

# Диалоговые системы

8 декабря 2016 г.

- 1 Диалоговые системы и диалоговые менеджеры
- 2 Целеориентированные диалоговые менеджеры (Goal-oriented)
- 3 General conversation

Диалоговый менеджер моделирует взаимодействие пользователя и «агента» (программа, бот) путем последовательности сообщений на естественном языке:  $(a_1, u_1), (a_2, u_2), \dots (a_t, u_t)$ . Задача агента — правильно выбрать следующее действие или реплику  $a_{t+1}$ .

- Диалоги:
  - Целеориентированные (Goal-oriented): решение конкретной задачи пользователя (звонок в колл-центр, заказ авиабилетов),
  - Беседа на общие темы (General conversation): выдача уместных реплик (чат-боты, тесты Тьюринга)
- Интерфейс: речь, текст

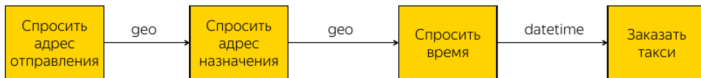
- Базовая структура: приветствие → основная тема → завершение
- Модели:
  - Stack-model: диалог группируется по интендам пользователя
  - Cache-model: диалог группируется по лингвистическим фрагментам
- Инициативы: системная, пользовательская и смешанная
- Интерпретация пользовательских запросов: использование контекста диалога, стратегии подтверждения

# Проблемы моделирования диалогов

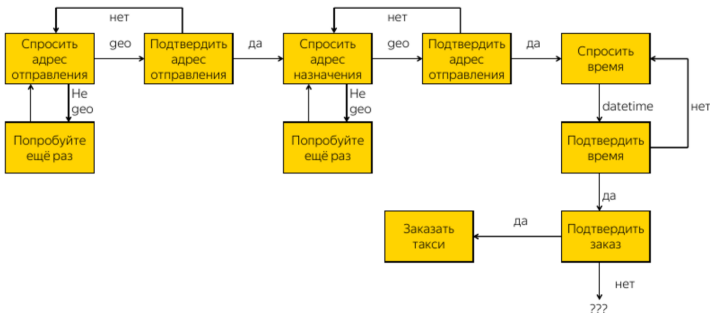
- Эллиптические конструкции
  - A Какая погода в Москве?
  - B +10 C
  - A А завтра?
- Анафоры и кореференции: местоимения, отсылки к упомянутым сущностям (там/здесь, тот/этот)
- Вставные реплики:
  - A Ты куда идешь?
  - B А зачем тебе знать?
  - A Хочу пойти с тобой
  - B Я иду в магазин.
- Ненаправленность:
  - A Телефон.
  - B Я в ванной.
  - A Хорошо.
- Речевые акты: например, "Здесь холодно."
- Несмысловые реплики (ага, угу, ммм)

# Goal-oriented

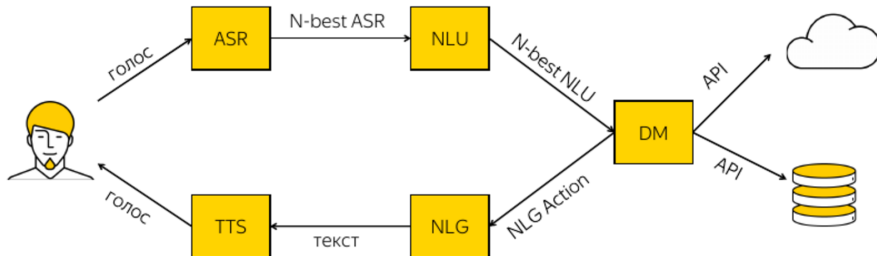
- Цель - решить конкретную задачу пользователя, например, заказать такси:



- Сценарии могут быть непростыми:



# Goal-oriented: основные компоненты



# Основные компоненты: Automatic Speech Recognition (ASR)

- По наблюдаемому речевому сигналу  $O = o_1, o_2, \dots$  вычислить наиболее вероятную последовательность слов  $W = w_1, w_2, \dots, w_T$ :

$$W = \operatorname{argmax}_w P(O|W)P(W)$$

- Чтобы учитывать ошибку распознавания, необходимо иметь  $N$ -лучших гипотез  $W_1, W_2, \dots, W_N$ , и соответствующие оценки для вероятностей высказывания  $P(W_1|O), P(W_2|O), \dots, P(W_N|O)$



# Основные компоненты: Natural Language Understanding (NLU)

Для каждой последовательности слов  $W_t$  вычислять семантическое представление  $s_t$ , содержащее информацию об **интенте** и его параметрах - **слотах**, например

Мне нужно такси на Льва Толстого, 16 через 30 минут

- **Интент**: заказ такси
- **Откуда**: улица Льва Толстого, дом 16
- **Когда**: через 30 минут
- **Куда**: ?

Также, как и в случае с ASR, может быть несколько гипотез.

# Основные компоненты: Natural Language Understanding (NLU)

- **Классификатор интенгов**

- подойдет любой текстовый классификатор (например, на основе Bag-of-Words),
- важное отличие - классификация по коротким высказываниям, может помочь посимвольная обработка <sup>1</sup>,
- для разрешения кореференций и эллипсов используются отдельные модели<sup>2</sup>

- **Семантический теггер** или Слот-филлер

- Каждому слову  $w_i$  ставит тег соответствующего слота  $slot_i$  или пустой тег  $\emptyset$
- Используется BIO-нотация: **B**egin, **I**nside, **O**utput
- Модели: Conditional Random Fields (CRF) или рекуррентные нейросети<sup>3</sup>

---

<sup>1</sup>X. Zhang, Character-level Convolutional Networks for Text Classification, 2015

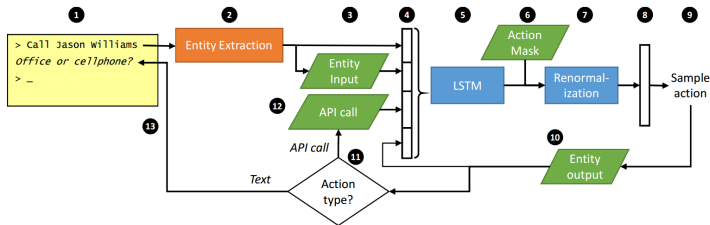
<sup>2</sup>S. Wizeman et al. Learning Global Features for Coreference Resolution, 2016

<sup>3</sup>G. Mesnil et al. Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding, 2015

На основании семантического представления  $s_t$  выбирает действие  $a_t$

- Простейший подход - табличный выбор  $s_t \rightarrow a_t$
- Supervised learning:  $a_t = \operatorname{argmax}_a p(a|s_{1:t})$
- Reinforcement learning: Markov Decision Processes (MDP) или Partially Observable MDP (POMDP)

- Предсказание по последовательности<sup>4</sup>



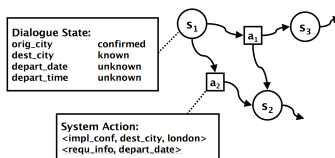
- Необходимо много данных - симуляция пользователя

<sup>4</sup>J. D. Williams End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning, 2016

# Reinforcement learning

Пусть каждому выбранному действию  $a_t$  соответствует награда  $r_t$ .  
Задача агента - выбрать такую стратегию действий  $\pi$  (policy), которая максимизирует суммарный выигрыш.

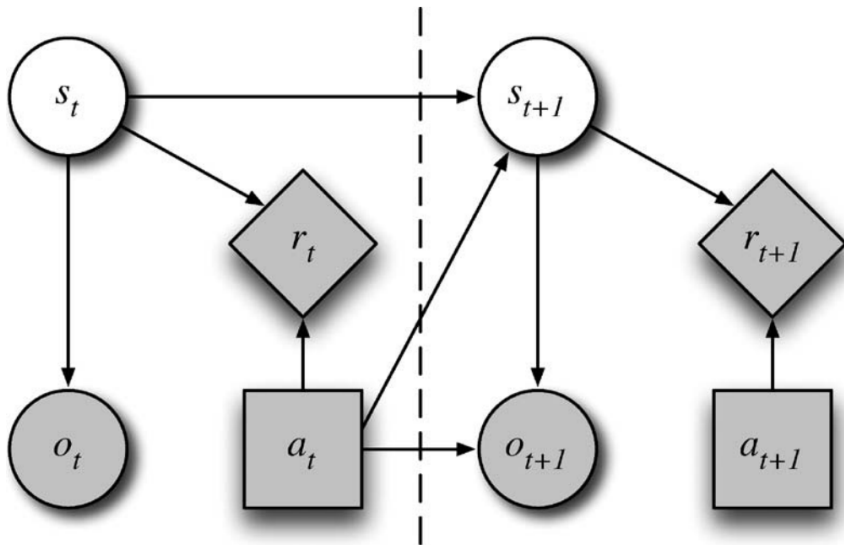
- MDP:  $\mathcal{S}, \mathcal{A}, P(s'|a, s), s_0, R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$



- POMDP:  $\mathcal{S}, \mathcal{A}, \mathcal{O}, b_0, P(s'|a, s), P(o_t|s_t, a_t), R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$

$$\begin{aligned} b_{t+1}(s_{t+1}) &= P(s_{t+1}|o_{t+1}, a_t, b_t) \\ &= \nu P(o_{t+1}|s_{t+1}, a_t) \sum_{s_t} P(s_{t+1}|s_t, a_t) b_t(s_t) \end{aligned}$$

# Reinforcement learning: POMDP <sup>5</sup>



<sup>5</sup>S. Young et al. POMDP-Based Statistical Spoken Dialog Systems: A Review, 2013

- Суммарный выигрыш:  $R_t = \sum_{k=1}^{\infty} \gamma^k r_{t+k} = r_t + \gamma R_{t+1}$ ,  $0 \leq \gamma \leq 1$
- Пусть есть некоторая стратегия  $\pi$ . Q-функцией называется функция

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}(R | s_t = s, a_t = a)$$

- Задача - найти оптимальную стратегию

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$$

- Итерационный алгоритм:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a'))$$

---

<sup>6</sup>S. Russel, P. Norvig Artificial intelligence (3<sup>rd</sup> edition), глава 21

- $\epsilon$ -стратегия: вместо жадной стратегии выбора  $a'$ , максимизирующего  $Q$ -функцию, с вероятностью  $\epsilon$  выбираем случайное действие (exploration-exploitation)
- Если пространство  $\mathcal{S}$  слишком велико - применяем нейронные сети:

$$Loss(a|s) = \frac{1}{2} \left( r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)^2$$



- Идея: вместо того, чтобы сначала строить Q-функцию, можно сразу искать оптимальную стратегию <sup>7</sup> среди вероятностных распределений, т.е.  $\pi \sim p(a|s_t, \theta)$

$$J(\theta) = \mathbb{E}(R) = \sum_{a_{1:T} \in \mathcal{A}} p_{\theta}(a_{1:T}) R(a_{1:T}) \rightarrow \max$$

- На практике используется приближение с помощью стохастического метода градиентного спуска:

$$\begin{aligned} \nabla J(\theta) &= \sum_{a_{1:T} \in \mathcal{A}} \nabla p_{\theta}(a_{1:T}) R(a_{1:T}) \\ &= \sum_{a_{1:T} \in \mathcal{A}} p_{\theta}(a_{1:T}) \nabla \log p_{\theta}(a_{1:T}) R(a_{1:T}) \\ &= \mathbb{E}_{a_1} \mathbb{E}_{a_2} \dots \mathbb{E}_{a_T} \nabla \log p_{\theta}(a_t | a_{1:(t-1)}) \sum_{t=1}^T r(a_{1:t}) \end{aligned}$$

---

<sup>7</sup>J. Li et al. Deep Reinforcement Learning for Dialogue Generation, 2016

# Основные компоненты: Natural Language Generator (NLG) & Text-to-speech synthesis (TTS)

- Цель: исходя из полученного действия, сгенерировать текстовую реплику агента (NLG):  $a_t \rightarrow l_t$  и речевой сигнал (TTS)
- Для генерации  $l_t$  используются контекстные грамматики, реже - нейронные сети для генерации шаблонов <sup>8</sup>

---

<sup>8</sup>Tsung-Hsien Wen et al. Multi-domain Neural Network Language Generation for Spoken Dialogue Systems

- Цель - на каждом шаге диалога агентом должна быть сгенерирована реплика (reply), которая:
  - Синтаксически корректна
  - Связана по смыслу с предыдущим контекстом диалога (context)
  - По возможности не является слишком общим ответом (например, "я не знаю", "ага")
- Источники данных: OpenSubtitles, Twitter, различные мессенджеры
- Подходы:
  - **Порождающий**: моделируем  $P(reply|context)$
  - **Ранжирующий**: строим функцию  $Sim(reply, context)$

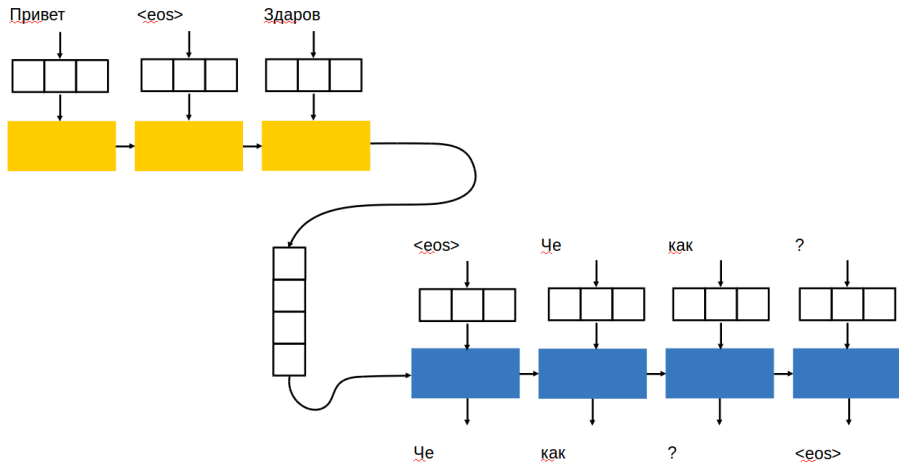
# General conversation: порождающий подход

- Модели заимствованы из технологий машинного перевода
- Стандартный подход: sequence-to-sequence<sup>9</sup>
- Рекуррентная encoder-decoder архитектура:
  - encoder прочитывает исходный контекст
  - decoder генерирует результат, опираясь на выход encoder'a

---

<sup>9</sup>Sutskever I. Vinyals O. Le Q. Sequence to Sequence Learning with Neural Networks, 2014

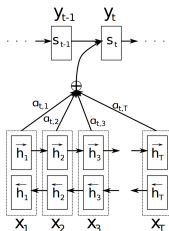
# General conversation: порождающий подход



- Обучение по методу максимального правдоподобия

$$\prod_{t=1}^T p_{\theta}(\text{reply}_t | \text{reply}_{1:(t-1)}, \text{context}) \rightarrow \max_{\theta}$$

- Генерация реплик:  $\text{argmax}_{\text{reply}} p_{\theta}(\text{reply} | \text{reply}_{1:(t-1)}, \text{context})$  - используется метод beam-search.
- Стандартные модификации: LSTM, GRU рекуррентные сети, двунаправленный encoder, сеть с вниманием<sup>10</sup>



<sup>10</sup>D. Bahdanau, Neural Machine Translation by Jointly Learning to Align and Translate, 2015

- Чтобы штрафовать обшие ответы, вместо максимума правдоподобия используется критерий совместной информации (MMI) <sup>11</sup>:
  - $\log P(reply|context) - \alpha \log P(reply)$
  - $(1 - \alpha) \log P(reply|context) + \alpha \log P(context|reply)$
- Для построения таких последовательностей можно также Монте-Карло оценку:

$$\log P(reply|context) - \frac{\alpha}{K} \sum_{k=1}^K \log P(reply|context_k)$$

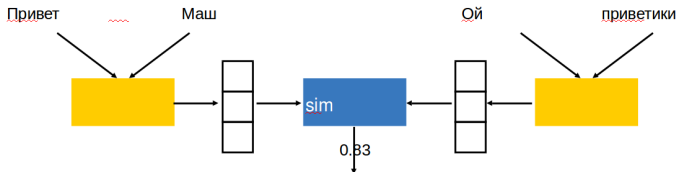
- Для обучения применяется метод, аналогичный policy-gradient: MMI функционал используется в расчете функции награды  $R$  <sup>12</sup>

---

<sup>11</sup>J. Li et al. A Diversity-Promoting Objective Function for Neural Conversation Models, 2016

<sup>12</sup>J. Li et al. Deep Reinforcement Learning for Dialogue Generation, 2016

# General conversation: ранжирующий подход



- Вместо генерации ответов можно выбирать наиболее уместные из заранее заготовленных
- Плюсы
  - Меньше риски сгенерировать грамматически некорректный ответ или ответ с обценной лексикой
  - Обучение намного быстрее
  - Меньше проявляется проблема «общих» ответов
- Минусы
  - Множество реплик сильно ограничено



## Модели <sup>13, 14</sup>

- Энкодеры реплик — любые сети, которые по набору слов могут получить вектор, например
  - рекуррентные или сверточные по словам или символам,
  - полносвязные поверх BoW моделей
- Функция уместности

$$Sim(reply, context) = \frac{h_{reply}^T h_{context}}{\|h_{reply}\| \|h_{context}\|}$$

- Для обучения нужны отрицательные примеры - применяем случайное семплирование

---

<sup>13</sup>Huang P. S. et al. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data, 2013

<sup>14</sup>Inaba M., Takahashi K. Neural Utterance Ranking Model for Conversational Dialogue Systems, 2016