

# Когнитивный ассистент

February 6, 2020

## 1 Описание задачи

В данной работе мы описываем методику конструирования целеориентированного режима когнитивного ассистента (далее КА). Нас интересует построение интерактивной диалоговой системы, т.е. системы, способной анализировать запросы пользователя, задавать ему вопросы и правильно на них отвечать, которая помогает пользователю в построении его собственного плана достижения заранее (в самом начале диалога) обозначенной цели. А именно, цель пользователя является достаточно конкретной (напр., купить автомобиль), и КА должен в процессе диалога максимально конкретизировать цель (в случае автомобиля данной конкретизацией может быть выбор между отечественным автомобилем и иномаркой, между поддержанным и новым и т.д.) и провести пользователя по всем этапам достижения цели давая рекомендации.

Мотивацией для создания новой модели поведения когнитивного ассистента является следующее. На данный момент диалоговые системы не показывают результатов, сравнимых с результатом общения с экспертом. Во-первых, текущие технологии не заточены на интерактивное общение с пользователем. А именно, существуют неплохие вопросно ответные системы [?] и есть результаты в создании диалоговых систем для общения на свободную тему (chit-chat) [?], однако практически не разработаны системы, задающие вопрос пользователю и обрабатывающие ответ для достижения собственных целей. Во-вторых, существующие диалоговые системы плохо заточены для работы в предметной области. Для решения этих задач в частности используются базы знаний [?].

## 2 Модель ассистента

**Определение 2.1.** *Графом когнитивного ассистента называется ориентированный граф  $G = (V, E)$ , такой что выполнены следующие условия:*

1. *Есть две вершины  $s$  и  $t$ , такие что  $d_{in}(s) = d_{out}(t) = 0$ , т.е. в  $s$  не входит ни одного ребра, из  $t$  не выходит. Эти вершины называются стартовой и терминальной соответственно.*
2. *Каждой вершине  $v \in V$  соответствует набор алгоритмов  $(\mathcal{M}_v, \mathcal{G}_v, \mathcal{R}_v, \mathcal{I}_v, \mathcal{S}_v)$ . Алгоритм  $\mathcal{M}_v$  есть **главный алгоритм**, алгоритм  $\mathcal{G}_v$  есть **алгоритм генерации сообщения**, алгоритм  $\mathcal{R}_v$  есть **алгоритм обработки запроса**, алгоритм  $\mathcal{I}_v$  есть **алгоритм интерактивного общения**, алгоритм  $\mathcal{S}_v$  есть **алгоритм принятия решения**. Формальное описание будет дано ниже.*

Главный алгоритм  $\mathcal{M}_v$  определяет порядок запуска остальных четырех алгоритмов.

Алгоритм генерации сообщения  $\mathcal{G}_v$  есть алгоритм, который выводит единственное сообщение  $s$  пользователю. Это сообщение есть полезная информация, которую получит пользователь, в частности, этот алгоритм сообщит, что нужно сделать пользователю для достижения своей цели.

Алгоритм обработки запроса  $\mathcal{R}_v$  есть алгоритм, который получает на вход сообщение пользователя, обрабатывает его и, если обнаруживает некоторый запрос, то начинает его обработку. Считается, что все возможные обрабатываемые запросы обработаны заранее и их конечное число. Наиболее важным запросом является запрос на аргументацию одного из ранее полученных сообщений, т.е. запрос на генерацию  $a$ , который убедит, что одна информация в полученном ранее сообщении поможет пользователю в достижении цели.

Алгоритм принятия решения  $\mathcal{S}_v$  есть алгоритм, который используя алгоритм  $\mathcal{I}_v$  и всю ранее полученную информацию, принимает решение по какому ребру пойти.

Алгоритм интерактивного общения  $\mathcal{I}_v$  есть алгоритм, принимающий на вход три набора сообщений  $(\{s_k^1\}, \{s_k^2\}, \{s_k^3\})$  - набор сообщений (или информации, извлеченных из них), полученных от пользователя в пройденных вершинах, набор сообщений (или информации, извлеченных из них), полученных от пользователя в текущей вершине, и набор сообщений, которые характеризуют текущие потребности когнитивного ассистента.

Данный алгоритм будет вызываться в  $\mathcal{S}_v$  для пополнения информации о пользователе до такого уровня, что можно будет выбрать наилучшую для данного пользователя следующую вершину.

Существует четыре различных случая вершин и соответствующих им алгоритмов:

1. Конечная вершина
2. Вершина с одним исходящим ребром
3. Вершина с более чем одним исходящим ребром
4. Стартовая вершина

Мы считаем, что если пользователь дошел до конечной вершины  $t$ , то он достиг своей цели. Поэтому в конечной вершине алгоритмы имеют простую структуру. А именно,  $\mathcal{M}_t$  вызывает функцию  $\mathcal{G}_t$ , которая выводит некоторое прощальное сообщение и завершает работу. Остальные алгоритмы можно считать алгоритмами, которые завершают свою работу на любом входе за первый шаг, возвращая пустую строку (далее такие алгоритмы будем называть пустыми алгоритмами).

В случае вершины с одним исходящим ребром алгоритм  $\mathcal{M}_v$  устроен следующим образом. Во-первых, он выводит сообщение при помощи  $\mathcal{G}_v$ . Затем вызывает алгоритм  $\mathcal{S}_v$ , который возвращает единственный возможный вариант. Алгоритм  $\mathcal{I}_v$  есть пустой алгоритм. Алгоритм  $\mathcal{R}_v$  в этом и в оставшихся случаях мы обсудим позже.

Алгоритм  $\mathcal{M}_v$  в оставшихся пунктах устроен следующим образом. Сначала вызывается алгоритм  $\mathcal{S}_v$ , затем на основе принятого решения генерируется сообщение алгоритмом  $\mathcal{G}_v$ . Как принимается решение в  $\mathcal{S}_v$  и как происходит общение через  $\mathcal{I}_v$  мы обсудим после формализации пользователя.

### 3 Модель пользователя

**Определение 3.1.** Пусть  $v$  - вершина, из которой выходит более одного ребра. Пусть  $\Omega_v = \{u | (v, u) \in E\}$ . Тогда  $\Omega_v$  есть **множество решений**. Также считаем, что каждому элементу этого множества соответствует некоторый набор признаков  $x_v \in X_v$  и функция соответствия  $g_v : X_v \times F \rightarrow R_+$ , где  $F$  - особенности пользователя (см. определение 3.2).

Без ограничения общности здесь и далее мы будем считать, что  $X_v \subset \mathbb{R}$ . Действительно, какими бы не были признаки (бинарные, категориальные и т.д.) существуют способы приведения их к такому виду.

**Определение 3.2.** *Моделью пользователя мы будем называть тройку  $u = (x_-, x_+, f)$ , где 1.  $x_-, x_+ \in \times_{v: d_{\text{out}}(v) > 1} (X_v \cup \{\pm\infty\}^n)$  - некоторые ограничения, которые пользователь ставит на варианты решений для  $\Omega_v$ , т.е. пользователь хочет, чтобы  $x_-|_v < x_v^* < x_+|_v$ . Здесь  $x_v^*$  - решение, которое КА примет в вершине  $v$ ,  $x_{\pm}|_v$  - та часть вектора  $x_{\pm}$ , которая соответствует вершине  $v$ .*

*2.  $f \in F$  - некоторые особенности пользователя.*

Сделаем несколько пояснений к этому определению.

Во-первых, пользователь может не знать, что означает  $k$ -ый признак варианта в вершине  $v$  или он ему может быть не интересен. В обоих этих случаях мы будем считать  $x_{\pm}|_v^k = \pm\infty$ .

Во-вторых, мы добавили к возможным значениям ограничений из множеств  $X_v$  также  $\pm\infty$ . В данном определении их можно заменить на  $\max_{x \in X_v} x_k + \epsilon$  и  $\min_{x \in X_v} x_k - \epsilon$  для некоторого положительного  $\epsilon$ .

В-третьих, обсудим особенности пользователя  $F$ . Это некоторые характеристики пользователя, которые не связаны явно с характеристиками решений, однако можно установить связь, насколько решение соответствует этим характеристикам при помощи функции соответствия  $g_v$ . Примерами таких характеристик могут быть следующие:

1. В случае если КА помогает купить пользователю некоторый предмет, то таковой характеристикой может быть то, зачем пользователю этот предмет.

2. В случае если покупка нуждается в регулярном обслуживании, которое стоит денег, то таковой характеристикой может быть заработок пользователя и/или сколько он готов тратить на это.

Далее мы также будем считать, что  $F \subset \mathbb{R}_+^d$ .

## 4 Алгоритм принятия решения

Целью данного раздела является описание модели принятия решения. Рассмотрим некоторую вершину  $v$  со степенью  $d_{\text{out}}(v) > 1$ . Пусть  $X_v$  - множество признаков решений,  $g_v : X_v \times F \rightarrow \mathbb{R}_+$  - функция соответствия.

Пусть также мы общаемся с пользователем  $u = (x_-, x_+, f) \in U$ . Определим  $u|_v = (x_-|_v, x_+|_v, f) \in U|_v$ , т.е. отбросим все признаки, которые не влияют на принятие решения в данной вершине.

Допустим, у нас есть функция  $s_v : X_v \times U|_v \rightarrow \mathbb{R}$ . Она будет устроена следующим образом:

$$s_v(x, u) = g_v(x, f) + \sum_{k=1}^n \lambda_k^- L(x_-|_v^k, x_k) + \sum_{k=1}^n \lambda_k^+ G(x_+|_v^k, x_k),$$

где константы  $\lambda_k^\pm \geq 0$ , функции  $L$  и  $G$  есть барьерные функции, которые штрафуют за нарушение ограничений. Примерами таких функций могут быть индикаторные функции или их сглаженные аналоги.

**Определение 4.1.** *Признаки с индексами  $K = \{k_j\}_j$  назовем  $\epsilon$ -несущественным при заданных  $x^{k_i}, k_i \notin \{k_j\}_j$ , если множество  $\left\{v_k \mid v_k \in \Omega, s_v(u, v_k) \geq \max_{z \in \Omega} s_v(u, z) - \epsilon\right\}$  неизменно для любых  $x_{k_j}, k_j \in \{k_j\}_j$ .*

Если говорить неформально, то ясно, что задача алгоритма  $\mathcal{S}_v$  найти максимальное или близкое к нему значение  $s_v(x, u)$  при фиксированном  $u$  на множестве  $\{x_z \mid z \in \Omega_v\}$  при минимальном вызове функции общения  $\mathcal{I}_v$ . Рассмотрим два случая:

1. Множество  $\Omega_v$  небольшое, т.е.  $|\Omega_v| \sim 1$ .
2. Множество  $\Omega_v$  достаточно большое, т.е.  $|\Omega_v| \gg 1$

Существенная разница между этими случаями в следующем. Мы можем путем простого перебора элементов  $\Omega_v$  установить все  $\epsilon$ -несущественные признаки, если мы можем легко оптимизировать функционал  $s_v$ , и в таком случае мы можем уменьшить количество вопросов к пользователю. В случае, если  $\Omega_v$  большое, то перебор всех элементов и нахождения для каждого из него несущественных признаков, является достаточно трудоемкой задачей и мы не можем себе ее позволить.