УДК004.42

АРХІТЕКТУРА СИСТЕМИ ОЦІНКИ РЕЛЕВАНТНОСТІ

Астістова Т.І., кандидат технічних наук, доцент
Київський національний університет технологій та дизайну
Бабич А.А., студентка
Київський національний університет технологій та дизайну

Ключові слова: алгоритми, Python, TF-IDF, BM25, BERT, семантичні моделі, препроцесинг,релевантність, контент, ІПС.

У сучасному інформаційному просторі виникає необхідність ефективного аналізу та оптимізації контенту у пошукових системах (ІПС). ІПС базуються на алгоритмахпошуку та ранжуванні вебдокументів, що дає можливість визначати, наскільки зміст документа відповідає інформаційному запиту користувача.

Розрізняють кілька видів релевантності в залежності від критеріїв оцінки: тематична (відображає збіг тематики документа із запитом користувача); семантична (враховує значення і контекст вживання слів); контекстна (оцінює відповідність результату конкретній ситуації чи намірам користувача); персоналізована (адаптує результати пошуку до поведінкових особливостей користувача) [1,4].

Класичні алгоритми оцінювання релевантності, зокрема TF-IDF (TermFrequency-InverseDocumentFrequency - статистична міра) та BM25(BestMatching 25, - вдосконалена версія TF-IDF) дозволяють визначати важливість термінів у тексті на основі їх частотного розподілу. Ці підходи не враховують контекстуальні зв'язки між словами, синонімію та семантичні відтінки, через що результати пошуку часто втрачають точність[1].

Сучасні нейромережеві підходи, зокрема моделі на основі трансформерів семантика (BERT- двонаправлена нейромережа, RoBERTa, MUM- мультимодальні моделі), забезпечують глибоке контекстне розуміння тексту та точніше відображають його змістову структуру. Однак їх використання потребує значних обчислювальних ресурсів і часу, що обмежує можливість інтеграції таких моделей у прикладні інструменти аналізу контенту.

В роботі представлено архітектуру системи та моделі даних, які забезпечать узгоджене зберігання, обробку й аналіз інформації та полегшують вирішенню задачі оцінки релевантності контенту. .

Запропонований підхід ґрунтується на модульній архітектурі. Кожен модуль виконує окрему функцію, а взаємодія модулів описується за допомогою послідовних етапів обробки даних. Такий поділ спрощує реалізацію, тестування та подальше розширення системи [2]. Окремі модулі (пакети/скрипти) відповідають за завантаження даних, препроцесинг, обчислення текстових показників, семантичний аналіз, інтегральну оцінку та формування звітів.

Архітектуру системи доцільно подати у вигляді структурної схеми (рис. 1). На вхід системи подається тексти сторінок цільового вебсайту та семантичне ядро пошукових запитів. На виході - зведені таблиці й аналітичні матеріали, які містять значення показників релевантності для конкретних сторінок і кластерів запитів, а також рекомендації щодо оптимізації контенту. На рисунку 1. представлена структурна схема системи для аналізу текстів.

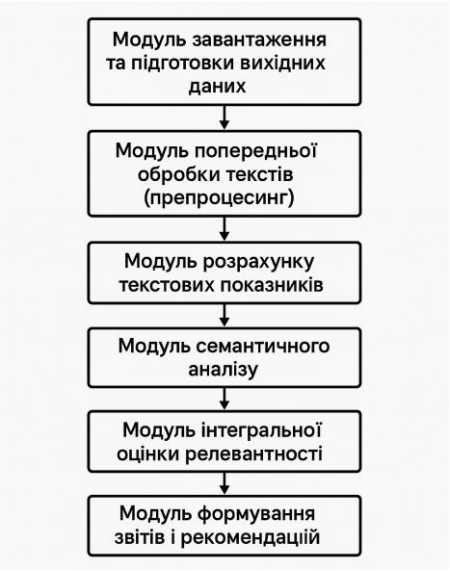

Рисунок 1 – Структурна схема системи аналізу текстів

Модуль завантаження даних забезпечує отримання текстів вебсторінок у зручному форматі (наприклад, CSV, JSON або бази даних) та семантичного ядра (список запитів та їх кластерів).

Модуль попередньої обробки текстів реалізує єдину для сторінок і запитів схему препроцесингу (нормалізацію регістру, токенізацію, видалення стоп-слів, лематизацію, фільтрацію шумових токенів). На цьому етапі формуються уніфіковані текстові подання, придатні для подальшого застосування частотних і семантичних моделей.

Модуль розрахунку текстових показників використовує результати препроцесингу для обчислення індексу покриття ключових слів, TF-IDF, BM25, а також структурних індикаторів (розташуванням ключових термінів у заголовках, підзаголовках, метатегах та основному тексті). Вихід цього модуля - матриця текстових показників для пар «кластер запитів - сторінка».

Модуль семантичного аналізу . На цьому етапі формуються семантичні індекси для пар «кластер - сторінка» або окремих запитів.

Модуль інтегральної оцінки релевантності поєднує результати двох попередніх модулів, розраховуючи інтегральний індекс текстової релевантності для кожної сторінки й кластера запитів. Модуль класифікує сторінки за рівнями релевантності (низький, середній, високий).

Модуль формування звітів і рекомендацій перетворює числові результати на інтерпретовані представлення (таблиці, діаграми, теплові карти.)

Для реалізації цієї архітектури було обрано середовище програмування Python, яке має розвинену екосистему бібліотек для аналізу текстів і побудови моделей обробки природної мови. Було проведено експериментальне тестування в середовищі Python із використанням бібліотек scikit-learn, sentence-transformers, numpy, pandas [3,4].На рисунку 2 представлено фрагмент розробки модуля препроцесингу текстів на мові Python.

Використання архітектури системи дозволяє:

1. гнучко працювати з різними форматами вхідних даних (CSV);
2. використовувати готові реалізації алгоритмів TF-IDF, BM25, ембеддінгів;
3. швидко будувати допоміжні візуалізації (таблиці, діаграми) для аналізу результатів.

```python
def __init__(self, lang: str = "uk", min_token_len: int = 2):
    self.lang = (lang or "uk").lower().strip()
    self.min_token_len = max(1, int(min_token_len))

    # Морфологічні аналізатори
    self.morph_uk = pymorphy3.MorphAnalyzer(lang="uk")
    self.morph_ru = pymorphy3.MorphAnalyzer(lang="ru")

    self.stop_words = STOP_UK if self.lang == "uk" else STOP_RU

def normalize(self, text: str) -> str:
    if not text:
        return ""
    text = text.lower()
    text = text.replace("’", "'").replace("`", "'")
    text = re.sub(r"\s+", " ", text).strip()
    return text

def tokenize(self, text: str) -> List[str]:
    if not text:
        return []
    return self.TOKEN_RE.findall(text)
```

Рисунок 2 - Фрагмент програмної реалізації препроцесингу текстів (Python)

Список використаних джерел

1. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. Pearson, 2023. 1248 p.

2. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond // Foundations and Trends in Information Retrieval. 2009. Vol. 3, No. 4. P. 333-389.

3. «Python frameworks, purpose» [Електронний ресурс]. – Режим доступу https://foxminded.ua/freimvorky-python/

4. Пріменко Д.Ю., Астістова Т.І. Алгоритмічні методи обробки текстових даних для оцінки та підвищення релевантності вебдокументів з питань енергоефективності.// Д.Ю. Пріменко Д.Ю.МгІТ-1-24, Т.І Астістова Т.І, Матеріали IV Міжнародної науково-практичної конференції «Енергоефективний університет» 23 жовтня 2025 року м. Київ.