Finetuning Neural Conversational Models for Auxiliary tasks with Deep Reinforcement Learning

Персиянов Дмитрий, группа 394

Аннотация

В современном мире большую популярность набрали диалоговые модели на основе рекуррентных нейронных сетей. Обучение моделей происходит на огромных корпусах текстов. К сожалению, для применения их в прикладных задачах (например, чат-поддержка в банке, персональный помощник пользователя) необходимо, чтобы диалоговые модели соответствовали определенным, иногда жестким требованиям. В данной работе предлагается способ дообучения диалоговых моделей под вспомогательные задачи на основе policy-gradient алгоритмов обучения с подкреплением. Предложенная методика не требует большого количества данных и они не требует от них никакого определенного вида, в отличие дообучения по методу максимального правдоподобия.

Содержание

1 Введение

Люди все больше взаимодействуют с компьютером через естественные диалоговые интерфейсы. Классические подходы для построения goal-oriented диалоговых систем (Amazon Alexa, Microsoft Cortana, Google Now, Apple Siri) базируются на понятиях интента и слотов. Они требуют данных из предметной области с соответствующей разметкой интентов и слотов ([1], [2], [3], [4])

В то же время есть open-domain conversational models, которые в основном работают на sequence-to-sequence моделях ([5], [6], [7]). Эти модели обучаются на огромных датасетах с диалогами не из предметной области, так как не всегда таковые имеются. Часто возникают задачи дообучать их под дополнительные задачи, которые формализуются с помощью функции награды за сгенерированный ответ.

Интерес к обучению с подкреплением снова вырос за последние два года. Успех его интеграции с глубоким обучением подкрепляется статьями DeepMind [8], [9]. Обучение с подкреплением позволяет работать с недифференцируемыми функциями наград, что открывает широкие возможности применения таких алгоритмов для дообучения диалоговых моделей ([10], [11], [12], [13]).

В данной работе предлагается подход, требующий относительно небольшое количество времени и данных для дообучения моделей и основанный на обучении с подкреплением. Рассматриваются задачи поощрения/запрета списка слов для модели, а также задача генерации ответов в стиле какой-либо персоны.

2 Нейросетевые диалоговые модели

Будем рассматривать sequence-to-sequence диалоговые модели на основе рекуррентных нейронных сетей ([6], [5]). Такие модели состоят из двух рекуррентных сетей – энкодер и декодер.

Энкодеру подается на вход предложение $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, где x_i – векторное представление i-го слова предложения. Энкодер поддерживает внутреннее состояние $\mathbf{h_t} = f(\mathbf{h_{t-1}}, x_t)$, которое изначально инициализируется нулями или случайными числами. После обработки предложения скрытое состояние энкодера $\mathbf{h_n}$ трактуется как латентное представление входного предложения и используется для инициализации скрытого состояния декодера.

Декодер инициализируется последним скрытым состоянием $\mathbf{h_n}$ энкодера, принимает на вход служебный токен **BOS** и генерирует последовательность слов $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m\}$, минимизируя кроссентропию между распределением $p(\hat{\mathbf{y}}|\mathbf{x})$ и истинным дискретным распределением ответа $p(\mathbf{y}|\mathbf{x})$. Слово $\hat{y}_i$ генерируется на основе скрытого состояния декодера $\mathbf{h_i^{dec}}$ и $i-1$-го слова из истинного ответа:

$$p(\hat{y}_i|y_{i-1}, \ldots, y_1, \mathbf{x}) = g(\mathbf{h_i^{dec}}, y_{i-1}) \tag{1}$$

Функцией потерь в задаче обучения диалоговой модели является кроссэнтропия:

$$L(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{|V|} y_{tj} \cdot \log(\hat{y}_{tj}), \tag{2}$$

где $|V|$ – размер словаря, T – длина последовательности, y_t – one-hot представление правильного t-го слова в ответе, а $\hat{y}_t$ – вероятностное распределение, полученное от модели.

3 Обучение с подкреплением и policy-gradient методы

В этом разделе ставится задача обучения с подкреплением, описываются policy-gradient методы для ее решения. Также описывается постановка данной задачи в рамках диалоговых моделей.

Назовем **средой** марковский решающий процесс $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, где $\mathcal{S}$ – множество (возможно бесконечное) состояний, $\mathcal{A}$ – множество (возможно бесконечное) допустимых действий агента, $P = P(s'|s, a)$ – динамика среды, $r = r(s, a)$ – средняя награда при совершении агентом действия a из состояния s, γ – фактор дисконтирования.

Назовем **агентом** (политикой) распределение $\pi(a|s) : \mathcal{A} \times \mathcal{S} \to [0, 1]$. Агент взаимодействует со средой во времени. В каждый момент времени t агент находится в состоянии s_t, совершает действие a_t, получает от среды награду r_t и переходит в следующее состояние s_{t+1}. Задача агента – максимизировать дисконтированную суммарную награду $G_0 = \sum_{i=0}^{T} \gamma^i r_i$. Введем обобщение суммарной награды $G_t = \sum_{i=t}^{T} \gamma^i r_i$.

Будем рассматривать параметризованные стохастические политики $\pi_\theta(a|s)$ и пытаться максимизировать суммарную награду. Функционал полезности политики $\eta(\pi) := \mathbb{E}[G_0]$. В ранних работах ([14], [15]) был установлен основной метод для такой оптимизации, основанный на policy-gradient теореме. Следуя ему, политику можно обновлять стохастическим градиентным спуском:

$$\Delta\theta = \sum_{t=0}^{T-1} G_t \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{3}$$

Обновления весов в точности по (3) приводят к методу REINFORCE. В таком обновлении есть недостаток, связанный с большой дисперсией G_t, что свойственно оценкам Монте-Карло. В [15] показано, что выражение

$$\Delta\theta = \sum_{t=0}^{T-1} (G_t - V(s)) \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{4}$$

также является градиентом $\eta(\pi) := \mathbb{E}[G_0]$. Добавка $V(s)$ является оценкой value-function, также называется бейзлайном. Обновление весов по (4) приводит к серии методов Actor-Critic (или A2C).

В нейросетевой диалоговой модели политикой естественно принять распределение на словах, полученное от декодера. Параметры политики это параметры всей диалоговой модели. Также можно параметризовать политику лишь параметрами декодера, а энкодер оставлять неизменным.

Действиями в данном контексте будут генерируемые слова, а состоянием агента – скрытое состояние декодера.

4 Постановка задачи

Пусть дана диалоговая модель $G_\theta : \mathbf{x} \to \hat{\mathbf{y}}$, отвечающая на сообщение $\mathbf{x}$ сообщением $\mathbf{y}$. Пусть задана функция награды за ответ $R(\mathbf{y}, \hat{\mathbf{y}})$. Нашей задачей будет являться оптимизация параметров θ с целью максимизации средней награды по выборке $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N$:

$$\frac{1}{N} \sum_{i=1}^{N} R(\mathbf{y}_i, G_\theta(\mathbf{x}_i)) \underset{\theta}{\to} \max \tag{5}$$

Заметим, что постановка задачи достаточно общая, что выражается в зависимости функции награды как от правильного ответа, так и от сгенерированного моделью. В наших экспериментах функция наград, как правило, будет содержать два слагаемых, первое из которых соответствует оптимизации модели под конкретную задачу, а второе будет соответствовать сохранению кроссэнтропии на выборке.

5 Эксперименты

В качестве диалоговой модели использовалась 1-слойная LSTM сеть с размером скрытого слоя 1024. Последнее состояние энкодера использовалось не только для инициализации

декодера, но и подавалось в него на каждом моменте времени. В качестве входа в энкодер подавались 3 последние предложения из контекста. Модель обучалась на английских субтитрах с opensubtitles.org. Размер датасета – 18 миллионов пар контекст-ответ.

Общая функция потерь состоит из двух слагаемых – RL функции потерь из 4 и LLH функции потерь из стандартной диалоговой модели с каким-то весом α (в экспериментах $\alpha = 5$). Это необходимо для того, чтобы распределение декодера не вырождалось и присутствовала языковая структура в ответах.

5.1 BePolite

В качестве первого эксперимента была поставлена следующая задача: по данному списку "запрещенных"слов дообучить модель с целью убрать "запрещенные"слова из ответов модели, при этов сохранив языковую структуру в ответе, то есть сохранив кроссэнтропию около значения, которое было достигнуто обучением диалоговой модели. Был собран список из 250 "запретных"слов. За генерацию любого слова из этого списка агенту давалась награда $r_t = -1$.

	Базовый seq2seq	A2C
All	-0.256	-0.024
Target list conditioned	-0.293	-0.028

Таблица 1: Средние награды базовой модели и дообученной с помощью A2C.

В таблице 1 приведены средние награды за ответ для базовой модели и ее дообученной версии. Строка "All"соответствует средним наградам по любым входным предложениям, строка "Target list conditioned"соответствует средним наградам за ответ по входным предложениям, которые содержат в себе хотя бы одно слово из списка. Базовая модель генерирует хотя бы одно "запрещенное"слово в 25 случаях из 100 (29 из 100 для "Target list conditioned"), а дообученная в 2 из 100 (3 из 100).

Время дообучения модели составило 0.5 часа на одной видеокарте GTX 1080. Было обработано 800 случайных минибатчей по 64 примера (50 тысяч примеров из 18 миллионов).

<TODO>: Вставить показатели перплексии до и после дообучения, показав что модель не сильно ухудшилась.

5.2 BeLikeX

Следующий эксперимент будет заключаться в том, чтобы заставить модель говорить как какой-то человек. Для этого берется дискриминатор $D(\mathbf{x})$, который выдает число – меру похожести реплики $\mathbf{x}$ на реплики персоны. Обучать его можно, имея примеры фраз нужной персоны и примеры фраз любых других персон. В дальнейшем этот дискриминатор используется как функция награды при обучении модели.

Список литературы

[1] Tiancheng Zhao and Maxine Eskénazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *CoRR*, abs/1606.02560, 2016.

[2] Jason Williams, Nobal B. Niraula, Pradeep Dasigi, Aparna Lakshmiratan, Carlos Garcia Jurado Suarez, Mouni Reddy, and Geoffrey Zweig. Rapidly scaling dialog systems with interactive learning. January 2015.

[3] Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei-hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. Multi-domain dialog state tracking using recurrent neural networks. *CoRR*, abs/1506.07190, 2015.

[4] Julien Perez. Dialog state tracking, a machine reading approach using a memory-enhanced neural network. *CoRR*, abs/1606.04052, 2016.

[5] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[6] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. *CoRR*, abs/1603.06155, 2016.

[7] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808, 2015.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[9] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[10] Kavosh Asadi and Jason D. Williams. Sample-efficient deep reinforcement learning for dialog control. *CoRR*, abs/1612.06000, 2016.

[11] Antoine Bordes and Jason Weston. Learning end-to-end goal-oriented dialog. *CoRR*, abs/1605.07683, 2016.

[12] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *CoRR*, abs/1606.01541, 2016.

[13] Zachary C. Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. Efficient exploration for dialog policy learning with deep BBQ networks \& replay buffer spiking. *CoRR*, abs/1608.05081, 2016.

[14] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. A Bradford book. Bradford Book, 1998.

[15] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, pages 1057–1063, 1999.