

# 文献阅读报告

## Deal or No Deal? End-to-end Learning for Negotiation Dialogues

张心泽

2017 年 6 月 24 日

### 文献摘要

大多数人与人之间的对话发生在半合作环境中，具有不同目标的代理人试图通过谈判就决策达成一致。虽然谈判需要复杂的沟通和推理技巧，但谈判的结果很容易区分，即“成功-失败”。如何进行谈判对于 AI 来说是一个很有趣的问题。

Facebook Artificial Intelligence Research（以下简称 FAIR）就此问题收集了一个关于多项目谈判任务的人与人之间（以下简称“人-人”）的对话数据集，建立了一个端到端学习（end to end learning）的模型训练机器进行谈判，并展示了谈判机器人的可能性。

### 1 介绍

在进行谈判时，智能代理<sup>①</sup>（以下简称代理）通常需要与怀有不同目标的他人合作，谈判的过程通常以自然语言为主题的对话来体现。具体来说，谈判是一个以自然语言为载体，通过推理与对话，最终达成或无法达成自己意图的过程。其中，谈判的资源为总数为 5-7 的书、帽子和球三种资源项目；主体为两位智能代理；价值函数<sup>②</sup>为对于给定的资源项目均有且只有唯一对应的价值；意图为通过对话使最终自己取得的资源价值最大，即文中所述“goals”。这样的过程要求代理能够理解、推理和通过语言表述来实现意图。

FAIR 以概率的形式描述代理在谈判中的行为，通过选取最大的行为概率来推理谈判行为，并以此构建了一个端到端的神经网络模型训练机器进行谈判。FAIR 首先建立了似然模型<sup>③</sup>，通过循环神经网络监督学习“人-人”谈判数据，使机器模仿人在谈判中的语言以达成交易；随后发现这种方法会导致机器为了达成交易不考虑自身意图，即过于妥协。因此，FAIR 在此之上提出了加强学习<sup>④</sup>和对话推演<sup>⑤</sup>两种模型优化机器的谈判能力，使其实现意图，而

---

<sup>①</sup> Intelligent Agents，即从事谈判工作且具有一定智能的代理。论文中没有指定其必须为人类，因此在定义上智能代理可为机器或人类。

<sup>②</sup> Value Function，即项目的价值函数。资源项目的总价值为 10 分，各项目单位价值为非负整数。项目价值函数在各智能代理处为随机生成，且在谈判前两位代理互不了解对方的价值函数因此会出现某些谈判中某资源项目对两位代理具有相同的价值，这也更接近现实。

<sup>③</sup> Likelihood Model，对应论文 Section 3。

<sup>④</sup> Goal-based Training，对应论文 Section 4。

<sup>⑤</sup> Goal-based Decoding，对应论文 Section 5。

不仅是模仿对话达成交易。

FAIR 针对所提的三种模型提出了自己的评价指标<sup>®</sup>：分数、成交率和帕累托最优率。原因经我归纳有三，一是本文虽使用了统计机器翻译（SMT）模型的方法，但侧重于实现用于谈判的 AI，因此 SMT 的 Blue Score 评价指标在此并不适用；二是本文对谈判过程进行了输入-对话-输出的表述，其中对输入中的项目建立了价值函数，对输出进行了成交-失败区分，故因此可得输出价值分数与成交率，所以在此提出了分数和成交率；三是在两人谈判模型中，不同的输出对应不同价值分数，这是一个可帕累托改善过程，故因此提出了帕累托最优率。




## 2 数据收集

FAIR 利用 Amazon Mechanical Turk 进行了两人间多项目谈判的自然语言数据收集任务，共收集了 5808 个对话数据集。在本篇论文中，FAIR 对于“两人间多项目谈判”的定义引用于 Fershtman<sup>[1]</sup>; Devault et al.<sup>[2]</sup>，若不实现阅读这两篇参考文献，读者很容易对谈判产生误解，误解为两位代理需要通过谈判交易对方的资源。实际上，在该谈判下，项目为两位代理要瓜分的资源，价值为随机生成；用户即代理仅知道自己的项目价值，另外一个代理的项目价值函数需要通过对话推理得出；输出为用户通过谈判结果所认为自己所应该得到项目部分。

值得注意的是，谈判结果可以为成交即“Deal”，也可以为失败即“No Deal”，那么如何确定谈判结果的标的？如项目和输出的定义，若谈判成交，则两位用户输出的各项目之和应当分别等于各项目总数。因此，FAIR 在本文中以“输出的各项目之和应当分别等于各项目总数”作为谈判结果的标的。

**Divide these objects between you and another Turker. Try hard to get as many points as you can!**

**Send a message now, or enter the agreed deal!**

Items	Value	Number You Get
	8	<input type="text" value="1"/>
	1	<input type="text" value="1"/>
	0	<input type="text" value="0"/>

Fellow Turker: I'd like all the balls

You: Ok, if I get everything else

Fellow Turker: If I get the book then you have a deal

You: No way - you can have one hat and all the balls

Fellow Turker: Ok deal

Type Message Here:

Message

图 1: 谈判对话采集界面，FAIR 用其收集“人-人”谈判文本数据集

FAIR 在 Das et al.<sup>[3]</sup> 的基础上建立谈判数据采集界面。如图1所示，在进行数据采集时，

<sup>®</sup> Comparison Systems，对应 Section 6。

界面会显示各项目的数量、价值、输出<sup>⑦</sup>和对话。其中，书、帽子和球的数量以 Items 下的图标数量表示，价值以 Values 下的数值表示，输出以 Number You Get 下的数值表示，对话以对话框形式表述和记录。以图1为例，项目的数量分别为 1、2 和 3，价值分别为 8、1 和 0，输出分别为 1、1 和 0，对话记录如对话框所示。

### 3 似然模型

FAIR 在本节中提出了一个谈判的基准模型。作为基准模型，机器至少应能根据输入的项目及其价值模仿人进行谈判。在此，首先对谈判过程和主体行动进行结构化的分析。

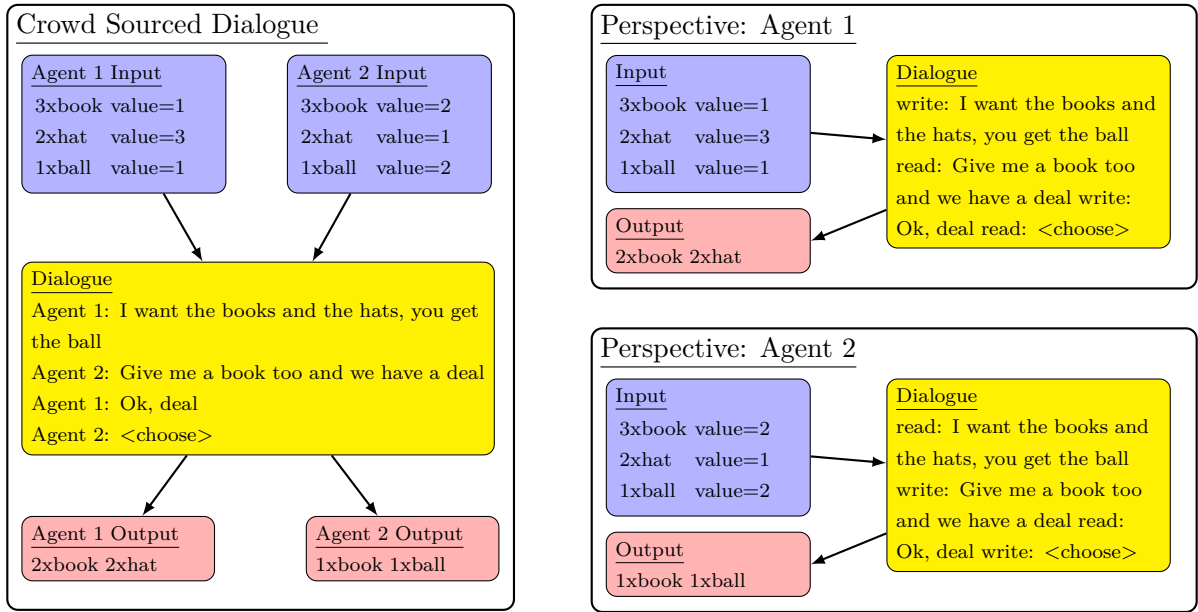


图 2: 谈判过程展示

#### 3.1 Date Representation

模型将各项目数量、各代理的价值函数作为输入，表示为 input goal  $g$ ，其中  $g$  由 6 个非负整数组成，依次对应各项目的数量和价值；将代理间的对话作为中间过程，表示为对话 dialogue  $x$ ，其中 tokens  $x_{0..T}$  表示对话  $x$  中依次进行的对话；将代理的输出作为模型输出，表示为 output  $o$ ，其中  $o$  由 6 个非负整数组成，表示三种项目分配给两位代理的数量。

以图2为例，从每个代理的角度将对话（图2左）分为了两个对话示例（图2右）。

模型的输入为：3 本书、2 顶帽子、1 个球和各代理的价值函数。其中，对于代理 1 即“Agent 1”，各项目的单位价值分数分别为：1、3 和 1，即输入为“3 1 2 3 1 1”；对于代理 2 即“Agent 2”，各项目的单位价值分数分别为：2、1 和 2，即输入为“3 2 2 1 1 2”；对于每

<sup>⑦</sup> Output，即输出，表示代理经过对话后认为自己应该取得的各项目数量。

一位代理，因谈判的目的是瓜分资源而不是交易资源，所以输入项目相同且项目总数在 5-7 之间，项目价值总分均为 10。

模型的中间过程根据代理的不同，划分为了两种情况。其中，对于代理 1，首句行为为写 “write”，对于代理 2，首句行为为读 “read”。

模型的输出为 “2 2 0 1 0 1”。其中前三位整数 “2 2 0” 表示代理 1 通过谈判认为自己可取得的项目数量，即 2 本书和 2 顶帽子；后三位整数 “1 0 1” 表示代理 2 通过谈判认为自己可取得的项目数量，即 1 本书和 1 个球。

值得注意的是，在图2中，因为两个代理输出中各项目选择加起来等于各项目总数，因此图2中的谈判视为达成，即 “Deal”。若出现两个代理输出中各项目选择加起来不等于各项目总数，如输出为 “3 2 0 1 0 1”，则表示谈判失败，即 “No Deal”。

### 3.2 GRUs RNN

FAIR 在本节中，提出了一个序列-序列网络，以根据代理的输入产生代理的对话，如图5a。

模型使用了 Cho et al.<sup>[4, 5]</sup> 中提出的循环神经网络，即 “recurrent neural network”，并借鉴了其文章中所提出的 “RNN Encoding-Decoding” 机制。该机制作为此篇论文处理对话自然语言的基础，且文中并未介绍，因此有必要在此就其关键概念进行阐述。

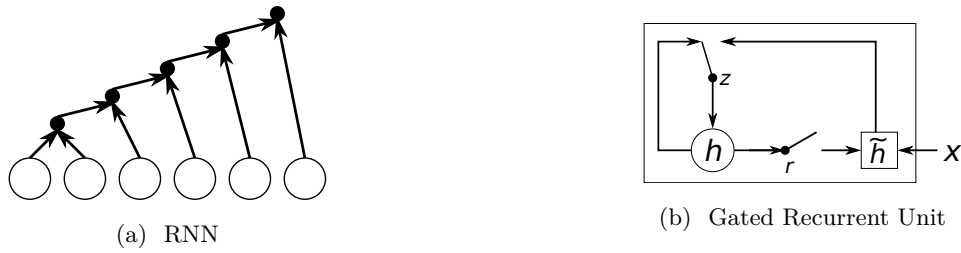


图 3: GRUs RNN

不同于传统的 FNN (Feed-forward Neural Networks, 前向反馈神经网络), RNN 引入了循环来处理序列数据。典型的序列数据如文本, 由一定顺序组成的词构成。在传统的神经网络模型中, 从输入层到隐藏层再到输出层, 层与层之间为全连接, 但每层之间的节点是无连接的。这种简单的神经网络对于很多问题无能为力, 以翻译模型为例, 要预测输出句子即输出词的组合, 需要考虑上文甚至上下文的情景, 因为句子中的前后单词并不是独立的。

RNN 之所以称为循环神经网络, 是因其序列的当前输出与前面的输出有关, 如图3a所示。具体表现形式为, 网络会通过激活函数和误差传递对前面的信息进行记忆并应用至当前输出的计算中, 即隐藏层之间的节点不再无连接而是有连接的。如对于可变长度序列  $x = (x_1, \dots, x_T)$ , 对于每一步  $t$ , 隐藏层单元  $h_{(t)}$  的更新公式为:

$$h_{(t)} = f(h_{(t-1)}, x_t) \quad (1)$$

其中  $f$  为非线性激活函数。通过将词的预测转化为词概率的计算问题，RNN 可以通过学习预测出下一个输出的所有词的概率，通过选择概率最大的词完成预测。如共有  $K$  个词，则第  $t$  个词  $x_{t,j}$  的计算公式为：

$$p(x_{t,j} = 1 \mid x_{t-1}, \dots, x_1) = \frac{\exp(w_j h_{(t)})}{\sum_{j'=1}^K \exp(w_{j'} h_{(t)})} \quad (2)$$

其中  $j$  表示所有可能的词， $j = 1, \dots, K$ ， $w_j$  是词权重矩阵  $W$  的列。通过组合词的概率，从而完成词序列  $x$  的预测。

$$p(x) = \prod_{t=1}^T p(x_t \mid x_{t-1}, \dots, x_1) \quad (3)$$

因为这一特性，RNN 及其改进型被广泛应用于机器翻译模型中。Cho et al.<sup>[5]</sup> 在此基础上提出了适用于统计机器翻译的词嵌入方法与 Encoder-Decoder 机制，如图4。在现实情况下，

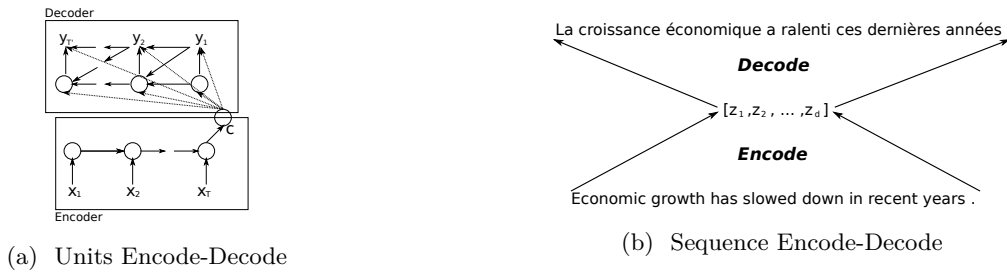


图 4: RNN Encode-Decode

序列数据的输入一般都是可变长度的。图4a展示了将可变长度的序列数据  $x = (x_1, \dots, x_T)$  作为输入嵌入至矩阵  $c$ ，然后再以  $c$  和目标序列数据  $y = (y_1, \dots, y_{T-1})$  作为输入，生成  $y = (y_1, \dots, y_{T'})$ 。

值得注意的是，这里的  $T$  和  $T'$  可能是不同的。

此时，隐藏层单元  $h_{(t)}$  的更新公式为：

$$h_{(t)} = f(h_{(t-1)}, y_{t-1}, c) \quad (4)$$

同样的，目标词  $y_t$  的计算公式为：

$$P(y_t | y_{t-1}, y_{t-2}, \dots, y_1, c) = g(h_{(t)}, y_{t-1}, c) \quad (5)$$

理论上，RNN 能对任何长度的序列数据进行处理，但是由于梯度消失和梯度爆炸问题，实际运用中多使用改进型 RNN，如图3b所示，便是 Gated Recurrent Unit(GRU)。结合 GRU 的改进型 Gated Recurrent Units RNN 即为此篇论文所使用 GRUs。

以图3b和图4b为例。以  $x = (x_1, x_2, \dots, x_T)$  表示输入的序列数据，其中  $x_t \in \mathbb{R}^d$ 。GRUs 由四个权重矩阵组成，分别为  $W^l$ 、 $W^r$ 、 $G^l$  和  $G^r$ 。对于任意一步  $t \in [1, T-1]$ ，第  $j$  个隐藏层单元  $h_j^{(t)}$  计算为：

$$h_j^{(t)} = \omega_c \tilde{h}_j^{(t)} + \omega_l h_{j-1}^{(t-1)} + \omega_r h_j^{(t-1)} \quad (6)$$

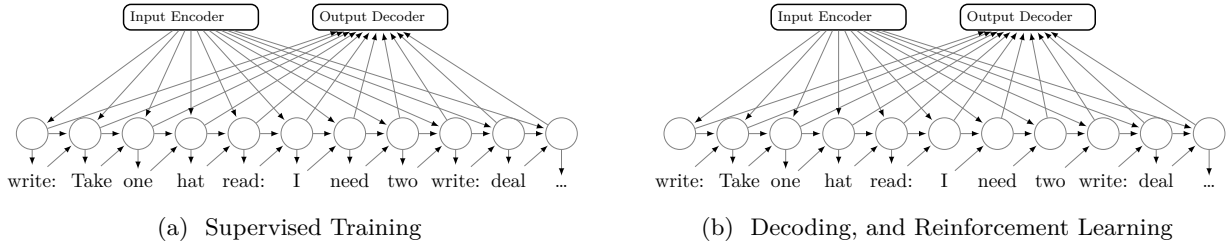


图 5: 端到端神经网络

其中  $\omega_c$ 、 $\omega_l$  和  $\omega_r$  值的和为 1。隐藏层单元初始化为:

$$h_j^{(0)} = Ux_j \quad (7)$$

其中  $U$  表示输入数据的隐藏层映射矩阵。

新的激活函数  $\tilde{h}_j^{(t)}$  计算为:

$$\tilde{h}_j^{(t)} = \phi \left( W^l h_{j-1}^{(t)} + W^r h_j^{(t)} \right), \quad (8)$$

其中  $\phi$  为非线性激活函数。

聚集系数  $\omega$  计算为:

$$\begin{bmatrix} \omega_c \\ \omega_l \\ \omega_r \end{bmatrix} = \frac{1}{Z} \exp \left( G^l h_{j-1}^{(t)} + G^r h_j^{(t)} \right) \quad (9)$$

其中  $G^l, G^r \in \mathbb{R}^{3 \times d}$  且

$$Z = \sum_{k=1}^3 \left[ \exp \left( G^l h_{j-1}^{(t)} + G^r h_j^{(t)} \right) \right]_k. \quad (10)$$

由此完成了基于 GRUs RNN 的词嵌入与 Encode-Decode 机制。Cho et al.<sup>[4, 5]</sup> 利用该机制实现了一个英译法的统计机器翻译模型。

### 3.3 Supervied Learning

如3.2中所提, FAIR 在本小节中, 提出了一个序列-序列网络, 以根据代理的输入产生代理的输出。其中, 该序列-序列网络, 或称为端到端网络, 是在 Cho et al.<sup>[4, 5]</sup> 所提的 GRUs RNN Encode-Decode 基础上的应用与改进<sup>®</sup>; 代理的输入包含初始项目数量、价值和对话; 代理的输出包含对话和选择。

因此此篇论文所提的端到端网络模型使用了四个 RNN:  $GRU_w, GRU_g, GRU_{\vec{o}}$  和  $GRU_{\overleftarrow{o}}$ 。

<sup>®</sup> 与 Cho et al.<sup>[4, 5]</sup> 相同的是, 此篇论文的对话输入和输出也均是文本序列数据; 不同在于, 此篇论文的输入中包含了项目数量和价值, 并希望最终输出取得最大的价值分数, 而不仅是流畅地进行谈判对话。

代理的输入  $g$  通过  $\text{GRU}_g$  进行 Encode, 对应的隐藏层单元为  $h^g$ 。模型基于  $h^g$  和  $x_{t-1}$  对  $x_t$  进行依次预测。对于每一步  $t$ ,  $\text{GRU}_w$  都将上一隐藏层单元  $h_{t-1}$ 、上一词  $x_{t-1}$ <sup>⑨</sup>和  $h^g$  作为神经网络的输入, 以更新当前的隐藏层单元  $h_t$ 。

$$h_t = \text{GRU}_w(h_{t-1}, [Ex_{t-1}, h^g]) \quad (11)$$

因此, 目标词  $x_t$  的计算关系为:

$$p_\theta(x_t|x_{0..t-1}, g) \propto \exp(E^T h_t) \quad (12)$$

值得注意的是, 此时的模型会同时预测两位代理的对话词, 并且模型依然是前馈的。

在对话的最后, 代理会输出选择, 以  $o$  表示。在这里, 模型对两位代理的选择通过相互独立的分类器独立预测<sup>⑩</sup>。分类器通过对话共享双向  $\text{GRU}_o$  模型和 Attention 机制<sup>[6]</sup>, 在对话和  $g$  的基础上进行词预测。

$$h_t^{\vec{o}} = \text{GRU}_{\vec{o}}(h_{t-1}^{\vec{o}}, [Ex_t, h_t]) \quad (13)$$

$$h_t^{\overleftarrow{o}} = \text{GRU}_{\overleftarrow{o}}(h_{t+1}^{\overleftarrow{o}}, [Ex_t, h_t]) \quad (14)$$

$$h_t^o = [h_t^{\overleftarrow{o}}, h_t^{\vec{o}}] \quad (15)$$

$$h_t^a = W[\tanh(W'h_t^o)] \quad (16)$$

$$\alpha_t = \frac{\exp(w \cdot h_t^a)}{\sum_{t'} \exp(w \cdot h_{t'}^a)} \quad (17)$$

$$h^s = \tanh(W^s[h^g, \sum_t \alpha_t h_t]) \quad (18)$$

最终的选择则在对话、 $g$  和  $h^s$  的基础上预测, 计算关系为:

$$p_\theta(o_i|x_{0..t}, g) \propto \exp(W^{o_i} h^s) \quad (19)$$

模型通过降低对话预测损失和选择预测损失来实现较好的预测效果。两个误差间的权重通过  $\alpha$  来调节。

$$L(\theta) = - \underbrace{\sum_{x,g} \sum_t \log p_\theta(x_t|x_{0..t-1}, g)}_{\text{Token prediction loss}} - \alpha \underbrace{\sum_{x,g,o} \sum_j \log p_\theta(o_j|x_{0..T}, g)}_{\text{Output choice prediction loss}} \quad (20)$$

不同于 Vinyals and Le<sup>[7]</sup> 的神经网络对话模型, FAIR 的方法考虑和共享了读取和写入词的所有参数。

<sup>⑨</sup> 通过矩阵  $E$  进行词嵌入。

<sup>⑩</sup> 这也是为什么要在3.1中需要对代理分开进行建立输入、中间过程和输出的原因。

### 3.4 Decoding

在 Decoding 的过程中, 模型必须要根据对话历史  $x_{0..t-1}$  和输入项目数量与价值  $g$  产生  $x_{0..T}$ 。如同 3.2 中的 Decode 过程, 也是通过选取最大概率词的方式进行预测。计算关系为:

$$x_t \sim p_\theta(x_t | x_{0..t-1}, g) \quad (21)$$

整个对话过程以任一方代理写出结束标志词<sup>①</sup>时结束。这时模型预测出选择  $o$ , 具体为从满足约束条件的可行输出选择集  $O$  中选择概率最高的那个。

$$o^* = \operatorname{argmax}_{o \in O} \prod_i p_\theta(o_i | x_{0..T}, g) \quad (22)$$

值得注意的是, 此篇论文在这里使用了“feasible set”一词。个人通过实验推测其原因为: 若不建立可行输出选择集, 直接选取概率最高的输出选择, 可能会出现该选择不满足约束条件, 即可能项目数量直接高于初始项目数量的情况。此时已不属于谈判失败, 而属于代理非智能。因此, FAIR 在这里建立了可行输出集。同时 FAIR 在初始项目数量时, 设置了总数在 5-7 间的规定, 有效降低了枚举可行输出选择的时间复杂度。因此 FAIR 在设立这一规定时就考虑了这一因素, 这个技巧非常巧妙, 值得借鉴。

## 4 强化学习

3 中的监督学习强调模仿谈判代理人的对话, 但并没有明确的表现出想要获取最大项目价值的意图。因为在这一节, FAIR 利用 3 中的学习结果进行强化学习。类似的方法在 Li et al.<sup>[8]</sup> 和 Das et al.<sup>[9]</sup> 中也有体现。

在强化学习阶段, 代理  $A$  通过读取代理  $B$  的表述来对自己的模型参数进行优化。虽然另外一个代理  $B$  可以是人, FAIR 先用之前监督学习出的模型代替人进行尝试。然而却发现, 当两个机器代理同时更新自身模型参数时, 它们的对话会与人类语言出现偏离<sup>②</sup>。因此 FAIR 对此节的强化学习模型进行了修正。

在该修正后模型中, 代理  $A$  首先读取它的输入项目数量与价值  $g$ , 然后产生表述  $x_{0..n}$ 。当  $x$  产生结束表述标记时, 接着从代理  $B$  那里读取它的回应  $x_{n+1..m}$ 。这样往复直到有一方代理产生了结束标志词。此时, 两位代理同时输出选择  $o$ , 并记录对应的价值总分<sup>③</sup> (以下简称分数)。为方便区分两位代理, 这里用  $X^A$  表示代理  $A$  的行为, 即产生的词和选择。

在一个完整的对话生成后, FAIR 根据谈判结果对代理  $A$  的模型参数进行更新。这里用  $r^A$  表示代理  $A$  最终分数,  $T$  表示对话长度。 $\gamma$  表示分数的影响因子, 该因子距离对话结束

• end-of-dialogue, 即结束标志词, 如 “Deal”、“OK” 和 “Okey” 等。

• 此处的偏离是指语法的偏离而非词汇的偏离, 词嵌入的特性使得 decode 过程不会出现新词。

• Reward, 即此处的价值总分。若两位代理的选择之和不等于各项目总数, 则认为谈判失败即 “Disagree”, 此时两位代理的价值总分均为 0。



越短影响则越强。 $\mu$  表示代理的平均谈判分数，由此 FAIR 定义了代理行为  $x_t \in X^A$  的期望谈判分数  $R$ ：

$$R(x_t) = \sum_{x_t \in X^A} \gamma^{T-t} (r^A(o) - \mu) \quad (23)$$

FAIR 在这里通过计算代理每一步的期望谈判分数来对参数进行优化：

$$L_\theta^{RL} = \mathbb{E}_{x_t \sim p_\theta(x_t | x_{0..t-1}, g)} [R(x_t)] \quad (24)$$

梯度的计算方法如 Williams<sup>[10]</sup>，为：

$$\nabla_\theta L_\theta^{RL} = \sum_{x_t \in X^A} \mathbb{E}_{x_t} [R(x_t) \nabla_\theta \log(p_\theta(x_t | x_{0..t-1}, g))] \quad (25)$$

值得注意的是，个人认为，FAIR 在这里将模型的优化问题转变成了在给定公式下搜索最优解问题，由此使用梯度下降求解最优，完成优化。

## 5 对话推演

小节3.4中所提的 Likelihood-based decoding 具有一定的缺陷。在谈判过程时，代理有两类策略，一是接受对方的要求，二是给出自己的要求，即“Counter Offer”。接受要求这一行为在 Likelihood-based decoding 中会具有更高的概率，因为接受要求比给出要求更易达成“Deal”。

本节中所提出的 Goal-based decoding 将允许更多的谈判策略。例如，在考虑项目价值和分数之后，代理可能会采用欺骗的手段<sup>⑥</sup>的获取较高价值的项目，从而拿到更高的分数。

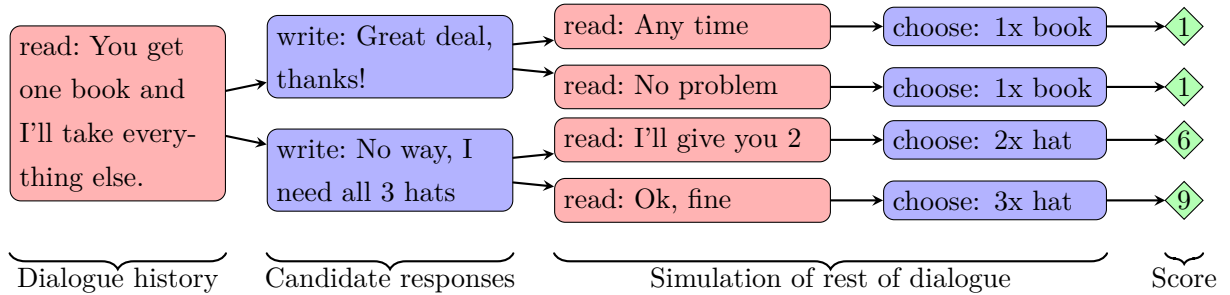


图 6: 对话推演

与强化学习不同，FAIR 在本节利用代理行为概率  $p_\theta$  计算期望谈判分数，并且这里的计算考虑代理的未来行为，如图6。FAIR 通过两个阶段来实现对话推演。首先，以  $U = u_{0..c}$  表示整个谈判过程中的表述<sup>⑥</sup>，以  $x_{0..n-1}$  表示当前的对话历史。通过计算  $x_{n+k+1,T}$  对应的  $p_\theta$  来获取  $u = x_{n,n+k}$ 。再通过计算  $u = x_{n,n+k}$  对应的  $R(u)$  来挑选最优的输出选择  $o$ 。因此该方法下的期望价值分数与对话中的表述相关：

$$R(x_{n..n+k}) = \mathbb{E}_{x_{(n+k+1..T;o)} \sim p_\theta} [r(o)p_\theta(o|x_{0..T})] \quad (26)$$

• 在人-人谈判训练集中，人的表述往往是诚实的，能够较直观的反应自己的需求。

• utterances，包括已出现和可能出现的表述。

Model	vs. likelihood				vs. Human			
	Score (all)	Score (agreed)	% Agreed	% Pareto Optimal	Score (all)	Score (agreed)	% Agreed	% Pareto Optimal
likelihood	5.4 vs. 5.5	6.2 vs. 6.2	87.9	49.6	4.7 vs. 5.8	6.2 vs. 7.6	76.5	66.2
rl	7.1 vs. 4.2	7.9 vs. 4.7	89.9	58.6	4.3 vs. 5.0	6.4 vs. 7.5	67.3	69.1
rollouts	7.3 vs. 5.1	7.9 vs. 5.5	92.9	63.7	5.2 vs. 5.4	7.1 vs. 7.4	72.1	78.3
rl+rollouts	8.3 vs. 4.2	8.8 vs. 4.5	94.4	74.8	4.6 vs. 4.2	8.0 vs. 7.1	57.2	82.4

表 1: 模型谈判结果

接下来返回最大期望价值分数对应的表述:

$$u^* = \operatorname{argmax}_{u \in U} R(u) \quad (27)$$

FAIR 的对话推演为 5 次, 预测轮数为 10 轮。

## 6 实验

### 6.1 Training Details

FAIR 在实验中使用的科学计算工具为 PyTorch。输入的项目数量和价值被嵌入至 64 维的线性空间, 对话中的词则被嵌入至 256 维的线性空间<sup>⑥</sup>。对应的  $\text{GRU}_w$ ,  $\text{GRU}_g$ ,  $\text{GRU}_{\vec{d}}$  和  $\text{GRU}_{\vec{v}}$  的隐藏层单元分别为 64、128、256 和 256。

在小节3.3Supervised Learning 中, 使用了随机梯度下降的方法搜索预测损失的最小值, 其中最小批次采样单元为 16 个, 以此避免落入局域陷阱, 公式20中的  $\alpha$  为 0.5。模型的初始学习速度为 1.0, Nesterov 动量为 0.1, 梯度速度为 0.5。在此基础训练了 30 次后, 然挑选效果最后的模型结果, 并开始对学习速度进行退火处理。

此外, 训练与测试数据集中并没有包含人-人谈判失败的情况, 且出现次数少于 20 次的词作废词处理。

在节4强化学习中, 学习速度为 0.1, 梯度速度为 1.0, 公式23中的  $\gamma$  为 0.95。在 4 次强化学习后, FAIR 同样采用随机梯度下降进行优化, 学习速度为 0.5。

### 6.2 Comparison Systems

FAIR 进行了以下模型谈判比较: LIKELIHOOD 使用了节3中的 supervised training 和 decoding; RL 使用了节4中的 goal-based selfplay; ROLLOUTS 使用了节3中的 supervised training 和节5中的 goal-based decoding; RL+ROLLOUTS 使用了节5中的 rollout。

Metric	Dataset
Number of Dialogues	5808
Average Turns per Dialogue	6.6
Average Words per Turn	7.6
% Agreed	80.1
Average Score (/10)	6.0
% Pareto Optimal	76.9

表 2: 人-人谈判数据集信息

Model	Valid PPL	Test PPL	Test Avg. Rank
likelihood	5.62	5.47	521.8
rl	6.03	5.86	517.6
rollouts	-	-	844.1
rl+rollouts	-	-	859.8

表 3: 词汇复杂度和平均谈判轮数排名

### 6.3 Evaluation

表1展示了 4 种模型分别 LIKELIHOOD 模型及人类谈判的结果。评价指标有 4 种, 分别为平均总体得分、平均成交得分、成交率和帕累托最优率。其中, 若两位代理的最终选择出现分歧, 即选择的各项目之和不等于初始各项目之和, 则视为谈判失败, 所以得分 “Score” 分为平均总体得分和平均成交得分; 成交率为谈判成功的次数与总谈判次数的比值; 帕累托最优率为两位代理的输出均已达到价值分数最高, 其中任何一方均没有办法再进行提交的次数与谈判成功次数的比值。

表2展示了人-人谈话数据集的结果。

表3展示了各模型与人类的差别, 其中 Valid PPL、Test PPL 和 Test Avg. Rank 分别代表了 LIKELIHOOD 模型产生回答的有效复杂度、测试复杂度和平均谈判轮数, 指标越低表示越接近人类。由表3可知, LIKELIHOOD 模型最接近人。值得注意的是, 模型间的差异有可能是由于 RL、ROLLOUTS 和 RL+ROLLOUTS 更复杂化的谈判策略造成的。

通过表1可知, ROLLOUTS 和 RL+ROLLOUTS, 尤其是 RL+ROLLOUTS 与 LIKELIHOOD 模型相比具有明显优势。RL+ROLLOUTS 模型帕累托最优率的提高展示了更好的选择方案, 这种结果验证了强化学习与对话推演比单纯模仿人类在谈判上更具有优势。

## 7 现象

RL+ROLLOUTS 模型的谈判明显要比 LIKELIHOOD 模型花费更多的轮数, 这表明 RL+ROLLOUTS 谈判起来更加努力。但在一些情景下, RL+ROLLOUTS 模型更倾向在每一轮中坚持价值最大的需求仅做出表述方式上的改变。这在实际与人类谈判时, 人类更有可能在较少轮数内直接放弃谈判而不会像机器一样接受或是最终 disagree。

欺骗是一种高级的谈判策略, FAIR 发现 RL+ROLLOUTS 模型在某些情景中假装对价值底的项目感兴趣, 最终 “妥协” 取得高价值项目, 如表4。

FAIR 认为此篇论文所提出的模型能够产生流畅的对话语句。不过个人通过观察数据集认为, 首先训练集和测试集中的人-人对话语句内词汇量较少, 语法结构较为简单。因此流畅的对话语句不足为奇。FAIR 也提出会在将来扩充表述的多样性。

• 具体的词嵌入方法请参考小节3.2和 Cho et al.<sup>[4, 5]</sup>。

Input		
rl+rollouts	4xbook value=0 2xhat value=5 1xball value=0	
Human	4xbook value=1 2xhat value=2 1xball value=2	
rl+rollouts	You can have the ball if I can have everything else.	
Human	No can't do that the ball is not going to give me anything	
rl+rollouts	I need the hats and you can have the rest	
Human	Deal	
rl+rollouts	Great! Thank you!	
Output		Reward
rl+rollouts	2xhat	10/10
Human	4xbook 1xball	6/10

表 4: 模型示例

FAIR 发现 RL+ROLLOUTS 模型的一种语言错误普遍发生在 agreement 的语境下却提出了更进一步的要求即 counter offer。这一行为很少在人类身上发生。FAIR 认为出现这种情况的原因在于，在训练集中，agree 往往处于谈判结束前，而且对方代理很少就 agree 再进行谈判。因此，RL+ROLLOUTS 模型可能认为在谈判快要提出 agree 时进行 counter offer 有助于对方接收该要求。

## 参考文献

- [1] Chaim Fershtman. The importance of the agenda in bargaining. *Games and Economic Behavior*, 2(3):224 – 238, 1990.
- [2] David Devault, Johnathan Mell, and Jonathan Gratch. Toward natural turn-taking in a virtual human negotiation agent. In *national conference on artificial intelligence, national conference on artificial intelligence*, 2015.
- [3] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. *arxiv preprint. arXiv preprint arXiv:1611.08669*, 1, 2016.
- [4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
- [5] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *empirical methods in natural language processing, empirical methods in natural language processing*, pages 1724–1734, 2014.
- [6] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, and Valentin Robu. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198:73–103, 2013.
- [7] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.
- [8] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.
- [9] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. *arXiv preprint arXiv:1703.06585*, 2017.
- [10] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.
- [11] Victoria Talwar and Kang Lee. Development of lying to conceal a transgression: Children’s control of expressive behaviour during verbal deception. *International Journal of*

- Behavioral Development, 26(5):436–444, 2002. doi: 10.1080/01650250143000373. identifier: CDY8W3RB90T0VM9L.
- [12] Jason D. Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech Language*, 21(2):393–422, 2007.
- [13] David Traum, Stacy C. Marsella, Jonathan Gratch, Jina Lee, and Arno Hartholt. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents, 2008.
- [14] Nicholas Asher, Alex Lascarides, Oliver Lemon, Markus Guhe, Verena Rieser, Philippe Muller, Stergos Afantenos, Farah Benamara, Laure Vieu, and Pascal Denis. Modelling strategic conversation: The stac project. *Proceedings of SemDial*, page 27, 2012.
- [15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [16] Matthew Henderson, Blaise Thomson, and Jason Williams. The second dialog state tracking challenge. volume 263, 2014.
- [17] Junhua Mao, Xu Wei, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan L. Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. pages 2533–2541, 2015.
- [18] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.
- [19] Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016.
- [20] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. *arXiv preprint arXiv:1704.07130*, 2017.