Міністерство освіти, науки, молоді та спорту України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики
Магістерська програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
проф., д.ф.-м.н.

_____М. М. Глибовець
(підпис)
10 листопада 2013 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на дипломну роботу
(магістерську тезу)

студенту 2 року навчання МП «Інформаційні управляючі системи та технології»

*Решетньову Ігорю Володимировичу*
на тему:
**Розробка алгоритму ітеративної побудови термінології на базі колекцій текстів наукової тематики**

Зміст текстової частини роботи (ТЧ)
Вступ (обгрунтування та постановка задачі)

1. Роль автоматизованої побудови тезаурусів як предмету дослідження в галузі інформаційного пошуку
    1. Проблеми інформаційного пошуку
    2. Розвиток онтологічних і семантичних систем аналізу текстів
    3. RDF граф як формат опису залежностей між ресурсами в семантичному вебі
2. Огляд існуючих методів побудови термінології і складання тезаурусів
    1. Статистичні методи
    2. Лексикографічні методи
3. Розробка ітеративного методу побудови термінології за допомогою комбінації лексикографічного і статистичного методів
    1. Математична модель
    2. Опис алгоритму, якісна та кількісна оцінки
    3. Застосування алгоритму для побудови термінології у вигляді RDF схеми з використанням синтетичних та реальних даних україномовної наукової періодики.
        1. Розробка інтерфейсу користувача і RESTful API до системи
        2. Схема тестування та оцінка результатів

Висновок (досягнені результати згідно з поставленої мети)
Джерела
Додатки

Дата видачі 20 жовтня 2013 р.

Керівник А. М. Глібовець
(підпис)
Завдання отримав І.В.Решетньов
(підпис)

**Власний доробок**

1. Використання статистичного методу. Відбір значимих темінів в документі за метрикою TF-IDF. Idf отримати за допомогою взяття з Google Search.
    a. Розбір pdf документів у формат, прийнятний для індексування.
    b. Початкова індексація документів за допомогою рішення Apache Lucene.
    c. Вирішення проблеми низької документарної частоти терімнів, що виникає через неповноту і обмеженість індивідуальних тематичних колекцій документів, за рахунок заміни її документарною частотою з великої пошукової системи.
    d. Використання Google Custom Search API для пошуку документарної частоти термінів.
    e. Використання MongoDB в якості кешу для зберігання отриманих документарних частот термінів, вирішення проблеми обмеженої квоти використання Google Search API.
    f. Пошук і застосування рішення щодо лематизації вхідних слів документу. Бібліотека JLemmaGen.
    g. Проблема сортування і відкидання загальновживаних слів з вихідного документу.
2. Формалізація і реалізація ітеративного кроку алгоритму. Для кожного специфічного терміну в новому документі пошук в документах колекції характеристичних фрагментів тексту, що описують термін.
    a. Пошук характеристичних фрагментів у вигляді повних речень

документу. Інтерпретація речення як одиниці зв’язку між термінами.

b. Представлення характеристичного фрагменту у вигляді вікна навколо згаданого терміну, що в залежності від розміру вікна може містити від декількох словосполучень до декількох речень. Застосування лексикографічного принципу цитатної картотеки.
c. Врахування ваги - частоти згадувань терміна, під час пошуку характеристичних фрагментів.

3. Використання лексикографічних методів під час пошуку характеристичних фрагментів тексту.
    a. Огляд і відбір існуючих відношень між термінами у сфері побудови тезаурусів.
        i. асоціація
        ii. відношення “is a”
        iii. відношення “part of”
    b. Огляд лексикографічних принципів формування означень. Відбір лексикографічних шаблонів, що відповідають означенням одних термінів через інші
        i. слова-марекери: тире, “це”, “значить”...
        ii. по частинах мови, частинах речення.
    c. Part-of-Speech тегування. Пошук рішення і застосування бібліотеки Java Language Tool.
    d. Реалізація лексикографічних шаблонів
    e. Зберігання відносин між термінами.
4. Використання документарної бази MongoDB в якості ефективного сховища для даних проміжних кроків роботи алгоритму.
    a. Зберігання документарної частоти термінів.
    b. Зберігання специфічних термінів колекції.
    c. Зберігання відношень між термінами
    d. Зберігання характеристичних фрагментів тексту для подальшого використання в пошуковій системі.
5. Ітеративний крок алгоритму: пошук зв’язків в характеристичних фрагментах тексту, застосування розроблених лексикографічних

шаблонів.

6. RESTful API для роботи з системою. Передбачено наступні елементи інтерфейсу:
    1. Надати список всіх колекцій документів
    2. Створити нову колекцію документів за ім’ям
    3. Додати до колекції документ
    4. Переглянути список документів колекції
    5. Переглянути відомості про документ колекції
    6. Завантажити зміст документу з колекції
    7. Переглянути всі терміни колекції
    8. Завантажити RDF граф термінів колекції. Використати формат JSON-LD.
    9. Навігація по зв’язках між термінами (посилання в стилі HATEOAS, види посилань - “is a”, “part of”, інші)
    10. Перегляд терміну колекції, характеристичних фрагментів тексту з посиланнями на документи.
7. Огляд існуючих підходів (google knowledge graph, wordnet, майже всі підходи - ручна розмітка експертами). Наш підхід - максимальне використання існуючих рішень для вирішення поставленої задачі. Детально описати всі кроки, гіпотезу, формалізацію алгоритму. Чому це має спрацювати, обгрунтування кроків. Трохи теорії по tf-idf і лексикографічним методам. Як успіх - мати таку робочу програму.
8. Протестувати на наукових записках НаУКМА і соціологічних дослідженнях.
9. Після передзахисту - інтерфейс користувача для демонстрації можливостей автоматичної побудови, навігації по визначеннях. Дослідження варіацій алгоритму, тестування на різних колекціях. План для майбутніх доробок.