forgues, Pineau J., Larcheveque J., et al. ootstrapping Dialog Systems with Word Embeddings[A]. Workshop on Modern Machine Learning and Natural Language Processing[C]., 2014.

[52] Danescu-Niculescu-Mizil C., Lee L., Pang B., et al. Echoes of power: language effects and power differences in social interaction[A]. Proceedings of the 21st World Wide Web Conference 2012[C]., 2012:699–708.

[53] Danescu-Niculescu-Mizil C., Sudhof M., Jurafsky D., et al. A computational approach to politeness with application to social factors[A]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics[C]., 2013:250–259.

[54] Fu L., Danescu-Niculescu-Mizil C., Lee L. Tie-breaker: Using language models to quantify gender bias in sports journalism[J]. CoRR, 2016, abs/1607.03895.

[55] Zhang J., Spirling A., Danescu-Niculescu-Mizil C. Asking too much? The rhetorical role of questions in political discourse[A]. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing[C]., 2017:1558–1572.

[56] Zhang J., Chang J. P., Danescu-Niculescu-Mizil C., et al. Conversations Gone Awry: Detecting Early Signs of Conversational Failure[A]. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics[C]., 2018:1350–1361.

[57] Danescu-Niculescu-Mizil C., Lee L. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs[J]. CoRR, 2011, abs/1106.3077.

[58] Li Y., Su H., Shen X., et al. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset[A]. Proceedings of the Eighth International Joint Conference on Natural Language Processing[C]., 2017:986–995.

[59] Tiedemann J. News from OPUS—A Collection of Multilingual Parallel Corpora with

Tools and Interfaces[M]. Vol. 5, 2009: 237–248.

[60] Banchs R. E. Movie-DiC: a Movie Dialogue Corpus for Research and Development[A]. The 50th Annual Meeting of the Association for Computational Linguistics[C]., 2012:203–207.

[61] Ameixa D., Coheur L., Fialho P., et al. Luke, I am Your Father: Dealing with Out-of-Domain Requests by Using Movies Subtitles[A]. Intelligent Virtual Agents - 14th International Conference[C]., 2014:13–21.

[62] Wang H., Lu Z., Li H., et al. A Dataset for Research on Short-Text Conversations[A]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing[C]., 2013:935–945.

[63] Li J., Monroe W., Jurafsky D. Learning to Decode for Future Success[J]. CoRR, 2017, abs/1701.06549.

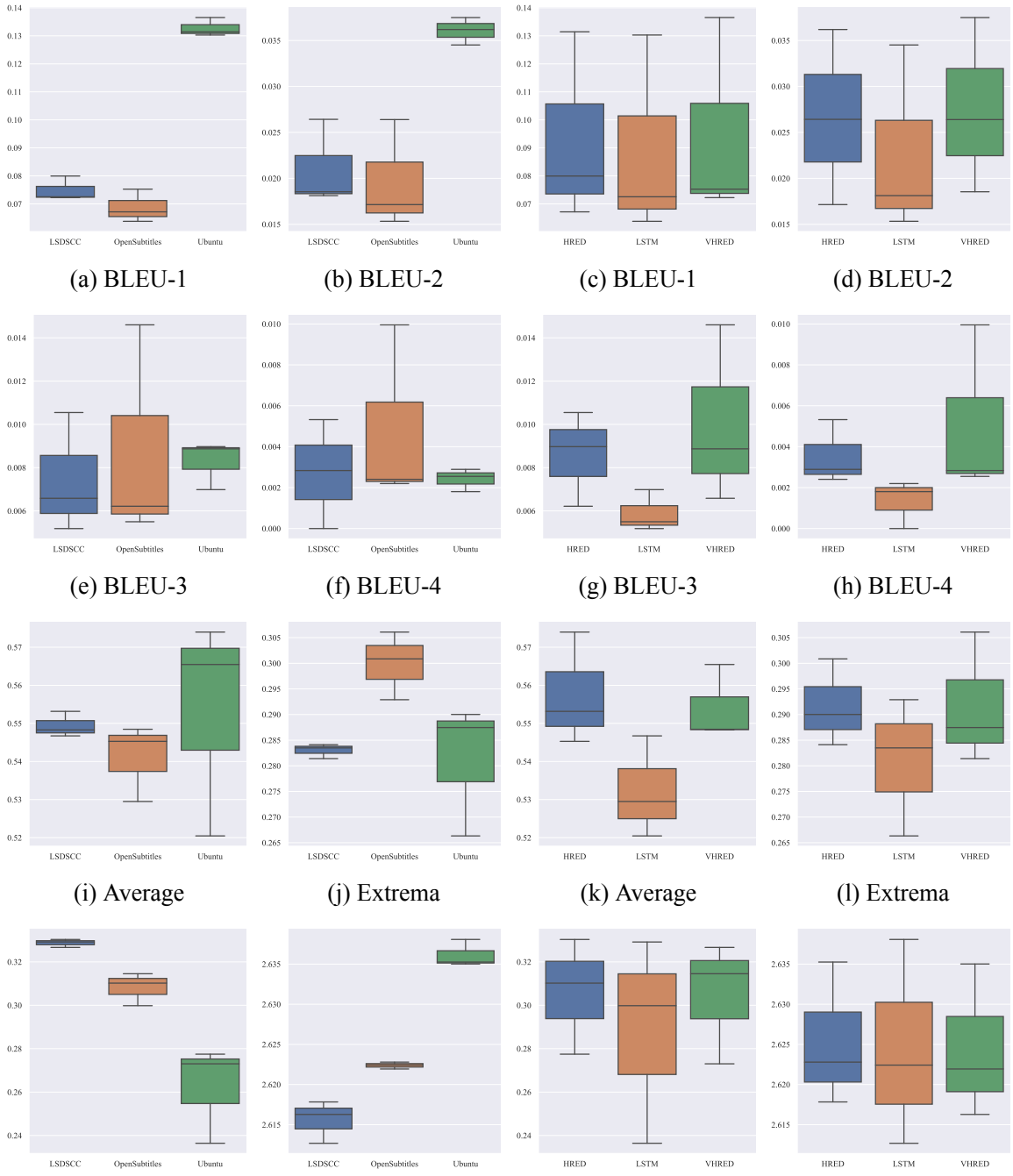
[64] Zhou H., Huang M., Zhang T., et al. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory[A]. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence[C]., 2018:730–739.

[65] Serban I. V., Lowe R., Charlin L., et al. Generative Deep Neural Networks for Dialogue: A Short Review[J]. CoRR, 2016, abs/1611.06216.

[66] Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization[A]. 3rd International Conference on Learning Representations[C]., 2015.

[67] Vaswani A., Shazeer N., Parmar N., et al. Attention Is All You Need[J]. CoRR, 2017, abs/1706.03762.

附录 A　系统得分在数据集和模型上的分布



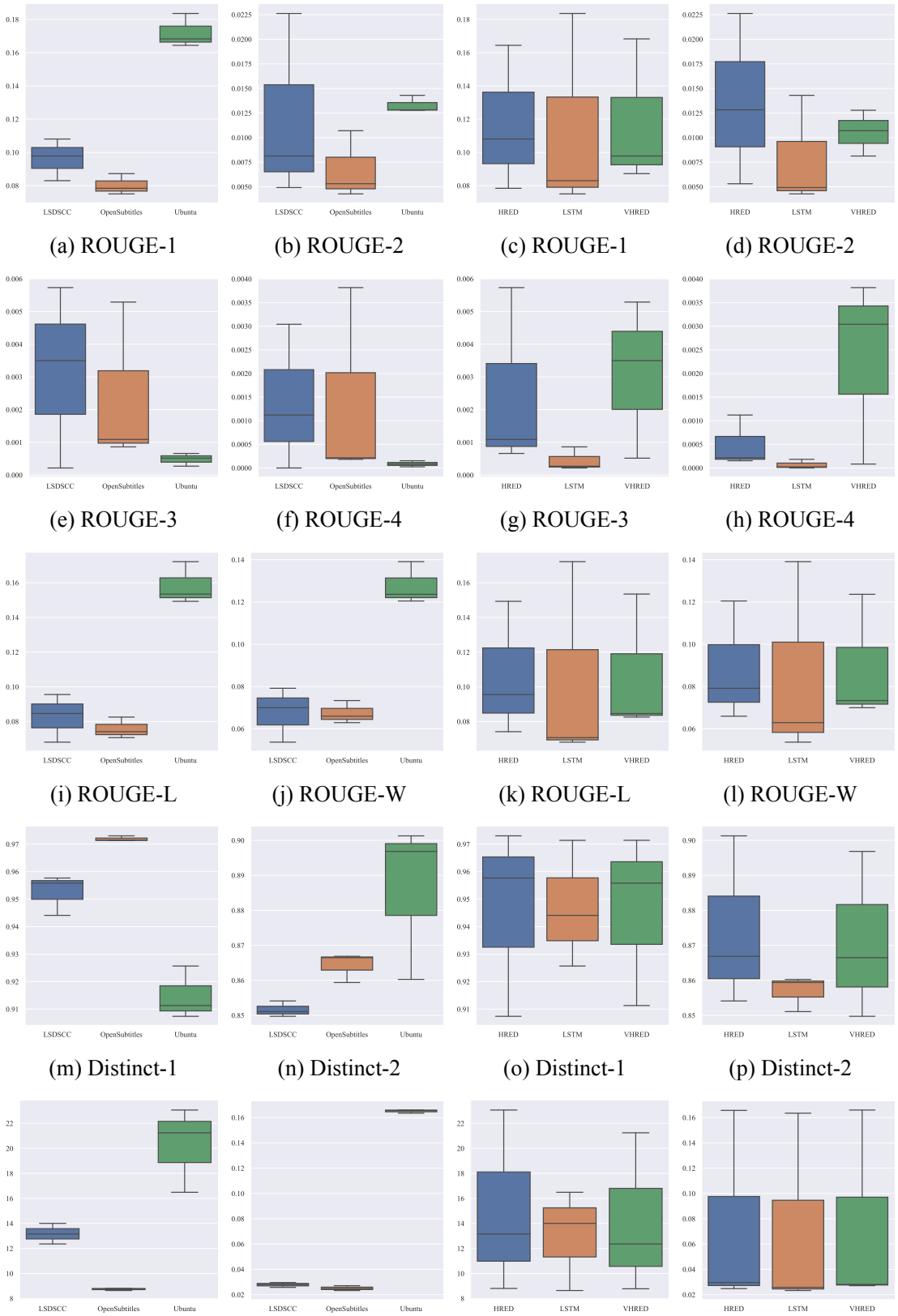
(a) BLEU-1　(b) BLEU-2　(c) BLEU-1　(d) BLEU-2

(e) BLEU-3　(f) BLEU-4　(g) BLEU-3　(h) BLEU-4

(i) Average　(j) Extrema　(k) Average　(l) Extrema

(m) Greedy　(n) ADEM　(o) Greedy　(p) ADEM

图 A.1　BLEU 等指标在不同数据集上的分布（左）和在不同模型上的分布（右）



(a) ROUGE-1 (b) ROUGE-2 (c) ROUGE-1 (d) ROUGE-2

(e) ROUGE-3 (f) ROUGE-4 (g) ROUGE-3 (h) ROUGE-4

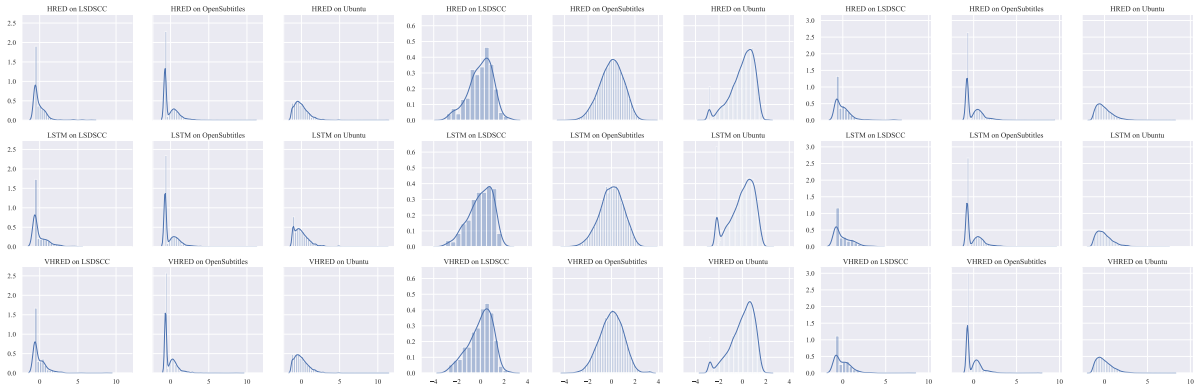
(i) ROUGE-L (j) ROUGE-W (k) ROUGE-L (l) ROUGE-W

(m) Distinct-1 (n) Distinct-2 (o) Distinct-1 (p) Distinct-2

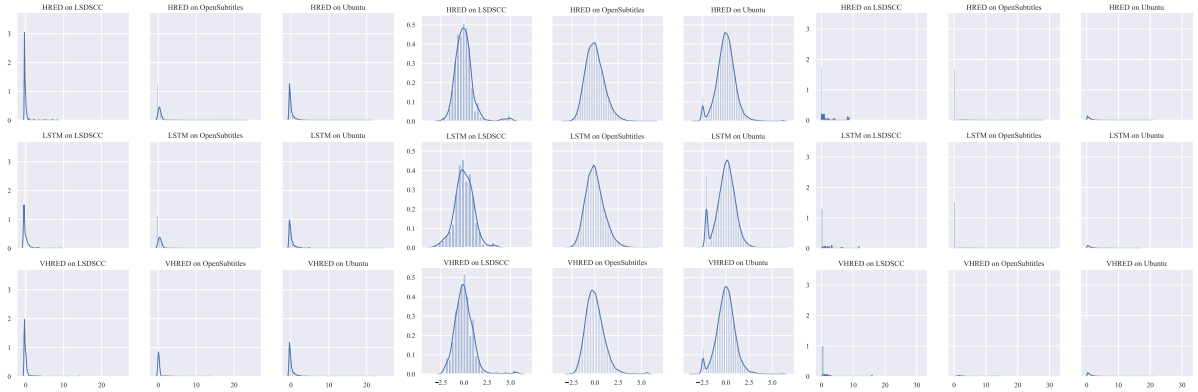
(q) #words (r) METEOR (s) #words (t) METEOR

图 A.2 ROUGE 等指标在不同数据集上的分布（左）和在不同模型上的分布（右）

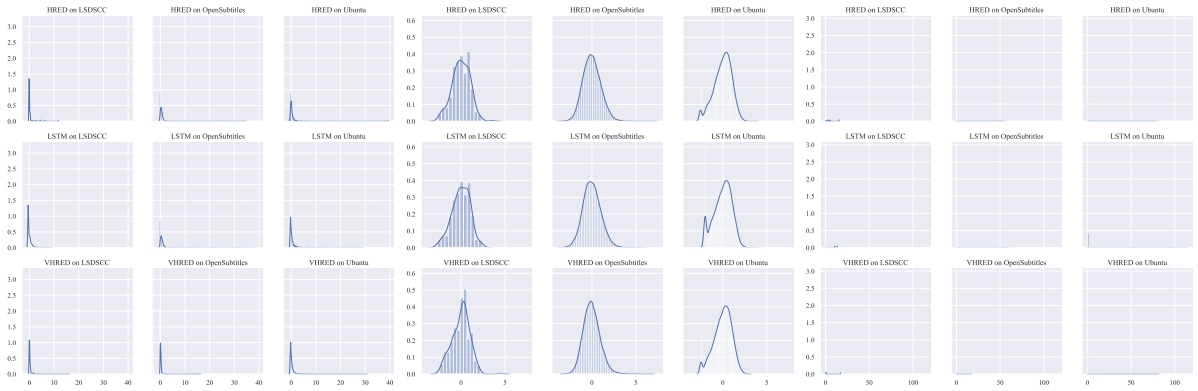
附录 B　指标的句子层面得分的分布



(a) BLEU-1　(b) Average　(c) ROUGE-1

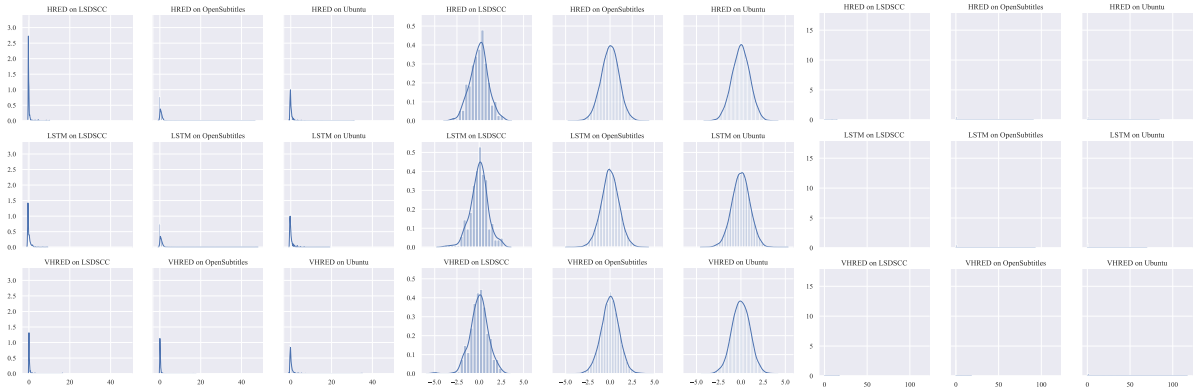


(d) BLEU-2　(e) Extrema　(f) ROUGE-2

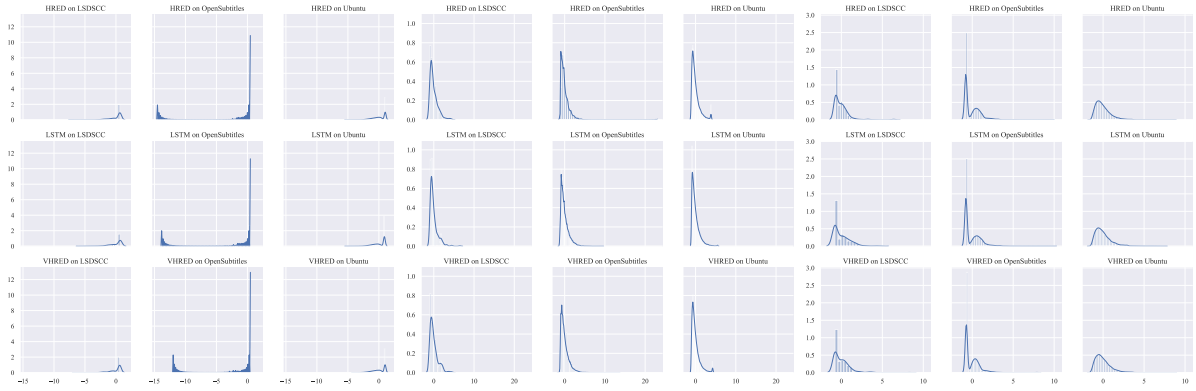


(g) BLEU-3　(h) Greedy　(i) ROUGE-3

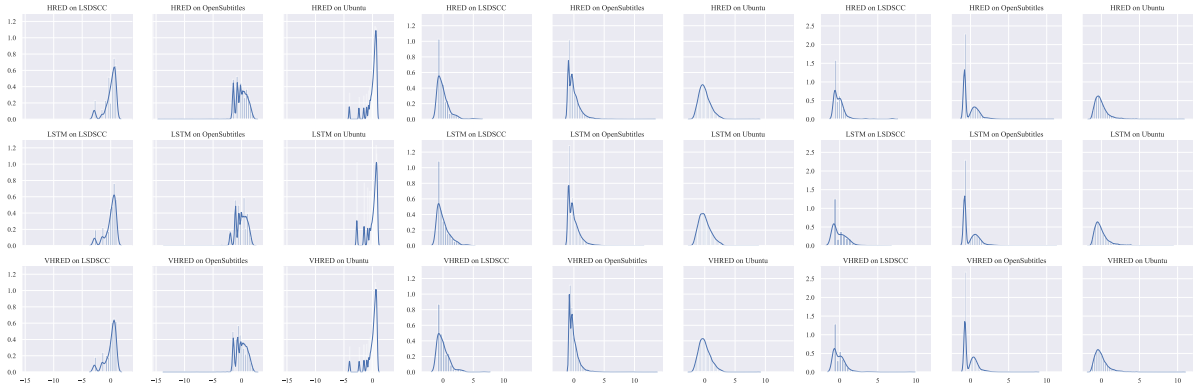
图 B.1　句子得分分布一



(a) BLEU-4 (b) ADEM (c) ROUGE-4



(d) Distinct-1 (e) #words (f) ROUGE-L



(g) Distinct-2 (h) METEOR (i) ROUGE-W

图 B.2 句子得分分布二