

УДК 004.8

ПРИМЕНЕНИЕ МОДЕЛЕЙ МЫШЛЕНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМАХ

А.С. Тощев

*Высшая школа информационных технологий и информационных систем
Казанского (Приволжского) федерального университета
atoschev@kpfu.ru*

Аннотация

Описана эволюция моделей мышления в рамках решения задачи построения интеллектуальной вопросно-ответной системы для автоматизации обработки запросов пользователей на естественном языке, начиная от простой модели на основе деревьев решений и заканчивая полноценной моделью мышления, основанной на модели мышления человека Марвина Мински. Каждая модель разработана и протестирована. Приведены результаты экспериментов и сделаны выводы о состоятельности каждой из моделей.

Ключевые слова: искусственный интеллект, машинное обучение, системный анализ, машинное мышление, обработка естественного языка, деревья решений

ВВЕДЕНИЕ

В настоящее время широко востребованным становится применение технологий искусственного интеллекта для автоматизации работы людей-специалистов, под которыми в настоящей статье подразумеваются агенты технической поддержки. За время развития области информационных технологий сконструировано множество различных информационных систем, которые необходимо поддерживать, так как ими продолжают пользоваться. К такой категории систем можно отнести работу электронных помощников [1]. Если рассматривать различные заявки, поступающие от пользователей, то среди них есть множество простых в исполнении [2], например, «Установите мне приложение», «Как найти такую-то статью на вашем сайте?», «Не работает раздел X на сайте». Целью исследования,

результаты которого представлены ниже, было создание системы, использующей принципы искусственного интеллекта и семантического анализа входящей информации [3] (с. 230-260). Важными критериями таких систем являются способности обучения и использования простейшей логики: мышление по аналогии; экстраполяция результата. Работы над системой начались в 2009 году, при этом были исследованы и использованы следующие модели: IDP, Menta 0.1, Menta 0.2-0.3, TU 1.0.

МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА

Модель IDP. Одной из первых задач создаваемой системы была обработка документов для дальнейшего интеллектуального поиска по ним (IDP –Intellectual Document Processing). В систему загружались документы, после чего система их разбирала. Ответ на поставленный вопрос формировался при помощи деревьев принятия решений [4]. Система не поддерживала обучения на естественном языке, а также требовала значительного количества документов для функционирования.

Модель Menta 0.1. Следующим шагом было построение интеллектуальной системы. Для этого были использованы следующие компоненты: обработка запросов на естественном языке; поиск решения; применение решения.

На этом этапе система была ориентирована на выполнение простых команд, например, «Добавить поле к форме». Основная функция модели представлена следующим потоком действий:

- получение и формализация запроса;
- генерация Action (специального объекта, который содержит информацию для следующего шага);
- изменение модели приложения в формате OWL [5] согласно Action;
- генерация и компиляция приложения.

Основными проблемами были: отсутствие устойчивости к ошибкам входной информации (грамматическим и содержательным), например, входной файл не имел отношения к программной системе, модель которой была представлена в базе знаний в формате OWL; система поиска решения работала только в рамках модели одной программы и только на прямых командах; отсутствовала функция обучения.

Модель Menta 0.2-0.3. Следующим шагом стало использование генетических алгоритмов в сочетании с модулями логики. В результате были сформированы основные компоненты будущей итоговой модели: критерии приемки (Acceptance Criteria); формат данных OWL; использование логических вычислений. Система содержала в себе модель приложения. При помощи генетического алгоритма модель строила из отдельных частей новую систему и при помощи логического движка [6] проверяла ее соответствие входным критериям приемки. Основным недостатком такого подхода оказалось отсутствие обучения и обработки естественного языка. Кроме того, апробация показала, что критерии приемки практически описывают необходимое решение, что являлось недопустимым. Подробнее данный подход описан в [7].

Модель TU – это модель, построенная с применением модели мышления Марвина Мински [8]. Она реализовала основные концепции предыдущих моделей и показала свою состоятельность на контрольных примерах. Характерные особенности системы: Acceptance Criteria, обучение, поиск и применение решения, отсутствие обработки естественного языка. Данная модель является более абстрактной и представляет собой верхнеуровневую архитектуру обработки запроса (мышления), ее компонентами стали лучшие части предыдущих систем.

Краеугольным постулатом модели Мински является конструкция-триплет: критик, селектор, путь мышления. Критик – это вероятностный предикат. В упрощенном виде критик – это датчик-переключатель, который срабатывает при определенных условиях. После активации определенным событием критик с помощью своего предиката проверяет, реагирует ли система на это событие или нет. После срабатывания критик ищет сопоставленный ему селектор и возвращает его. В реализованной системе критиками являются: «Критик входящих запросов», который срабатывает при появлении запроса в системе; «Критик обработки естественного языка», который срабатывает при построении семантической сети входящего запроса.

Селектор является компонентом работы с данными и производит подготовку запроса для выборки информации. Селектор может вернуть запрос для активации в системе других компонентов: либо другого критика, либо другого селектора. Селектор подготавливает атрибуты запроса. В реализованной системе

объектами типа селектор является, например, объект «Классификация проблем с прямыми инструкциями», который активируется «Критиком входящих запросов».

Путь мышления – это компонент, который производит работу с данными и решает заявленную проблему. Примерами путей мышления в области решения проблем информационных технологий могут служить: «приобретённое знание» – система знает, как решить проблему, используя уже накопленные знания; «адаптация» – применение существующего «похожего» решения; «реформуляция» – перевод проблемы после «адаптации» в «приобретённое знание»; «зов к помощи» – путь, при помощи которого система обращается к специалисту-человеку за помощью (во время активации этого пути система переходит в режим обучения).

Вторым важным постулатом теории Мински является концепция уровней мышления, каждый из которых определяет более комплексное поведение и включает в себя сложность предыдущего. Эти уровни таковы: инстинктивный; уровень приобретенных знаний; уровень мышления; рефлексивный уровень; саморефлексивный уровень; самосознательный уровень. Важно отметить, что в своей работе Мински дает лишь рамочное описание этих уровней с точки зрения человека. Нами эта концепция была расширена и перенесена на компьютерную систему.

Первый (самый низкий) уровень включает врожденные инстинкты, высший уровень – идеалы и персональные цели. На первом уровне система опирается на входжение ключевых фраз в текстовое сообщение и, применяя регулярные выражения, пытается «инстинктивно» понять проблему, например, в случае шаблонных запросов, создаваемых сторонними пользователями. Если система не смогла найти решение на первом уровне, она переходит на второй уровень; здесь происходит построение семантической сети входящего запроса, активируется «критик классификации проблем».

Третий уровень контролирует, не найдено ли решение текущей проблемы, ставит системе новые цели. Базовая цель – «помочь пользователю». Начиная с нее, система анализирует подцели: понять запрос, понять проблему, найти решение.

Четвертый уровень контролирует время выполнения входящего запроса и, если это время превышает определённую планку, производит перераспределение ресурсов.

На пятом уровне происходит инициализация контекста запросов, происходят коммуникации с пользователем.

Шестой уровень контролирует общее состояние системы, ресурсов, проблемы функционирования аппаратного комплекса и выставляет общий статус системы. Если все запросы укладываются в отведенное время, то выставляется положительный статус, иначе выставляется отрицательный статус. По общему статусу можно определить, необходимо ли внешнее вмешательство в работу системы: замена компонентов, увеличение ресурсов и пр.

Важно отметить, что каждый последующий уровень контролирует предыдущие. Под контролем понимается доступ к информационным параметрам управления предыдущими уровнями.

Для обмена информацией между уровнями и запросами была разработана концепция краткосрочной и долгосрочной памяти. Пути мышления работают с краткосрочной памятью и модифицируют данные в ней. После успешной обработки запроса память переписывается в долгосрочную и сохраняется уже в общей базе знаний. Таким образом, в базу знаний попадает только апробированная информация, исключая ошибки нахождения неверного решения.

Модель данных системы является семантической сетью, построенной над нереляционной базой данных [9], и описывает «знания» системы: решения, проблемы, пути мышления, критики, селекторы и т. д. Таким образом критики, селекторы, пути мышления могут быть «приобретенными знаниями».

Необходимо отметить возможность обучения системы. Как и концепция долгосрочной и краткосрочной памяти, обучение системы является нашим расширением модели мышления Мински. На начальном этапе система содержит базовые концепции: объект, действие. С помощью обучения в систему можно ввести новые концепции. Обучение также проводится на естественном языке. Например, «веб-браузер – это объект. Firefox – это веб-браузер». Теперь система знает две новых концепции – веб-браузер и Firefox, а также то, что Firefox – это веб-браузер.

Модель Мински является важным постулатом системы, но лишь небольшой ее частью – нами разработаны архитектура, а также важные теоретические и концептуальные системы. В рамках проведенного исследования модель была дополнена и расширена специальными уровнями мышления, критиками, селекторами. Практическое применение системы весьма обширно и может служить как для обработки запросов поддержки пользователей, так и электронным помощником – при обучении – и вспомогательным компонентом системы – для взаимодействия с пользователями.

ЗАКЛЮЧЕНИЕ

По результатам анализа функциональности всех построенных интеллектуальных вопросно-ответных систем была выбрана та из них, которая базируется на модели мышления Марвина Мински: она дала наиболее оптимальное сочетание возможности прикладной реализации и учета универсальной гибкости человеческого мышления. Иными словами, модель не копирует процесс человеческого мышления, а дает возможную его интерпретацию, тем самым позволяя реализовать ее в условиях ограниченных вычислительных ресурсов.

Благодарности

Работа выполнена при финансовой поддержке программы НИЛ OpenLab Высшей школы информационных технологий и информационных систем Казанского (Приволжского) федерального университета.

СПИСОК ЛИТЕРАТУРЫ

1. *Wikipedia*. Virtual assistant. URL: https://en.wikipedia.org/wiki/Intelligent_personal_assistant.
2. *Тощев А.С., Таланов М.О.* Результаты обработки инцидентов. URL: <http://tu-project.com/for-business/>.
3. *Russel S., Norvig P.* Artificial intelligence. An modern approach. Willams, 2007. 1408 p.
4. *Deng H., Runger G., Tuv E.* Bias of importance measures for multi-valued attributes and solutions // Proceedings ICANN'11 Proceedings of the 21st international conference on Artificial neural networks. 2011. Part II. P. 293-300.

5. *Lesly L.* OWL: Representing information using the Web Ontology Language. — 47403, Blumington, Liberty drive 1663: Trafford publishing, 2005. 302 p.

6. *Wang P.* Non-axiomatic logic a model of intelligent reasoning. USA: World Scientific Publishing Company, 2013. 276 p.

7. *Maxim Talanov, Aidar Makhmutov, Andrei Krekhov.* Automating programming via concept mining, probabilistic reasoning over semantic knowledge base of SE domain // 2010 6th Central and Eastern European Software Engineering Conference (CEE-SECR) Russian Federation, Moscow, 13–15 October 2010. P. 30-35.

8. *Minsky M.* The emotion machine. Simon & Shuster Paperbacks, 2007. 400 p.

9. *Wikipedia.* NoSQL databases. URL: <https://en.wikipedia.org/wiki/NoSQL>.

APPLICATION OF THINKING MODEL IN INTELLECTUAL QUESTION-ANSWER SYSTEMS

A.S. Toshev

High School of Information Technology and Information Systems of Kazan Federal University

atoshev@kpfu.ru

Abstract

We described an evolution of thinking model in application with building intellectual question-answer system for automation processing user requests in natural language, starting with simple decision trees and finished with human thinking model. Every model has been developed, prototyped and tested. Experimental data and conclusions for every model provided.

Keywords: *artificial intelligence, machine learning, system analysis, machine thinking, natural language processing, decision trees*

REFERENCES

1. *Wikipedia*. Virtual assistant. URL: https://en.wikipedia.org/wiki/Intelligent_personal_assistant.
2. *Toshev A., Talanov M.* Incident request processing analysis. URL: <http://tu-project.com/for-business/>.
3. *Russel S., Norvig P.* Artificial intelligence. An modern approach. Willams, 2007. 1408 p.
4. *Deng H., Runger G., Tuv E.* Bias of importance measures for multi-valued attributes and solutions // Proceedings ICANN'11 Proceedings of the 21st international conference on Artificial neural networks. 2011. Part II. P. 293-300.
5. *Lesly L.* OWL: Representing information using the Web Ontology Language. — 47403, Blumington, Liberty drive 1663: Trafford publishing, 2005. 302 p.
6. *Wang P.* Non-axiomatic logic a model of intelligent reasoning. USA: World Scientific Publishing Company, 2013. 276 p.
7. *Maxim Talanov, Aidar Makhmutov, Andrei Krekhov.* Automating programming via concept mining, probabilistic reasoning over semantic knowledge base of SE domain // 2010 6th Central and Eastern European Software Engineering Conference (CEE-SECR) Russian Federation, Moscow, 13–15 October 2010. P. 30-35.
8. *Minsky M.* The emotion machine. Simon & Shuster Paperbacks, 2007. 400 p.
9. *Wikipedia*. NoSQL databases. URL: <https://en.wikipedia.org/wiki/NoSQL>.

СВЕДЕНИЯ ОБ АВТОРЕ



ТОЩЕВ Александр Сергеевич – аспирант Института математики и механики им. Н.И. Лобачевского Казанского (Приволжского) федерального университета (КФУ), младший научный сотрудник Лаборатории машинного понимания Высшей школы информационных технологий и информационных систем КФУ.

Alexander Sergeevich TOSCHEV, received MS degree in mathematics and economics from Kazan Federal University (2011). Currently is a graduate student at the N.I. Lobachevskii Institute of Mathematics and Mechanics of Kazan Federal University. Current scientific interests: data mining, artificial intelligence, machine learning.

email: atoshev@kpfu.ru

Материал поступил в редакцию 15 мая 2015 года