



РОЗРОБКА ЕКСПЕРТНИХ СИСТЕМ, ЗАСНОВАНИХ НА НЕЙРОННИХ МЕРЕЖАХ

УДК 004.75:004.272.26:004.032.26:519.1

КАТЕРИНИЧ Лариса Олександрівна

к.ф.-м.н., асистент кафедри інформаційних систем Київського національного університету імені Тараса Шевченка.

Наукові інтереси: штучний інтелект, системи штучного інтелекту, інтернет-технології, програмування.

e-mail: katerinich@rambler.ru

ГАЛКІН Олександр Володимирович

к.ф.-м.н., доцент кафедри інформаційних систем Київського національного університету імені Тараса Шевченка.

Наукові інтереси: методи об'єктно-орієнтованого програмування, системи штучного інтелекту, Java.

e-mail: galkin@unicyb.kiev.ua

ВЕРЕС Максим Миколайович

к.ф.-м.н., доцент кафедри інформаційних систем Київського національного університету імені Тараса Шевченка.

Наукові інтереси: управління інформацією, штучний інтелект, розподілені

обчислення, паралельні розрахунки на кластерних системах, методи об'єктно-орієнтованого програмування.

e-mail: veres@ukr.net

ВСТУП

Останнім часом найбільш розповсюдженим і живим став термін інтелектуальні системи (ІС), оскільки для розв'язання складних задач вони використовують методи штучного інтелекту, що ґрунтуються на використанні знань. Розвиток методів штучного інтелекту призвів до розробки практично діючих ІС. Структура таких систем досить складна, оскільки повинна складатися з таких обов'язкових компонент, як база знань і підсистем придбання знань, формування цілі, виведення нових знань, діалогового спілкування, обробка внутрішньої і зовнішньої інформації, навчання і самонавчання, контролю та діагностики.

ПОСТАНОВКА ЗАДАЧІ

Необхідно побудувати практично діючу інтелектуальну систему, яка використовуючи технологічні можливості середовища розробки та знання експерта предметної області продукує рекомендації

на рівні експерта. Дана система повинна швидко працювати з великою кількістю взаємозв'язаних даних.

Мета роботи полягає у розробці експертної системи Н-Гомеопат, яка побудована та функціонує на нейромережевій структурі, що у свою чергу дозволяє прискорити в цілому реакцію всієї системи на представлену інформацію. Це досягається за допомогою використання баз даних та спеціальних евристичних методів.

ОСНОВНИЙ МАТЕРІАЛ

Знання з конкретної предметної області складають основу ІС. У межах предметної області системи розв'язують задачі шляхом обробки знань. Отже, знання — це система (модель) подання предметної області у вигляді фактів і їх відношень або об'єктів та їх зв'язків. Задача з конкретної області може бути представлена конкретними сутностями і їх поданням у формі, що визначається обраною моделлю знань. Система здатна розв'язувати таку задачу, якщо вона має засоби обробки знань в обраній формі. Знання відрізняються від

звичайних даних присутністю специфічної структури і додатковими властивостями, серед яких можна виділити наступні:

- *інтерпретованість*. Знання мають змістовну інтерпретацію, а дані завжди інтерпретуються конкретною програмою;
- *відношення класифікації*. Знання дозволяють описати структуру знань «клас-підклас». Клас об'єднує однакову інформацію для всіх «підкласів» і у разі необхідності передає «підкласу»;
- *ситуаційний зв'язок*. Співпадіння окремих фактів в одній предметній області визначають ситуаційні зв'язки. Вони дозволяють будувати процедури аналізу знань. При використанні даних це важко організувати.

Знання в конкретній предметній області при використанні їх у системі об'єднуються у бази знань (БЗ), так само як дані у бази даних (БД). Знання не відкидають дані. БЗ і БД розглядаються як різні рівні представлення інформації.

Одним з різновидів інтелектуальних систем є експертні системи (ЕС). Архітектура ЕС суттєво відрізняється від усіх інших обов'язково наявністю додаткових модулів обробки БЗ, пояснень та накопичення знань.

Для функціонування системи БЗ має бути наповнена знаннями. Для цього запрошують висококваліфікованих спеціалістів у тій галузі, для якої розробляється система (вони відіграють роль експертів). Їх завданням є — описати всі відомі знання для функціонування ЕС. У БЗ мають бути наявні знання першого та другого родів. Знання першого роду — це загальновідомі факти, явища та закономірності, які є в даній предметній області. Знання другого роду — це набір емпіричних правил та інтуїтивних висновків, якими користуються спеціалісти, приймаючи рішення в умовах невизначеності за наявності неповної суперечливої інформації. Відомості про ці знання, як правило, не опубліковані. У базі знань ЕС переважно містяться знання першого роду, але мають бути й знання другого роду. Якщо ці знання відсутні, то це означає поганий вибір експертів (вони не вміють формулювати свої знання чи навпаки: не хочуть цього робити, щоб зберегти за собою статус унікальних спеціалістів).

Інженер з питань знань має такі обов'язки: знання, що їх подали експерти, він структурує і записує в базу знань з урахуванням правил побудови моделі знань ЕС, що проектується. Усі знання, які подані в базі знань поділяються на інтенціональні та екстенціональні.

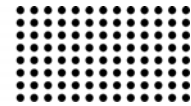
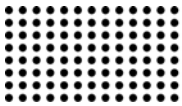
Інтенціональні, або *абстрактні*, знання являють собою понятійні (концептуальні) знання про об'єкти предметної області і зв'язки між ними. *Екстенціональні* (*конкретні*) знання — це кількісні характеристики інтенціональної частини знань, тобто база даних ЕС.

Блок рішень необхідний для пошуку та побудови логічних висновків, які видає користувачеві ЕС. Дії цього блока подібні до міркувань людини-експерта, яка оцінює проблему і пропонує її гіпотетичне вирішення. Цей блок виконує функції управління процесом пошуку розв'язків, тобто він визначає спосіб і послідовність використання різних правил та процедур. Кожна ЕС має містити певну кількість таких правил та процедур. Кількість правил, що їх містить середня ЕС, перевищує 500, а для великої ЕС може перевищувати й 1000. Здебільшого блок розв'язків складається з двох частин: блоку логічного виведення та управляючого блоку.

Блок логічного виведення виконує дії, аналогічні до інтелектуальної діяльності спеціаліста, коли той приймає рішення. Функції цього блока — побудова логічного висновку на базі існуючих знань, які зберігаються в БЗ.

Блок управління керує процесом пошуку розв'язання, тобто визначає послідовність використання різних правил і процедур маніпулювання знаннями.

Блок пояснень слугує для видачі за запитом користувача послідовності логічних висновків та міркувань, якими оперувала система у процесі пошуку рішення. Наявність такого блоку в ЕС дає змогу використовувати її не лише для прийняття рішень, а й для процесу навчання як навчальну систему. Проблема пояснень та обґрунтування правильності міркувань — важлива й складна задача. Адже попри те, що система містить знання експертів та надає поради, відповідальність за прийняте рішення несе особисто користувач. Досі немає правових актів, які б визначали відповідальність експертів за знання, надані системі, а також за рішення, які приймаються в результаті консультування користувачів з ЕС, яка містить знання даного експерта чи групи експертів.



Оцінка ЕС користувачем значною мірою залежить від того, наскільки праця з експертною системою схожа на співробітництво з експертом, і, відповідно, істотно залежить від якості пояснень, що їх надає система користувачеві на ті запитання, які викликають у нього сумнів. Усі питання, які можуть виникнути в користувача при роботі з ЕС, можна поділити на такі групи: пов'язані з процесом рішення проблеми (як і чому? з якою метою? з яким результатом? з чого це впливає?); стосовно значень термінів, які прийняті в ЕС при організації діалогу з користувачем; про причини виникнення певного запитання у процесі експертизи; стосовно наслідків, які впливають із даної користувачем відповіді на запитання, поставлене системою (наприклад, що буде, коли?). Для того щоб ЕС змогла швидко і якісно пояснювати правильність своїх відповідей, а також доцільність поставлених запитань, вона записує в робочій пам'яті хід своїх міркувань та їх послідовність.

Блок спілкування з користувачем або інтерфейс користувача необхідний для організації діалогової взаємодії між системою і користувачем. Основна вимога до цього блока — це реалізація спілкування природною мовою користувача.

Блок нагромадження знань. ЕС здебільшого будується для таких предметних областей, які характеризуються необхідністю актуалізації та розширення знань. З огляду на це ЕС містить блок, який дає змогу експерту завантажувати базу знань, а також виконувати редагування знань, які зібрані в базі. Усе більший інтерес викликає питання автоматизованого набуття знань експертною системою (актуалізація, коригування та розширення знань ЕС через процес навчання). Проблему навчання можна звести до створення нових понять та правил на базі існуючих, а також підключення їх в базу знань таким чином, аби не було суперечливості знань.

Отже, функція цього блока полягає у формуванні емпіричних залежностей із неповних знань, тобто здобуття знань першого роду на основі знань другого роду. Але через складність реалізації цих функцій не всі ЕС мають у своєму складі такий блок.

Створення нових знань повністю автоматизованим способом — це дуже проблематичний підхід, запропонований занадто захопленими прибічниками штучного

інтелекту [1]. Будь-які знання (особливо нові) потребують дуже ретельної перевірки, яку можуть виконати лише експерти.

Основані на знаннях (інтелектуальні) або експертні комп'ютерні системи мають значні переваги у порівнянні з людиною (що не є експертом) в плані продуктивності праці при розв'язанні різних проблем. Це досягається за допомогою використання баз знань та спеціальних евристичних методів.

При проектуванні системи Н-Гомеопат висувались вимоги до:

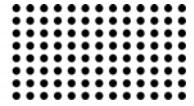
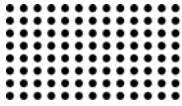
- архітектури;
- типів користувачів;
- моделі даних;
- функцій користувача (use-cases);
- інтерфейсу користувача (опис екранів та форм);
- опису як все це інтегрується з іншим програмним та апаратним забезпеченням (включаючи точну версію ОС, БД та веб-серверів, по потребі)
- безпеки та конфіденційності;
- ліцензійної політики, де документується як продукт буде розповсюджуватись замовником і, відповідно, які сторонні компоненти сумісні по ліцензійним міркуванням;
- інші вимоги, за потребою.

Виходячи з огляду існуючих програмних продуктів можна зазначити, що при розробці системи слід враховувати наступну інформацію:

- опитування пацієнта і оформлення історії захворювання;
- пошук по симптоматичному та нозологічному принципу;
- сортування та аналіз знайденої інформації;
- визначення захворювання і формування гомеопатичного рецепту;
- бази даних найбільш часто вживаних запитів;
- банк даних рецептурних прописів;
- щоденник лікаря.

Поряд с цим програмний продукт має бути легко налаштовуватися під потреби користувача:

- вільна зміна розміщення основних та додаткових вікон;
- зміна відображуваної інформації безпосередньо під час роботи програми



– програма не повинна бути занадто велика за обсягами

– легко інсталюватись та бути легко поширюваною.

Також для оптимальної роботи і зручності лікаря-гомеопата база даних повинна мати наступні функціональні можливості, щодо перегляду та редагування інформації:

– формування рубрик та підрубрик;

– встановлення перехресних посилань;

– визначати синоніми;

– додавати та вилучати ліки та посилання на авторів;

– змінювати бал вказаних ліків.

Нейронні мережі (НМ) мають широке коло застосувань, серед яких, як зазначалося, значне місце посідають експертні системи. Експертна система – це програмний комплекс, яка використовуючи технологічні можливості середовища розробки та знання експерта предметної області продукує рекомендації на рівні експерта.

Важливою рисою, що відрізняє ЕС від інших систем є те, що знання, яким оперує ЕС, вона отримує в процесі спілкування з експертом [2]. ЕС, які використовують НМ для отримання знань, проходять ще один додатковий етап, що певною мірою покращує їх рівень знань. Вони проходять стадію навчання з використанням певних алгоритмів і відповідно самостійно підвищують свій рівень знань. Оскільки принцип побудови та організація роботи НМ були запозичені з людських аналогів, то навчання мережі так як і людини не може завжди бути стовідсотковим. Але відкидаючи певні обмеження і суперечності, щодо вимог до НМ [3], можна використовуючи ці дві складові навчання – експерта і самонавчання мережі, суттєво підвищити ефективність роботи ЕС. Така система отримала назву нейронна експертна система (НЕС).

Концептуальні вимоги до системи Н-Гомеопат. У експертній системі Н-Гомеопат потрібно за представленими симптоми захворювання пацієнта встановити діагноз і призначити лікування.

Впровадження НМ, як засіб для розв'язання цієї задачі вимагає не лише технічної реалізації, а й оптимально побудованого робочого місця лікаря-гомеопата. Оскільки лікар не є фахівцем у сфері засобів розробки відповідних систем, програмні продукти повинні мати інтуїтивно зрозумілий інтерфейс.

Виходячи з цього та огляду відомих програмних продуктів, можна зазначити, що при розробці системи в першу чергу слід враховувати наступне:

– опитування пацієнта і оформленні історії захворювання;

– пошук по симптоматичному та нозологічному принципу;

– сортування та аналіз знайденої інформації;

– визначення захворювання і формування гомеопатичного рецепту;

– БД найбільш часто вживаних запитів;

– банк даних рецептурних прописів;

– щоденник лікаря.

Поряд с цим програмний продукт має бути легко налаштовуватися під потреби користувача, а відтак забезпечувати:

– вільну зміну розміщення основних та додаткових вікон;

– зміну відображуваної інформації безпосередньо під час роботи програми;

– оптимальні розміри програмного коду;

– легку інсталяцію та легку поширюваність.

Також для оптимальної роботи і зручності лікаря-гомеопата база даних повинна мати наступні функціональні можливості щодо перегляду та редагування інформації:

– формування рубрик та підрубрик;

– встановлення перехресних посилань;

– визначення синонімів;

– актуалізація основних понять (ліки, посилання, характеристичні числа тощо).

Навчання системи відбувається за обраним алгоритмом навчання НМ, зокрема алгоритмом зворотного поширення помилки [4,5]. Первинні данні (симптоми та/або гомеопатичні препарати) для навчання спеціалізованих НМ готують і вводять до системи Н-Гомеопат експерти відповідної предметної області. НМ повинна бути навчена навчаючих вибірках (формує експерт), а після навчання (тобто у робочому режимі роботи) до системи вхідні данні (симптоми) вводяться користувачем системи.

Всі дії з обробки даних, які будуть передані до спеціалізованих НМ виконує комутаційний елемент. Комутаційний елемент реалізований за допомогою НМ [6-8]. Завданням такого елементу у загальному випад-

ку є класифікація вхідної інформації з подальшим розподілом її по мережах, які оптимально розв'язують задачі конкретної предметної області. Вхідною інформацією на першому етапі будуть коректні симптоми захворювань.

Експертна система Н-Гомеопат побудована та функціонує на нейромережевій структурі, що у свою чергу дозволяє прискорити в цілому реакцію всієї системи на представлену інформацію. Комутаційний елемент через свої функціональні властивості дозволяє класифікувати представлену на вхід інформацію і на виході надавати інформацію з тією структурою даних, яка є вхідною для НМ, що розв'язує задачі певного класу. Також слід зазначити, що однією з ключових функцій комутаційного елемента є можливість обробки вхідної інформації з подальшим вибором алгоритму і архітектури мережі.

ОСНОВНІ ЕТАПИ РОЗРОБКИ ЕС Н-ГОМЕОПАТ

Відомо, що створення різного роду діагностичних систем пов'язане з використання системи управління базами даних (СУБД). Це обумовлено тим, що процес діагностики передбачає обробку великої кількості інформації, яка з метою забезпечення ефективності використання повинна мати чітку структуру. Процес побудови діагностичної системи ЕС Н-Гомеопат складається з наступних етапів:

1. Постановка цілей і задач.
2. Створення БД, яка містить всю необхідну для діагностики інформацію.
3. Розробка стратегії діагностики.
4. Створення програмного забезпечення, використовуючи сучасні СУБД.

Для вирішення поставлених задач було детально вивчено і проаналізовано предметну область.

Базові множини, які визначають процес діагностики в системі є множини СИМПТОМИ і ПРЕПАРАТИ. Множина СИМПТОМИ – це опис відхилень від нормального стану організму. Характеристиками елементів цієї множини можна вважати множину, яка складається з областей прояву недугу (наприклад, органи травлення, зору, слуху, серце, судини, кров та ін.), а також множину ключових слів, які лаконічно характеризують симптоми.

Множина ПРЕПАРАТИ містить лікувальні засоби. Її характеристиками можна вважати множини латинських, російських, народних назв, а також спосіб приготування, подібні засоби лікування, характерні ознаки та ін. Крім того, кожний препарат пов'язаний з конкретним набором симптомів, а кожен симптом пов'язаний з не менше ніж одним препаратом.

Для проектування ЕС Н-Гомеопат використовувалась СУБД Microsoft SQL Server 2008 та інтегроване середовище розробки програмного забезпечення Microsoft Visual Studio 2008, а саме компонент Visual C#.

ЕС Н-Гомеопат містить наступні БД:

- base_NeuronNetwork - використовується для зберігання вхідних та вихідних векторів;
- Gomeopat - використовується для зберігання інформації про пацієнтів, симптомів, препаратів, діагнозів;
- NeuronNetwork - використовується для зберігання проміжних результатів навчання НМ.

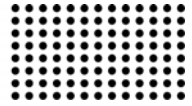
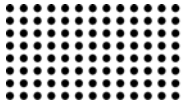
АЛГОРИТМ ФУНКЦІОНУВАННЯ ЕС Н-ГОМЕОПАТ

Розробка стратегії полягає в аналізі зв'язків між базовими множинами та їх характеристиками з метою визначення відповідних відношень, характер яких впливає на процес діагностики.

В основу стратегії діагностики ЕС Н-Гомеопат покладено метод покрокових уточнень, суть якого полягає в тому, що кінцева мета - призначення препаратів, досягається шляхом послідовного уточнення простору пошуку. Цей метод виправдовує своє застосування при роботі з великою кількістю взаємозв'язаних даних.

Для визначення необхідного препарату здійснюється детальний аналіз зв'язків між елементами базових множин СИМПТОМИ і ПРЕПАРАТИ та встановлюється логічний ланцюжок пошуку потрібного препарату за наявністю набору симптомів.

Відповідно до даного методу, спочатку визначається область прояву нездужання. Потім із запропонованого переліку ключових слів обирається слово, яке найкраще визначає відхилення від нормального стану організму. З цим словом пов'язана певна кількість симптомів, повний опис яких можна отримати, обравши ключове слово. В свою чергу кожний симптом пов'язаний з одним або декількома препаратами, а, отже, і з іншими симптомами. В такий спосіб будується



логічний ланцюжок, який має вигляд: область → слово → симптоми → препарати → симптоми → препарат.

Цей ланцюжок будується за рахунок встановлених зв'язків між множинами СИМПТОМИ і ПРЕПАРАТИ.

Симптоми є характеристиками препаратів, тому в залежності від наявності симптомів, пов'язаних з препаратом, можна або призначити препарат, або ні. Так як обраний симптом може бути характеристикою більше ніж одного препарату, то наявність деякого симпто-

му у пацієнта пов'язана з можливим існуванням інших (прихованих) симптомів, пов'язаних з попередньо обраним за рахунок препаратів. Отже, область пошуку необхідного препарату дещо розширюється. Це дозволяє визначити додаткові симптоми, які в свою чергу детально аналізуються. Так продовжується до тих пір, поки не будуть враховані всі симптоми, пов'язані з обраними.

ЛІТЕРАТУРА:

1. Андон Ф.И., Балабанова А.С. Выявление знаний и изыскания в базах данных: подходы, модели, методы и системы //Проблемы программирования. – 2000. – №1-2. – С.513-526.
2. Элти Дж., Кумбс М. Экспертные системы: концепции и примеры. – М.: Финансы м статистика, 1987. – 191 с.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М: Горячая линия – Телеком, 2006. – 452 с.
4. Круглов В. В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети: Учеб. пособие. – М.: Издательство физико-математической литературы, 2001. – 224 с.
5. Уинстон П. Искусственный интеллект. – М.: Мир, 1980. – 519 с.
6. Bargiela, Andrzej and Pedrycz, Witold. Granular Computing: An introduction. – Kluwer Academic Publishers, 2003. – 5 p.
7. Hall L.O. The choice of ply operator in fuzzy intelligent systems. Fuzzy Sets and Systems. – 1990. –V.34. – P.135-144.
8. Jacek Leski. Systemy neuronowo-rozmyte. – Warszawa: Wydawnictwo Naukowo-Techniczne, 2008. – 686 s.