Моделирование и анализ информационных систем. Т. 25, № 4 (2018), с. 411–420
Modeling and Analysis of Information Systems. Vol. 25, No 4 (2018), pp. 411–420

Вопросно-ответные системы

DOI: 10.18255/1818-1015-2018-4-411-420

УДК 517.9

Вопросно-ответная система для поддержки абитуриентов с использованием современных мессенджеров

Филонов Д. Р., Чалый Д. Ю.[1], Мурин Д. М., Дурнев В. Г., Соколов В. А.[1]

получена 31 июля 2018

Аннотация. В настоящее время растет интерес пользователей к приложениям для мгновенного обмена сообщениями, мессенджерам. Они позволяют не только общаться с другими пользователями, но и включают в себя функционал, помогающий создавать автоматических собеседников, автоматизирующих отдельные бизнес-процессы либо удовлетворяющих информационные потребности пользователей. В статье рассматривается узкоспециализированная вопросно-ответная система, которая использует инфраструктуру, предоставляемую современными мессенджерами для обмена сообщениями и предназначенную для информационной поддержки абитуриентов, поступающих в университет. В процессе разработки системы был накоплен корпус характерных вопросов абитуриентов, а также разработана модель, которая позволяет осуществлять поиск близких вопросов по этому корпусу. При этом абитуриент может формулировать свои вопросы на естественном языке без изучения предварительно заданных шаблонов или специальных правил построения сообщений. Для получения ответов на вопросы, которых нет в корпусе системы, привлекаются специалисты приемной комиссии, имеющие свой интерфейс. Система была реализована с использованием современных облачных технологий, предоставляемых компанией Amazon. Среди них бессерверные (serverless) вычисления и NoSQL-базы данных. Для этого была разработана архитектура сервиса, удовлетворяющая этой модели вычислений. Поскольку система может оперировать с чувствительными данными, включая персональные данные, а также предоставлять персонализированный сервис, были проанализированы методы обеспечения безопасности системы, а также подходы к авторизации пользователей. В настоящее время система проходит тестирование и оценку используемых алгоритмов информационного поиска.

Ключевые слова: вопросно-ответная система, мессенджер, информационный поиск, облачные системы и сервисы, безопасность

Для цитирования: Филонов Д. Р., Чалый Д. Ю., Мурин Д. М., Дурнев В. Г., Соколов В. А., "Вопросно-ответная система для поддержки абитуриентов с использованием современных мессенджеров", *Моделирование и анализ информационных систем*, **25**:4 (2018), 411–420.

Об авторах: Филонов Дмитрий Русланович, orcid.org/0000-0002-6402-5114, магистр,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: frbdima@gmail.com

Чалый Дмитрий Юрьевич, orcid.org/0002-0007-1896-0876, канд. физ.-мат. наук, доцент,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: chaly@uniyar.ac.ru

Мурин Дмитрий Михайлович, orcid.org/0000-0002-8068-0784, канд. физ.-мат. наук, доцент,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: nirum87@mail.ru

Дурнев Валерий Георгиевич, д-р физ.-мат. наук, профессор,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: durnev@uniyar.ac.ru

Соколов Валерий Анатольевич, orcid.org/0000-0003-1427-4937, д-р физ.-мат. наук, профессор,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: sokolov@uniyar.ac.ru

Благодарности:
[1] Работа выполнена при поддержке проекта «Моделирование и анализ информационных систем» (инициативный проект ЯрГУ № АААА-А16-116070610022-6).

Введение

Поиск информации является одной из базовых потребностей человека в современном информационном обществе. Начиная с середины 60-х годов [2, 8] развивались системы, перед которыми ставились задачи обработки текстов на естественном языке с целью выделения в них смысловых единиц, которые потом использовались для удовлетворения информационных потребностей пользователей. Одним из видов таких систем являются вопросно-ответные системы, которые в настоящее время стали перспективным направлением исследований [3, 6, 7]. При этом выделяются системы [8], основанные на подходах, используемых в информационном поиске (IR, information retrieval) [1], а также системы, отвечающие на вопросы, используя знания о фактах, скажем, статистического характера. Первая парадигма, IR-системы, полагается на то, что существует большой корпус документов, в котором содержится ответ на вопрос пользователя. Поиск ответа во многом похож на то, как работают поисковые системы: вопрос пользователя преобразуется в запрос для поисковой системы с учетом контекста, а система выдает ранжированный список документов, из которых извлекается ответ. Вторая парадигма представляет вопрос в виде логического предиката, а ответ – значение свободной переменной этого предиката, которая делает его истинным.

В нашей работе мы применяем подходы, характерные для IR-парадигмы вопросно-ответных систем. Своей задачей мы ставим разработку узкоспециализированной системы для удовлетворения информационных потребностей абитуриентов, поступающих в университет. Поскольку большинство абитуриентов делают это первый раз, то у них возникает много вопросов, касающихся процесса поступления, формальных требований к абитуриентам, текущей конкурсной ситуации, информации о процессе обучения, культурной, спортивной и творческой жизни в университете. Для поиска ответов на эти вопросы можно использовать сайт университета, группы в соцсетях и другие информационные источники, включая общение с представителями приемной комиссии, а также выпускниками и студентами университета. Однако это требует существенных усилий и может приводить к получению недостоверной информации. Поэтому наличие ассистента, который способен находить ответы в достоверных источниках, могло бы существенно упростить процесс получения информации, включая персонализированную справку о текущей конкурсной ситуации. Это, в частности, поднимает вопросы безопасности, включая то, каким образом должен авторизоваться абитуриент, как хранить чувствительную информа-

цию в анонимизированном виде и как безопасно передавать ее по коммуникационным сетям. В нашей работе мы обобщаем и развиваем подход, который реализуется в работе [4]. При этом мы включаем членов приемной комиссии как экспертов, которые могут знать ответы на те вопросы, которые еще не внесены в систему. Это становится возможным благодаря интерфейсам, существующим в современных мессенджерах.

1. Модель вопросно-ответной системы

Целью вопросно-ответной системы, которая основывается на подходах, характерных для информационного поиска, является поиск кратких текстовых фрагментов в корпусе документов, которые являются ответами на вопрос пользователей. В контексте таких систем подобные ответы еще называют фактоидами. Наш опыт показывает, что существенная часть вопросов абитуриентов имеет именно такой характер. Согласно Тейлору [2] все информационные запросы можно разделить в зависимости от их четкости на следующие категории:

1. «Идеальный вопрос», который пользователь не может сформулировать реально, но присутствующий в его подсознании неоформленно; осознанное отношение к информационной потребности, но в неопределённой форме (при этой степени понимания индивид может обратиться к знающему человеку, который поможет ему оформить запрос).

2. Вопрос исследователя, осознающего, что он ищет и в каком виде.

3. Вопрос в таком виде, в каком он представлен в информационной системе.

Вопросно-ответные системы на первом шаге пытаются определить тип вопроса и вычленить смысловые сущности, которые должен содержать ответ. Наша система предназначена прежде всего для решения случая 3 и в какой-то степени случая 2, когда вопрос содержит термины, прямо сигнализирующие об информационной потребности пользователя. Для решения вопросов, которые относятся к категории 1, мы включаем во взаимодействие внутри информационной системы представителей приемной комиссии.

Важным компонентом рассматриваемого типа вопросно-ответных систем является корпус документов, по которым производится поиск ответов. В течение приемных кампаний 2016 и 2017 года в группе ВКонтакте для абитуриентов факультета ИВТ производилась активная работа по информированию абитуриентов и ответам на возникающие вопросы. Поскольку ответы публиковались в формате один пост – один ответ, это позволило накопить базу вопросов и ответов на них. Все такие публикации были скачаны автоматизированным образом, используя API ВКонтакте. В результате была сформирована база из вопросов (а также соответствующих им ответов) в количестве 141. Эти данные являются базой знаний нашей системы. Их отличает краткость — вопросы чаще всего содержат небольшое количество терминов. Наша экспертная оценка, которая основывается на повторяемости вопросов в каждый год приема, показывает, что данная база вопросов содержит существенную долю информационных потребностей, в которых ответы являются фактоидами.

Каждый вопрос рассматривается в виде «мешка слов» и при помощи модели векторного пространства [1] представляется в виде многомерного вектора. Это одна из фундаментальных моделей информационного поиска, которая позволяет осуществлять поиск документов по запросу, ранжирование, классификацию и кластеризацию документов [1]. Использование этой модели возможно и для вопросно-ответных систем [5]. Размерность векторного пространства определяется мощностью словаря терминов коллекции документов. Рассмотрим подход, который мы используем для преобразования вопросов на естественном языке в вектор.

Сначала из вопроса убираются стоп-слова, т.е. такие термины, которые являются общеупотребительными и не несут существенной смысловой нагрузки (например, предлоги). Координаты преобразованного в виде вектора документа можно представить несколькими способами. Так, i-я координата может быть равна:

1. Значению «истина», если i-й термин словаря присутствует в вопросе, и «ложь» в противном случае.

2. Частоте вхождения i-го термина в документ.

3. Частоте вхождения i-го термина в документ, умноженной на обратную документную частоту. Документная частота равна количеству документов, содержащих i-й термин.

Для данной коллекции документов наиболее рациональным методом взвешивания координат векторов является использование только частот терминов. Домножать документы на их обратную частоту не целесообразно, так как тексты достаточно короткие и в них отсутствуют широкоупотребительные слова (а те, что встречаются, фильтруются списком стоп-слов). Использование исключительно логических значений также не рационально, так как некоторые термины могут встречаться несколько раз, поэтому частота термина является оптимальным показателем релевантности. Для снижения размерности векторов можно использовать стемминг или лемматизацию [1]. В итоге мы получаем набор векторов, которые моделируют известные системе информационные потребности.

Вопрос пользователя при помощи аналогичного подхода преобразуется в вектор, и при помощи косинусной меры сходства выбирается наиболее близкий известный системе вопрос. В качестве результата система выдает известный ей ответ на этот вопрос.

2. Архитектура вопросно-ответной системы

Разработка системы велась на языке программирования Python 3.6 с использованием фреймворка Serverless для разворачивания на сервисе бессерверных вычислений AWS Lambda [9]. Для хранения данных и обновления индекса системы используется база данных Amazon DynamoDB [10]. Все это позволяет реализовать масштабируемый и отказоустойчивый сервис.

AWS Lambda — это управляемая событиями (event-driven), бессерверная вычислительная платформа, предоставляемая в составе Amazon Web Services. Это вычислительная служба, которая запускает программный код в ответ на события

и автоматически управляет вычислительными ресурсами, которые требуются для выполнения этого кода.

Serverless — это архитектурный стиль, в котором приложения в значительной степени разрабатываются с использованием подходов Backend-as-a-Service (BaaS) и Functions-as-a-Service (FaaS). BaaS — это когда в клиентской части приложений используются сторонние удаленные сервисы. А FaaS — это когда в серверной части приложений логика представляет собой функции, запускаемые в облаке.

После загрузки программного кода сервис AWS Lambda обеспечивает все ресурсы, необходимые для его выполнения и масштабирования, с высокой степенью доступности. Можно настроить автоматический запуск программного кода из других сервисов AWS или непосредственно из любого мобильного или веб-приложения. При бессерверных вычислениях приложение по-прежнему работает на серверах, но AWS полностью берет на себя управление этими серверами. Таким образом, бессерверные приложения обладают тремя ключевыми преимуществами:

- отсутствие необходимости управлять серверами;
- гибкость масштабирования;
- автоматическое обеспечение высокой доступности.

Для хранения базы знаний, извлечения из нее промежуточных данных, пула «горячих» вопросов (т.е. таких, которые адресованы экспертам приемной комиссии) абитуриентов и таблицы с позициями экспертов в этом пуле используется нереляционная база данных Amazon DynamoDB, которая представляет собой быстрый и гибкий сервис баз данных NoSQL. Взамен привычных реляционных баз данных DynamoDB предлагает концепцию таблиц «ключ – значение», которые качественно отличаются от реляционного способа организации данных, так как используют менее строгие схемы описания данных. Каждый элемент (объект, запись) в таблице может отличаться от других количеством и типом атрибутов, при этом поддерживаются первичный и вторичный механизмы индексации, что позволяет избежать потерь в производительности запросов. Значения атрибутов таблицы могут быть строковыми, числовыми, двоичными, а также наборами значений. Размер одной записи не должен превышать 64 Кб. Так как единого интерфейса к NoSQL-продуктам не существует, пользователям необходим оригинальный интерфейс взаимодействия с этой СУБД. Поэтому в архитектуре проекта были развернуты два взаимодействующих уровня Data Access Layer (для реализации функций CRUD: Create, Read, Update, Delete) и уровня сервисов для работы с различного рода данными для каждой из таблиц проекта.

DynamoDB позиционируется как простая в эксплуатации СУБД для онлайновых систем, которые могут масштабироваться до тысяч запросов в секунду с сохранением короткого времени отклика. При этом она, конечно, не сравнится по функциональности с реляционными СУБД и не умеет выполнять сложные запросы, хотя поддерживается ряд атомарных операций, которые, например, изменяют значение конкретного числового поля в записи. Главное, DynamoDB позволяет максимально быстро развернуть онлайновую систему, которой требуется база с достаточно простой организацией данных.

Некоторые преимущества использования DynamoDB:

- быстрая и стабильная работа. Решение Amazon DynamoDB работает стабильно и быстро в системах любого масштаба в любой области применения. Среднее время обработки запроса на сервере составляет несколько миллисекунд. По мере роста объемов данных и повышения необходимой производительности система Amazon DynamoDB обеспечивает соответствие требованиям к пропускной способности и времени обработки запроса с помощью технологий автоматического разбиения на разделы и SSD в системах любого масштаба;

- высокая масштабируемость. При создании таблицы нужно просто указать требуемый объем запросов в единицу времени. Amazon DynamoDB выполняет все операции по масштабированию в скрытом режиме и в ходе их выполнения продолжает обеспечивать соответствие установленным требованиям к пропускной способности;

- гибкость. Amazon DynamoDB поддерживает работу со структурами данных на основе как документов, так и пар «ключ-значение», благодаря чему возможно выбрать оптимальную структуру базы данных с учетом особенностей разрабатываемой системы;

- событийно-ориентированное программирование: Amazon DynamoDB интегрируется с сервисом AWS Lambda для создания триггеров, которые позволяют разрабатывать приложения, автоматически реагирующие на изменение данных.

Общая архитектура сервиса изображена на рис. 1, она состоит из нескольких взаимодействующих уровней:

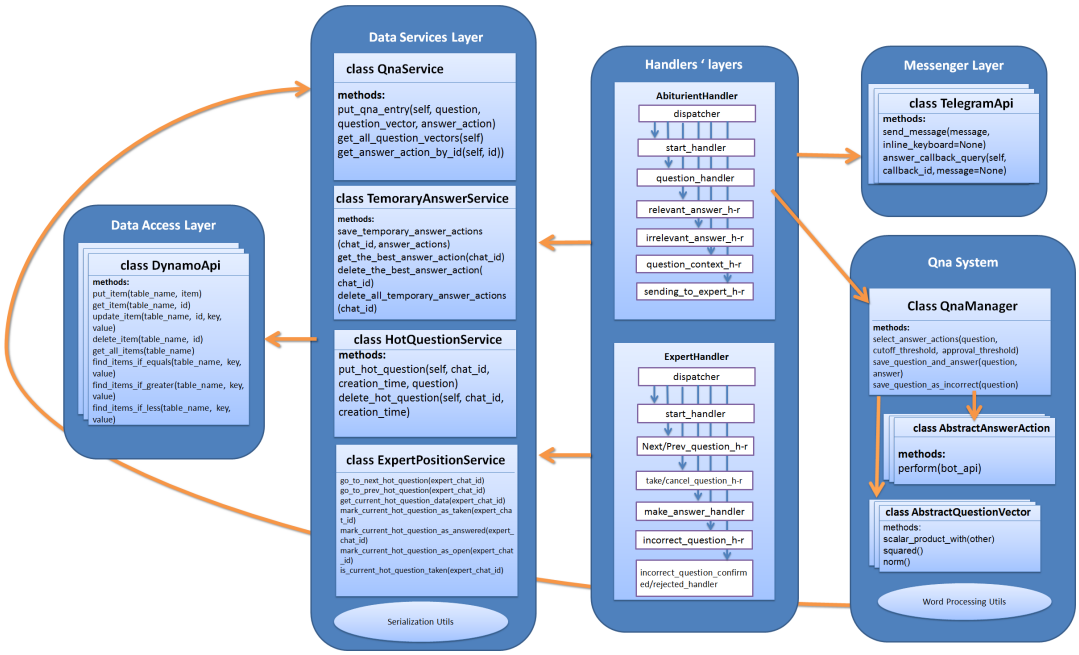

Рис. 1. Схема классов вопросно-ответной системы

Fig. 1. Question answering system class scheme

- уровень вопросно-ответной системы (*Qna System*). Здесь реализуется модель векторного пространства и поиск наиболее релевантных документов в ней. Для обработки естественного языка используются библиотеки NLTK [11] и pymorhy 2 [12]. Класс QnaManager реализует внешний интерфейс взаимодействия с базой знаний;

- уровень доступа к данным (*Data Access Layer*). Слой доступа к данным представлен набором классов, реализующих общий интерфейс взаимодействия с хранилищем данных. В настоящее время реализован класс DynamoAPI, взаимодействующий с нереляционной базой данных Amazon Dynamo DB;

- уровень сервисов для работы с данными (*Data Services Layer*) предназначен для работы с конкретными объектами вопросно-ответной системы: базой вопросов и ответов, очередью релевантных ответов, пулом вопросов пользователей, требующих ответа, а также навигации эксперта приемной комиссии по пулу вопросов абитуриентов, требующих ответа;

- уровень для работы с мессенджерами (*Messenger Layer*), который реализует функционал, требуемый для взаимодействия с конкретным мессенджером;

- уровень обработчиков событий системы (*Handlers' Layer*). Этот уровень реализован с использованием AWS Lambda и включает реакцию системы на различные события, например, получение вопроса от абитуриента, действия экспертов приемной комиссии и т.п.

3. Вопросы безопасности

Каждая система или сервис должны удовлетворять требованиям безопасности, исключающим утечку чувствительных данных. Проанализируем архитектуру разработанной системы с точки зрения безопасности. Система построена с помощью архитектуры «клиент – сервер», в которой клиентом является приложение-мессенджер, а сервер реализован при помощи бессерверного подхода с использованием сервисов Amazon. Безопасность клиентского приложения обеспечивается компанией-разработчиком мессенджера, которая должна своевременно проводить обновления, решающие проблемы безопасности. Серверная часть должна быть реализована с использованием принципов разделения ролей, авторизующих каждый используемый сервис. Это можно реализовать при помощи сервиса Amazon IAM. В нашей системе такой подход пока не реализован, и это является одним из направлений дальнейших исследований. Клиентская и серверная части взаимодействуют между собой через сеть Интернет по специализированному протоколу. Обеспечение корректности и безопасности работы такого протокола также является ответственностью разработчика мессенджера.

Разработанная система может предоставлять персонализированный сервис абитуриентам, однако вопросы авторизации и аутентификации не были нами рассмотрены при обсуждении архитектуры системы. Отсутствие таких возможностей позволяет пользователям получать доступ к закрытой информации, которая должна быть доступна только определенным адресатам. Кроме того, система не должна

хранить в явном виде персональные данные абитуриентов, поскольку ее работа происходит в облаке, принадлежащем сторонней организации. К таким персональным данным относятся, например, номер телефона или адрес электронной почты, которые сообщаются при подаче документов. Очевидно, что такие данные легко позволяют аутентифицировать пользователей. В самом мессенджере пользователи идентифицируются по присвоенным во время регистрации уникальным идентификаторам. Таким образом, процесс аутентификации должен сопоставлять этим идентификаторам реальных абитуриентов, причем без раскрытия их персональных данных.

Мы предлагаем использовать следующую схему аутентификации пользователей, которая исключает хранение в явном виде персональных данных пользователей. При загрузке информации о пользователях в облако рассчитывается значение криптографической хеш-функции (например, с помощью алгоритма MD5) от тех данных, которые известны только пользователю и приемной комиссии. Такими данными могут быть фамилия, имя, отчество, номер телефона, персональный идентификатор абитуриента, который назначается ему при подаче документов. Пользователь аутентифицируется, ответив на несколько вопросов, которые касаются этих данных. При помощи той же самой хеш-функции ответы пользователя преобразуются в его идентификатор, который потом сравнивается с базой загруженных идентификаторов. При совпадении с одним из существующих идентификаторов пользователь аутентифицируется.

Архитектура нашей системы содержит часть, которая должна быть доступна только специалистам приемной комиссии. Мы предлагаем изолировать эту часть в виде отдельного канала используемого мессенджера, к которому дается доступ только для тех учетных записей, которые принадлежат данной категории пользователей.

Рассмотренные подходы позволяют обеспечить безопасность чувствительных данных и предложить безопасное взаимодействие между системой и всеми категориями пользователей.

Заключение

В результате выполненных работ была создана автоматизированная вопросно-ответная система, которая позволяет отвечать на вопросы абитуриентов, сформулированные на естественном языке, а в отдельных случаях привлекать специалистов приемной комиссии. Архитектура системы обеспечивает безопасное взаимодействие, а также является масштабируемой, поскольку реализована при помощи облачных сервисов Amazon. Система включает в себя дополнительный уровень абстракции, который позволяет использовать различные мессенджеры, базы данных или методы извлечения знаний из неструктурированных данных, представленных на естественном языке.

Дальнейшее развитие системы предполагает всестороннее тестирование, проведение оценок качества информационного поиска, подключение новых мессенджеров.

Список литературы / References

[1] Маннинг К.Д., Рагхаван П., Шютце Х., *Введение в информационный поиск*, Вильямс, М., 2011; In English: Manning C.D., Raghavan P., Schutze H., *Introduction to Information Retrieval*, Cambridge University Press New York, NY, USA, 2008.

[2] Taylor R.S., "The Process of Asking Questions", *American Documentation*, **13**:4 (1962), 391–396.

[3] Radev D.R., Prager J., Samn V., "Ranking suspected answers to natural language questions using predictive annotation", *Proceedings of the sixth conference on Applied natural language processing*, 2000.

[4] Филонов Д.Р., Тупикин В.И., "Чат-бот для Telegram для помощи абитуриентам", *Заметки по математике и информатике*, 2017, 152–156; [Filonov D.R., Tupikin V.I., "Chat-bot dlya Telegram dlya pomoschi abiturientam", *Zametki po matematike i informatike*, 2017, 152–156, (in Russian).]

[5] Jovita L., Hartawan A., Suhartono D., "Using Vector Space Model in Question Answering System", *Procedia Computer Science*, **59** (2015), 305—311.

[6] Mishra A., Jain S.K., "A Survey on Question Answering Systems with Classification", *Journal of King Saud University – Computer and Information Sciences*, **28**:3 (2016), 345–361.

[7] Bouziane A., Bouchiha D., Doumi N., Malki M., "Question Answering Systems: Survey and Trends", *Procedia Computer Science*, **73** (2015), 366–375.

[8] Jurafsky D., Martin J.H., *Speech and Language Processing*, Prentice-Hall, NJ, USA, 2009.

[9] Amazon Web Services, *Serverless Computing and Applications*. https://aws.amazon.com/serverless/.

[10] Amazon Web Services, *Amazon DynamoDB*, https://aws.amazon.com/dynamodb/.

[11] Perkins J., *Python Text Processing with NLTK 2.0 Cookbook*, Packt Publishing, 2010.

[12] Korobov M., "Morphological Analyzer and Generator for Russian and Ukrainian Languages", *Analysis of Images, Social Networks and Texts*, 2015, 320–332.

Filonov D. R., Chalyy D. Ju., Murin D. M., Durnev V. G., Sokolov V. A., "Question Answering System for Applicant Support by Using Modern Messengers", *Modeling and Analysis of Information Systems*, **25**:4 (2018), 411–420.

DOI: 10.18255/1818-1015-2018-4-411-420

Abstract. There is an increasing interest to the instant messaging applications, messengers. These applications allow us to interact with other users and include a functionality that can help us to implement bots that automate various business processes or provide information services. In this paper, we consider a specialized question answering system that uses today's messaging services infrastructure to support university applicants. We gathered a corpus of applicants questions throughout two years and developed an information retrieval model that helps us to find similar questions in the corpus. Applicants can type their questions using a natural language without any formal requirements to phrase construction or using special templates. If the system is unable to find a relevant answer, the user can directly address the question to representatives of the university. The system was implemented with the use of modern cloud services that are provided by Amazon. We used serverless computations and NoSQL data bases, so we had to develop an architecture of the system in that way. Since the system contains sensitive personal data and provide personalized service, we must focus our attention on security. We proposed the means that must improve the safety of the system, more specifically, authentification process that can be used without the explicit use of personal data, however, this is a future work. At present we test our system and evaluate its quality of information retrieval.

Keywords: question answering system, messenger, information retrieval, cloud systems and services, security

On the authors:
Dmitry R. Filonov, orcid.org/0000-0002-6402-5114, Master,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: frbdima@gmail.com

Dmitry Ju. Chalyy, orcid.org/0002-0007-1896-0876, PhD,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: chaly@uniyar.ac.ru

Dmitry M. Murin, orcid.org/0000-0002-8068-0784, PhD,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: nirum87@mail.ru

Valery G. Durnev, Prof.,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: durnev@uniyar.ac.ru

Valery A. Sokolov, orcid.org/0000-0003-1427-4937, Prof.,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: sokolov@uniyar.ac.ru

Acknowledgments:
[1] This work was supported by initiative project of P.G. Demidov Yaroslavl State University «Modeling and Analysis of Information Systems» No AAAA-A16-116070610022-6.