Національний університет «Києво – Могилянська академія»

Магістеріум факультету інформатики



**Магістерська теза**

на тему:

**«Розробка алгоритму ітеративної побудови термінологій в колекціях наукових текстів»**

*Підготував:*

студент ІУСТ-2,

Решетньов І.В.

*Науковий керівник:*

А.М.Глібовець

**Постановка завдання**

В рамках даної роботи буде досліджено алгоритм ітеративної побудови термінології у вигляді RDF графу за колекцією наукових текстів спільної тематики. Схема побудови зв’язків між термінами буде враховувати статистичні властивості термінів а також лексикографічні евристики пов’язані зі структурою україномовних документів. Принципова схема алгоритму складається з двох кроків: початкове виділення з тексту важливих термінів за статистичним методом, і послідуюче видобування з тексту пов’язаних термінів за рахунок застосування лексикографічних шаблонів. В рамках дослідження передбачається розробити структурну схему алгоритму, з подальшою формалізацією алгоритму, а також програмне забезпечення для моделювання роботи алгоритму. Кінцевою метою розробки є програма, що дозволить отримати RDF граф термінів колекції документів, наданої у форматі pdf. Результати даної роботи мають вирішити задачу складання термінології по колекції документів і можуть слугувати основою для створення інтелектуальних систем пошуку документів у семантичному вебі.

**Анотація**

Мета даної роботи – розробити метод і відповідне програмне забезпечення для вирішення задачі побудови термінологіі в колекціі текстів наукової тематики.

Дане дослідження є складовою частиною циклу робіт проведених на кафедрі з тематики, присвяченої побудові пошукової системи і репозіторію наукових праць.

З алгоритмічної точки зору результатом буде такий алгорітм, що на вході отримує колекцію документів у форматах pdf | doc, а на виході віддає файл що містить RDF граф з термінологією що зустрічається в даній колекції.

Як складовий модуль системи пошуку і каталогізації наукових праць, на модульному рівні дистрибутив програми буде надано у вигляді jar-архіву з відповідним API.

Для кінцевого користувача і для тестування ефективності побудови термінології буде розроблено веб-інтерфейс до компонентів програми, де в користувача будуть можливості завантажити в систему колекцію документів, запустити алгоритм побудови термінології, передивитись і скачати файл-результат у вигляді RDF, а також продивитись термінологію, її входження в документи колекції і пов'язані з результатом роботи алгоритму метадані термінів.

Функціональна схема роботи алгоритму побудови термінології буде включати в себе наступні етапи і технології:

* початкова індексація документів колекції за допомогою рішень з відкритим кодом (Apache Lucene);
* застосування алгоритмів розбору NLP (лінгвістичних аналізаторів) на реченнях і абзацах текстів, зберігання отриманих метаданих у спеціалізованих індексах;
* експериментальні дослідження щодо формулювання множини ефективних запитів до використаних пошукових систем і сховищ даних, що будуватимуть зв’язні висловлювання щодо термінів, опис і використання лінгвістичних евристик;
* розробка і застосування ітеративного алгоритму побудови термінології, що зможе покращувати результати при додаванні нових документів до колекції, а також після кожної ітерації;
* використання документарної бази MongoDB в якості сховища даних RDF;
* розробка RESTful API і веб-інтерфейсу користувача до досліджуваної системи;
* підбір тестових колекцій наукових документів і написання тестових пакетів для оцінки ефективності і точності алгоритму;
* експериментальні дослідження алгоритму з варіацією параметрів;
* встановлення залежностей між параметрами, публікація найбільш вдалих налаштувань як окремих методів API;
* розгортання готової системи на сервері для публічного доступу.

Зміст

Вступ 6

Розділ I 6

Роль автоматизованої побудови тезаурусів як предмету дослідження в галузі інформаційного пошуку 6

1.1. Проблеми інформаційного пошуку 6

1.2. Розвиток онтологічних і семантичних систем аналізу текстів 7

1.3. Сучасний стан опису ресурсів в семантичному вебі: WordNet, Google Knowlrdge Graph, Wikipedia. 7

1.4. RDF граф як формат опису залежностей між ресурсами в семантичному вебі 7

1.5. Огляд концептів і абстрактного синтаксису RDF 7

1.6. Огляд стандарту JSON-LD як конкретної специфікації RDF. 7

1.7. Дослідження існуючих методів побудови термінології і складання тезаурусів 7

1.7.1 Статистичні методи 8

1.7.2 Лексикографічні методи 9

Розділ II 11

2.1 Структурна схема і формалізація алгоритму 11

2.1.1 Структурна схема алгоритму 11

2.1.2 Математична модель і формалізація алгоритму 11

2.2 Оцінка складності алгоритму 11

Розділ III 12

Реалізація алгоритму 12

Вибір готових рішень для реалізації кроків алгоритму. 12

Реалізація системи обробки документів для побудови термінології 12

Реалізація прикладного програмного інтерфейсу доступу до системи. 12

Реалізація графічного інтерфейсу користувача у вигляді веб-сервісу. 12

Розділ IV 12

Схема тестування та оцінка результатів 12

Висновок 13

Використані джерела 14

Додатки 14

# 

# Вступ

# Розділ I

# Роль автоматизованої побудови тезаурусів як предмету дослідження в галузі інформаційного пошуку

# 1.1. Проблеми інформаційного пошуку

Зазвичай, інформаційна потреба користувача пошукової системи не відповідає термінам, що зустрічаються в документах, або користувач неправильно розуміє в конкретиний час в конкретному місці термінологію області знань, до котрої він здійснює пошук. За таких умов, одним із методів покращення пошукової видачі є використання тезаурусів термінів предметних областей. Тезауруси є таблицями термінів, що поєднують пов’язані між сосою терміни, зазвичай вказуючи тип зв’язку (NT, BT, USE, RT). Інформаційні системи можуть використовувати тезауруси на етапі індексації документів, для того щоб правильніше класифікувати документи по категоріях, або під час пошуку, розширюючи пошуковий запит користувача пов’язаними термінами.

Тут варто згадати експерименти, що показували, як люди тільки у 20% випадків вибирали однакові терміни для опису тих самих речей, а також заміри релевантності пошукової видачі з доданою інформацією з тезаурусу, що мають більші показники. Таким чином, можна вважати доведеним і очевидним, що використання тезаурусів в пошукових системах значно покращує їх якість.

Проблема полягає у тому, що для більшості комерційних баз даних, що поширють наукову інформацію, тезауруси складаються експертами з області знань, а також фахівцями зі складання тезаурусів, тобто власноруч. Якщо ж розглядати окремі найновіші області знань, де відповідна термінологія тільки формується і випускається велика кількість нових публікацій, то в данному випадку термінологічні словники дуже швидко застарівають, і їх треба оновлючати частіше, тобто знову залучати експертів. На противагу такому підходу, існують методи автоматизованої побулови тезаурусів, що в якості корпусу беруть усі найновіші публікації з теми, і будують по них зв’язки між термінами. Як наслідок, маючи таку систему оновлювати термінологічні зв’язки значно простіше і дешевше. В публікації Моніки Лассі було розглянуто основні методи автоматизованої побудови тезаурусів, такі як CO, LSI, SVD та метод баєсівських мереж. Дані методи мають як різну ефективність і часову оцінку складності, так і приниципи - статистичний і лексикографічний. Розробці і тестуванню нового методу, що б використовував комбінацію ідей, що зустрічаються в цих методах, і присвячено дану магістерську тезу.

## 1.2. Розвиток онтологічних і семантичних систем аналізу текстів

?

## 1.3. Сучасний стан опису ресурсів в семантичному вебі: WordNet, Google Knowlrdge Graph, Wikipedia.

Ворднет- корпус англійської мови, розмічений експертами.

## 1.4. RDF граф як формат опису залежностей між ресурсами в семантичному вебі

## 1.5. Огляд концептів і абстрактного синтаксису RDF

## 1.6. Огляд стандарту JSON-LD як конкретної специфікації RDF.

[**http://www.niso.org/schemas/iso25964/iso25964-1\_v1.4.xsd**](http://www.niso.org/schemas/iso25964/iso25964-1_v1.4.xsd)

Використати для контексту JSON-LD.

<http://json-ld.org/playground/> - ресурс для перевірки валідності формату і конвертації між видами серіалізації.

## 1.7. Дослідження існуючих методів побудови термінології і складання тезаурусів

**CO, LSI, SVD, Bayesian Networks.**

Chen et al. (1995, p. 178) declare that the research in the area of generating terms by co-occurrence analysis

done in the 1980’s and 1990’s have not given the results hoped for. Some experiments have resulted in poor

retrieval when terms were generated completely automatically. An experiment by Ekmekcioglu, Robertson &

Willetii in 1992 employed a user-directed approach where users were suggested terms for expanding their

queries. Four different approaches were used; original queries; query expansions generated by co-occurrence

data; Soundex codes, where the same phonetic code is assigned to words that sound the same; and strong

similarity measure based on similar character microstructure. The results showed no significant difference in

retrieval effectiveness between the initial queries and the expansions made. There was, however, a very small

degree of overlap between the relevant documents retrieved by the initial queries and the relevant documents

retrieved by the co-occurrence approach. The small degree of overlap of retrieved documents show that by

adding to the query terms from a thesaurus generated by co-occurrence analysis, **an increase in the number of**

**relevant documents retrieved can be accomplished**. (ibid, p. 178f)

### 1.7.1 Статистичні методи

Idf = 1/df

Метрика, що дозволяє сортувати терміни по важливості і специфічності в документах.

*Тут навести приклади досліджень, що сходяться до того, що проста метрика tf-idf настількі ж ефективна, як і складні ймовірностні моделі.*

Документарну частоту по невеликій колекції, а там паче по 1 документу, побудувати джуе важко. Набагато простіше взяти вже існуючу докумнтарну частоту терміну з пошукової системи. Як приклад було обрано пошукову систему Google, що разом з результатами звичйного пошуку завжди повертає приблизну кількість результатів. Даний показник і було взято за документарну частоту терміна. Під час побудови допоміжних методів щодо видобування даного показника з пошукової видачі, було прийнято рішення, що звичайний парсинг веб-сторінок пошукової видачі буде громіздким рішенням, оскільки задля одного числа прийдеться запитувати і розбирати цілу сторінку, що містить багато інших даних. Тому в рамках окремного програмного модулю дипломної роботи було написано споживач Google Custom Search RESTful API, з налаштуванням всього необхідного в консолі сервісу Google. Як результат було отримано метод, що за даним терміном повертає його документарну частоту, причому кількість інтернет трафіку для таких запитів є мінімальною. Проте виникає інша проблема для корустування Google API: квоти вільного використання, що становлять 100 запитів на день. Звичайно, щоб обробляти великі документи і знайти, які терміни в них є важливими, необхідно дізнатись документарну частоту кожного. Тому було прийнято рішення, по-перше, зберігати в локальному кеші всі знайдені документарні частоти, і в першу чергу звертатись до нього, що з бігом часу призведе до досить повної бази термінів, і по-друге, залишити можливість обробляти звичайні веб-сторінки пошуку, на випадок вичерпання квоти.

Кеш, як і інші дані проміжної роботи алгоритму, було вирішено зберігати в документарній базі MongoDB.

### 1.7.2 Лексикографічні методи

Гіпоніми і гіперніми

<http://en.wikipedia.org/wiki/Hyponymy_and_hypernymy>

Першим для пошуку гіпонів лексикографічні патерни використав Херст в 1992 році.

<http://en.wikipedia.org/wiki/Noun_phrase>

За Gregory Grefenstette, можна вибирати тільки сполучення іменник-іменник, як варіант.

Варто обмежетись іменниковими словосполученнями і фразами як у роботі Херста, під час пошуку і транслювання лексикографічних патернів.

weighted Jaccard measurement

Патерни (з роботи херста)

such NP as {NP ,}\* {(or | and)} NP

NP {, NP} \* {,} or other NP

NP {, NP}\* {,} and other NP

NP {,} (including | especially) {NP,}\* {or | and} NP

**Пошук фразових іменникових словосполучень**

Для реалізації алгоритму пошуку гіпонімів, спершу необхідно навчити систему розпізнавати фразові словосполучення.

Пропонований підхід - фікскація іменників у реченні, з послідуючим добиранням навколишніх слів за правилами.

За Херстом, відношення між термінами “загальне-чатскове” відповідають гіпонімічним зв’язкам у тексті. І даному досліднику вдалося виокремити підмножну шаблонів, що була б достатньо точною і виконуваною для більшості текстів, враховуючи відмінності в формуванні термінології в окремих сферах інтересів.

Вважаючи на схожість наукового стилю на міжнародному рівні, здається вдалою думка про локалізацію знайдених шаблонів для української мови, з доданням нових.

Для того щоб звузити рамки дослідження і досягти певного результату для специфічних, проте найбільш уживаних способах творення термінології, до розгляду було залучено тільки терміни-іменники і іменникові словосполучення. Таке рішення було прийнято, виходячи з тих припущень, що більшість темінів утворено саме поодинокоми іменниками або термінологічними словосполученнями, тому і зв’язки в тексті слід шукати між іменниками і словосполученнями.

При цьому пропонується використати каскадний підхід, за котрим термінологія складається з найбільш широких зв’язків між усіма словами в реченні, потім серед іменників в реченні, потім між оточуючими ці іменники словосполученнями.

Під позначенням NP в записі шаблону слід розуміти іменникове словосполучення, що може складатись як з одного іменника, так і набувати набагато складніших форм.

З шаблонів, що відповідають за зв’язки між термінами у реченні, було обрано наступні категорії:

1. Прямі означання і дефініції, з використанням характерних для української мови знаків пунктуації і слів-зв’язок ( тире, слова “є”, “вважається”, “слід розуміти” і т. д.)
2. Шаблони за Херстом, а саме: + примеры
   1. такий NP як {NP ,}\* {(і | або)} NP
   2. NP {, NP} \* {,} або інший NP
   3. NP {, NP}\* {,} і інші NP
   4. NP {,} (включаючи | особливо) {NP,}\* {і | або} NP
3. Шаблони на позначення зв’язків частина-ціле
   1. NP є частиною NP

Всі подані шаблони розширюються синнімічними і схожими за вживанням словами в формулах шаблону. Під час співставення речень з шаблоном відбувається приведення всіх слів до нормільної форми, що дозволяє зменшити необхідну кількість варіацій шаблону.

**Етапи впровадження лексикографічних підходів**

в даній роботі наступні:

1. Відповідно до оглянутої літератури і відомостей з лексикографії, побудова і тестування на простих прикладах - реченнях шаблонів, що дозволять виокремити термінологічні словосполучення

2. Тестування на простих реченнях шаблонів, що виокремлюють зв’язки між термінами (темрінологічними словосполученнями)

3. Тестування побудованих шаблонів на визначеннях з термінологіних словників. Оцінка точності запропонованих методів.

4. Тестування методів на текстах вікіпедії. Порівняння з вже розміченими зв’язками між термінами.

5. Тестування на вільних колекціях

**Деталі реалізації методу співставлення з шаблоном.**

Під час застосування правил, враховується їх черговість, таким чином в першу чергу віднаходяться і потрапляють в якості елементів співпадіння ті іменникові словосполучення, що є ширшими за кількістю слів, а отже рідшими за вживаням.

Таким чином для налаштування підисистеми пошуку термінологічних словосполучень, необхідно правильно підібрати не тільки правила їх формування по частинах мови, а ще й послідовність застосування.

В якості тестового словника використано <http://ipp.lp.edu.ua/Library/004/004.html>

*Кизименко Л.Д., Бєдна Л.М.* Словник-довідник соціального працівника

*Для студентів та соціальних працівників*

# Розділ II

Розробка ітеративного методу побудови термінології за допомогою комбінації лексикографічних і статистичних методів

## 2.1 Структурна схема і формалізація алгоритму

### 2.1.1 Структурна схема алгоритму

**Індексування**

Процес побудови термінології з колекції текстів можна розкласти на два принципових кроки: по-перше, видобування зі всіх слів, що зустрічаються в тексті документів, таких, що відповідають термінам в області знань відповідних документів, і по-друге – встановлення на множині даних термінів відношень, що використовуються в тезаурусах, зокрема симетричне відношення пов’язаних термінів (RT – *Related Term*), і асиметричні відношення типу “частина-ціле” (BT – *Broader Term*, і NT – *Narrower Term* відповідно).

Завдання виокремлення термінів з

### 2.1.2 Математична модель і формалізація алгоритму

## 2.2 Оцінка складності алгоритму

# Розділ III

Практична частина

## Реалізація алгоритму

### Вибір готових рішень для реалізації кроків алгоритму.

## Реалізація системи обробки документів для побудови термінології

## Реалізація прикладного програмного інтерфейсу доступу до системи.

## Реалізація графічного інтерфейсу користувача у вигляді веб-сервісу.

# Розділ IV

Застосування алгоритму для побудови термінології у вигляді RDF схеми з використанням синтетичних та реальних даних україномовної наукової періодики.

## Схема тестування та оцінка результатів

Налаштування конфігурації бази для тестування

<http://bits-and-kites.blogspot.com/2014/01/spring-mongodb-tutorial.html>

Для тестування роботи алгоритму було вирішено розробити систему конфігурацій для швидкого прогону різних варіацій алгоритму на різних даних, та в одночас з необхідною оптимізацією швидкості таких тестувань, а також із заміром основних характеристик виконання окремих кроків алгоритму.

Найбільше для даного випадку підходить реалізація окремих інтеграційних тестів, що містимуть незалежні конфігурації оточення, такі як реалізація документарної бази у памяті.

1. Тестування алгоритму на словниках. Беремо термінологічний словник, зв’язвний, і дивимось чи знаходить алгоритм ці терміни і зв’язки

2. Тестування на вікіпедії. Там ми вже маємо готові зв’язки між термінами. Порівняти ті зв’язки з тими, що побудує алгоритм по документу з вихідного тексту.

3. Тестування на реальних даних.

Схема тестування є необхідною як для дослідження якості знайдених темінологічних зв’язків між термінами, так і для розгляду модифікацій алгоритму задля віднайдення найбільш ефективного рішення, що буде включено в заключну реалізацію системи як веб-сервісу.

Для цього є пропозиція віднайти:

* реалізації алгоритмів CO, LSI, SVD, Bayesian Networks, для того щоб будувати тезауруси за їх допомогою, і потім порівнюват між собою.
* єдиний формат теазурусу для порівняння.
* зафіксувати модифікації власного алгоритму
* налаштувати тестове середовище, що дозволяло б запускати прогони алгоритмів.
* знайти сукупність реальних текстових наукових матеріалів для порівняння роботи на них алгоритмів.
* якщо алгоритм допускає недетермінованість, роботи декілька прогонів, і потім у порівннянні використовувати усереднену метрику
* розробити метрику порівнняння.

# Висновок

(досягнені результати згідно з поставленої мети)

# Використані джерела

**"Designing and Implementing RESTful Web Services with Spring." *Tutorial ·*. Web. 12 May 2014. <https://spring.io/guides/tutorials/rest/2/>.**

**"JSON-LD 1.0." *JSON-LD 1.0*. Web. 12 May 2014. <http://www.w3.org/TR/json-ld/#data-model>.**

**Landau, Sidney I. *Dictionaries: The Art and Craft of Lexicography*. New York: Scribner, 1984. Print.**

**"RDF 1.1 Concepts and Abstract Syntax." *RDF 1.1 Concepts and Abstract Syntax*. Web. 12 May 2014. <**[**http://www.w3.org/TR/2014/PR-rdf11-concepts-20140109/**](http://www.w3.org/TR/2014/PR-rdf11-concepts-20140109/)**>.**

**Towards a Comprehensive Theory of Lexicographic Definitions.**

[**http://wordnet.princeton.edu/**](http://wordnet.princeton.edu/)

[**http://www.w3.org/wiki/RdfThesaurus**](http://www.w3.org/wiki/RdfThesaurus)

[**http://www.imsglobal.org/vocabularies/iso2788\_relations.xml**](http://www.imsglobal.org/vocabularies/iso2788_relations.xml)

[**http://www.niso.org/schemas/iso25964/**](http://www.niso.org/schemas/iso25964/) **- newest thesauri schema standard**

[**http://www.niso.org/schemas/iso25964/iso25964-1\_v1.4.xsd**](http://www.niso.org/schemas/iso25964/iso25964-1_v1.4.xsd)

# Додатки