

Лик Э.В.  
 Университет «Туран»  
 Алматы, Казахстан  
 Научный руководитель: Ким Е.Р.

## ЭКСПЕРТНЫЕ СИСТЕМЫ КАК ОБЛАСТЬ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

**Аннотация.** В данной статье рассмотрена экспертная система и приведена ее упрощенная архитектура. Функционирование экспертной системы проанализировано на примере библиотеки «*experta*» на языке программирования Python, основной задачей которой является выявление диабета у детей.

**Ключевые слова:** *expert, python, Knowledge Engine.*

В задачи улучшения различных сфер жизнедеятельности человека входит планомерное вытеснение человеческого фактора для более точных оценочных действий. Одни из современных систем, которые способны переработать специфику деятельности компетентных лиц, являются экспертные системы [1].

Экспертная система – интеллектуальная система, которая может имитировать способность человека-эксперта принимать решения. Подобные системы направлены на решения сложных задач путем аргументации. Одной из основных характеристик экспертной системы является способность анализа и выдачи заключения рассматриваемой проблемы. Также стоит отметить, что современные экспертные системы не способны учиться на опыте. Эта широко распространенная критика, как правило, в значительной степени верна. За исключением простых систем классификации, не использующих учебный компонент для построения частей своих баз знаний из библиотеки ранее решенных задач. Способности к обучению необходимы для интеллектуальных систем, которые могут оставаться полезными перед лицом изменения окружающей среды или изменения стандартов экспертизы [1-3].

В основе функционирования экспертных систем лежит принцип взаимодействия эксперта с базой знаний. Оптимальное архитектурное построение помогает избежать излишних операций при расчете оценки. Для взаимодействия пользователя с системой существует модуль пользовательского интерфейса, который через механизм логического вывода использует базу знаний. На рисунке 1 демонстрируется упрощенная архитектура экспертной системы [4-6].



Рисунок 1 – Архитектура экспертной системы

В зависимости от реализуемой задачи, в модуль «механизма логического вывода» интегрируют модель представления данных и база данных. Модель представления данных – это структура хранимых знаний эксперта и способа взаимодействия с ними. А база данных служит для сбора совокупности идей с дальнейшим добавлением их в базу знаний. Тем самым, есть потенциальная возможность модернизации интеллектуальной системы для последующего обучения в процессе эксплуатации.

В качестве исследования функционирования экспертной системы будет использована одна из реализаций библиотеки «*experta*» на языке программирования Python. Ее основная задача – выявление диабета у детей. Основными анализируемыми признаками служат уровень сахара в крови, гликемический индекс и наличие диабета у родителей. Выбор программной среды для разработки экспертной системы производился с учетом критерия простоты реализации и схожести с системой «*CLIPS*». Схожесть данной библиотеки предполагает близкое представление логических процессов к человеческому мышлению.

Для формирования логического вывода, экспертная система руководствуется специализированной информацией, называемой фактом. Синтаксической интерпретацией факта в рамках библиотеки «*experta*» служит класс «*Fact*». Данный класс, как и все другие используемые классы, импортируется из библиотеки. Первоначальным действием после импорта, следует инициализация пустого класса «*Human*», наследующего класс «*Fact*». «*Human*» - класс, содержащий все необходимые факты для вывода заключения. После всех вышеописанных действий, осталось записать знания о предметной области, которые называются правилами. Их запись в библиотеке осуществляется с использованием оберточных функций – декораторов «*Rule*». Все правила исследуемой области записываются в произвольно названный «основной» класс с наследуемым функционалом «*Knowledge-Engine*». Также в декораторы «*Rule*» необходимо вписать факты предметной области и функцию вызова. Именно в функции вызова записывается наше заключение. Такая структура является минимальной для организации набора правил. После окончания реализации «основного» класса, осталось создать его экземпляр в переменную с любым именем. И внести в метод «*declare*» экземпляр «*Human*» с перечислением в параметре необходимых признаков, таких как: возраст, уровень глюкозы в крови, наличие тремора, наличие голода, наличие пота, наличие родственников с диабетом, наличие слабости, наличие деуринации, наличие жажды, наличие тумана в глазах, наличие сухости во рту, наличие галитоза и наличие аритмичного дыхания. После чего, необходимо вызвать у созданного экземпляра метод «*run*».

В итоге, в зависимости от внесенных признаков, пользователь увидит рекомендацию от экспертной системы. Например, при вводе неполной спектра параметров, а именно «возраст = 3» и «уровень глюкозы в крови = 8», система выдает следующую рекомендацию: «Осторожно! Высокая вероятность наличия диабета». Наличие и специфика ответов зависит от выраженности и количества признаков.

Проводя сравнение данной системы со среднестатистическими возможностями человека, были выявлены некоторые характеристики сторон двух категорий. В качестве тестируемых лиц-экспертов случайным образом были отобраны 5 студентов и 5 профессоров медицинского института. Каждому из ряда тестируемых лиц, были предоставлены признаки, по которым необходимо было сформировать рекомендацию, относительно наличия или отсутствия диабета у ребенка. Ниже, в таблице 1 изображены обобщенные сравнительные характеристики тестируемого лица и экспертной системы.

Главная причина, по которой результаты экспертной системы будут лучше, чем подходы тестируемых, заключается в способности системы точно следовать ранее заложенным инструкциям эксперта. Причем, такая система не подвержена усталости. А скорость ее лишь зависит от параметров вычислительного устройства, на котором она функционирует.

Цель данной системы, основанная на знаниях – сделать критическую информацию, необходимую для работы системы, явной. В традиционной компьютерной программе логика встроена в код, который, как правило, может проверять только специалист по компьютерным технологиям. При использовании экспертной системы важно было определить правила в формате, которые были бы интуитивно понятными специалисту в области рассматриваемой проблемы. Преимуществами этого явного представления знаний были быстрое развитие и простота обслуживания.

**Таблица 1 – Сравнительная характеристика тестируемого эксперта и экспертной системы**

| Характеристика                    | Тестируемый эксперт | Экспертная система |
|-----------------------------------|---------------------|--------------------|
| <b>Скорость</b>                   | Изменчивая/Низкая   | Постоянная/Высокая |
| <b>Производительность</b>         | Изменчивая/Низкая   | Постоянная/Высокая |
| <b>Расположение</b>               | Определенное        | Везде              |
| <b>Необходимость перепроверки</b> | Есть                | Нет                |
| <b>Вероятность ошибки</b>         | Средняя             | Низкая             |

Простота обслуживания является наиболее очевидным преимуществом. Это было достигнуто двумя способами. Во-первых, устраняя необходимость в написании обычного кода, можно избежать многих обычных проблем, которые могут быть вызваны даже небольшими изменениями в системе. Это также послужило причиной второго преимущества: быстрого прототипирования. С помощью экспертной системной оболочки можно было ввести несколько правил и создать прототип за несколько дней, а не месяцев или лет, обычно связанных со сложными проектами сферы информационных технологий [4, 6].

Проведенное исследование явным образом показало преимущественные стороны экспертной системы в сравнении с экспертом определенной области. Обобщая результаты, можно предположить наличие схожих выводов в исследовании с другой предметной областью.

**ЛИТЕРАТУРА**

1. Feathers M. Working Effectively with Legacy Code / Feathers M. – Upper Saddle River: Prentice Hall, 2004. – 464.
2. Cornelius T. Expert Systems: The Technology of Knowledge Management and Decision Making for the 21st Century / Cornelius T. – Academic Press, 2001. – 1947.
3. Schreiber G. Knowledge Engineering and Management: The Common KADS Methodology / Schreiber G., Akkermans H., Anjewierden A., De Hoog R., Shadbolt N. – The MIT Press, 1999. – 471.
4. Jackson P. Introduction to Expert Systems (International Computer Science Series) / Jackson P. – Addison-Wesley Pub, 1990. – 526.
5. Domingos P. The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World / Domingos P. - Basic Books, 2018. – 352.
6. Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin. – Upper Saddle River: Prentice Hall, 2017. – 430.

**Лик Э.В.**

**Ғылыми жетекші: Ким Е.Р.**

**Сараптамалық жүйелер зияткерлік ақпараттық жүйелер өрісі ретінде**

**Аңдатпа.** Бұл мақалада сараптама жүйесі және оның жеңілдетілген архитектурасы қарастырылады. Сараптамалық жүйенің жұмысына Python бағдарламалау тіліндегі сараптама

кітапханасының мысалында талдау жасалады, оның негізгі міндеті балалардағы қант диабетін анықтау болып табылады.

**Кілт сөздер:** expert, python, Knowledge Engine.

**Lik E.V.**

**Scientific supervisor: Kim E.R.**

**Expert systems as the field of intelligent information systems**

**Abstract.** In this article the expert system is reviewed and its simplified architecture is given. The functioning of the expert system is analyzed with the example of «experta» library in thePython programming language, the main task of which is to detect diabetes in children.

**Keywords:** expert, python, Knowledge Engine

**Сведения об авторах:**

**Лик Эдуард Валерьевич**, студент второго курса университета «Туран».

**Ким Екатерина Р.**, к.т.н., доцент кафедры ИТ университета «Туран».

УДК 530.1, 681.3.06

**Zimanova D.A, Sadir A.K, Kassymov B.M**

Kazakh -British technical university

Almaty, Kazakhstan

Scientific supervisor: Shamo P.S.

**SENTIMENT ANALYSIS OF SOCIAL NETWORKING SERVICES  
DATA USING FUZZY SETS AND LOGIC**

**Abstract.** This paper is devoted to usage of fuzzy logic in sentiment analysis of social network comments. The main feature of our work is that we aim to provide not only novel algorithm for sentiment analysis, but also to approve the correctness of our method, we implemented a prototype application on iOS platform. Our target audience is bloggers, business owners, SMM managers, artists and even government workers.

In today's world, huge value of information is created every day. However, all these data is unstructured, which means that, despite that Internet is a transparent platform for reading data, it is impossible to make sense out of big data. Using our application user will be able to access generalized public mood towards some topics from social network posts.

**Key words:** Sentiment analysis, fuzzy logic, fuzzy sets, NLTK, text processing, Emoji Sentiment Ranking, emoji processing, fuzzification

Nowadays text processing technologies have attracted more attention than ever before and social networks have a vast influence on human life. To this end, Sentiment Analysis is a rapidly developing field in Artificial Intelligence (AI), consisting of widely differing tools and evaluation methods which target fetching emotional tones and subjective point of view of a speaker or a writer in some piece of text. In other words, sentiment analysis refers to a set of technologies to measure how positive or negative the text is. In the current paper, on the one hand, we focus on introducing a novel algorithm for sentiment analysis which mainly relies on python's Natural Language Toolkit (NLTK), fuzzy mathematics and Emoji Sentiment Ranking. We propose the processing of emojis (picture-like symbols commonly used to express someone's emotions) to boost sentiment analysis of a certain text [1].