

– класифікація за методом Роккі (Rocchio classification), яка заснована на векторній моделі і, в якій використовуються відгуки про релевантність вхідних елементів з метою підвищення точності рекомендаційної системи.

У цілому, можна відмітити про існування множини алгоритмів навчання для рекомендаційних систем, які застосовуються для машинного навчання, а також, які в будь-якому разі, потребують подальшого вдосконалення для отримання ще більшої точності надання рекомендацій.

Список використаних джерел:

1. Melville P., Mooney R., Nagarajan R. Content-Boosted Collaborative Filtering for Improved Recommendations. *National Conference on Artificial Intelligence : «AAAI-2002»* : materials (20-25 July 2016, Edmonton, Canada). Edmonton, Canada : AAAI, 2002. P. 187-192.
2. Linden G., Smith B., and York J. Item-to-Item Collaborative Filtering. Los Alamitos, CA, USA: IEEE Internet Computing, 2003. P. 76-80.
2. Офіційний блог для розробників компанії IBM. URL: <https://www.ibm.com/developerworks/ru/library/os-recommender1/> (дата звернення: 25.04.2019).
3. Мазурік О.Ю. Порівняльний аналіз моделей оцінювання в рекомендаційних системах. *Системний аналіз та інформаційні технології*. Київ : НТУУ «КПІ», 2016. С. 113.

Омельченко П.В.

студент,

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АНАЛІЗ ПРОБЛЕМИ ГЕНЕРАЦІЇ БАЗИ ДАНИХ НА ОСНОВІ ОНТОЛОГІЇ ПРЕДМЕТНОЇ

Онтологія (за визначенням Н. Грубера) – специфікація концептуалізації [1]. Це спроба всеохоплюючого і детального формального опису певної області людських знань, за допомогою концептуальної схеми. Онтології придатні для розуміння людьми, але в той же час всебічний та формальний опис, придатний для автоматичної обробки спеціальними програмами. Онтології можуть використовуватись для:

- формування і фіксації загального знання певної предметної області або її частини;
- явної концептуалізації предметної області, що дозволяє описати семантику даних;
- опису певних компонентів, які можуть використовуватись як специфікація для певної задачі;
- розробки сховищ даних в якості високорівневого інтерфейсу;
- покращення взаємодії між програмними агентами та розробниками;
- уніфікації обміну даними;
- формалізації процесів специфікації.

Таким чином онтології використовуються для розв'язання великого спектру завдань, та досить поширені в інформаційних системах.

Елементами онтології є:

– екземпляри (індивіди) – це основні, низкорівневі компоненти онтології, які являють собою або фізичні об'єкти (книги, тварини, споруди) або абстрактні (слова, числа). Є не обов'язковими елементами онтології, проте оскільки однією з основних задач онтологій є класифікація екземплярів, то зазвичай вони наявні;

– класи (поняття) – абстрактні групи, набори або колекції певних об'єктів. Класи можуть містити екземпляри, класи нащадки, або і те і інше;

– атрибути – містяться в об'єктах онтології, використовуються для зберігання інформації про об'єкт, такої як ім'я та значення, також можуть містити специфічну інформацію, яка властива лише об'єктам певного типу. Значення атрибуту може бути складеним типом даних, тобто списком простих типів даних;

– відношення (залежності) – атрибут значення якого є інший об'єкт. Використовуються для визначення залежностей між об'єктами, наприклад відношення наслідування.[2]

Основними проблемами при генерації схеми бази даних на основі онтології заданої предметної області є:

– в структурі онтології зв'язки існують не тільки між класами (класи представляють таблиці у базі даних), а й між самими даними;

– в структурі онтології існують класи, які використовуються тільки як предки для інших класів, і не містять ніяких даних;

– складність встановлення відповідності між даними на класом до якого вони належать;

– значна відмінність зв'язків між класами онтології та зв'язків між таблицями в базі даних.

Для аналізу онтології рекомендується скористатися схемою представленою на рисунку 1.

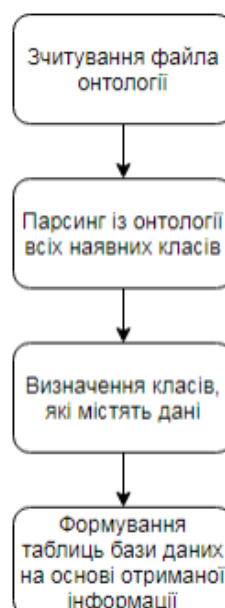


Рис. 1. Схема аналізу онтології

Система відкидає із розпаршеної колекції екземпляри, оскільки для генерації таблиць необхідні тільки класи, а екземпляри лише зберігають інформацію, яка на даному етапі розробки системи ніяк не використовується. Після відбору класів, постає проблема визначення на основі яких з них генерувати таблиці бази даних. Для зменшення логічних помилок та покращення швидкодії системи було прийнято рішення генерувати таблиці на основі лише тих класів які містять екземпляри, адже класи які використовуються лише як предки для інших класів, власних екземплярів не містять. Наступним логічним кроком є побудова таблиць бази даних на основі відібраних класів онтології. При створенні таблиці назва класу стає назвою таблиці, атрибути класу стають полями таблиці, при цьому тип поля залишається таким яким був тип атрибуту в онтології.

Список використаних джерел:

1. Лапшин В.А. Онтологии в компьютерных системах. – М.: Научный мир, 2014.
2. Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения. – М.: Бином. Лаборатория знаний, 2015. – 173 с. – ISBN 978-5-9963-0007-5