Гибридная интеллектуальная русскоязычная диалоговая информационная система на основе метаграфового подхода

1. **Гапанюк Ю.Е. (Gapanyuk Yu.E.), доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, gapyu@bmstu.ru**
2. **Леонтьев А.В. (Leontiev A.V.), аспирант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, alekseyl@list.ru**
3. **Латкин И.И. (Latkin I.I.), аспирант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, igor.latkin@outlook.com**
4. **Чернобровкин С.В. (Chernobrovkin S.V.), аспирант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, sergey.chernobrovkin@inbox.ru**
5. **Белянова М.А. (Belyanova M.A.), магистрант кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, flerchy@gmail.com**
6. **Морозенков О.Н. (Morozenkov O.N.), студент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана, m@oleg.rocks**

Аннотация. В статье рассмотрена информационная система, позволяющая отвечать на часто задаваемые вопросы, вопросы по базе знаний и вести активный диалог с пользователем. Рассмотрены методы машинного обучения, применяемые для создания диалоговых информационных систем. Предложена структура гибридной диалоговой информационной системы. Рассмотрена реализация модуля ответов на часто задаваемые вопросы на основе методов машинного обучения. Рассмотрена реализация модуля обработки базы знаний на основе метаграфового подхода. Приведены результаты экспериментов.

Ключевые слова: диалоговая информационная система, чатбот, метафон, гибридная интеллектуальная информационная система (ГИИС), метаграф, метавершина.

1. Введение

В настоящее время все более популярными становятся диалоговые системы. Сейчас почти невозможно встретить сайт компании, на котором пользователю не предлагалось бы задать вопрос «консультанту», который является диалоговой системой. В англоязычной литературе системы такого класса часто называются чатботами (chatbots).

К сожалению, сегмент задач, в котором такие диалоговые системы хорошо выполняют свои функции, пока остается достаточно узким. В основном, современные диалоговые системы удовлетворительно справляются только с задачей ответа на часто задаваемые вопросы (в русском языке используется аббревиатура «ЧАВО», а в английском «F.A.Q.» – Frequently Asked Questions). Системам такого класса посвящено большое количество публикаций, в частности [1].

Как правило, «ЧАВО»-модуль является требованием заказчика №1 к диалоговой системе. Наиболее часто встречающееся требование №2 – разработка модуля, который отвечает на вопросы пользователя на основе базы знаний. Для базы знаний могут быть использованы следующие варианты представления:

1. База знаний представлена в виде текста. Это вариант, который достаточно часто встречается на практике, но при этом является наиболее трудоемким с точки зрения реализации и тестирования. Трудность реализации заключается в том, что необходимо использовать полноценный синтаксический анализатор, который позволяет извлекать непротиворечивые факты из текста. Тестирование и проверка полученной базы знаний является трудоемкой процедурой, которая требует значительных затрат ресурсов со стороны заказчика.
2. База знаний представлена в виде схемы реляционной БД, структурированной коллекции документов нереляционной БД, хранилища данных. В этом случае данные хранятся в БД в структурированном виде. Однако, извлечение знаний из БД в виде разнородных фактов и их тестирование также может представлять собой процедуру, трудоемкую для заказчика.
3. База знаний представлена в виде денормализованной таблицы (витрины данных). Фактически в этом случае данные представлены подобно тому, как они представляются в пакетах электронных таблиц. Идентификаторы не

используются, ключевые слова (названия товаров и т.д.) могут повторяться в ячейках таблицы. Подобный подход может вызывать неудобства при разработке вследствие денормализации, но, как показывает практика, он наиболее приемлем для заказчика по следующим причинам:

- Большинство заказчиков, вне зависимости от специальности, используют электронные таблицы, и модель представления знаний в виде электронной таблицы является для них естественной.
- Каждая строка денормализованной таблицы фактически представляет собой запись (факт, ситуацию) в базе знаний. Поэтому заказчик может достаточно легко осуществить тестирование такой модели знаний.

На текущий момент многие пользователи готовы предоставить список «ЧАВО» и данные своей предметной области в виде денормализованной таблицы (набора денормализованных таблиц). Типичными требованиями заказчика к диалоговой системе в этом случае являются следующие требования:

I. Ответы на вопросы из списка «ЧАВО».
II. Ответы на вопросы из базы знаний, представленной в виде денормализованной таблицы.
III. Активный диалог с пользователем. Если пользователь задает вопрос по базе знаний, то диалоговая система пытается задать встречные вопросы, соответствующие недостающим параметрам, пытается предложить товар или услугу, информация о которых хранится в базе знаний.

В данной статье мы рассматриваем структуру и основные принципы работы информационной системы, реализующей данные требования.

2. Использование методов машинного обучения для создания диалоговых систем

В настоящее время методы машинного обучения рассматриваются как основное направление развития искусственного интеллекта. Прежде всего это методы, связанные с нейронными сетями и глубоким обучением. В диалоговых системах данные методы также применяются очень активно.

Для реализации функциональности «ЧАВО» используются методы предобработки текста на основе Word2Vec [2]. Задача ответа на вопрос пользователя обычно решается как задача классификации, где в качестве целевых классов выступают ответы, а возможные вопросы используются в качестве признаков. Подобные решения широко описаны в литературе (например, в монографии [3]). Также существуют коммерческие системы, реализующие функциональность «ЧАВО» методами машинного обучения, в частности Microsoft QnA Maker.

Но, к сожалению, для ответов на вопросы по базе знаний методы машинного обучения не предлагают подобных устойчивых решений. Наиболее близкими аналогами таких методов являются рекуррентные нейронные сети (топологии LSTM [4], Seq2Seq [5]). Эти топологии активно используются в машинном переводе, а также при ответах на вопросы общего характера. Основной проблемой при использовании этой технологии для ответов на вопросы по базе знаний является неустойчивость нейронной сети к схожим вопросам. Небольшая вариация слов во входном вопросе может привести к принципиально различным ответам.

Таким образом, функциональность «ЧАВО» может быть полностью реализована с помощью методов машинного обучения, тогда как для ответов на вопросы по базе знаний в целом необходимо использовать другие методы, методы машинного обучения можно применять для этой задачи только в виде вспомогательных инструментов.

3. Предлагаемая структура гибридной диалоговой системы

Основная сложность гибридизации системы состоит в том, что фактически система содержит два разнородных хранилища знаний – «ЧАВО» и базу знаний в виде

денормализованной таблицы. При этом интеграция знаний из обоих хранилищ должна выглядеть для пользователя бесшовной, у пользователя не должно возникать ощущения, что он общается фактически с двумя разными диалоговыми системами.

В организации бесшовной интеграции помогает тот факт, что «ЧАВО», как правило, содержит справочные вопросы общего характера, тогда как база знаний содержит детальную информацию о товарах и услугах.

Структура системы представлена на рис. 1.

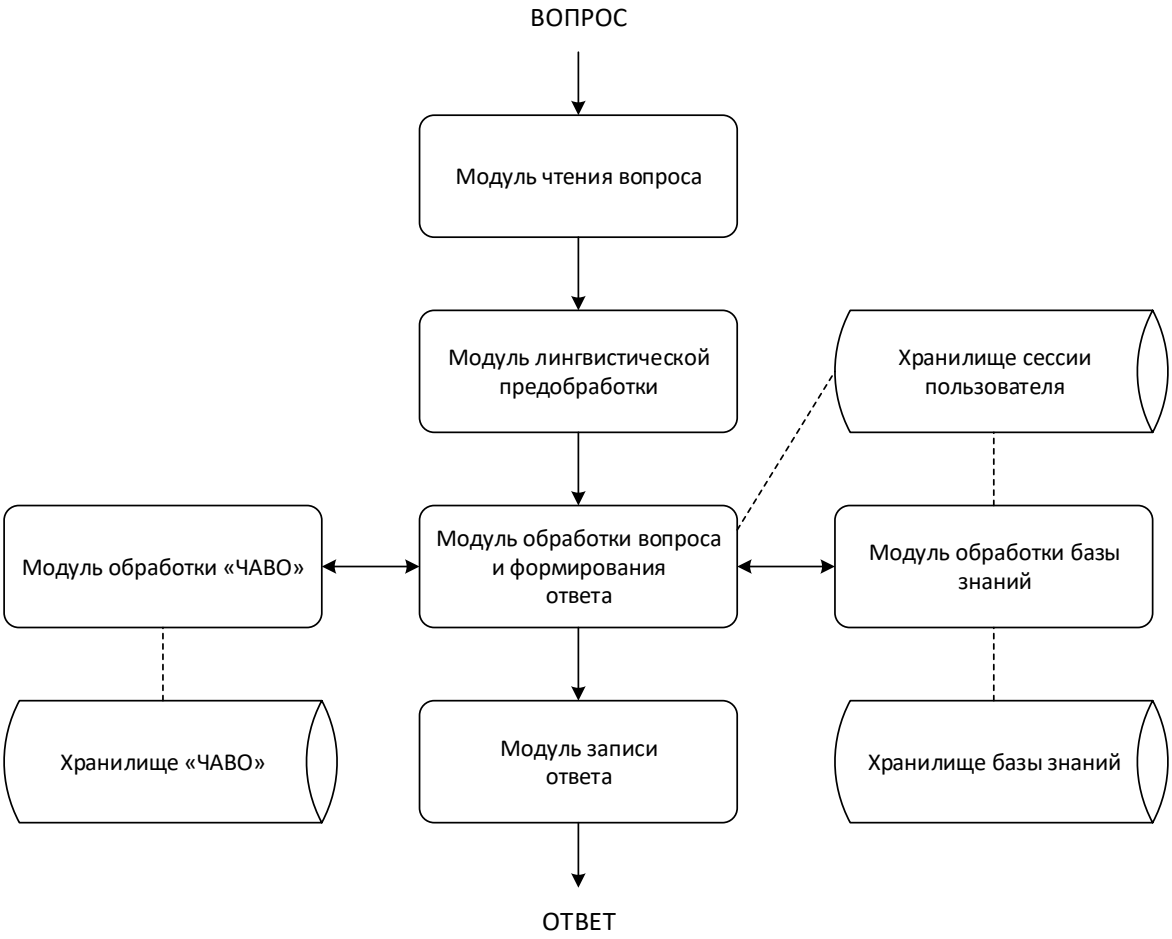

Рис. 1. Структура гибридной диалоговой системы.

Модули чтения вопроса и записи ответа предназначены для интеграции с API сайта или мессенджера, для получения вопроса и записи сформированного ответа.

Модуль лингвистической предобработки осуществляет выделение из текста вопроса лингвистической информации, необходимой для формирования ответа. С точки зрения

архитектуры ГИИС, предложенной в [6] данный модуль является «подсознанием» ГИИС. При этом функцию среды здесь выполняет исходный текст вопроса пользователя. Все остальные модули осуществляют логическую обработку и реализуют «сознание» ГИИС.

Модуль обработки «ЧАВО» реализует требование I и осуществляет формирование ответов на основе хранилища «ЧАВО».

Модуль обработки базы знаний реализует требования II и III, осуществляет формирование ответов на основе хранилища базы знаний и реализует активный диалог с пользователем.

Модуль обработки вопроса и формирования ответа осуществляет вызов модулей обработки «ЧАВО» и базы знаний и интеграцию их ответов. На выходе данного модуля формируется итоговый ответ. Хранилище сессии пользователя позволяет накапливать историю диалога и отвечать на новые вопросы с учетом накопленной истории.

Рассмотрим работу данных модулей подробнее.

4. Модуль обработки «ЧАВО»

Задача данного модуля может быть сформулирована как задача классификации, где в качестве классифицируемого объекта выступает сообщение пользователя, а в качестве меток – множество вопросов из таблицы «вопрос-ответ». Таким образом задача модуля сводится к нахождению в таблице «вопрос-ответ» вопроса, максимально похожего на заданный пользователем вопрос, и возврату соответствующего ему ответа.

Для того, чтобы сравнить схожесть пары предложений нужно использовать некоторую метрику и преобразовать предложения в векторное представление. Для векторизации предложений использовались алгоритмы TF-IDF и Doc2Vec. В качестве метрик использовалось косинусное и евклидово расстояние.

С технической точки зрения можно выделить два подхода к решению поставленной задачи. Первый подход заключается в построении большой матрицы, строками которой

являются отдельные вопросы из базы знаний, а столбцами – веса слов TF-IDF. В результаты мы получаем матрицу T размерностью $N \times M$, где N – количество вопросов, M – число уникальных слов в словаре.

Для правильной работы TF-IDF, его необходимо обучать на большом корпусе текстов. Чем больше корпус, тем лучше учитывается специфичность (важность) слов – параметр, за который и отвечает множитель IDF. Для обучения TF-IDF использовался неразмеченный корпус Open Corpora. Корпус содержит около 108 тысяч предложений и порядка 57 тысяч уникальных слов (с учетом стемминга). Нахождение наиболее похожего ответа в случае использования косинусного расстояния сводится к нахождению максимума от произведения матрицы T на транспонированный вектор искомого вопроса v: $id = \arg\max M * v^T$, где *id* – это индекс наиболее подходящего вопроса в списке «ЧАВО».

Таким образом, все предложения списка «ЧАВО» представлены в одном векторном пространстве размерностью M. Матрица T при этом сильно разрежена, ведь в большинстве предложений 5-7 слов, но никак не 57 тысяч. Плюсом данного подхода является возможность применения хорошо оптимизированных матричных операций. Минусом подхода являются временные издержки на обучение модели и формирование матрицы, а также сложность реализации дообучения, так как при изменениях пар «вопрос-ответ» матрицу нужно перестраивать.

Второй, более гибкий подход к реализации данной задачи исходит из следующей идеи: так как вектора, полученные от TF-IDF далее используются для определения косинусного расстояния, то нет никакого смысла сравнивать и хранить нулевые значения в векторах. Следовательно, нет необходимости переводить все предложения в одно общее пространство размерности M. Достаточно последовательно, на каждом шаге сравнения двух предложений,

создавать небольшие пространства, образованные объединением множеств слов, присутствующих в обоих предложениях.

С математической точки зрения первый и второй подходы эквивалентны, так как перемножения нулевых элементов (которые соответствуют словам, отсутствующим и в текущем предложении таблицы «вопрос-ответ», и в пользовательском вопросе) никак не влияют на результат. С технической точки зрения второй подход хорош тем, что нет необходимости синхронизировать изменения в базе знаний и в полученной матрице. Это обусловлено тем, что в данном подходе нет матрицы – вместо матрицы вектора небольших размерностей строятся непосредственно в момент поиска вопроса и их размерность зависит от вопроса пользователя. Недостатком второго подхода является невозможность оптимизации вычислений путем применения матричных операций, так как вектора не находятся в одном векторном пространстве.

Для векторизации текстовых предложений применялись две модели: TF-IDF и Doc2Vec. Ниже приведены результаты сравнения этих двух моделей, а также их взвешенного ансамбля.

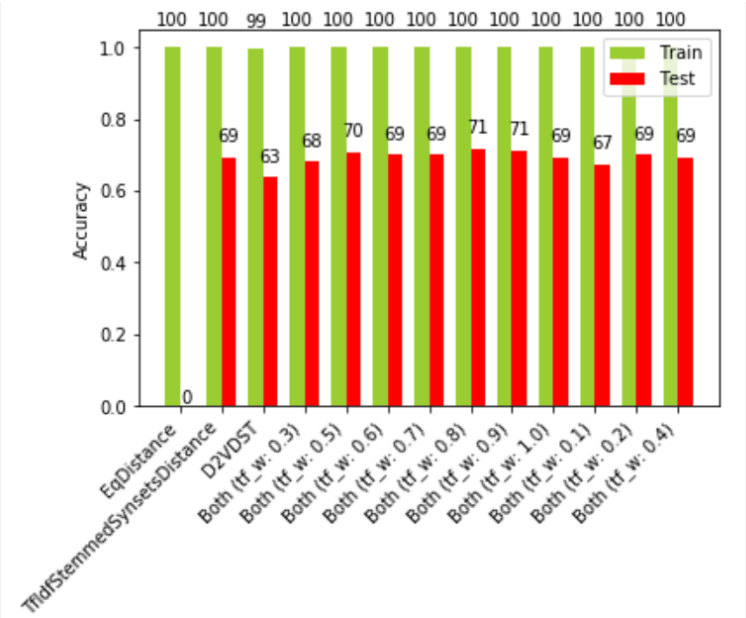

Рис. 2. Сравнение TF-IDF и Doc2Vec.

Таблица 1. Сравнение TF-IDF и Doc2Vec.

Модель	**Точность на обучающей выборке**	**Точность на тестовой выборке**
EqDistance	1.00	0.00
TfldfStemmedSynsetsDistance	1.00	0.69
D2VDST	0.99	0.64
Both (tf_w: 0.1)	1.00	0.67
Both (tf_w: 0.2)	1.00	0.70
Both (tf_w: 0.3)	1.00	0.68
Both (tf_w: 0.4)	1.00	0.69
Both (tf_w: 0.5)	1.00	0.71
Both (tf_w: 0.6)	1.00	0.70
Both (tf_w: 0.7)	1.00	0.70
Both (tf_w: 0.8)	1.00	**0.72**
Both (tf_w: 0.9)	1.00	0.71
Both (tf_w: 1.0)	1.00	0.69

EqDistance – это простая модель, которая сравнивает вопрос на равенство строк, служит для контроля правильности работы оценивающих модели алгоритмов. На обучающей выборке ее точность составляет 100%, а на тестовой составляет 0%.

TfIdfStemmedSynsetsDistance – это модель, на базе TF-IDF.

D2VDST – модель Doc2Vec.

Both (tf_w: 0.3) – это ансамбль из обеих моделей (TF-IDF и Doc2Vec), в скобках отмечен баланс (вес) модели TF-IDF.

Как видно из графика и таблицы наилучший результат – 0.72 на тестовой выборке показал ансамбль из двух моделей: TF-IDF с весом 0.8 и Doc2Vec с весом 0.2.

Также было проведено сравнение двух метрик – косинусного и евклидова расстояний. Результаты представлены в таблице 2.

Таблица 2. Сравнение косинусного и евклидова расстояний.

Метрика	**Точность на обучающей выборке**	**Точность на тестовой выборке**
Косинусное расстояние	1.00	**0.71**
Евклидово расстояние	1.00	0.64

Лучший результат показало использование косинусного расстояния.

Таким образом, для реализации модуля ЧАВО была использована комбинация мер TF-IDF с весом 0.8 и Doc2Vec с весом 0.2 а также косинусное расстояние.

5. Модуль обработки базы знаний

Логическая модель базы знаний представляет собой денормализованную таблицу, содержащую концепты. Столбцы таблицы могут быть ключевыми и неключевыми (зависимыми от ключевых). Комбинация ключевых столбцов таблицы определяет уникальность строки таблицы. Формально таблицу можно описать следующим образом:

$$TL = \left\langle \{t_i\}, \{kc_j\}, \{nkc_k\} \right\rangle, t_i = \{kd_{ij}\} \to \{nkd_{ik}\}, kc_j = \{kd_{ij}\}, nkc_k = \{nkd_{ik}\},$$

где TL – денормализованная таблица; t_i – i-я строка денормализованной таблицы; kc_j – j-й ключевой столбец; nkc_k – k-й неключевой столбец; kd_{ij} – данные i-й строки j-го ключевого столбца; nkd_{ik} – данные i-й строки k-го неключевого столбца.

Таким образом, строка таблицы представляет собой множество данных ключевых столбцов, от которых зависит множество данных неключевых столбцов.

Пример денормализованной таблицы приведен на рис. 3.

	kc_1	kc_2	nkc_1
	Товар	**Марка**	**Стоимость (руб.)**
t_1	шариковая ручка		200
t_2	карандаш	Crayola	120
t_3	карандаш	Kores	150

Рис. 3. Пример денормализованной таблицы.

Не смотря на простоту модели денормализованной таблицы с точки зрения пользователя, данная модель не может быть использована напрямую в качестве физической модели базы

знаний, так как не позволяет достаточно эффективно хранить и обрабатывать данные в информационной системе. Внутри информационной системы денормализованная таблица представляется с использованием метаграфовой модели.

Метаграфовая модель [6, 7] может быть охарактеризована как разновидность «сети с эмерджентностью», где фрагмент сети, состоящий из вершин и связей, может выступать как отдельное понятие, называемое метавершиной.

Наличие у метавершин собственных атрибутов и связей с другими вершинами является важной особенностью метаграфов. Это соответствует принципу эмерджентности, то есть приданию понятию нового качества, несводимости понятия к сумме его составных частей. Фактически, как только вводится новое понятие в виде метавершины, оно «получает право» на собственные свойства, связи и т.д., так как в соответствии с принципом эмерджентности новое понятие обладает новым качеством и не может быть сведено к подграфу базовых понятий.

Рассмотрим представление примера денормализованной таблицы (приведен на рис. 3) в виде фрагмента метаграфа (приведен на рис. 4).

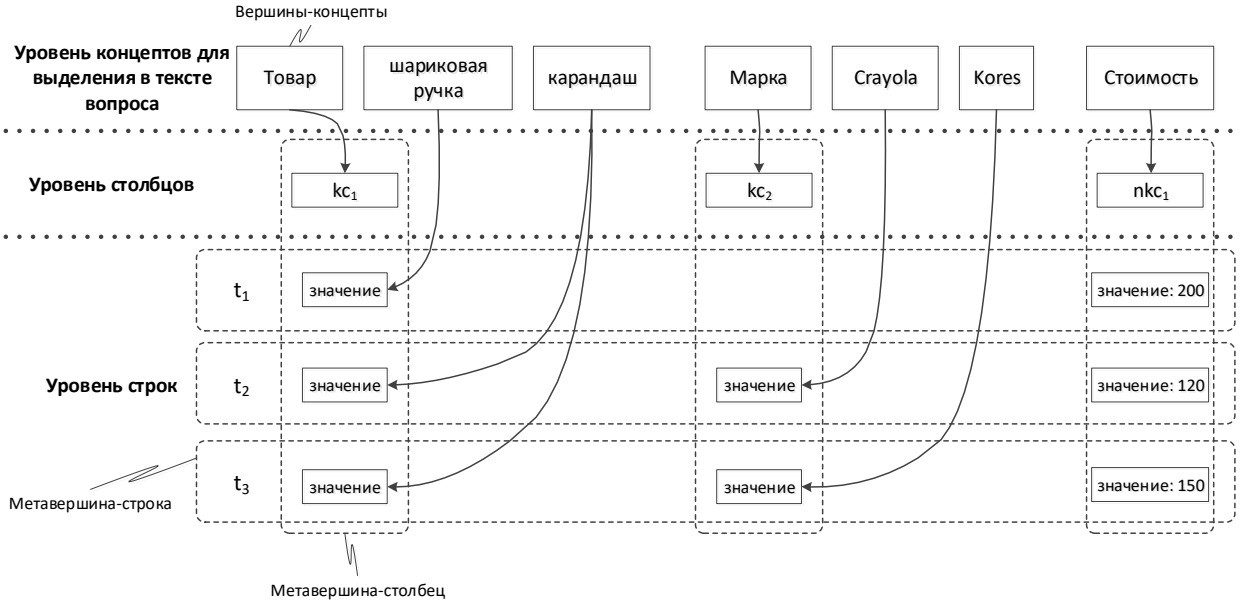

Рис. 4. Пример денормализованной таблицы, представленной в виде фрагмента метаграфа.

Представленное на рисунке «послойное» хранение концептов-метавершин соответствует логике их обработки. Сначала в вопросе пользователя распознаются концепты, соответствующие названиям и значениям столбцов, затем они фильтруются по различным правилам и используются для формирования ответа.

Необходимо также отметить, что каждый концепт-вершина, представленный на рис. 4, содержит множество атрибутов (в частности транслитерированный и фонетизированный варианты слов), которые используется для обработки данных (не показаны, чтобы не загромождать рис. 4). Наличие одновременно сложных связей и атрибутов делает модель денормализованной таблицы совершенно неприменимой в качестве физической модели базы знаний без вспомогательного представления с помощью метаграфовой модели.

Представление базы знаний с использованием метаграфовой модели позволяет отвечать на вопросы пользователя. Но для ведения активного диалога с пользователем необходимо вести историю диалога и определять цели диалога.

История диалога не обязательно должна быть полным списком реплик пользователя. Достаточно хранить список высказанных пожеланий пользователя, наложенный на базу знаний. В нашем случае эта информация хранится в сессии пользователя в формате

$$Session_{User} = \left\langle \left\{ kc_j, kd_{ij} \right\} \right\rangle,$$

где $Session_{User}$ – сессия пользователя *User*; kc_j – j-й ключевой столбец; kd_{ij} – данные i-й строки j-го ключевого столбца.

Таким образом, в сессии хранится информация в виде пар «ключевой столбец»: «данные ключевого столбца». Пример сессионной информации, основанный на рис. 3 может быть следующим: *{Товар: карандаш, Марка: Crayola}*.

Для ведения истории диалога используются следующие правила обработки вопроса пользователя:

- Если вопрос пользователя содержит концепты, соответствующие kd_{ij} (данные i-й строки j-го ключевого столбца), то для каждого из этих концептов ищется соответствующий ключевой столбец kc_j и пара $kc_j : kd_{ij}$ сохраняется в сессию пользователя. Если сессия содержала пару, соответствующую столбцу kc_j, то эта пара предварительно удаляется. Например, если вопрос пользователя содержит концепт «Crayola», то в сессию сохраняется пара *{Марка: Crayola}*, если до этого пользователь указывал другую марку, то эта информация удаляется.
- Если вопрос пользователя содержит концепты, соответствующие ключевому столбцу kc_j, то выводится множество значений этого столбца с учетом данных сессии. Например, если вопрос пользователя содержит концепт «товары», то система выведет в ответ список товаров: «шариковая ручка», «карандаш». Но если в сессии хранится пара *{Товар: карандаш}*, то пользователю будет выдано сообщение о том, что он заказывает карандаши.

Ведение истории диалога является вспомогательным механизмом, который используется для реализации целей диалога. В текущей версии системы основной целью диалога является детализация запроса до уровня одной строки таблицы. Цель реализуется следующим образом:

- После обработки вопроса пользователя и сохранения в сессии найденных концептов производится фильтрация метавершин на основе данных сессии,

соответствующих строкам денормализованной таблицы и оценивается количество найденных строк.

- Если найдена одна строка, то цель считается выполненной и пользователю выводится информация о неключевых параметрах найденной строки, например, информация о стоимости товара.
- Если найдено более одной строки, то система реализует активный диалог с пользователем и пытается выполнить цель, задавая вспомогательные вопросы. Для этого определяется ключевой столбец с минимальным разнообразием, то есть определяется какой столбец содержит минимальное количество возможных концептов с учетом фильтра на основе данных текущей сессии. Например, если в сессии хранится пара *{Товар: карандаш}*, то столбцом с минимальным разнообразием будет столбец «Марка», содержащий концепты «Crayola» и «Kores». Система выведет вспомогательный вопрос со списком найденных марок и попросит пользователя выбрать марку. Выбранная марка будет занесена в сессию. Активный диалог будет продолжаться до тех пор, пока цель не будет выполнена, то есть пока не будет однозначно определена строка денормализованной таблицы.
- Если пользователь в процессе диалога задает несовместимый набор концептов, который приводит к нулевому количеству строк в результате фильтрации, то выводится сообщение о том, что для данных параметров не найдены данные и производится очистка сессии пользователя.

Таким образом, применение метаграфового подхода для хранения базы знаний позволяет использовать метавершины и как элементы данных для ответа на вопросы и как информационные элементы, используемые для реализации активного диалога.

6. Модуль лингвистической предобработки

Данный модуль должен выделить как признаки, используемые для методов машинного обучения модуля обработки «ЧАВО», так и концепты, используемые в модуле обработки базы знаний.

Чтобы нивелировать склонения русского языка и орфографические ошибки в вопросе пользователя, текст приводится к нормальной форме. Последовательность действий по предобработке вопроса приведена на рис. 5.

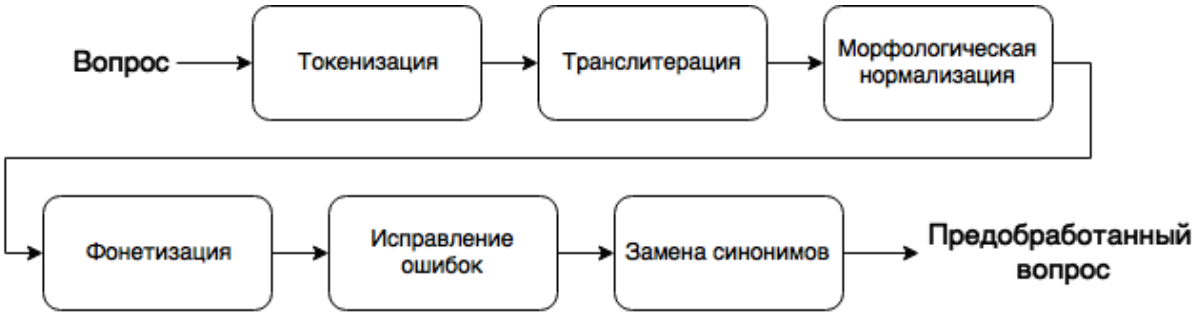

Рис. 5. Последовательность действий по предобработке вопроса.

Рассмотрим более подробно каждое действие, представленное на рис. 5.

1. Токенизация. Предложение разбивается на отдельные слова с учетом пробелов, переноса строк и пунктуации.
2. Транслитерация. Все латинские символы заменяются на аналогичные кириллические.
3. Морфологическая нормализация. Слова приводятся к начальной форме:
    - Существительные, числительные и прилагательные приводятся к именительному падежу, единственному числу.
    - Глаголы, причастия и деепричастия приводятся к инфинитивной форме глагола.
4. Фонетизация. Использовалась адаптация алгоритма Metaphone [8] под русский язык. Алгоритм фонетизации описан ниже.

5. Исправление ошибок. Алгоритм исправления ошибок описан ниже.
6. Замена синонимов. Были разработаны списки синонимов, соответствующие предметной области. Если слово или словосочетание из вопроса пользователя оказывалось в этих списках, то оно заменялось на уникальный идентификатор, который использовался в дальнейшей обработке.

Рассмотрим более подробно алгоритмы фонетизации и исправления ошибок.

Алгоритм фонетизации:

1. Гласные заменяются на те, что слышатся на их месте в безударном слоге. Используется следующая таблица перевода:
    - О, Ы, А, Я → А;
    - Ю, У → У;
    - Е, Ё, Э, И → И.
2. Исключаются повторяющиеся символы.
3. Оглушаются согласные на слабой позиции. Слабой считается такая позиция (место в слове) звука, при которой он слышится неясно, неотчётливо. Такими позициями для согласных звуков являются:
    - расположение согласного звука в конце слова: дуб (дуп), верблюд (вирблют);
    - расположение согласного звука перед другим согласным (кроме сонорных) – при так называемом стечении согласных, когда их несколько в слове: пробка (пропка), скобка (скопка).
4. Повторно исключаются повторяющиеся символы.
5. Сжимаются окончания. Схожие окончания заменяются на один уникальный спецсимвол, соответствующий списку схожих окончаний:
    - -ук, -юк;

- -ик, -ек;
- -ов, -ев, -иев;
- -ых, -их;
- -ий, -ый;
- -ова, -ева, -иева.

6. Исключаются твердый и мягкий знаки.

Алгоритм исправления ошибок:

1. Предварительно на этапе запуска приложения загружается корпус текстов. Использовался корпус Open Corpora.
2. Если слово из вопроса пользователя уже есть в корпусе, то алгоритм заканчивает свою работу.
3. Корпус сортируется по популярности слов по убыванию.
4. Для каждого слова из корпуса вычисляется расстояние Левенштейна между этим словом и словом из вопроса пользователя.
5. Выбирается замена с минимальным расстоянием Левенштейна. Если кандидатов несколько, то из них выбирается слово с максимальной популярностью в корпусе.
6. Вычисляется процент отличия слова по формуле $\rho = d/S$, где ρ – процент отличия слова; d – расстояние Левенштейна между словом из вопроса пользователя и словом из корпуса; S – длина слова из вопроса пользователя.
7. Если процент отличия слова не превышает заданный порог ρ_T, то возвращается найденное слово из корпуса. Иначе возвращается исходное слово вопроса пользователя без изменений.

Для предобработки базы знаний используется аналогичная последовательность действий, но с выключенным блоком исправления ошибок. Предполагается, что текстовая

информация, загружаемая в базу знаний, составлена корректно и при загрузке сопоставление с корпусом текстов не требуется. В то же время, вопрос пользователя может содержать ошибки, которые требуют сопоставления с корпусом текстов.

7. Модуль обработки вопроса и формирования ответа

Модуль работает на основе следующих правил:

- Текст вопроса передается модулю обработки «ЧАВО». Если текст вопроса с высокой степенью соответствия найден в «ЧАВО», то осуществляется ответ из модуля обработки «ЧАВО». Считается, что это общий вопрос и он имеет приоритет перед частными вопросами из базы знаний.
- Если текст вопроса не найден в «ЧАВО» или найден с низкой степенью соответствия, то найденные концепты передаются в модуль обработки базы знаний и ответом на вопрос считается ответ этого модуля.
- Если концепты не найдены или модуль обработки базы знаний вернул пустой ответ, то выводится стандартный ответ, содержащий просьбу уточнить вопрос к системе.

Таким образом, осуществляется интеграция модулей обработки «ЧАВО» и базы знаний. В случае общих вопросов приоритет отдается модулю обработки «ЧАВО».

8. Результаты экспериментов – пример диалога с системой

Рассмотрим пример диалога с системой. В диалоге используются следующие обозначения: «П» – пользователь, «С» – диалоговая система, «К» – комментарии по обработке вопросов и ответов.

С: Здравствуйте. Я буду рада ответить на Ваши вопросы по ассортименту товаров.

П: Я хочу купить ручку.

К: Слово «ручка» распознано как слово, входящее в концепт «шариковая ручка». База знаний содержит единственную запись с таким концептом.

С: Мы можем предложить Вам шариковую ручку по цене 200 руб. Оформить заказ?

П: Нет.

С: Спасибо. Я готова ответить на другие вопросы.

П: Я хочу купить карандаш.

К: Система выделила концепт «карандаш», определила что записей с таким концептом несколько и пытается предложить марку карандаша.

С: У нас есть карандаши следующих марок: «Crayola» и «Kores». Какой карандаш Вы желаете приобрести?

П: В чем особенность карандашей Crayola?

К: Система выделила концепт «Crayola». Но при этом с высокой степенью соответствия выделено сходство вопроса пользователя с вопросом «ЧАВО». Производится ответ на основе «ЧАВО».

С: Особенностью карандашей Crayola является …

П: Тогда хочу купить карандаш Crayola.

К: Система выделила концепт «Crayola». «ЧАВО» не содержит вопросов, похожих на вопрос пользователя. База знаний содержит единственную запись с концептами «карандаш» и «Crayola».

С: Карандаш марки «Crayola» стоит 120 руб. Оформить заказ?

П: Да.

К: Далее следует процедура оформления заказа.

9. Выводы и дальнейшие направления работы

В статье рассмотрен прототип гибридной диалоговой системы, реализующий ответы на вопросы на основе «ЧАВО» и базы знаний.

Сложность гибридизации состоит в том, что модуль обработки «ЧАВО» реализован с использованием алгоритмов машинного обучения, тогда как модуль обработки базы знаний реализован на основе метаграфового подхода.

Применение метаграфового подхода для хранения базы знаний позволяет использовать метавершины и как элементы данных для ответа на вопросы и как информационные элементы, используемые для реализации активного диалога.

Прототип системы успешно реализован, приведены результаты экспериментов.

К недостаткам существующей версии системы и направлениям дальнейшей работы можно отнести следующее:

- В настоящее время в системе не предусмотрены операции группировки на основе базы знаний – вычисление количества, суммы и т.д.
- Не осуществляется полный синтаксический разбор вопросов пользователя и, как следствие, не учитывается иерархическая связь между концептами.

Литература

1. Ranoliya B.R., Raghuwanshi N., Singh S. Chatbot for university related FAQs, International Conference on Advances in Computing, Communications and Informatics (ICACCI’2017), Udupi, 2017, pp. 1525-1530. doi: 10.1109/ICACCI.2017.8126057

2. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2013, pp. 3111-3119.

3. R. Khan, A. Das. Build Better Chatbots, Apress, 2018.

4. Sundermeyer M., Ney H., Schlüter R. 2015. From feedforward to recurrent LSTM neural networks for language modeling. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 23(3) 2015, pp. 517-529. doi: http://dx.doi.org/10.1109/TASLP.2015.2400218

5. Sutskever I., Vinyals O., Le Q.V. Sequence to sequence learning with neural networks. Advances in neural information processing systems 2014, pp. 3104–3112.

6. Черненький В.М., Терехов В.И., Гапанюк Ю.Е. Структура гибридной интеллектуальной информационной системы на основе метаграфов. Нейрокомпьютеры: разработка, применение. 2016. Выпуск №9. С. 3-14.

7. Черненький В.М., Гапанюк Ю.Е., Ревунков Г.И., Терехов В.И., Каганов Ю.Т. Метаграфовый подход для описания гибридных интеллектуальных информационных систем. Прикладная информатика. 2017. № 3 (69). Том 12. С. 57–79.

8. Parmar V.P., Kumbharana C.K. Study existing various phonetic algorithms and designing and development of a working model for the new developed algorithm and comparison by implementing it with existing algorithms. J. Comput. Appl. 98(19) 2014, pp. 45–49.