Інтелектуальні методи видобування ключових словосполучень із тексту для побудови онтологічних моделей інформаційно-пошукових систем

Мокін В.Б., Бондалєтов К.О.
(Вінницький національний технічний університет,
E-mail: vbmokin@vntu.edu.ua)

В сучасних інформаційно-пошукових системах пошук здійснюється на основі онтологічних моделей, які, у свою чергу, будуються із понять-онтологій [1].

Зазвичай, онтологічні моделі побудовані на окремих ключових словах, а не на словосполученнях. Це дещо уповільнює пошук. Наприклад, прив'язка даних до слова «море» значно ширша, ніж до «Чорне море». Або англійською мовою «bug» є значно ширшим поняттям, аніж – «Southern Bug» (у пошукових системах, як правило, ігнорується регістр слів). Однак, якщо методи автоматичного видобування окремих слів – добре розвинені, то методів видобування комбінацій слів – значно менше і вони ще потребують удосконалення [2].

Більшість методів, наприклад TF.IDF, KP-Miner, RAKE, YAKE, KEA, TextRank, SingleRank, ExpandRank, TopicRank, TopicalPageRank, PositionRank і MultipartiteRank та інші, використовують статистичні підходи щодо заданого тексту [3], але це ускладнює їх застосування до великої кількості текстів, які можуть бути ще й – різними мовами. Було б більш доцільно використовувати єдиний великий корпус слів, на якому відпрацьовувати алгоритми і потім лише адаптувати під кожний текст окремо. Таку можливість в останні роки надає розроблена у Google багатомовна модель BERT (Bidirectional Encoder Representations from Transformers), натренована на великих корпусах, передусім на Вікіпедії. Для неї є готові передтреновані моделі (https://huggingface.co/models) 104

мовами, у т.ч. українською. Якщо застосувати BERT до оригінального тексту, то на виході ми отримаємо вектори чисел (ембеддинги), які зможемо далі обробляти. Цінність такого підходу полягає у можливості співставлення результатів для різних текстів.

На BERT основано відомий пакет програм KeyBERT [2], який реалізує декілька методів. В одному методі спочатку знаходять можливі словосполучення-кандидати, а потім серед них вибирають найкращі, які найчастіше зустрічаються, використовуючи як критерій для порівняння косинусну подібність cosine_similarity. У другому методі (MaxSum) максимальна сумарна відстань між парами заданих словосполучень визначається як пари даних, для яких відстань між ними є максимальною, тобто максимально збільшується схожість кандидата з документом, мінімізуючи подібність між кандидатами. У третьому методі (MMR) використовується максимальна релевантність маржі, яка намагається звести до мінімуму надмірність і максимізувати різноманітність результатів [2]. Все це автоматизовано у бібліотеці KeyBERT на Python.

Ми дослідили ефективність цих методів на прикладі різних текстів про стан та якість водних ресурсів р. Південний Буг, у т.ч. з веб-ресурсів, один з прикладів таких результатів наведено у [4] (рис. 1).

Аналіз показав, що ключова слова, отримані за методами MaxSum та MMR є більш релевантними, а за методом MMR – дійсно більш різноманітні, але це не завжди корисно. А ще важливо добре здійснювати передоброблення тексту (видалення чисел, стоп-слів, скорочень тощо) чи додаткове фільтрування вибраних ключових словосполучень у результатах видобування (видалення дублікатів у вигляді перестановки слів та ін.).

	sim
keywords	
річки південний	0.173172
південний буг	0.168476
гідрографічна мережа	0.156652
чорного моря	0.107028
малі річки	0.093931

Рис. 1. Перші 5 найкращих ключових словосполучень із двох слів, отримані методом MMR з усередненням функції cosine_similarity на усі речення, видобуті зі сторінки веб-сайту БУВР Південного Бугу про гідрографічну мережу (https://buvrpb.davr.gov.ua/vodni-resursy/hidrohrafichna-merezha) [6]

Список використаних джерел

1. Стрижак О. Є. Онтологічні інформаційно-аналітичні системи / Олександр Євгенійович Стрижак // Радіоелектронні і комп'ютерні системи. 2014. № 3. С. 71-76.
2. M. Grootendorst. Keyword Extraction with BERT // Towards Data Science. — 2022 — https://towardsdatascience.com/keyword-extraction-with-bert-724efca412ea
3. LIAAD Yet Another Keyword Extractor (Yake) / Laboratory of Artificial Intelligence and Decision Support. — https://github.com/LIAAD/yake
4. Vitalii Mokin, Kostyantyn Bondaletov. Kaggle Notebook "NLP for UA : KeyBERT - keywords extractions"—

https://www.kaggle.com/code/vbmokin/nlp-for-ua-keybert-keywords-extractions