Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Московский физико-технический институт (государственный университет)»

Факультет инноваций и высоких технологий

Кафедра анализа данных

Персиянов Дмитрий Андреевич

Обучение с подкреплением нейросетевых диалоговых моделей для вспомогательных целей

Выпускная квалификационная работа бакалавра

Направление подготовки: 01.03.02 Прикладные математика и информатика

Зам. зав. кафедрой	_____	/Бунина Е.И./
Научный руководитель	_____	/Устюжанин А.Е./
Студент	_____	/Персиянов Д.А./

г. Москва
2017

Обучение с подкреплением нейросетевых диалоговых моделей для вспомогательных целей

Д.А. Персиянов

Аннотация. В современном мире большую популярность набрали диалоговые модели на основе рекуррентных нейронных сетей. Обучение моделей происходит на огромных корпусах текстов. Однако, для применения их в прикладных задачах (чат-поддержка в банке, персональный помощник пользователя и т.д.) необходимо, чтобы модель соответствовала определенным, иногда жестким требованиям. В работе предлагается способ дообучения диалоговых моделей под эти требования на основе policy-gradient алгоритмов обучения с подкреплением. Предложенный подход не требует большого количества данных и не накладывает ограничений на их вид, в отличие от дообучения по методу максимального правдоподобия. На примере двух задач: запрета обсценной лексики в ответе и требования отвечать в "стиле" какой-либо персоны демонстрируется эффективность метода.

Содержание

1. Введение

Люди все больше взаимодействуют с компьютером через естественные диалоговые интерфейсы, которые помогают пользователю с различными повседневными задачами. Такие системы называют целеориентированными (англ. goal-oriented) диалоговыми системами. Классические подходы для построения целеориентированных диалоговых систем (Amazon Alexa, Microsoft Cortana, Google Assistant, Apple Siri, Яндекс Алиса) основаны на модуле отслеживании состояния диалога (англ. dialog state tracking), а также модуле распознавания естественного языка (англ. natural language understanding). Обучение этих компонент требует данных из предметной области со сложной разметкой ([1] [2] [3]).

С другой стороны, бурное развитие получили диалоговые нейросетевые модели, направленные на общение на общие темы (англ. open-domain). Их можно разделить на две категории: к первой относятся генеративные модели, основанные на рекуррентных сетях ([4] [5] [6]), ко второй модели, ранжирующие большой индекс возможных ответов ([7] [8]). Обучение происходит на огромных датасетах с диалогами и структурированная разметка, как правило, не требуется. В данной работе рассматриваются только генеративные диалоговые модели, для краткости будем называть их просто диалоговыми моделями (системами).

Для обучения диалоговых моделей требуются выборки с миллионами пар контекст-ответ, над которыми часто нет контроля. Таким образом модель может обучиться отвечать не так, как хотелось бы. Например, в ответах может присутствовать обсценная лексика или модель может отвечать оскорбительно на контексты, касающиеся культурных, расовых проблем и т.п. В связи с этим возникает задача обучения модели так, чтобы она удовлетворяла такого рода требованиям.

Следующая задача, возникающая при отсутствии большого специфичного датасета с репликами, состоит в том, чтобы научить модель общаться как какая-то персона и отвечать в стиле этой персоны. Обе вышеперечисленные задачи можно сформулировать на языке обучения с подкреплением, то есть в виде функции наград за сгенерированный ответ.

Интерес к обучению с подкреплением снова вырос в последние несколько лет. Успех его интеграции с глубоким обучением подкрепляется статьями группы DeepMind ([9] [10]). Обучение с подкреплением позволяет работать с недифференцируемыми функциями наград, что открывает широкие возможности применения таких алгоритмов для дообучения диалоговых моделей ([11] [12] [13] [14] [15]).

В данной работе предлагается подход, основанный на глубоком обучении с подкреплением и требующий небольшое количество времени и данных для обучения моделей. Рассматриваются задачи запрета списка слов для модели, а также задача генерации ответов в стиле какой-либо персоны. Предложенный подход значительно превосходит бейзлайны без обучения с подкреплением.

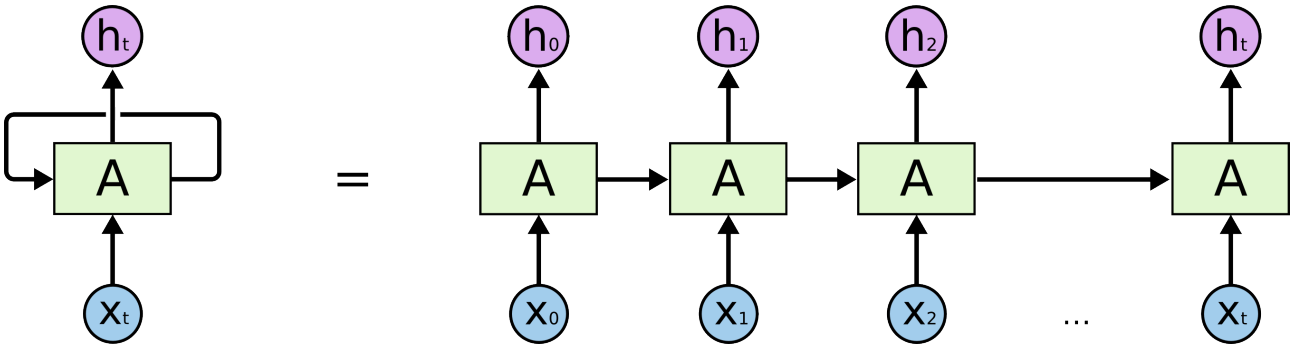
2. Нейросетевые диалоговые модели

В данной главе представлены базовые знания о рекуррентных сетях и sequence-to-sequence подходе для построения диалоговых систем.

2.1. **Рекуррентные нейронные сети.**

Рекуррентные нейронные сети органично подходят для работы с данными, в которых наблюдается последовательная структура. Примерами таких данных являются тексты (последовательность слов), временные ряды, видеоряды (последовательность кадров), звук.

Рис. 1. Схематичное устройство рекуррентной нейронной сети.



На Рис. 1 обозначено примерное устройство рекуррентной сети. Более формально, на вход сети подается последовательность входных векторов $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$, а она в свою очередь обрабатывает эту последовательность, поддерживая вектор скрытого состояния $\mathbf{h_t}$. Изначально (до обработки первого вектора входной последовательности) скрытое состояние инициализируется, например, нулями ($\mathbf{h_0} = \mathbf{0}$). Скрытое состояние обновляется следующим образом:

$$\mathbf{h_t} = \sigma(W^{(hh)}\mathbf{h_{t-1}} + W^{(hx)}\mathbf{x_t}), \tag{1}$$

где $\sigma(x) = \frac{1}{1+\exp^{-x}}$ – сигмоидная функция активации (для вектора применяется поточечно), а $W^{(hh)}$, $W^{(hx)}$ – матрицы обучаемых параметров рекуррентного слоя. Из формулы

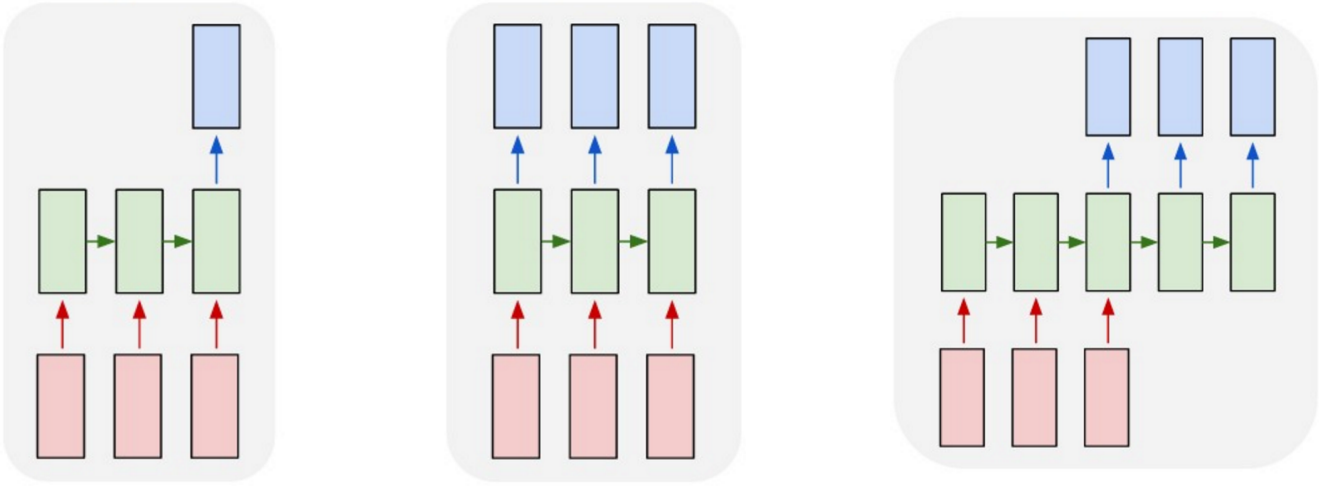
(1) понятно, что размерность вектора скрытого состояния можно варьировать и тем самым влиять на архитектуру сети и количество обучаемых параметров. В экспериментах в данной работе эта размерность будет равна 1024.

Преимущество рекуррентной архитектуры в возможности предсказывать целевые переменные различных типов. В общем случае, если входной последовательности $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ соответствует последовательность $y = \{y_1, y_2, \ldots, y_n\}$ целевых переменных для каждого элемента $\mathbf{x_t}$, как, например, в задаче POS-таггинга, предсказать такую последовательность $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ можно, используя скрытое состояние на каждом шаге:

$$\hat{y}_t = \text{softmax}(U\mathbf{h_t}), \tag{2}$$

где $\text{softmax}(\mathbf{v}) = \left(\frac{e^{v_1}}{\sum e^{v_i}}, \frac{e^{v_2}}{\sum e^{v_i}}, \ldots, \frac{e^{v_n}}{\sum e^{v_i}}\right)$ – классификационная функция активации, $U \in \mathbb{R}^K \times \mathbb{R}^{dim\mathbf{h_t}}$ – матрица параметров, K – количество классов.

Также возможно предсказывать целевую переменную после обработки входной последовательности, используя финальное скрытое состояние сети $\mathbf{h_n}$.



(A) (B) (C)

Рис. 2. Варианты структуры целевой переменной и ее предсказание.

Таким образом, рекуррентные сети подходят для широкого спектра задач предсказания на данных, имеющих последовательную структуру. На Рис. 2(a) проиллюстрирован

вариант предсказания целевой переменной после обработки входной последовательности, на Рис. 2(b) предсказание последовательности целевых переменных на каждом шаге, на Рис. 2(c) предсказание последовательности целевых переменных после обработки входной последовательности.

Вышеупомянутый способ (1) (т.н. "vanilla" ячейка) обновления скрытого состояния сети имеет свои недостатки, в числе которых популярная проблема затухания и "взрыва" градиентов ([16]). В связи с этим стали популярны другие виды рекуррентных ячеек, такие как LSTM и GRU. В данной работе во всех экспериментах используются ячейки LSTM, обновление в которых происходит следующим образом:

$$\mathbf{i_t} = \sigma(W^{(i)}\mathbf{x_t} + U^{(i)}\mathbf{h_{t-1}})$$

$$\mathbf{f_t} = \sigma(W^{(f)}\mathbf{x_t} + U^{(f)}\mathbf{h_{t-1}})$$

$$\mathbf{o_t} = \sigma(W^{(o)}\mathbf{x_t} + U^{(o)}\mathbf{h_{t-1}})$$

$$\tilde{\mathbf{c_t}} = \tanh(W^{(c)}\mathbf{x_t} + U^{(c)}\mathbf{h_{t-1}})$$

$$\mathbf{c_t} = \mathbf{f_t} \circ \mathbf{c_{t-1}} + \mathbf{i_t} \circ \tilde{\mathbf{c_t}}$$

$$\mathbf{h_t} = \mathbf{o_t} \circ \tanh(\mathbf{c_t})$$

Здесь $W^{(i)}$, $W^{(f)}$, $W^{(o)}$, $W^{(c)}$, $U^{(i)}$, $U^{(f)}$, $U^{(o)}$, $U^{(c)}$ – обучаемые матрицы параметров.

2.2. **Векторные представления слов в рекуррентных сетях.** Рекуррентные сети принимают на вход последовательности переменной длины, элементами которой являются векторы. Для того, чтобы обрабатывать текстовые данные рекуррентными сетями, необходимо уметь представлять слова в виде векторов. Формально, теперь на вход сети подается последовательность *целых чисел* $x = \{x_1, x_2, \ldots, x_n\}$, элементы которой отображаются в семантическое векторное пространство. Здесь x_i – номер в словаре V, соответствующий i-му слову в предложении. Словарь конечен, часто его размер составляет 10-200 тысяч

слов. Слова, которых нет в словаре, заменяются на специальный токен **UNK**, который кладется в словарь отдельно.

Каждому слову в словаре сопоставляется вектор фиксированной размерности d (типичные размерности 100-500). Таким образом, имеем **матрицу эмбеддингов** $W \in \mathbb{R}^{|V|} \times \mathbb{R}^d$.

Существуют три варианта построения векторных представлений для слов:

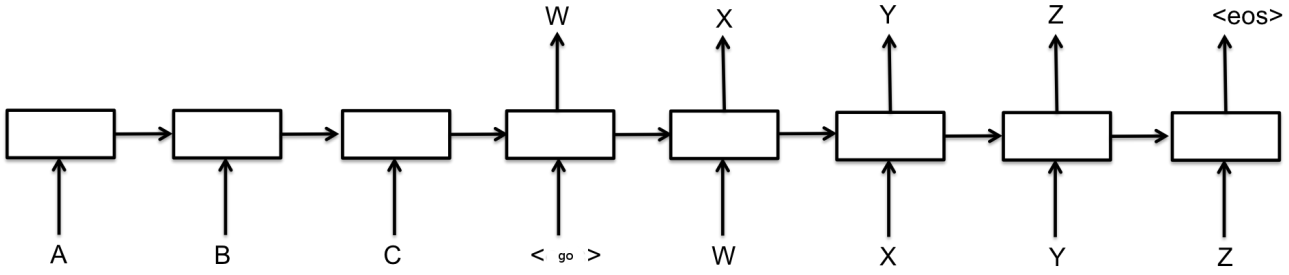
(1) Предобучить векторные представления на большом корпусе текстов, не обязательно том, на котором будет обучаться сеть. Обучение можно производить, например, алгоритмами word2vec, glove и их вариациями ([17] [18] [19]). Во время обучения сети векторные представления "замораживаются" и не обучаются вместе с параметрами сети.
(2) Инициализировать векторные представления случайными числами (аналогично параметрам сети) и обучать их вместе с параметрами сети на целевом датасете.
(3) Предобучить векторные представления как в первом пункте, а затем дообучать их во время обучения сети.

В настоящей работе используется второй вариант, размерности векторных представлений во всех моделях равны 512. Размер словаря 50000 наиболее частых слов в корпусах.

2.3. **Sequence-to-sequence подход.** Рассмотрим подход к построению диалоговых систем, называемый sequence-to-sequence. В его основе лежат две рекуррентные сети: **кодировщик** и **декодировщик**. Подход был впервые применен в машинном переводе ([20]), а впоследствии модифицирован под диалоговые системы ([4]).

На вход кодировщику поступает входное предложение – последовательность векторных представлений слов $\mathbf{x} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$. Кодировщик обрабатывает последовательность, в результате получая латентное представление $\mathbf{h_n^{enc}}$ предложения, которое впоследствии используется для инициализации скрытого состояния декодировщика. Схематичное представление модели можно видеть на Рис. 3.

Рис. 3. Схематичное устройство sequence-to-sequence диалоговой модели.



Пусть входному предложению $\mathbf{x}$ соответствует ответ из выборки $y = \{y_1, y_2, \ldots, y_m\}$. Ему соответствует последовательность векторных представлений $\mathbf{y} = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_m}\}$.

Скрытое состояние $\mathbf{h_0^{dec}}$ декодировщика инициализируется финальным скрытым состоянием кодировщика, то есть $\mathbf{h_0^{dec}} = \mathbf{h_n^{enc}}$. Процесс генерации ответа выглядит следующим образом: вначале декодировщик принимает на вход служебный токен **BOS** (точнее его векторное представление) и генерирует первое слово $\hat{y}_1$, семплируя из распределения $\hat{p}_1 = \text{softmax}(U\mathbf{h_1^{dec}})$. Затем векторное представление $\mathbf{\hat{y}_1}$ этого слова подается в качестве входа на следующем шаге, генерируется следующее слово и так далее. Процесс продолжается до тех пор, пока модель не сгенерирует служебный токен **EOS**. Для того, чтобы это случилось, на этапе обучения этот токен добавляется ко всем ответам в конец.

Однако, процесс обучения модели отличается от вышеописанного процесса генерации. На t-ом шаге генерации на вход сети подается векторное представление $(t-1)$-го слова из **правильного** ответа, то есть $\mathbf{y_{t-1}}$, а не слово, сгенерированное моделью на предыдущем шаге, как в случае с процессом генерации.

Формально, на этапе обучения модель моделирует факторизацию распределения "языковой модели":

$$p(y|x) = p(y_1|x)p(y_2|y_1, x) \cdot \cdots \cdot p(y_m|y_1, y_2, \ldots, y_{m-1}, x) \tag{3}$$

С помощью следующего параметризованного распределения:

$$p_\theta(y_t|y_1, y_2, \ldots, y_{t-1}, x) \approx p(y_t|y_1, y_2, \ldots, y_{t-1}, x), \tag{4}$$

где θ – все параметры диалоговой модели. В то же время, на этапе тестирования, распределение выглядит иначе:

$$p_\theta(y_t|\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{t-1}, x) \tag{5}$$

Эта разница между распределениями является известной проблемой, называющейся **несходством распределений** (англ. distribution discrepancy) и один из методов её решения предложен в статье [21]. Тем не менее, это не является большой проблемой для данной работы, но стоит заметить, что методы обучения с подкреплением неявно обходят эту проблему, так как оптимизируют модель используя **сгенерированные** ей ответы, а не правильные ответы из обучающей выборки.

Опишем функцию потерь на одном примере (x, y) для обучения диалоговой модели:

$$L(\theta) = -\frac{1}{m}\sum_{t=1}^{m} \log \hat{p}_{ty_t} \underset{\theta}{\to} \min, \tag{6}$$

где $|V|$ – размер словаря, m – длина последовательности, а $\hat{p}_{ty_t}$ – вероятность слова y_t на шаге t. Эта функция потерь – кроссэнтропия между распределениями $p_\theta(y_t|y_1, y_2, \ldots, y_{t-1}, x)$ и $p(y_t|y_1, y_2, \ldots, y_{t-1}, x)$.

На этапе тестирования модели, ответ генерируется жадным образом. На каждом шаге декодирование выбирается слово с максимальной вероятностью. Существуют улучшения данного метода, такие как beam-search ([22]).

3. Обучение с подкреплением

В этой главе описываются основы обучения с подкреплением, а также необходимый для дальнейшего понимания материал о policy-gradient методах.

3.1. **Введение в обучение с подкреплением.**

Обучение с подкреплением – класс методов для построения т.н. **агентов**, сущностей, взаимодействующих со **средой**. Агент может "чувствовать" среду (учитывать **состояния** среды) и действовать в ней. Целью агента является выбор оптимальных действий для достижения цели. Обучение с подкреплением отличается от классического машинного обучения, где присутствует выборка с правильными (оптимальными) ответами. Вместо этого, агенту необходимо исследовать среду методом проб и ошибок и понимать, как действовать в различных ситуациях. Например, на роботе установлены различные сенсоры в виде камер и дальномеров, с помощью которых он получает состояние среды, при этом робот может выбирать, в какую сторону ему двигаться. За каждое действие агент получает награду, таким образом понимая, что "хорошо" и что "плохо". Задачей агента является выучить **стратегию** выбора оптимальных действий в долгосрочной перспективе для достижения цели. Для этого агент поддерживает **кумулятивную награду** каждого состояния или каждой пары состояние-действие.

Обучение с подкреплением отличается от классического машинного обучения двумя свойствами:

(1) **Метод проб и ошибок.** Агент пробует различные действия в одном и том же состоянии, чтобы понять, какое из них ведёт к наибольшей кумулятивной награде. В классическом машинном обучении оптимальные метки даны сразу.

(2) **Запаздывающая награда.** Выполненные агентом действия влияют не только на награду, полученную в этот момент времени, но и на награды в будущем. В классическом машинном обучении предполагается, что все объекты выборки независимы.

Завершая введение, агент в обучении с подкреплением использует свой собственный прошлый опыт взаимодействия со средой, чтобы улучшить свою стратегию в долгосрочной перспективе.

3.2. **Марковский процесс принятия решений.** Назовем **средой** марковский процесс принятия решений $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, где $\mathcal{S}$ – множество (возможно бесконечное) состояний, $\mathcal{A}$ – множество (возможно бесконечное) допустимых действий агента, $P = P(s'|s, a)$ – динамика среды, $r = r(s'|s, a)$ – награда при совершении агентом действия a и переходе среды из состояния s в s', $\gamma \in [0, 1]$ – фактор дисконтирования. Вероятность $P(s'|s, a)$ есть вероятность среды перейти в состояние s' при условии, что до этого она находилась в состоянии s и агент совершил действие a. Фактор дисконтирования γ определяет значимость наград в будущем: награда со значением r, полученная на k моментов времени позже, будет в γ^{k-1} раз "дешевле", чем полученная немедленно.

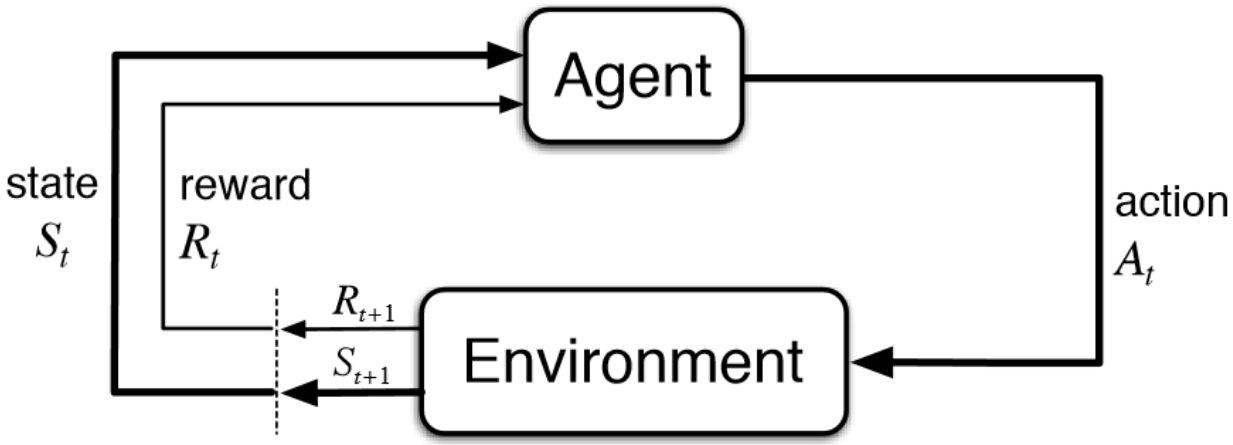
Введём сущность **агента**. Агент взаимодействует со средой во времени. В каждый момент времени t агент находится в состоянии s_t, совершает действие a_t, получает от среды награду r_t и переходит в следующее состояние s_{t+1}. Задача агента – максимизировать дисконтированную суммарную награду $G_0 =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

Общее представление о взаимодействии агента со средой представлено на Рис. 4

Важным предположением во всём обучении с подкреплением является **гипотеза о марковости среды**:

$$p(s_{t+1}, r_{t+1}|s_0, a_0, r_1 \ldots, s_{t-1}, a_{t-1}, r_t, s_t, a_t) = p(s_{t+1}, r_{t+1}|s_t, a_t) \tag{7}$$

Рис. 4. Взаимодействие агента со средой.



Это свойство позволяет предсказать следующее состояние, имея текущее состояние и совершенное агентом действие. Большинство доказательств алгоритмов обучения с подкреплением используют это свойство. В действительности агент не имеет доступа к истинному состоянию среды, которое обладает свойством марковости. Агент лишь получает *наблюдения* от среды, тем самым обновляя своё состояние s. Справедливости ради, стоит заметить, что в большинстве случаев можно добиться того, что условие марковости для состояния агента будет "почти" выполнятся и на практике алгоритмы будут работать.

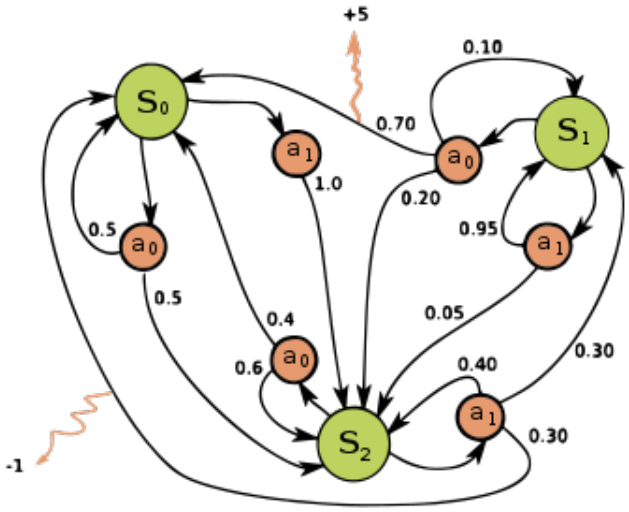
На Рис. 5 показана иллюстрация к марковскому процессу принятия решений.

3.3. **Функции полезности состояния. Уравнения Беллмана.** Почти все алгоритмы обучения с подкреплением включают в себя оценку т. н. **функций полезности** – функций, которые принимают на вход состояние и выдают оценку того, насколько агенту выгодно находится в этом состоянии (или принимают на вход пару состояние-действие, на выходе оценка того, насколько выгодно агенту совершить действие в данном состоянии).

Формально, определим **функцию полезности состояния** для политики $\pi(a|s)$:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s\right] \tag{8}$$

Рис. 5. Пример марковского процесса принятия решений.



В правой части стоит ожидаемая кумулятивная награда, если агент действует согласно политике π из состояния s.

Аналогично, определим **функцию полезности состояния-действия**:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t|S_t=s,\ A_t=a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t=s,\ A_t=a\right] \tag{9}$$

Фундаментальным свойством, используемым повсюду в обучении с подкреплением, является тот факт ([23]), что они удовлетворяют следующим рекуррентным соотношениям, называющимися **уравнениями Беллмана**:

$$v_\pi(s) = \sum_a \pi(a|s)\sum_{s'} p(s'|s,a)[r(s'|s,a)+\gamma v_\pi(s')], \tag{10}$$

$$q_\pi(s,a) = \sum_{s'} p(s'|s,a)[r(s'|s,a)+\gamma v_\pi(s')] \tag{11}$$

Оптимальная стратегия π^* может быть получена из **уравнений оптимальности Беллмана**:

$$v^*(s) = \max_\pi v_\pi(s) = \max_a \sum_{s'} p(s'|s,a)[r(s'|s,a) + \gamma v^*(s')], \tag{12}$$

$$q^*(s,a) = \max_\pi q_\pi(s,a) = \sum_{s'} p(s'|s,a)[r(s'|s,a) + \gamma \max_{a'} q^*(s',a')] \tag{13}$$

Тогда оптимальная стратегия есть $\pi^*(s) = \arg\max_a q^*(s,a)$.

Методы для поиска оптимальной политики, использующие уравнения (12) и (13), принадлежат так называемому классу value-based методов обучения с подкреплением. Примеры: Q-learning, SARSA ([23]).

3.4. **Policy-gradient алгоритмы обучения с подкреплением.** В данном разделе рассматривается некая противоположность value-based подхода, а именно класс policy-based методов, идея которых заключается в параметризации стратегии и её оптимизации напрямую. Также, часто этот класс называют policy-gradient в связи с оптимизацией политики напрямую методом градиентного спуска.

Будем рассматривать параметризованные стохастические политики $\pi_\theta(a|s)$, $\theta \in \mathbb{R}^n$ и пытаться максимизировать кумулятивную награду G_0. **Функционалом полезности политики** назовем

$$\eta(\theta) \doteq \mathbb{E}_{\pi_\theta}[G_0] = v_{\pi_\theta}(s_0) \tag{14}$$

В ранних работах ([23], [24]) был установлен основной результат, позволяющий максимизировать (14). Следуя ему, можно записать:

$$\nabla_\theta \eta = \mathbb{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(a|s) \cdot q_{\pi_\theta}(s) \tag{15}$$

На практике матожидание в (15) заменяется семплированием из текущей политики, а функция полезности $q_{\pi_\theta}(s)$ заменяется на её оценку.

Рассмотрим различные методы, получающиеся из (15).

REINFORCE. Так как G_t является несмещённой оценкой $q_{\pi_\theta}(s)$, то производя замену, получаем метод REINFORCE: $\Delta\theta = \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G_t$.

REINFORCE with baseline. На практике часто бывает, что G_t имеет большую дисперсию. Показано ([23], [24]), что $\sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot (G_t - b(s_t))$ также является градиентом функционала полезности политики. Такой метод называется REINFORCE with baseline. Добавка $b(s)$, называемая бейзлайном, позволяет уменьшить дисперсию оценки функции полезности при обучении и не влияет на градиент (15).

Actor-Critic. В качестве бейзлайна можно взять параметризованную v_π, которая будет обучаться вместо с политикой согласно Q-learning обновлениям. Таким образом, получаем метод Actor-Critic и формулу обновления весов $\Delta\theta = \sum_t (G_t - v(s_t)) \nabla_\theta \log \pi_\theta(a_t|s_t)$.

Advantage Actor-Critic. Метод Advantage Actor-Critic (A2C) также использует факт, что $r_{t+1} + \gamma v_\pi(s_{t+1})$ также является несмещенной оценкой $q_\pi(s_t, a_t)$ (следует из (11)). Метод Actor-Critic комбинирует value-based и policy-based подходы. Формула обновления выглядит следующим образом: $\Delta\theta = \sum_t (r_{t+1} + \gamma v(s_{t+1}) - v(s_t)) \nabla_\theta \log \pi_\theta(a_t|s_t)$.

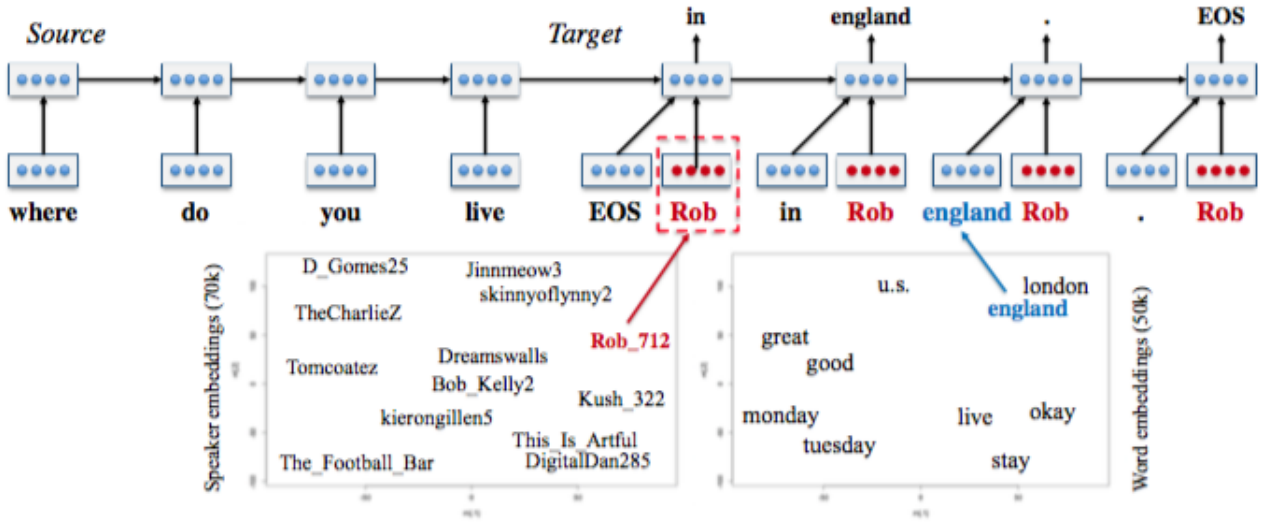
3.5. **Обучение с подкреплением в нейросетевых диалоговых моделях.** В нейросетевой диалоговой модели естественно принять распределение на словах $\hat{p}_t$ в качестве политики. Параметрами политики будут все параметры диалоговой модели. Также можно параметризовать политику лишь параметрами декодировщика, а кодировщик оставлять неизменным.

В такой постановке, действиями будут являться генерируемые декодировщиком слова, а состояниями агента – вектора скрытых состояний $\mathbf{h_t}$. Таким образом, можно задавать различные функции наград за генерируемые слова и дообучать модель одним из вышеперечисленных методов.

4. Обзор литературы

A Persona-Based Neural Conversational Model. Основная работа, пытающаяся решить задачу консистентности спикера. Основная идея заключается в обуславливании декодировщика не только на контекст предложения, но и на векторное представление пользователя. Векторное представление подается в ячейку рекуррентной сети на каждом шаге декодирования ответа и выучивается вместе с сетью на обучающей выборке. Архитектура нейронной сети представлена на Рис. 6.

Рис. 6. Архитектура Persona-based Conversational Model.



Отличие эксперимента **BeLikeX** в настоящей работе в исследовании применимости обучения с подкреплением для данной задачи, а также скорости сходимости обучения.

Deep Reinforcement Learning for Dialogue Generation. Работа не имеет прямого отношения к настоящей, но интересна своим подходом к решению основных проблем нейросетевых диалоговых моделей, таких как:

- **Общие ответы.** Реплики "я не знаю "без понятия "не понимаю о чем вы говорите"и т.п. имеют обладают высокой вероятностью по языковой модели, из-за чего модель генерирует их на большое множество контекстов.

- **Зацикливание модели.** Если модель запустить в режиме разговора самой с собой, то она быстро зацикливается, чередуя две реплики друг с другом. Это происходит из-за отсутствия новой информации в ответе модели. Авторы стремятся увеличить поток информации в ответах модели на каждом шаге диалога.

Авторская попытка решить эти проблемы заключается во введении трёх функций наград за сгенерированные моделью ответы. Дообучение происходит с помощью policy-gradient метода REINFORCE.

Качественное сравнение авторской модели с бейзлайном можно видеть на Рис. 7.

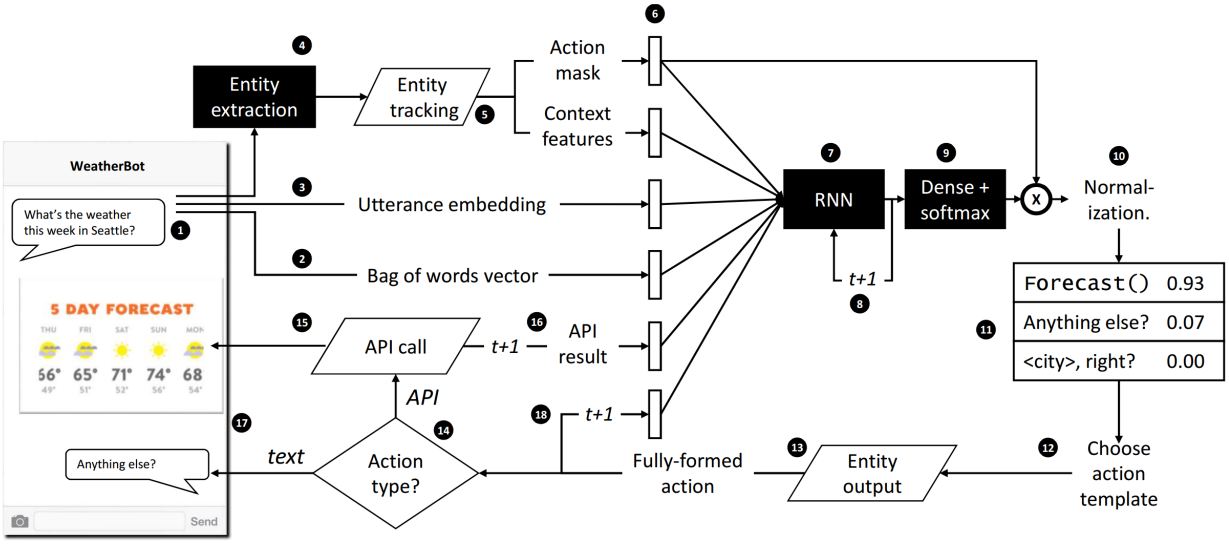
Рис. 7. Ответы бейзлайна и модели из статьи DRL for Dialogue Generation.

Baseline mutual information model (Li et al. 2015)	**Proposed reinforcement learning model**
A: Where are you going? (1) B: I'm going to the restroom. (2) A: See you later. (3) B: See you later. (4) A: See you later. (5) B: See you later. (6)	A: Where are you going? (1) B: I'm going to the police station. (2) A: I'll come with you. (3) B: No, no, no, no, you're not going anywhere. (4) A: Why? (5) B: I need you to stay here. (6) A: I don't know what you are talking about. (7) ...
A: how old are you? (1) B: I'm 16. (2) A: 16? (3) B: I don't know what you are talking about. (4) A: You don't know what you are saying. (5) B: I don't know what you are talking about . (6) A: You don't know what you are saying. (7) ...	A: How old are you? (1) B: I'm 16. Why are you asking? (2) A I thought you were 12. (3) B: What made you think so? (4) A: I don't know what you are talking about. (5) B: You don't know what you are saying. (6) ... ...

Sample-efficient Deep Reinforcement Learning for Dialog Control. Основной целью статьи является уменьшение необходимого количества объектов в обучающей выборке для обучения целеориентированной диалоговой системы. Авторы исследуют методы REINFORCE, REINFORCE with baseline, Actor-Critic применительно к этой задаче, а также демонстрируют пользу техники Experience Replay ([9]).

Hybrid Code Networks. Авторы демонстрируют нейросетевую архитектуру для целеориентированных диалоговых систем, которая позволяет интегрировать компоненты, написанные разработчиком, напрямую в нейросеть. Также авторы показывают, как дообучать такую систему с помощью обучения с подкреплением в online-режиме. Архитектура сети представлена на Рис. 8.

Рис. 8. Архитектура из статьи Hybric Code Networks.



Self-critical Sequence Training for Image Captioning. В данной статье вводится принципиально новый policy-gradient алгоритм, называющийся **Self-critical Sequence Training** (SCST). Авторы проводят эксперименты на задаче описания изображений, используя классический датасет MSCOCO ([25]), и устанавливают лучший результат в сравнении с прошлыми работами.

Алгоритм напрямую оптимизирует целевую метрику CIDEr, по которой измеряется качество модели на тестовой выборке. Рассмотрим этот подход, так как именно он будет использоваться в эксперименте **BeLikeX**.

На этапе тестирования последовательность слов генерируется жадным образом. Таким образом, сгенерированный ответ зависит только от начального состояния декодировщика $\mathbf{h}_0^{\text{dec}}$. Это позволяет использовать значение метрики на таком ответе в качестве бейзлайна

в алгоритме REINFORCE with baseline. Таким образом, обновление весов происходит по следующей схеме:

$$\Delta\theta = \nabla_\theta \log p_\theta(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m | \mathbf{h}_\mathbf{0}^{\text{dec}}) \cdot (r(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) - r(\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m)), \tag{16}$$

где $\hat{y}_t$ – t-ое слово ответа, сгенерированного семплированием, $\overline{y}_t$ – t-ое слово "жадного" ответа, $r(y)$ – функция награды (в статье используется CIDEr).

5. Эксперимент BePolite

5.1. **Постановка задачи.** Целью данного эксперимента является уменьшение обсценной лексики в ответах диалоговой модели. При этом требуется сохранить грамматическую и смысловую структуры ответа.

Обозначим обучающую выборку пар контекст-ответ как $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, а множество обсценных слов как $\mathcal{S} = \{w_1, w_2, \ldots, w_M\}$. Также обозначим диалоговую модель с набором параметров θ как $G_\theta(x) \to \hat{y}$.

Введём функцию награды за сгенерированное слово $r(\hat{y}_t) = -\mathbb{I}(\hat{y}_t \in \mathcal{S})$. Тогда кумулятивная награда за сгенерированный ответ есть $R(\hat{y}) = \sum_{t=1}^{m} r(\hat{y}_t)$.

Теперь можно записать функционал потерь на одном примере обучающей выборки:

$$GL(\theta) = \mathbb{E}_{x,y\sim\mathcal{D}}\Big[\alpha \sum_t \log p_\theta(y_t|y_1, y_2, \ldots, y_{t-1}, x) + R(G_\theta(x))\Big] \underset{\theta}{\to} \max \tag{17}$$

В реализации матожидание оценивается через среднее по обучающей выборке.

Заметим, что функция $R(\hat{y})$ недифференцируема по параметрам сети θ, но обучение с подкреплением позволяет обойти эту проблему. Число α есть вес "классического" кросс-энтропийного функционала потерь.

5.2. **Входные данные.** Обучающей выборкой являются субтритры к англоязычным фильмам с сайта opensubtitles.org. Размер обучеющей выборки составляет 18 миллионов пар контекст-ответ. Размер валидационной – 100 тысяч пар. Длина контекста равна двум репликам. Примеры обучающих пар можно видеть в Табл. 1.

Размер словаря с обсценной лексикой составил 500 слов.

5.3. **Метрики качества.** Результат оценивается по двум метрикам на валидационной выборке:

(1) **Перплексия**. $\text{PPL} = \exp^{-\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y^{(i)}|x^{(i)})}$.

Таблица 1. Примеры обучающих фраз из датасета *opensubtitles*.

Контекст	Ответ
Hey, do we know anyone who has strawberry milk? **EOS** Yeah, I think Donnie's got some. **EOS**	**BOS** Donnie Freckles? **EOS**
Is it true that you have a magic ring that can make you invisible? **EOS** Uh, yeah. **EOS**	**BOS** That is so cool! Hey, I just thought of something. **EOS**
I need you to start catching for me. **EOS** What? **EOS**	**BOS** Yeah, I need you to be my catcher, at least until **EOS**
Can I help you? **EOS** We want to play the winners. The losers of this game are playing the winners. **EOS**	**BOS** But that is not fair. **EOS**
Do you really need a diploma? **EOS** Jeff learned how to print diplomas. **EOS**	**BOS** As well as the diplomacy and perseverance. **EOS**
Come here. **EOS** What is that, Alex? **EOS**	**BOS** I don't know. **EOS**

(2) **Средняя награда**. $\text{avgR} = \frac{1}{N}\sum_{i=1}^{N} R(G_\theta(x^{(i)}))$.

Перплексия является стандартной метрикой для оценки языковых моделей. Вторая метрика – среднее количество обсценных слов в ответах модели.

5.4. **Базовое решение.** В качестве диалоговой модели была обучена 1-слойная LSTM сеть с размером скрытого слоя 1024. Размер словаря составил 50000 наиболее частотных слов. Последнее состояние кодировщика использовалось не только для инициализации декодировщика, но и подавалось в него на каждом шаге декодирования. Размерность векторных представлений 500, представления обучались вместе с сетью. Обучение длилось, пока значение функционала потерь падало на валидационной выборке (около 4 суток).

5.5. **Решение.** В качестве основного подхода к дообучению модели был выбран метод Advantade Actor-Critic (A2C). Модель, аппроксимирующая функцию полезности состояния, принимает на вход скрытое состояние декодировщика на каждом шаге, пропускает его через два полносвязных слоя по 512 нейронов, затем через слой в 256 нейронов. Функция активации – ReLu. Наконец, линейная комбинация из 256 нейронов используется как оценка value-function. Архитектура отображена на Рис. 9.

Дообучение происходило на той же обучающей выборке, на 32000 случайных примерах.

5.6. **Результаты.** В Табл. 2 представлены значения метрик для базовой модели.

Рис. 9. Архитектура ”критика” в эксперименте BePolite.

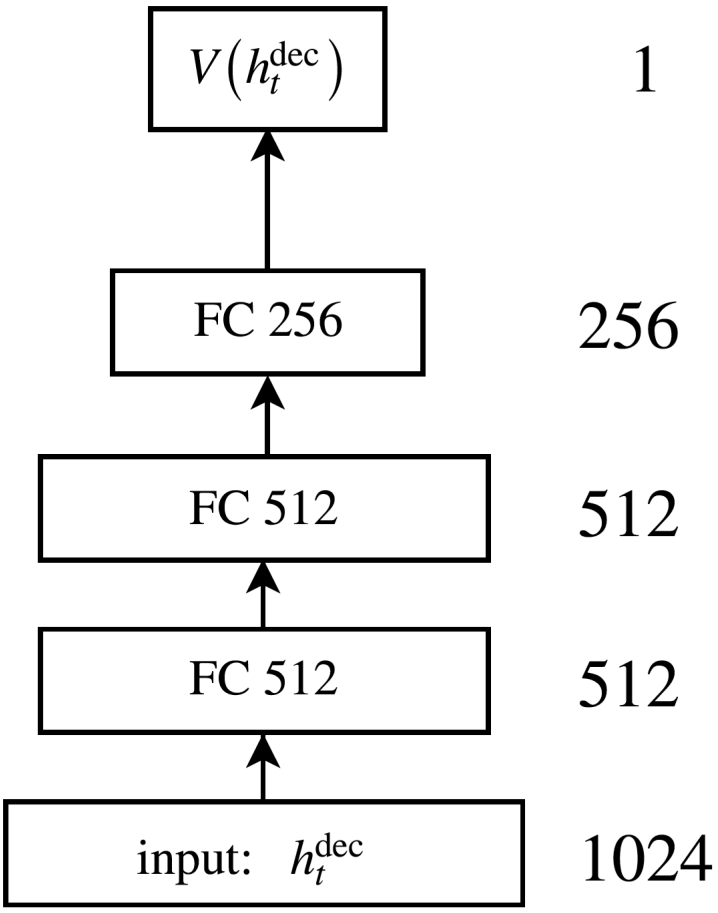


Таблица 2. Значения метрик для базовой модели.

PPL	**avgR**
3.142	-0.136

В Табл. 3 показаны результаты модели после дообучения по методу A2C для разных значений α.

Таким образом, наблюдается снижение обсценной лексики в 6-7 раз. Базовая модель генерирует в среднем 130 обсценных слов на 1000 ответов, а модель дообученная по методу A2C всего 20 слов на 1000 ответов. При этом значения перплексии ухудшаются незначительно.

α	**PPL**	**avgR**
5	3.297	-0.021
20	3.270	-0.065

Таблица 3. Значения метрик A2C модели.

6. Эксперимент BeLikeX

Цель данного эксперимента – заставить модель генерировать ответы, похожие на ответы какой-либо персоны.

6.1. **Постановка задачи.** Опять, обозначим обучающую выборку пар контекст-ответ как $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, а диалоговую модель с набором параметров θ как $G_\theta(x) \to \hat{y}$.

Пусть $D_{\text{uid}}(y) \to [-1, 1]$, – функция, принимающая на вход предложение (вектор натуральных чисел) y и выдающая "меру похожести" предложения y на типичный ответ пользователем uid. Будем считать эту функцию наградой за генерацию ответа y моделью G_θ. Конкретная реализация данной функции будет приведена ниже.

Эта функция наград никак не учитывает контекст, получаемый на вход моделью, поэтому, если максимизировать только её, модель может выродиться и генерировать один и тот же ответ пользователя на все контексты. В связи с этим, в функционал потерь необходимо добавить функционал от языковой модели, аналогично (17).

Теперь можно записать оптимизационную задачу:

$$GL(\theta) = \mathbb{E}_{x,y\sim\mathcal{D}}\Big[\alpha \sum_t \log p_\theta(y_t|y_1, y_2, \ldots, y_{t-1}, x) + D_{\text{uid}}(G_\theta(x))\Big] \underset{\theta}{\to} \max. \tag{18}$$

В реализации матожидание оценивается через среднее по обучающей выборке. Число α играет ту же роль, что и в эксперименте BePolite.

6.2. **Входные данные.** Опишем данные, использовавшиеся в эксперименте:

(1) Выборка $\mathcal{D}_g$. Данные из русскоязычной социальной сети Twitter. Объем – 50 миллионов пар контекст-ответ. Необходима для обучения базовой диалоговой модели. Объём валидационной выборки – 100 тысяч примеров. Примеры диалогов представлены в Табл. 4.

(2) Выборка $\mathcal{D}_u$. Получена из отдельной выборки из Twitter'а, в которой к каждому сообщению привязан идентификатор пользователя, путём отбора пар контекст-ответ, ответ в которых принадлежит одному из 1000 пользователей с наибольшими количествами ответов. Объём – 1,2 миллиона обучающих примеров. Необходима для обучения модели, соответствующей функции похожести. Архитектура модели описана ниже.

(3) Выборка $\mathcal{D}_s$. Получена из $\mathcal{D}_u$ путём отбора реплик, принадлежащих пользователю, который был идентифицирован как техническая поддержка сотового оператора. Объём – 3000 обучающих примеров (500 примеров в валидационной выборке). Примеры диалогов представлены в Табл. 5.

Таблица 4. Примеры обучающих фраз из выборки $\mathcal{D}_g$.

Контекст	Ответ
Слава инстаграму, героям iii слава! Не забывай про геймеров. **EOS** Поизвращаемся - героям 4 (!!!) слава. **EOS**	**BOS** Матерь божья! **EOS**
Я тоже самое маме сказал **EOS** Брат по несчастью. **EOS**	**BOS** Я не про свой город сказал. Прости :с **EOS**
Я ринусь в бой, достойный схватки, последней на мой век. Живи и умирай в сей день. Живи...и умирай...в сей день. **EOS** Особенно печальный момент, когда он бумажники все в стопочку складывает и вспоминает свою семью... Ах. Я рыдала **EOS**	**BOS** Да. Я тоже! Весь фильм со слезами на глазах... до сих пор не отошла, все думаю, как бы я поступила... смогла бы...? **EOS**
точно не скоро:3 **EOS** ахах это почему? :d **EOS**	**BOS** сериалыыыы:3 **EOS**
Так блин это фильм а это сериалы:d **EOS** Все равно ты непостоянная смирись **EOS**	**BOS** ахахаа лол **EOS**
Родители смотрят рождественские встречи с пугачевой а я ржу ибо вспоминаю ту упоротость со стайлсом и пугачевой **EOS** По первому, да ? Сочувствую, мои тоже **EOS**	**BOS** Я с украины и у меня по интеру идет. Вообщем аля первый только в украине **EOS**

6.3. **Метрики качества.** Перечислим метрики, по которым будем оценивать качество решения:

(1) **Перплексия**. $\text{PPL} = \exp^{-\frac{1}{N}\sum_{i=1}^{N}\log p_\theta(y^{(i)}|x^{(i)})}$.

(2) **Средняя похожесть на пользователя**. $\text{avgD} = \frac{1}{N}\sum_{i=1}^{N} D_{\text{uid}}(G_\theta(x^{(i)}))$.

(3) **Человеческая оценка**. Будем смотреть на ответы разных моделей глазами.

Аналогично эксперименту BePolite, дообучая модель по функционалу потерь, где присутствует не только слагаемое от языковой модели, значение перплексии будет неизменно ухудшаться. Но так как перплексия не является целевой метрикой задачи, то достаточно следить, что её значения остаются в пределах нормы.

Так как целевым пользователем была выбрана тех. поддержка, то целесообразно считать метрики на выборке $\mathcal{D}_s$ помимо $\mathcal{D}_g$.

6.4. **Базовое решение.** Базовое решение состоит из двух частей: диалоговой модели и реализации функции "похожести" $D(y)$. Опишем каждую из них.

- **Диалоговая модель**. Диалоговая модель имеет такую же архитектуру, как модель из предыдущего эксперимента, только обучается на выборке $\mathcal{D}_g$. Словарь – так же 50000 наиболее частотных слов в выборке $\mathcal{D}_g$. Обучение длилось около 7 суток.
- **Функция похожести**. Для реализации функции похожести была обучена модель из класса, часто называемый *Deep Semantic Structured Model* или *Deep Semantic Similarity Model* ([7] [8] [26] [27] [28]). Архитектура модели представлена на Рис. 10. Модель состоит из двух "башен".

  Первая башня принимает на вход идентификатор пользователя uid и отображает его в семантическое пространство пользователей-фраз посредством двух слоёв.

Таблица 5. Примеры обучающих фраз из выборки $\mathcal{D}_s$.

Контекст	Ответ
Жаль, что с я связан корпоративным тарифом. Очень хочется перейти на оператора с нормальным покрытием. **EOS** и мегафон здесь не берёт, и папа джонс не везёт пиццу **EOS**	**BOS** Добрый вечер! Укажите, пожалуйста, точный адрес (нас. пункт, улицу, номер дома), в чем именно заключается сложность. Проверим. **EOS**
на модем который используется как роутер списаны звонки, и всякое такое чего у меня в тарифе нету... **EOS** сервис на высшем уровне! Такое сейчас время, не своруешь не проживёшь , жаль ! Хорошая была компания! **EOS**	**BOS** коллеги, обратите внимание. **EOS**
Добрый вечер! С какими сложностями Вы столкнулись, скажите, пожалуйста? Постараемся Вам помочь. **EOS** интернет работал только "Е"полтора часа,уже все хорошо. **EOS**	**BOS** Рады, что Ваш вопрос решился. Возникнут вопросы - обращайтесь! =) **EOS**
Добрый день! Для получения корректного ответа от специалиста уточните свой регион. **EOS** мурманская область **EOS**	**BOS** Коллеги, просьба проверить. **EOS**
Добрый день! Если сложности еще сохраняются, пожалуйста, уточните Ваш регион. **EOS** эти сложности были в начале Тверской в период массового скопления граждан. **EOS**	**BOS** Если есть такая возможность, то уточните, пожалуйста, адрес более подробно - для проверки требуются название улицы и номер дома.**EOS**
Без согласия Абонента услуга не отключается. **EOS** Зато без согласия абонента услуга подключается. **EOS**	**BOS** Платные услуги подключаются либо самим абонентом, либо сотрудником компании с согласия абонента. **EOS**
здравствуйте! Это платный контент. Для отключения введите команду на своем телефонном аппарате. **EOS** Сейчас буду разбираться, подключилась или нет "в тихую"эта платная дрянь. **EOS**	**BOS** просьба обратить внимание . **EOS**
какой у Вас регион? **EOS** Пермь **EOS**	**BOS** Коллеги просьба проверить **EOS**
здравствуйте! Уточните, пожалуйста, по какому адресу наблюдаются сложности с интернетом **EOS** по всему городу Томску. Фрунзе 120, Лазо 24, Мира 35. Проблема не локальная и не частная. **EOS**	**BOS** Коллеги обратите, пожалуйста, внимание **EOS**
более чем. Dial up быстрее. **EOS** это вот мегафон у меня так работает(беда-беда огорчение **EOS**	**BOS** Добрый вечер. Какой у Вас регион? **EOS**
Позвонил в Рязанский Россельхозбанк. Так приятно никогда не общался с поддержкой. Всякие мегафоны отдыхают в стороне **EOS** ты общался с техподдержкой мегафона?? ОНА СУЩЕСТВУЕТ??! **EOS**	**BOS** Обратите, пожалуйста, внимание **EOS**

Вначале действует эмбеддинг слой, отображающний идентификатор в его векторное представление размерности 128. Затем этот вектор проходит через полносвязный слой с 256 нейронами. Получившийся вектор u и считается представлением пользователя в семантическом пространстве.

Вторая башня принимает на вход предложение и пропускается через рекуррентный кодировщик с размером скрытого слоя 1024. Финальное скрытое состояние кодировщика проходит через полносвязный слой с 256 нейронами. Получившийся вектор a интепретируется как представление фразы в семантическом пространстве.

Мера похожести фразы y на стиль пользователя uid вычисляется как скалярное произведение соответствующих векторов в евклидовом пространстве. Во избежание слишком больших по модулю скалярных произведений, вектора u и a нормируются на сферу:

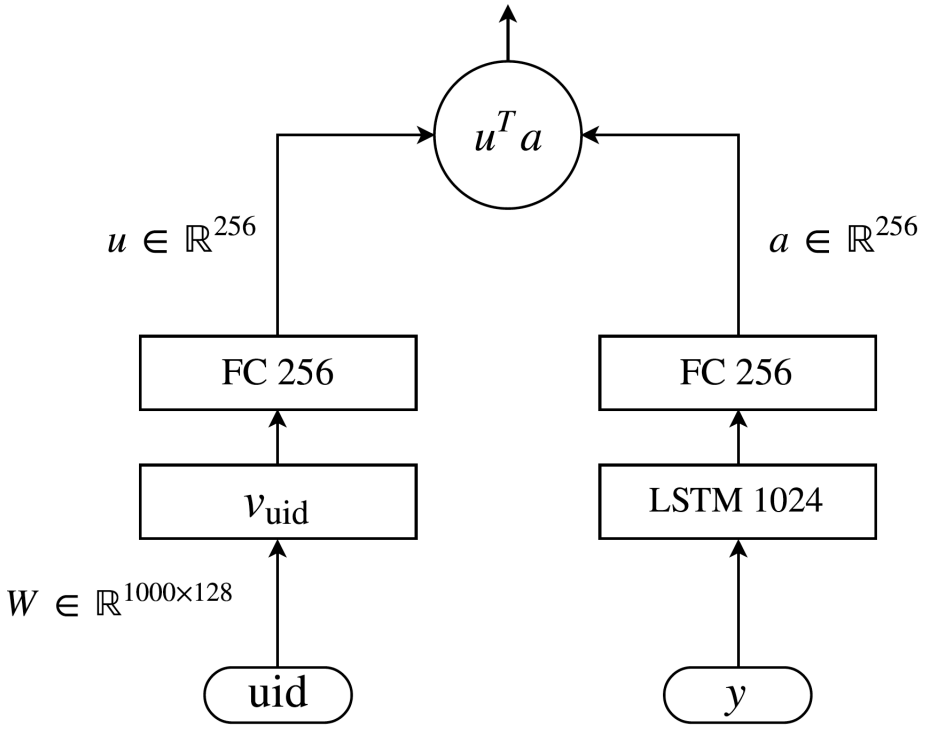
$$D_\psi(\text{uid}, y) = \frac{u^T a}{\|u\|\|a\|}, \tag{19}$$

где ψ – множество параметров обеих башен. Чтобы обучать такую модель, необходимо сформировать выборку из т.н. триплетов. Триплет это тройка $(\text{uid}, y_{\text{pos}}, y_{\text{neg}})$, где y_{pos} – фраза пользователя uid, а y_{neg} – случайная фраза другого пользователя. Обучение модели происходит с помощью max-margin функционала потерь:

$$L(\psi) = \mathbb{E}_{\text{uid}, y_{\text{pos}}, y_{\text{neg}} \sim \mathcal{D}_u} \big(\max(0, 0.5 - D_\psi(\text{uid}, y_{\text{pos}}) + D_\psi(\text{uid}, y_{\text{neg}})) \big) \underset{\psi}{\to} \min\,. \tag{20}$$

Заметим, что модель похожести обучается один раз и фиксируется. Обозначим диалоговую модель, обученную на $\mathcal{D}_g$ за G_g. Для широты эксперимента модель G_g была дообучена на выборке $\mathcal{D}_s$ классическим способом (см. (6)). Обозначим её за G_s.

Рис. 10. Архитектура модели "похожести" $D_\psi(\text{uid}, y)$.



В дальнейшем зафиксируем идентификатор пользователя в модели $D_\psi(\text{uid}, y)$ равным идентификатору тех. поддержки, а также её веса ψ и будем писать $D(y)$.

6.5. **Решение.** В данном эксперименте методом обучения с подкреплением был выбран Self-critical Sequence Training (SCST), где в качестве награды выступает $D(\hat{y})$.

Модель G_g была дообучена в трёх различных вариациях на различных выборках:

(1) **SCST-ON-SUPPORT**. На выборке $\mathcal{D}_s$ модель дообучена по аналогии с G_s, но по функционалу потерь (18). Значение $\alpha = 1.0$.

(2) **SCST-ON-USERS**. Аналогично предыдущему пункту, но на выборке $\mathcal{D}_u$.

(3) **SCST-ON-HIGH-REWARDED**. Из выборки $\mathcal{D}_u$ были отобраны только те пары (x, y), для которых выполнялось условие $D(y) > 0.5$. Объём результирующей выборки составил 22000 примеров. Значение $\alpha = 2.0$.

Три выборки в этих вариациях содержат разное количество ”хороших” реплик, то есть реплик, похожих на пользователя тех. поддержки. В $\mathcal{D}_s$ количество высоконаграждаемых реплик минимально, в $\mathcal{D}_u$ максимально. Третий вариант промежуточный.

6.6. **Результаты.** В Табл. 6 находятся примеры фраз y с наибольшими значениями $D(y) = 1.0$ от пользователей, отличных от выбранного в эксперименте. Видно, что модель хорошо выучила отображение в семантическое пространство и пригодна для использования в качестве функции награды.

Таблица 6. Фразы из выборки $\mathcal{D}_u$ с максимальной похожестью 1.0.

Фраза
Ответили в ДМ.
Приносим извинения за возможные неудобства.
сейчас все решим, приносим извинения за неудобства.
позвольте еще раз принести Вам наши извинения за доставленные неудобства.
Спасибо, что помогли нам провести работу над ошибками.
Если Вы оставляли заявку на тестирование, появится после праздников.
Ответили вам в ЛС.
Оператор видит ситуацию происходящую с номером и сможет вам помочь.
наши специалисты делают все возможное, чтобы решить данный вопрос в кратчайшие сроки.
Также мы акцентировали внимание специалистов на важности оперативного обслуживания покупателей.
Можно Ваши ФИО и дату рождения нам в ЛС? Разберемся
Вы также находитесь в Плесецком районе? Если да, то приносим извинения за временные неудобства :(
город Астана, верно? Вы часто там бываете? Если мы примем запрос по номеру для проверки?
вас соединят с оператором, который видит всю ситуацию с номером и вам обязательно поможет.
ожалуйста, расскажите о проблеме подробнее. Какой именно сбой и где (нас. пункт, улица, дом)?

	$\mathcal{D}_g$	$\mathcal{D}_s$
BASELINE	6.330	14.269
LLH-FINETUNED	24.308	1.040
SCST-ON-SUPPORT	17.574	1.175
SCST-ON-USERS	8.178	26.691
SCST-ON-HIGH-REWARDED	24.305	1.283

Таблица 7. Значения перплексии для моделей из эксперимента BeLikeX.

	$\mathcal{D}_g$	$\mathcal{D}_s$
BASELINE	0.0130	0.0484
LLH-FINETUNED	0.6743	0.9546
SCST-ON-SUPPORT	0.7798	0.8967
SCST-ON-USERS	0.0192	0.0183
SCST-ON-HIGH-REWARDED	0.3357	0.9331

Таблица 8. Значения avgD для моделей из эксперимента BeLikeX.

В Табл. 7 представлены значения перплексии для всех моделей на валидационных выборках, соответствующих $\mathcal{D}_g$ и $\mathcal{D}_s$. Строка BASELINE соответствует модели G_g, а строка LLH-FINETUNED – модели G_s. В Табл. 8 представлены значения метрики avgD.

Модель, дообученная методом SCST на всей выборке $\mathcal{D}_u$, даёт неудовлетворительные результаты, сравнимые с бейзлайном. Связать это можно с минимальным процентом реплик в $\mathcal{D}_u$ с высокой наградой. Модель, дообученная на $\mathcal{D}_s$ выигрывает у бейзлайна LLH-FINETUNED на выборке $\mathcal{D}_g$ за счёт эффективного исследования при обучении стратегии.

Интересным результатом с промежуточными значениями метрик является модель, дообученная на выборке ответов с высокими наградами. При специфическом контексте из области тех. поддержки модель уверенно отвечает в нужном стиле. Однако и на общей выборке она пытается генерировать фразы, похожие на ответы тех. поддержки.

Для качественного анализа в Табл. 9 приведены ответы моделей SCST-ON-SUPPORT и SCST-ON-HIGH-REWARDED на различные контексты. Первая модель переобучилась по выборку $\mathcal{D}_s$ и на общие контексты генерирует неуместные ответы в стиле тех. поддержки. Вторая же не потеряла "знание" об общих репликах и генерирует уместные ответы на общие контексты, при этом отвечая в нужном стиле на контексты области тех. поддержки.

Таблица 9. Ответы различных моделей в эксперименте BeLikeX.

Контекст	SCST-ON-SUPPORT	SCST-ON-HIGH-REWARDED
привет! **EOS** как дела?	коллеги помогите , пожалуйста , только в личные сообщения (город , улица , поселок , лк)	у нас нет света , но хорошо .
интернет не работает что делать?' **EOS**	коллеги помогите , пожалуйста , более подробно какие сложности с использованием .	здравствуйте ! если возникли сложности , то вы можете проверить баланс ?
что такое любовь ? **EOS**	коллеги, пожалуйста , базовые всей компании	все ахахахаххахаха
здравствуйте ! с какими сложностями столкнулись ? **EOS** уже все хорошо , спасибо	если возникнут вопросы , пожалуйста , обращайтесь	если будут вопросы , пишите !
какой у вас регион ? **EOS** томская область	какой у вас регион ?	проверьте , пожалуйста
какой у вас регион ? **EOS** как дела ?	проверьте , пожалуйста	:d все хорошо , но ограничения к сведению
ты любишь учиться ? **EOS** я обожаю ! а ты ?	какой у вас регион ?	нет , но к сожалению , не могу сказать одно время , да .

7. Заключение

7.1. **Итоги работы.** В данной работе предложен новый подход для дообучения нейросетевых диалоговых моделей с помощью методов обучения с подкреплением. На двух задачах: запрета обсценной лексики в ответе и требования отвечать "в стиле" технической поддержки продемонстрирована эффективность метода в сравнении с классическими бейзлайнами. Подведём итоги по проделанной работе:

(1) Была обучена рекуррентная диалоговая модель на выборке субтитров к англоязычным фильмам. Такое базовое решение генерировало 136 обсценных слов в среднем на 1000 предложений.
(2) Затем, модель была дообучена с помощью A2C метода обучения с подкреплением, продемонстрировав снижение обсценной лексики в 6 раз, до 21 слова в среднем на 1000 предложений. При этом потеря в перплексии составила всего 4%, что является незначительной потерей. Таким образом, A2C метод позволил эффективно дообучить диалоговую модель под поставленную задачу за счёт исследования в процессе генерации слов.
(3) Была обучена диалоговая модель на выборке русскоязычных диалогов из социальной сети Twitter.
(4) Была обучена модель "похожести" фразы на пользователя. Качественно показано, что она хорошо отбирает реплики, похожие по смыслу и лексике на фразы технической поддержки.
(5) Диалоговая модель была дообучена по методу SCST, в качестве награды была взята модель "похожести". Продемонстрирована эффективность дообучения, нужный результат был достигнут.

7.2. **Дальнейшие исследования.** Дальнейшие исследования можно разделить в соответствии с двумя проведёнными экспериментами:

(1) **BePolite**. На практике, собранный список обсценных слов может не покрыть всё их множество. Поэтому интересно проанализировать, насколько модель воспринимает семантику того, что ей запрещают говорить. Например, можно взять кластер синонимичных обсценных слов, часть из которых оставить в списке, а вторую часть убрать. После дообучения можно посмотреть, насколько реже стали использоваться те слова, которые убрали из списка.

(2) **BeLikeX**. В проведенном исследовании модель похожести на вход принимала только индентификатор пользователя и реплику, но никак не учитывала контекст в диалоге. Возможно, обуславливание модели на контекст поможет при дообучении.

Также, интересно использовать наработки из соперничающих генеративных сетей, а именно, трактовать модель похожести как дискриминатор, а диалоговую модель как генератор, и дообучать обоих по соперничающему функционалу потерь.

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] 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.

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

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

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

[6] 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.

[7] *Semantic Parsing for Single-Relation Question Answering*. Association for Computational Linguistics, June 2014.

[8] *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*. ACM International Conference on Information and Knowledge Management (CIKM), October 2013.

[9] 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.

[10] 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.

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

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

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

[14] 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.

[15] Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Çelikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue system via hierarchical deep reinforcement learning. *CoRR*, abs/1704.03084, 2017.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[19] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[20] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[21] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099, 2015.

[22] Hal Daumé III and Daniel Marcu. Learning as search optimization: Approximate large margin methods for structured prediction. *CoRR*, abs/0907.0809, 2009.

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

[24] 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.

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[26] Jianfeng Gao, Xiaodong He, Scott Wen-tau Yih, and Li Deng. Learning continuous phrase representations for translation modeling. Association for Computational Linguistics, June 2014.

[27] *Learning Semantic Representations Using Convolutional Neural Networks for Web Search*. WWW 2014, April 2014.

[28] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. CIKM, November 2014.

E-mail address: persiyanov@phystech.edu

Персиянов Дмитрий, каф. анализа данных, Московский физико-технический институт (государственный университет)