

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця
на правах рукопису

Коса Вікторія Вікторівна

УДК: 004.421:81'276.6:002.1(043.5)

ДИСЕРТАЦІЯ

МЕТОД ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ
ТЕРМІНОЛОГІЧНОГО НАСИЧЕННЯ В КОЛЕКЦІЯХ ДОКУМЕНТІВ
ДЛЯ ЗДОБУТТЯ ЗНАНЬ

122 – Комп'ютерні науки

12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ В. В. Коса

Науковий керівник: Єрмолаєв Вадим Анатолійович,
кандидат фізико-математичних наук,
доцент

Запоріжжя – 2021

© 2021 Коса Вікторія Вікторівна. Усі права захищені.

Розповсюджується Запорізьким національним університетом
Міністерства освіти і науки України, шляхом розміщення
у власному відкритому репозиторії,
за умовами ліцензії
Creative Commons License Attribution 4.0 International
(CC BY 4.0).

Ця дисертація доступна онлайн за: <http://phd.znu.edu.ua/page/1367.ukr.html>

АНОТАЦІЯ

Коса В. В. Метод експериментального дослідження термінологічного насичення в колекціях документів для здобуття знань. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 - Комп'ютерні науки. Запорізький національний університет Міністерства освіти і науки України, Запоріжжя, 2021.

Об'єктом дослідження є процес автоматизованого здобуття, з колекцій релевантних документів, наборів термінів, що характеризують довільну професійну предметну область, для подальшої побудови онтологій цієї предметної області, з урахуванням впливу явища термінологічного насичення.

Предметом дослідження є метод експериментального дослідження термінологічного насичення в колекціях документів для здобуття знань у процесі вивчення онтологій довільного домену.

Метою роботи є підвищення репрезентативності, ефективності та результативності здобуття термінології з колекцій професійних документів у межах довільної предметної області для подальшої побудови онтологій, шляхом розробки комплексного обчислювального методу виявлення та вимірювання термінологічного насичення в колекціях професійних текстових документів, що описують предметну область.

У вступі обґрунтовано актуальність теми дисертаційної роботи, зазначено зв'язок роботи з науково-технічними проектами, сформульовано мету і завдання дослідження, визначено об'єкт, предмет та методи дослідження, показано наукову новизну та практичне значення отриманих результатів, наведено інформацію про практичне використання доробку, особистий внесок здобувача, апробацію результатів дослідження та їх висвітлення у наукових публікаціях. Приводяться відомості щодо структури та обсягу дисертаційної роботи.

У розділі 1 розглянуто та проаналізовано сучасний стан досліджень за тематикою роботи. Це допомогло розробити підхід до моделювання процесу

термінологічного насичення, виявлення та вимірювання результатів цього процесу. Розділ починається з викладу методології, яку було використано для відбору літературних джерел для нашого систематичного огляду і аналізу (розділ 1.1). Далі, у розділах 1.2 - 1.11 розглянуто та проаналізовано сучасний стан досліджень у релевантних наукових галузях, починаючи з розробки онтологій (Ontology Engineering) та вивчення онтологій (Ontology Learning) і закінчуючи якісними дослідженнями (Qualitative Research). У розділі 1.12 резюмовано виявлені прогалини у сучасному стані досліджень та мотивацію щодо зменшення цих прогалин. Базуючись на виявлених прогалинах, у розділі 1.13 запропоновано «дорожню карту» для вирішення виявлених відкритих питань. У розділі запропоновано бачення та окреслено підхід для вирішення цих питань дослідження. Крім того, сформульовано питання дослідження, на які необхідно відповісти у рамках окресленого підходу, щоб отримати ефективний та результативний обчислювальний метод. На базі питань дослідження поставлено завдання дослідження, вирішення яких має призвести до досягнення його мети. У розділі 1.14 підсумовано представлені результати огляду та аналізу сучасного стану досліджень.

Розділ 2 сфокусовано на виконанні першого завдання дослідження – розробці формального фреймворку для методу виявлення та вимірювання термінологічного насичення. Розділ починається з базових відомостей (розділ 2.1), що містять визначення, необхідні для окреслення фокусу обраного формального теоретичного підходу. Далі представлено гіпотези дослідження, які необхідно перевірити в контексті теоретичного підходу. Ці гіпотези сформульовано на основі питань дослідження, що згруповані у першому завданні роботи (розділ 1.13). Розділи 2.3 та 2.4 зосереджено на формальному введенні функції термінологічної різниці (*thd*) та доведенні її метричних властивостей у просторі усіх можливих колекцій документів для домену. У розділі 2.5 сформульовано та доведено теорему, що окреслює достатні умови існування термінологічного насичення. На додаток, у розділі 2.6, досліджено, чи можна оптимізувати метод вимірювання та виявлення термінологічного насичення з точки зору зменшення часу на обчислення.

У результаті, представлено розроблений вдосконалений метод, що використовує розділення (partitioning) колекції документів на частки, що не перетинаються. Доведено, що вдосконалений метод дає ті ж самі значення вимірювань *thd*, але вимагає значно менше часу для обчислень і не є обмеженим за обсягом колекції. Крім того, обґрунтовано, що вдосконалений метод можна розпаралелити. Отже, на додаток до поліпшеної ефективності, вдосконалений метод, на відміну від базового (розділи 2.1 – 2.5), є результативним при обробці великих колекцій документів, що мають реальні промислові обсяги.

У розділі 3 виконується друге завдання дослідження – розробка нових або вдосконалення раніше розроблених алгоритмів, що матеріалізують розроблений формальний фреймворк (розділ 2) у вигляді обчислювального методу. У розділі 3.1 представлено конвеєр обробки колекцій документів. Як подання високого рівня, представлено робочий процес (workflow) для вимірювання термінологічного насичення. Подальша деталізація цього процесу наведена у функціональній блок-схемі, яка розкриває модульну структуру нашого набору алгоритмів і залежності між модулями. Алгоритми, що є функціональним змістом модулів, представлено наступним чином. У розділі 3.2 представлені алгоритми, які розроблено для інструментальної підготовки даних. Вони включають генерацію каталогу колекції документів та завантаження загальнодоступних повнотекстових документів. У розділі 3.3 представлені алгоритми, розроблені для фази перед-обробки даних у робочому процесі. Вони призначені для перетворення PDF у плоский текст (plain text) та генерування наборів даних на основі визначених параметрів конфігурації. Оптимізований алгоритм здобуття термінів для статистичної частини конвеєру та алгоритм для обчислення об'єднаних часткових C-value детально описані у розділі 3.4. Базовий алгоритм для обчислення термінологічної різниці між двома наборами термінів представлено у розділі 3.5. У розділі 3.6 цей базовий алгоритм вдосконалюється шляхом врахування розробленої техніки групування термінів та алгоритмів вимірювання подібності символічних строк. Алгоритм видалення регулярного накопиченого шуму з наборів термінів наведено у розділі 3.7. У розділі 3.8 описується реалізація розроблених алгоритмів у програмному забезпеченні та

наводиться посилання на це програмне забезпечення, що є загальнодоступним для використання у академічних дослідженнях.

Завданням розділу 4 є третє завдання дослідження – експериментальна оцінка розробленого методу (розділ 2), матеріалізованого у наборі алгоритмів (розділ 3), для виявлення та вимірювання термінологічного насичення. У розділі 4.1 сформульовані завдання експериментального дослідження. План експериментів викладено у розділі 4.2. У розділах 4.3-4.7 повідомляються та обговорюються результати проведених експериментів. Як пояснюється у розділі 4, результати доводять правильність, незалежність від домену, ефективність та масштабованість розробленого обчислювального методу, набору алгоритмів та програмного забезпечення.

Розділ 5 виконує четверте завдання роботи – презентує досвід використання та візію того, як представлений науковий доробок доцільно впроваджувати в академічну та промислову практики. У розділі 5.1, представлено досвід використання розробленого програмного забезпечення в промисловому проєкті для крос-перевірки прогнозу Гартнер (Gartner) про тенденції впровадження технологій на прикладі технології генеративних змагальних мереж (Generative Adversarial Networks) у поглибленому машинному навчанні (Deep Learning). У розділі 5.2 повідомлено про академічне використання нашого методу та програмного забезпечення для проведення пошукових досліджень літератури студентами магістратури з метою написання оглядів релевантних джерел для своїх магістерських робіт. У розділі 5.3 узагальнено досвід практичного використання результатів роботи, представлений у розділах 5.1 та 5.2, шляхом висвітлення потенційних переваг розробленого методу та програмного забезпечення для промислових користувачів. У розділі 5.4, ми представляємо потенційні бізнес-сценарії щодо застосування результатів роботи у галузі наукового видавництва. Досвід та перспективи практичного використання результатів роботи підсумовано у розділі 5.5.

У розділі 6 представлені загальні висновки по роботі та плани щодо подальшого розвитку науково-технічного доробку.

Ключові слова: предметна область, колекція документів, термінологічне насичення, виявлення термінологічного насичення, вимірювання термінологічного насичення, послідовне наближення, обчислювальний метод, набір алгоритмів, ефективність, результативність.

ABSTRACT

Kosa, V. V. A Method of Experimental Study of Terminological Saturation in Document Collections for Knowledge Elicitation. PhD Thesis. Manuscript.

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy. Study program: 122 – Computer Science. Zaporizhzhia National University of the Ministry of Education and Science of Ukraine, Zaporizhzhia, 2021.

Object of research: the process of automated extraction, from the collections of relevant documents, of the sets of terms that characterize an arbitrary subject domain, for the further development of the ontologies for this domain, with an account for the phenomenon of terminological saturation.

Subject of research: a method of experimental study of terminological saturation in documents collections for knowledge elicitation to be represented with respect to an arbitrary domain of discourse.

The goal of this thesis is the improvement, in representativeness, efficiency, and effectiveness, of eliciting knowledge, from the collections of professional documents bounded for an arbitrary domain, for the further development of ontologies, by developing a complex computational method for detecting and measuring terminological saturation in the collections of professional textual documents that describe the domain.

The Introduction substantiates the topicality of the thesis, outlines its relationship to scientific and technical research projects. It formulates the research goal and objectives, specifies the object, subject, and methods of research, and highlights the scientific novelty and practical value of the obtained results. It sketches out how the research results were used in practical cases. Further, it summarizes the personal contribution of the applicant, and presents how the approbation and publication of the contributed results were done. Finally, the Introduction provides the quantitative information about the structure of the thesis.

Chapter 1 reviews the related work that helped develop the approach to model the process of terminological saturation, detect, and measure the result of this process. We start with the outline of the methodology used to select the literature sources for our

systematic literature review in Sect. 1.1. Further, in Sect. 1.2 to 1.11 we review and analyse the relevant State-of-the-Art in various research fields ranging from Ontology Engineering and Ontology Learning to Qualitative Research. The research gaps found in this analysis are summarized and our motivation to narrow these gaps explained in Sect. 1.12. Based on the outlined research gaps, Sect. 1.13 offers a roadmap for resolving the discovered open issues. It proposes a vision of and outlines the possible approach to the solution for narrowing the research gaps. Further, it formulates the research questions to be answered in the envisioned approach. Based on the research questions, it formulated the research objectives of the presented study. Sect. 1.14 concludes the chapter by summarizing the presented findings.

Chapter 2 focuses on the elaboration of the formal framework for terminological saturation measurement as a method. The chapter begins with the preliminaries (Sect. 2.1) that include the definitions helping us frame out theoretical focus. Next, we present the research hypotheses that need to be checked with regard to our theoretical framework. These hypotheses are formulated based on the research questions (Sect. 1.13). In Sect. 2.3 and 2.4, we focus on the formal introduction of the terminological difference function (*thd*) and the proof of its metric properties in the space of all possible document collections for a subject domain. In Sect. 2.5, we formulate and prove the theorem shaping the sufficient conditions for the existence of terminological saturation. Finally, in Sect. 2.6, we investigate if the method for measuring and detecting terminological saturation could be optimized in terms of lowering the incurred computational overhead. As a result, we elaborate the refined method that uses the partitioning of a document collection. We prove that the refined method yields the same *thd* measurement values, but takes substantially less time for computation. Furthermore, we argue that the refined method is straightforwardly parallelisable. Hence, in addition to its efficiency, it is free of the constraints inherent for any straightforward implementation of an incremental way of computing saturation measurements using the elaborated baseline method, (Sect. 2.1 to 2.5).

In Chapter 3, we develop new or improve the formerly developed algorithms that operationalize our formal framework (Chapter 2) for computation. We begin with the

elaboration of the processing pipeline in Sect. 3.1. For that, we propose the workflow for measuring terminological saturation. The workflow is further detailed in the functional block diagram that reveals the modular structure of our algorithmic suite and dependencies between its modules. The algorithms representing the modules are further developed as follows. Sect. 3.2 presents the algorithms developed to instrument data preparation. These include the generation of a document collection catalogue and the download of the publicly available full-text documents. Sect. 3.3 presents the algorithms developed for the pre-processing phase of the workflow. These are for PDF to plain text conversion and dataset generation based on the defined configuration parameters. The optimized algorithm for the statistical part of the term extraction pipeline and the algorithm for merging partial C-values are elaborated in Sect. 3.4. The baseline algorithm for computing terminological difference between two bags of terms is presented in Sect. 3.5. Sect. 3.6 refines this baseline algorithm by incorporating the developed term grouping technique, including the algorithms for string similarity measurement and the refined algorithm for measuring terminological difference. The algorithm for removing regular accumulated noise from the bags of terms is reported in Sect. 3.7. Finally, Sect. 3.8 outlines the implementation of the algorithmic suite in software and offers the links to this software, made publicly available for academic research use.

In Chapter 4, we present our experimental evaluation of the developed method (Chapter 2) and algorithmic suite (Chapter 3) for terminological saturation measurement. We start with formulating the experimental objectives in Sect. 4.1. We continue with the set-up of our experiments in Sect. 4.2. Sect. 4.3 to 4.7 report and discuss the results of the performed experiments. As explained in the Chapter, the results prove the correctness, domain neutrality, efficiency, and scalability of the developed method and algorithmic suite.

Chapter 5 describes our experience of the use and views on how the presented approach could be more broadly transferred into academic and industrial practice. In Sect. 5.1, we present our experience of exploiting the developed software pipeline in an industrial project for cross-checking the prognosis, by Gartner, on the trends of technology uptake with respect to the Generative Adversarial Networks technology of

Deep Learning. In Sect. 5.2, we report on the academic use of our method and software for instrumenting the exploratory research of Master students with their aim to write the reviews of the related work for their Master theses. In Sect. 5.3, we extend the use case experience, reported in Sect. 5.1 and 5.2 by outlining the practical benefits for the early industrial adopters of our method and software. In Sect. 5.4, we present our view on the potential business scenarios and applications in the scholarly publishing industry. The summary of the experience of and prospects for the practical use of our contribution is given in Sect. 5.5.

Finally, we conclude on our findings and present the plans for the future work in Chapter 6.

Keywords: subject domain, document collection, terminological saturation, terminological difference, terminological saturation detection, terminological saturation measurement, successive approximation, computational method, algorithmic suite, efficiency, effectiveness.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. **Kosa, V.**, Chugunenko, A., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., Birukou, A.: Semantic saturation in retrospective text document collections. CEUR-WS, vol. 1851, 1–8 (2017) ISSN: 1613-0073. SCOPUS
2. **Kosa, V.**, Chaves-Fraga, D., Naumenko, D., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., Birukou, A.: Cross-evaluation of automated term extraction tools by measuring terminological saturation. Revised selected papers of ICTERI 2017. Cham, Germany: Springer-Verlag, CCIS vol. 826, 135–163 (2018) doi: 10.1007/978-3-319-76168-8_7, ISSN: 1865-0929. SCOPUS
3. Chugunenko, A., **Kosa, V.**, Popov, R., Chaves-Fraga, D., Ermolayev, V.: Refining terminological saturation using string similarity measures. CEUR-WS vol. 2105 3–18 (2018) ISSN: 1613-0073. SCOPUS
4. **Kosa, V.**, Chaves-Fraga, D., Keberle, N., Birukou, A.: Similar terms grouping yields faster terminological saturation. Revised selected papers of ICTERI 2018. Cham, Germany: Springer-Verlag, CCIS vol. 1007, 43–70. (2019) doi: 10.1007/978-3-030-13929-2_3, ISSN: 1865-0929. SCOPUS
5. **Kosa, V.**, Chaves-Fraga, D., Dobrovolskyi, H., Fedorenko, E., Ermolayev, V.: Optimizing automated term extraction for terminological saturation measurement. CEUR-WS, vol. 2387, 1–16 (2019) ISSN: 1613-0073. SCOPUS
6. **Kosa, V.**, Chaves-Fraga, D., Dobrovolskiy, H., Ermolayev, V.: Optimized term extraction method based on computing merged partial C-values. Revised selected papers of ICTERI 2019. Cham, Germany: Springer-Verlag, CCIS vol. 1175, 24–49. (2020) doi: 10.1007/978-3-030-39459-2_2, ISSN: 1865-0929. SCOPUS
7. **Kosa, V.**, Ermolayev, V.: Toward a theoretical framework of terminological saturation for ontology learning from texts. CEUR-WS vol. 2566, 40–51 (2020) ISSN: 1613-0073. SCOPUS

ПОДЯКИ

Навчання на PhD програмі та дослідницьку роботу автора цієї дисертації було фінансово забезпечено стипендією від Міністерства освіти і науки України. Частково, виконання представлених у роботі досліджень було профінансовано стипендією Кабінету міністрів України. Автор є вдячною за державну підтримку її досліджень цими двома урядовими інституціями.

Дослідження, що призвели до написання цієї дисертації, частково виконувалися у кооперації між кафедрою комп'ютерних наук Запорізького національного університету (ЗНУ) Міністерства освіти і науки України та групи інжинірингу онтологій (OEG) Політехнічного університету Мадриду. Автор висловлює подяку колегам з OEG – Давіду Чавесу-Фрага, Карлосу Баденесу-Олмедо, проф. Оскару Корчо – за їх вагому допомогу у спільних дослідженнях та участь у підготовці публікацій.

Частина цих досліджень, на початковій фазі, виконувалася у рамках проекту SemData (<http://www.semdata-project.eu/>, grant agreement No PIRSES-GA-2013-612551) 7ї Рамкової програми Європейського Союзу.

Колекцію повних текстів журнальних статей у галузі менеджменту знань (Knowledge Management) було надано компанією Springer-Verlag GmbH для використання у дослідницьких цілях. Автор вдячна компанії за надання можливості використання цього цінного ресурсу. Автор також висловлює подяку др. Аляксандру Бірюкову за його підтримку цих досліджень та внесок в статті, що було видано в результаті плідної кооперації з Springer-Verlag GmbH.

Автор висловлює подяку усім колегам з кафедри комп'ютерних наук ЗНУ за їх щире допомогу та поради під час виконання досліджень та написання дисертаційної роботи. Особливо, автор є вдячною Генадію Добровольському за його допомогу в імплементації програмного забезпечення та Альоні Чугуненко, чий магістерський проект допоміг у розробці техніки групування подібних термінів.

Автор висловлює щире подяку Євгену Ющенко, Дмитру Науменко та усім іншим колегам з компанії ТОВ ГРУПБВТ, які допомогли у розробці деякого

спеціалізованого програмного забезпечення та впровадженні кейсу використання щодо перевірки прогнозу Гартнер.

Автор щиро цінує можливість використання розробленого методу та програмного забезпечення, що було надано міжнародною магістерською програмою з Комп'ютерних наук і науки про дані Українського католицького університету (УКУ) під керівництвом Олексія Молчановського. Вона також вдячна за надану УКУ фінансову підтримку її участі у симпозиумі з Advances in Data Mining, Machine Learning, and Computer Vision (MS-AMLV 2019), що проводився в УКУ.

Автор дуже радіє можливості висловити щирі подяку її PhD ментору, доц. Вадиму Анатолійовичу Єрмолаєву, який був її провідником по дослідницькому проекту та вклав значні зусилля щоб забезпечити його успішне завершення.

Автор щиро вдячна своїй родині за їх підтримку та терпіння у важкі часи навчання та роботи над цією дисертацією.

ЗМІСТ

АНОТАЦІЯ	3
ABSTRACT	8
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА.....	12
ПОДЯКИ	13
СПИСОК СКОРОЧЕНЬ.....	18
СПИСОК АЛГОРИТМІВ	22
СПИСОК РИСУНКІВ	23
СПИСОК ТАБЛИЦЬ.....	26
ВСТУП.....	28
1 СУЧАСНИЙ СТАН ДОСЛІДЖЕНЬ У КОНТЕКСТІ РОБОТИ.....	39
1.1 Методологія пошуку та відбору релевантних публікацій	39
1.2 Розробка онтологій домену та здобуття вимог	40
1.3 Вивчення онтологій з текстів та консенсус спільноти.....	41
1.4 Збір релевантних документів доброї якості	43
1.5 Термінологічне насичення та репрезентативність	44
1.6 Теоретичне насичення та вивчення онтологій.....	45
1.7 Впорядкування документів для обробки: відмітки часу та вплив	46
1.8 Автоматизовані методи здобуття термінів.....	48
1.9 Програмні реалізації методів АЗТ.....	52
1.10 Вимірювання подібності текстів	54
1.11 Ефективне співставлення строк для пошуку вкладених термінів	57
1.12 Прогалини у наявних результатах досліджень та мотивація	58
1.13 Питання та завдання дослідження.....	61
1.14 Висновок	70
2 ФОРМАЛЬНИЙ ФРЕЙМВОРК МЕТОДУ ВИЯВЛЕННЯ ТА ВИМІРЮВАННЯ ТЕРМІНОЛОГІЧНОГО НАСИЧЕННЯ	73
2.1 Базові відомості.....	73
2.2 Гіпотези дослідження	79

2.3 Функція термінологічної різниці (<i>thd</i>).....	80
2.4 Метричні властивості функції <i>thd</i>	83
2.5 Умови існування термінологічного насичення.....	87
2.6 Масштабованість та оптимізація	89
2.7 Висновок	94
3 АЛГОРИТМИ ДЛЯ ВИЯВЛЕННЯ ТА ВИМІРЮВАННЯ ТЕРМІНОЛОГІЧНОГО НАСИЧЕННЯ	97
3.1 Потік обчислень для виявлення та вимірювання термінологічного насичення	98
3.2 Підготовчі кроки та алгоритми.....	102
3.3 Кроки та алгоритми перед-обробки	104
3.4 Алгоритми оптимізованого обчислювального конвеєру	107
3.5 Базовий алгоритм вимірювання термінологічної різниці.....	107
3.6 Алгоритми групування термінів.....	108
3.7 Алгоритм видалення накопиченого регулярного шуму	115
3.8 Реалізація у програмному забезпеченні.....	116
3.9 Висновок	116
4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА РОЗРОБЛЕНОГО МЕТОДУ	118
4.1 Завдання експериментів	119
4.2 Загальні налаштування експериментів	120
4.3 Перевірка коректності методу на синтетичних колекціях	128
4.4 Вибір програмного забезпечення для АЗТ	133
4.5 Вплив впорядкування документів	140
4.6 Вплив групування термінів.....	155
4.7 Валідність та масштабованість оптимізованого конвеєру здобуття термінів.....	166
4.8 Висновок	175
5 ПРАКТИЧНЕ ВИКОРИСТАННЯ РОЗРОБЛЕНОГО МЕТОДУ	177
5.1 Перевірка технологічного прогнозу Гартнер за допомогою термінологічного аналізу	177

5.2 Інструментальне забезпечення виконання магістрами огляду літератури	182
5.3 Практичні переваги	185
5.4 Потенційні сценарії застосувань у галузі наукового видавництва	187
5.5 Висновок	188
6 ВИСНОВКИ	190
6.1 Підсумки науково-технічного доробку	191
6.2 Напрямки подальшої роботи	193
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	195
ДОДАТКИ	210
Додаток А. Сертифікати та дипломи, що підтверджують апробацію результатів	210
Додаток Б. Приклад згенерованого каталогу колекції документів	215
Додаток В. Алгоритми, що імплементують обчислювальний метод	216
Додаток Г. Характеристики модулів розробленого програмного забезпечення	230
Додаток Д. Довідки про впровадження розробленого методу та програмного забезпечення	233
Додаток Е. Питання та метод перевірки прогнозу Гартнер	235
Додаток Ж. Публікація, апробація та використання результатів роботи ...	240

СПИСОК СКОРОЧЕНЬ

- 1DOC – (one document collection) колекція з одного документу
- ANSI – (American National Standards Institute) Американський інститут національних стандартів
- API – (Application Programming Interface) інтерфейс прикладного програмування
- ARN – (accumulated regular noise) накопичений регулярний шум
- ASCII – (American Standard Code for Information Interchange) Американський стандартний код для інформаційного обміну
- ATF – (average TF) середня частота терміну
- B – (bag of terms) набір термінів
- BPMN – (Business Process Model та Notation) модель і нотація опису бізнес процесу
- CC BY – Creative Commons License Attribution International
- CDC – (complete document collection) повна колекція документів
- CDS – (documents space) простір документів
- c.f. – (лат.: confer/conferatur) порівняйте з
- CSO – (Computer Science Ontology) онтологія комп'ютерних наук
- CSS – (Characteristic Scores and Scales) характеристичні бали та шкали
- CVM – (the method for computing C-value) метод для обчислення C-value
- DAC – (Design Automation Conference) конференція з автоматизації проектування
- DC – (document collection) колекція документів
- DMKD – (Data Mining and Knowledge Discovery) майнінг даних та виявлення знань
- DO – (document ordering) впорядкування документів:
- bd – (bi-directional ordering) двоспрямоване впорядкування
- ch – (chronological ordering) хронологічне впорядкування
- dcf – (descending citation frequency ordering) впорядкування за зменшенням частоти цитування

rch	– (reversed chronological ordering) зворотнє хронологічне впорядкування
rn	– (random ordering) випадкове впорядкування
Doc	– (document) документ
DSC	– (document sub-collection) підколекція документів
DOI, doi	– (digital object identifier) ідентифікатор цифрового об'єкту
Dom	– (domain) домен, предметна область
DP	– (document collection partitioning) розділення колекції документів на частини
eps	– (individual term significance threshold) індивідуальний поріг значущості терміну
FN	– (full negative) повністю негативний
FP	– (full positive) повністю позитивний
FSM	– (Final State Machine) скінчений автомат
GAN	– (Generative Adversarial Network) Генеративна змагальна мережа
GmbH	– (Gesellschaft mit beschränkter Haftung) товариство з обмеженою відповідальністю (нім.)
GPU	– (Graphical Processing Unit) графічний процесор
INC, inc	– інкремент, приріст
ISSN	– (International Standard Serial Number) міжнародний стандартний періодичний номер
IT	– (Information Technology) інформаційна технологія
KM	– (Knowledge Management) управління знаннями
LLC	– (Limited Liability Company) товариство з обмеженою відповідальністю (англ.)
MPCV	– (Merged partial C-value) злита часткова C-value
MSR	– Microsoft Research
NaCTeM	– (National Centre for Text Mining) Національний центр з майнінгу текстів
ns	– (normalized significance score) нормалізований показник значущості

OEG	– (Ontology Engineering Group) група інжинірингу онтологій
PC	– (personal computer) персональний комп'ютер
PCV	– (partial C-value) часткова C-value
PDF	– (portable data format) переносний формат даних
PhD	– (philosophy doctor) доктор філософії
PN	– (partial negative) частково негативний
POS	– (part of speech) частина мови
PP	– (partial positive) частково позитивний
R&D	– (research and development) дослідження та розробка
RAW	– (random articles from Wikipedia) випадкові статті з Вікіпедії
RIDF	– (residual IDF) залишковий IDF
\mathbb{R}^+	– (set of real positive numbers) множина дійсних додатних чисел
SAGOIT-IT	– Strategic Analysis of R&D Gaps and Opportunities for Industrial Uptake in Trending IT Fields
SotA	– (State-of-the-Art) сучасний стан досліджень
SotT	– (State-of-the-Technology) сучасний стан технології
STG	– (similar term grouping) групування подібних термінів
SVM	– (Support Vector Machine) підтримка векторної машини
T	– (bag of retained significant terms) набір збережених значущих термінів
TB	– (terminological basis) термінологічний базис
TCSC	– (terminological core subcollection) підколекція термінологічного ядра
TF	– (term frequency) частота терміну
TF-IDF	– (TF with inverse document frequency) TTF у поєднанні з оберненою частотою документу
thd	– функція термінологічної різниці
TSM	– (term similarity measure) міра подібності терміну
TST	– (term similarity threshold) поріг подібності термінів
TTF	– (total TF) загальна частота терміну
TXT	– (plain text format) формат плоского тексту

UNICODE	– (UNified CODing) УНІфіковане КОДування – промисловий стандарт
UPM	– (Universidad Politécnica de Madrid) Політехнічний університет Мадриду
URL	– (Unified Remote Locator) уніфікований віддалений локатор
UTG	– (use of terms grouping) використання групування термінів
VSM	– (vector space model) модель векторного простору
A3T	– автоматизоване здобуття термінів (automated term extraction)
ЗД	– завдання дослідження (research objective)
ЗНУ	– Запорізький національний університет
МСП	– міра строкової подібності (string similarity measure)
НП	– прогалина у наявних результатах (research gap)
ПД	– питання дослідження (research question)
УДК	– універсальна десяткова класифікація
УКУ	– Український католицький університет

СПИСОК АЛГОРИТМІВ

Алг. 3.1. Алгоритм GСAT генерації каталогу колекції документів.....	216
Алг. 3.2. Алгоритм DDL для завантаження повнотекстових документів з репозиторію документів	217
Алг. 3.3. Алгоритм PDF2ТХТ для здобуття тексту з набору повнотекстових файлів PDF у звичайні текстові файли.	218
Алг. 3.4. Алгоритм GDS для генерації набору даних для поточної ітерації потоку обчислень.	219
Алг. 3.5. Алгоритм AC-CV для оптимізованого обчислення C-values за допомогою Aho-Corascik Corascik (Aho and Corasick 1975) для швидкого співставлення декількох строк.....	220
Алг. 3.6. Алгоритм MPCV для злиття часткових C-value з двох наборів термінів	221
Алг. 3.7. Базовий алгоритм THD, адаптований з (Tatarintseva et al. 2013).....	222
Алг. 3.8. Алгоритм групування подібних термінів (STG).....	223
Алг. 3.9. Алгоритм (M-JR) для обчислення міри строкової подібності Жаро	224
Алг. 3.10. Алгоритм (M-JW) для обчислення міри строкової подібності Жаро-Вінклера.....	225
Алг. 3.11. Алгоритм (M-JA) для обчислення міри строкової подібності Жакара	226
Алг. 3.12. Алгоритм (M-SD) для обчислення міри строкової подібності Соренсена-Дайса.....	227
Алг. 3.13. Вдосконалення (R-THD) базового алгоритму THD (розділ 4.5).....	228
Алг. 3.14. Алгоритм ARNR для видалення строк ARN з наборів термінів	229

СПИСОК РИСУНКІВ

Рис. 1.1. Візія методу здобуття насиченого набору термінів з колекції документів, що є релевантними для опису домену.....	62
Рис. 1.2. Зміна інкрементального збільшення наборів даних на злиття наборів термінів для кращої масштабованості та підвищення продуктивності АЗТ	67
Рис. 3.1. Робочий процес для виявлення та вимірювання термінологічного насичення. Активності пофарбовані у білий колір повністю автоматизовані, світло-сірі вимагають взаємодії з користувачем, темно-сірі виконуються вручну.	99
Рис. 3.2 Приклад автоматично зформованого звіту про термінологічну різницю для послідовності інкрементально збільшених наборів даних.	100
Рис. 3.3. Структурована блок-схема обчислень у процесі здобуття насиченої термінології. Світло-сірі модулі та підпроцеси вимагають взаємодії з користувачем або виконуються частково вручну. Темно-сірі модулі та підпроцеси виконуються вручну.	101
Рис. 4.1. Експериментальний робочий процес. Необов'язкові завдання виділено сірим кольором.	121
Рис. 4.2. Розподіл статей у журналах колекції КМ (Kosa et al. 2017a). Вісь Y представляє роки публікації, а вісь X відповідає журналам (скорочення). Цифри в стовпцях: кількість томів, кількість випусків, та загальна кількість статей.....	125
Рис. 4.3. Візуалізація вимірювань насичення на наборах даних 1DOC	129
Рис. 4.4. Порівняння збережених наборів термінів, здобутих з колекції 1DOC програмами UPM Extractor та TerMine	130
Рис. 4.5. Візуалізація вимірювань насичення на наборах термінів колекції RAW здобутих за допомогою TerMine. Діаграма праворуч представляє більш детальний вигляд закругленого прямокутника на діаграмі зліва.	131
Рис. 4.6. Візуалізація вимірювань насичення на наборах термінів колекції RAW здобутих за допомогою UPM Extractor.	133
Рис. 4.7. Вимірювання насиченості в наборах даних DMKD на основі наборів термінів, здобутих за допомогою TerMine	134
Рис. 4.8. Вимірювання насиченості в наборах даних DMKD на основі наборів термінів, здобутих за допомогою UPM Extractor.....	135
Рис. 4.9. Порівняння збережених наборів термінів, здобутих з колекції DMKD за допомогою UPM Extractor та TerMine	136
Рис. 4.10. Вимірювання насичення для колекції TIME	137
Рис. 4.11. Вимірювання насичення для колекції DAC	138

Рис. 4.12. Індивідуальні пороги значущості термінів (<i>eps</i>) для різних впорядкувань додавання документів до наборів даних.	147
Рис. 4.13. Термінологічні відмінності (<i>thd</i>) для різних впорядкувань додавання документів до наборів даних.	148
Рис. 4.14. Середня частка збережених до всіх здобутих термінів для різних впорядкувань додавання документів до наборів даних.	149
Рис. 4.15. Інтегральні волатильності <i>thd</i> у зонах насичення для різних впорядкувань додавання документів до наборів даних. Ці ранги базуються на сумах значень точкової волатильності у відповідних зонах насичення. Найменш волатильне впорядкування є найкращим.	150
Рис. 4.16. Приклад накопичення регулярного шуму в наборах збережених значущих термінів, здобутих з DAC Naturelle з використанням впорядкування dcf : (а) верхня частина <i>T4</i> , в якій регулярний шум, хоча і присутній (підкреслений), ще не накопичений; (б) верхня частина <i>T5</i> в якій накопичився регулярний шум. На (<i>D5, D4</i>) спостерігається пік <i>thd</i> – збільшення з 65.42 до 149.31.....	152
Рис. 4.17. Чутливість до накопичення регулярного шуму з використанням різних порядків додавання документів до наборів даних. Впорядкування вказуються легендами, сполученими до кривих стрілками.	152
Рис. 4.18. Вимірювання термінологічного насичення на TIME.....	157
Рис. 4.19. Вимірювання термінологічного насичення на DMKD.....	157
Рис. 4.20 Вимірювання термінологічного насичення на DAC Cleaned.....	158
Рис. 4.21. Час (сек) витрачений алгоритмом STG для групування подібних термінів на наборах термінів TIME	159
Рис. 4.22. Час (сек) витрачений алгоритмом STG для групування подібних термінів на наборах термінів DMKD.....	159
Рис. 4.23. Час (сек) витрачений алгоритмом STG для групування подібних термінів на наборах термінів DAC Cleaned	160
Рис. 4.24. Пропорції збережених значущих до всіх здобутих термінів для різних мір подібності термінів по кожній колекції	160
Рис. 4.25. Оцінка реалізації вдосконаленого R-THD з різними МСП при <i>th</i> = 1.00 на наборах термінів DMKD. Вертикальні штрихові лінії позначають точку термінологічного насичення.	162
Рис. 4.26. Показники продуктивності оцінених МСП по пороговим значенням подібності термінів з (а) та без (б) урахування часу виконання. Точки в пунктирних прямокутниках представляють середні значення для всіх порогів.	165
Рис. 4.27. Послідовність виконання оцінюваних експериментів.....	169

Рис. 4.28. DMKD: Злиті часткові C-value обчислені оптимізованим конвеєром практично однакові з C-value обчисленими за допомогою базового конвеєру....	171
Рис. 4.29. Час виконання базового (Incremental) та оптимізованого (Merged Partial) конвеєрів.....	171
Рис. 4.30. Криві термінологічної різниці, обчислені для наборів термінів зі звичайно обчисленими C-value (Incremental) та зі злитими частковими C-value (Merged Partial) практично однакові. Результати для колекції TIME.....	172
Рис. 4.31. Динаміка впливу регулярного шуму на різницю між T_i та T_{im}	173
Рис. 4.32. Накопичені хибно позитивні елементи (регулярний шум, виділено сірим кольором) що спостерігаються у найбільшому наборі термінів, здобутому за допомогою оптимізованого конвеєру ($B22m$). Різниці ($ cv - mpcv $) для хибно позитивних елементів у середньому вищі, ніж для правдиво позитивних елементів.	173
Рис. 4.33. Вимірювання насичення для колекції КМ.....	174
Рис. 5.1. Вимірювання термінологічного насичення: thd (суцільна крива) по відношенню до eps (пунктирна крива).....	181
Рис. 5.2. Вимірювання волатильності термінологічних різниць	181
Рис. 5.3. Криві термінологічної різниці для пар частин колекції: thd-a-c – для академічної та спільної; thd-a-i – для академічної та промислової; thd-c-i – для спільної та промислової; eps – крива максимального індивідуального порогу значущості терміну.	181
Рис. Г.1. Процес вимірювання термінологічної різниці для виявлення насичення в окремій частині колекції.....	237
Рис. Г.2. Обчислювальний процес для оцінки термінологічних різниць між наборами збережених значущих термінів різних частин колекції.....	238

СПИСОК ТАБЛИЦЬ

Таблиця 1.1. Порівняння продуктивності найбільш широко використовуваних методів АЗТ на основі еталонних реалізацій та експериментів, опублікованих у (Astrakhantsev 2016; Zhang et al 2008)	50
Таблиця 1.2. Вільнодоступні програмні засоби АЗТ (перелічені в алфавітному порядку).....	53
Таблиця 1.3. Огляд вимірювань текстової подібності/відстані	55
Таблиця 1.4. Порівняння продуктивності алгоритмів, релевантних для пошуку вкладених термінів, за (Chowdhury and Farrell 2019). Для порівняння було використано асимптотичне позначення $O(.)$	57
Таблиця 1.5. Завдання дослідження, питання дослідження та відповіді	70
Таблиця 3.1. Атрибути документів у каталозі колекції	102
Таблиця 3.2. Параметри конфігурації для конвеєру здобуття насиченої термінології.....	105
Таблиця 3.3. Міри подібності для різних тестових випадків	113
Таблиця 3.4. Середні значення вимірювань подібності для різних категорій пар термінів з тестового набору	114
Таблиця 3.5. Пороги подібності термінів, що вибрані для експериментальної оцінки.....	114
Таблиця 4.1. Особливості використовуваних колекцій документів та наборів даних	126
Таблиця 4.2. Вимірювані аспекти, використані в експериментах	126
Таблиця 4.3. Порівняння вимірювань насичення у їх точках насичення для всіх порядків	146
Таблиця 4.4. DAC Naturelle. Порівняння показників шуму для всіх порядків ...	151
Таблиця 4.5. Узагальнення крос-оцінювання різних впорядкувань додавання документів до наборів даних на основі зазначених вимірюваних аспектів.....	154
Таблиця 4.6. Рейтинг оцінюваних МСП.....	163
Таблиця 4.7. Показники продуктивності оцінених МСП з урахуванням та без урахування часу виконання.....	164
Таблиця 4.8. Вимірювання <i>thd</i> та часу виконання для звичайного и оптимізованого конвеєрів (колекція DMKD).....	170
Таблиця 6.1. Підсумок зменшення обсягів колекцій документів та здобутих термінів.....	192

Таблиця Д.1. Повнота публікації результатів дослідження та цитування.....	241
Таблиця Д.2. Результати у загальнодоступних технічних звітах	242
Таблиця Д.3. Результати, що використані організаціями та приватними особами	243

ВСТУП

Дисертаційну роботу присвячено вирішенню актуальної науково-практичної задачі – розробці ефективного та результативного комплексного обчислювального методу для експериментального дослідження колекцій професійних документів у межах довільної предметної області (домену) з метою виявлення та вимірювання термінологічного насичення.

Обґрунтування вибору та актуальності теми дослідження. Наведемо пояснення про мотивацію вибору теми роботи, та її актуальність для науки та практики. Для цього представимо причини необхідності виявлення репрезентативних колекцій документів для здобуття знань у предметній області. Таке виявлення безпосередньо зв'язане з виявленням термінологічного насичення. Потім більш детально викладемо, як виявлення репрезентативних піднаборів підвищує ефективність аналізу великих даних та результативність здобуття потреб для побудови онтологій. Також представимо феномен насичення у більш широкому контексті різних галузей знань та вкажемо місце доробку цієї дисертації у наведеному контексті.

Актуальність виявлення та вимірювання термінологічного насичення. Колекції професійних документів, що описують ту чи іншу предметну область, все частіше використовуються для здобуття з них знань для побудови описових теорій (онтологій) для предметної області. Проблема у цьому контексті полягає у відсутності об'єктивного методу забезпечення репрезентативності вихідної колекції документів. Не можна гарантувати достатнє покриття онтологією цільового домену, якщо вихідні тексти не достатньо повно описують цей домен – тобто не є репрезентативними. Відсутність таких гарантій приводе до того, що онтології нечасто вважаються повними та якісними. Тому, їх впровадження у промисловість та академічні практики є не дуже розповсюдженим.

Ідея роботи базується на тому, що семантичним відбитком документу, у деякій професійній області, є набір використаних документом термінів. Тому, семантичним відбитком колекції документів є набір термінів, здобутих з усіх її документів. Виходячи з цього, колекція буде повно описувати домен тоді, коли

вона містить підколекцію з тим самим набором термінів, що і уся колекція. Це явище буде свідчити, що є частина колекції мінімального обсягу у якої спостерігається термінологічне насичення, тобто вона є репрезентативною щодо опису домену. Виявлення та вимірювання цього термінологічного насичення важливо тому, що: (i) дає змогу відокремити репрезентативну підколекцію мінімального обсягу, якщо вона існує, що гарантує якісний результат у подальшому процесі вивчення онтології з текстів; та (ii) підвищує ефективність здобуття термінів та знань, тому що реальні колекції документів для професійних предметних областей є дуже великими.

Актуальність виявлення репрезентативних піднаборів великих даних.

Сьогодні, метод та його програмна реалізація, що допомагають робити правильні і своєчасні висновки та рекомендації на основі аналізу великих вільно доступних даних, що містять професійні тексти у будь-якій галузі знань або домені, користуються значним попитом та знаходяться на стадії інтенсивного дослідження та розробки. Як зазначено у (Ermolayev et al. 2013), однією з проблем на цьому шляху є вирішення внутрішнього протиріччя між ефективністю та результативністю підходу. Складність завдання посилюється за декількома аспектами: розміру даних, об'єму даних, ефективності та результативності обробки даних, різноманітності та складності даних, правдивості даних, які повинні бути збалансовано прийняті до уваги. Наприклад, зі збільшенням об'єму ефективність знижується але результативність може зрости. На відміну від об'єму, збільшення ефективності знижує результативність.

У галузі здобуття інформації (Information Retrieval), вищезазначене протиріччя частково фіксується мірами точності та повноти (Baeza-Yates and Ribeiro-Neto 1999). Дійсно, зі зростанням кількості даних, повнота може збільшуватися, оскільки у більшій вибірці може бути більше істинно позитивних випадків. Однак точність може зменшитися, оскільки в більшій вибірці, разом із істинно позитивними випадками, може бути більше помилкових (хибно позитивних) випадків. Точність та повнота часто врівноважуються за допомогою *F*-міри (Baeza-Yates and Ribeiro-Neto 1999). На додаток до використання цієї

відомої збалансованої міри, відповідь на наступне слушне питання буде актуальною: яка вибірка даних мінімального розміру містить усі (суттєві) істинно-позитивні випадки? Відповідь на це питання дає можливість використовувати найбільш ефективний та результативний (та масштабований) метод здобуття інформації, оскільки він дозволяє відфільтрувати усі надлишкові дані. Така вибірка даних мінімального розміру є репрезентативною у контексті відповідного запиту на здобуття інформації.

Актуальність вирішення проблеми репрезентативності у побудові онтологій. Більш складне питання виникає, якщо ми розширимо сферу запиту з конкретного питання до певного обговорюваного домену: чи існує вибірка даних мінімального розміру для домену, яка є репрезентативною для будь-якого відповідного запиту в межах цього домену? Питання є складним, оскільки ми повинні знати всі істинно позитивні результати будь-якого запиту про домен. Щоб відповісти на це питання, потрібно побудувати описову теорію домену, яка відображає вимоги всіх, хто запитує інформацію в цьому домені. Одним із способів побудови такої теорії є розробка онтології домену, яка є достатньо повною для використання спільнотою стейкхолдерів знань у цьому домені. Отже, початкове питання трансформується в таке: як забезпечити повноту розробленої онтології домену?

Основною практикою в галузі розробки онтологій (Ontology Engineering) є збір релевантних компетентнісних питань (competency questions) шляхом організації систематичних інтерв'ю або проведення мозкових штурмів із групами експертів, обраних зі стейкхолдерів знань (див. Sure et al. 2003; Gómez-Pérez et al. 2004; Pinto et al. 2004; Schreiber et al. 1999; Suárez-Figueroa et al. 2012). Відповіді на компетентнісні питання пізніше використовуються як вимоги для побудови онтології. Однак, проблема полягає в тому, що ці методології не гарантують того, що обрана група експертів в достатній мірі повно віддзеркалює думку спільноти. Отже, зібраний набір вимог не може розглядатися як репрезентативний.

Порівняно більш сучасною тенденцією побудови описових теорій доменів є вивчення онтологій з текстів (Wong et al 2012). У цій галузі досліджень вимоги до

побудови онтологій здобуваються (вивчаються) з набору професійних документів, що описують домен, авторами яких є стейкхолдери знань у цьому домені. У даному контексті визнається, що вимоги є пов'язаними з термінами, що описують домен. Отже, якщо є репрезентативний набір релевантних термінів, то вимоги до повної онтології можуть бути розгорнуті з цих термінів. Отже, необхідним є метод для здобуття репрезентативного набору термінів, що описують домен. Для здобуття репрезентативного набору термінів повинна бути наявною термінологічно повна колекція документів, що описує домен.

У цій роботі ми пропонуємо такий метод, що дозволяє автоматично здобувати репрезентативний набір термінів мінімального розміру та формувати термінологічне ядро колекції документів мінімального розміру, що містить усі такі терміни. Для цього ми вводимо поняття досягнення термінологічної насиченості як процесу послідовного наближення, що веде до отримання термінологічного ядра колекції документів для довільного домену.

Феномен насичення. Насичення, як зазначено у Оксфордському Словнику (Oxford Learners Dictionaries)¹, це “стан або процес який відбувається, коли більше чогось не можна прийняти або додати, оскільки його вже досить багато або занадто багато”. Цей термін широко застосовується для позначення явищ у матеріальному та нематеріальному світах. Можна навести декілька прикладів насичення з різних галузей знань:

- У хімії насичення означає “ступінь, в якій щось поглинається чим-то іншим, виражене у відсотках від максимально можливого”¹
- У магнетизмі це означає “стан, коли магнітний матеріал є повністю намагнічений”²
- У термодинаміці, це вказує на “термодинамічний стан при нижній температурній межі перегрітої пари”²

¹ https://www.oxfordlearnersdictionaries.com/definition/american_english/saturation

² <https://en.wikipedia.org/wiki/Saturation>

- У комутативній алгебрі, “насичення відносно f ідеалу I у R є зворотнім зображенням $R_f I$ цього ідеалу при канонічному відображенні з R у R_f . Цей ідеал, що складається з усіх елементів R , добутки яких з якимось ступенем f належать I .”³

- У економіці, насичення ринку – це стан ринку, у якому продукт розподіляється на ринку до рівня природного споживання (Osenton 2004)

- У генетиці, насичення це “результат багаторазових заміщень у одній послідовності, або ідентичних заміщень у різній послідовності, так що очевидна швидкість дивергенції послідовностей нижча, від фактичної дивергенції” (Brinkmann et al. 2011)

- У якісних дослідженнях (Qualitative Research), насичення даними означає фазу якісного аналізу даних до якої дослідник продовжує вибірку та аналіз даних, тому що з’являються нові дані, що впливають на поліпшення якості розробки усіх концепцій теорії ... та чіткості опису їх зв’язків з іншими концепціями (Morse 2004)

У цій роботі ми вивчаємо термінологічне насичення як феномен в інкрементально зростаючій послідовності підколекцій документів повної колекції. Ми моделюємо процес термінологічного насичення як процес послідовного наближення. Ми експериментально знаходимо та формально доводимо, що за певних умов, термінологічне насичення призводить до насиченої підколекції документів, якщо тематика документів обмежена доменом. Ця насичена підколекція, яку ми називаємо підколекцією термінологічного ядра, є так само репрезентативною, як і вся колекція, що відображає думку стейкхолдерів знань щодо домену. Отже, набір термінів, здобутих з підколекції термінологічного ядра, можна розглядати як мінімальний репрезентативний набір термінів для подальшого виявлення вимог у конвеєрі вивчення онтології домену (domain ontology learning pipeline).

³ https://en.wikipedia.org/wiki/Gr%C3%B6bner_basis#Definition_of_the_saturation

Зв'язок роботи з науковими програмами, планами, темами. Дисертація є результатом участі автора у виконанні науково-дослідної тематики кафедри комп'ютерних наук Запорізького національного університету Міністерства освіти і науки України. Протягом навчання в аспірантурі та виконання дисертаційної роботи здобувач брав участь у наступних науково-дослідних проектах:

1. Проекті EU FP7 Marie Curie IRSES SemData project (<http://www.semdata-project.eu/>), grant agreement No PIRSES-GA-2013-612551. Строки виконання проекту: 10.2013 – 10.2017. Науковий керівник від України: к.ф.-м.н., доц. Єрмолаєв В. А. Участь здобувача – виконавець.

2. Промисловому проекті Strategic Analysis of R&D Gaps and Opportunities for Industrial Uptake in Trending IT Fields (SAGOIT-IT), фінансованому компанією ТОВ ГРУПБВТ. Строки виконання проекту: 09.2020 – 01.2021. Науковий керівник: к.ф.-м.н., доц. Єрмолаєв В. А. Участь здобувача – виконавець.

Об'єкт, предмет, мета і завдання дослідження.

Об'єктом дослідження є процес автоматизованого здобуття, з колекцій релевантних документів, наборів термінів, що характеризують довільну професійну предметну область, для подальшої побудови онтологій цієї предметної області, з урахуванням впливу явища термінологічного насичення.

Предметом дослідження є метод експериментального дослідження термінологічного насичення в колекціях документів для здобуття знань у процесі вивчення онтологій довільного домену.

Метою роботи є підвищення репрезентативності, ефективності та результативності здобуття термінології з колекцій професійних документів у межах довільної предметної області для подальшої побудови онтологій, шляхом розробки комплексного обчислювального методу виявлення та вимірювання термінологічного насичення в колекціях професійних текстових документів, що описують предметну область.

Для досягнення мети в роботі, на основі систематичного огляду та аналізу літературних джерел, що розкривають сучасний стан досліджень в обраній області (Розділ 2), поставлені та вирішені наступні **завдання**:

ЗД1: розробити формальний фреймворк методу виявлення та вимірювання термінологічного насичення, що охоплює визначення, формалізми та докази всіх його компонентів, необхідних для побудови якісного, ефективного та результативного обчислювального методу.

ЗД2: спираючись на розроблений формальний фреймворк, **розробити алгоритми** що матеріалізують **обчислювальний метод** для здобуття насиченої термінології; реалізувати ці алгоритми у програмному забезпеченні та надати програмне забезпечення для академічного та промислового використання.

ЗД3: використовуючи розроблене програмне забезпечення, **експериментально оцінити та перевірити розроблений обчислювальний метод** виявлення та вимірювання термінологічного насичення на синтетичних та реальних колекціях наукових публікацій, що належать до різних доменів.

ЗД4: проаналізувати, якими є потенційні **практичні наслідки (переваги) використання** розробленого методу на основі досвіду реалізації кейсів практичного використання в промисловості та академічній сфері; проаналізувати потенційні переваги впровадження науково-технічного доробку та потенційні бізнес-сценарії у цільовій промисловій галузі наукового видавництва.

Методи дослідження. Досягнення мети та виконання завдань дисертаційного дослідження здійснюється за допомогою комплексного використання: наукового методу; формальних математичних методів; методів та формальних мов специфікації обчислювальних процесів та алгоритмізації; методів та мов програмування; методів обробки текстових даних; методів та мір аналізу текстових даних і термінів; методів планування, виконання та аналізу обчислювальних експериментів.

Наукова новизна отриманих результатів.

Уперше:

1. Розроблено **комплексний обчислювальний метод виявлення та вимірювання термінологічного насичення** у послідовності інкрементально зростаючих підколекцій гіпотетично існуючої повної колекції професійних документів, що описують довільний домен. Розроблений метод є різновидом

методу послідовного наближення у Гільбертовому просторі колекцій документів, що є підмножинами повної колекції. Доведено теорему існування термінологічного насичення, що надає достатні умови існування. Розроблено алгоритми та програмне забезпечення для імплементації цього обчислювального методу.

Отримали подальший розвиток:

2. Формальне визначення **міри термінологічної різниці** між двома наборами термінів зі значеннями оцінок їх значущості на базі (Tatarintseva et al. 2013) – міра формально визначена як різновид відстані Манхеттен (Goma and Fahmy 2013); формально доведено її метричні властивості.

3. Обчислювальний **метод автоматичного здобуття термінів** на базі методу C-value (Frantzi and Ananiadou 1999) – запропоновано, замість обчислення C-value термінів, здобутих з усієї підколекції документів, обчислювати часткові C-value, здобуті з інкременту колекції документів, і потім зливати часткові C-value. Доведено, що злиті часткові C-value практично не відрізняються від C-value, обчислених за (Frantzi and Ananiadou 1999) – тобто модифікація методу не зменшує якість здобуття термінів. Показано, що модифікований таким чином метод є коректним, ефективним (за показником часу виконання), таким що очевидним чином розпаралелюється, та результативним з точки зору масштабованості для обробки текстів будь-якого великого обсягу.

Удосконалено:

4. Розроблений **обчислювальний конвеєр виявлення, вимірювання та аналізу термінологічного насичення**, що використовує розроблений метод виявлення та вимірювання термінологічного насичення та поліпшений метод автоматичного здобуття термінів з використанням злитих часткових C-value шляхом: залучення обчислювального методу для **відбору релевантних документів** до інкрементів колекції (Dobrovolskyi and Keberle 2018); використання розробленої техніки та алгоритмів **групування частково подібних термінів; впорядкування документів** для формування інкрементів колекції за зменшенням частоти цитування документів. За рахунок цих вдосконалень було підвищено якість та ефективність розробленого обчислювального методу.

Практичне значення отриманих результатів. Практичне значення дисертаційної роботи полягає у використанні її результатів, що підтверджується (Додаток Д) двома кейсами практичного використання у промисловості (розділ 6.1) та академічній сфері (розділ 6.2):

1. У проєкті SAGOIT-IT: Strategic Analysis of R&D Gaps and Opportunities for Industrial Uptake in Trending IT Fields компанії ТОВ ГРУПБВТ. У цьому проєкті розроблений вдосконалений обчислювальний конвеєр було використано для перевірки прогнозу Гартнер щодо перспектив впровадження технології генеративних змагальних мереж поглибленого навчання в промисловість шляхом аналізу термінологічного насичення.

2. У міжнародній магістерській програмі з Комп'ютерних наук та науки про дані Українського католицького університету. У цій програмі розроблений вдосконалений обчислювальний конвеєр було використано як програмний інструмент, що був рекомендований студентам на курсі «Академічне письмо» для відбору репрезентативного набору релевантних літературних джерел для написання огляду і аналізу сучасного стану досліджень за темами їх магістерських робіт.

Особистий внесок здобувача. Наукові положення і результати, що представлені в дисертаційній роботі, отримані здобувачем особисто. У наукових публікаціях результатів дисертаційної роботи, написаних у співавторстві, здобувачеві належить:

- (Kosa et al. 2017a): план дисертаційного проєкту; візія підходу щодо виявлення та вимірювання термінологічного насичення в колекціях професійних документів; зібрані та предоброблені колекції документів

- (Kosa et al. 2018a): збір та підготовка документів для синтетичних колекцій; експериментальна крос-оцінка програмних засобів для здобуття термінів; вибір базового методу та програмного засобу для автоматичного здобуття термінів за результатами крос-оцінки

- (Chugunenko et al. 2018, Kosa et al. 2019a): алгоритми мір подібності, пороги подібності, алгоритм групування термінів **STG**, вдосконалений алгоритм **R-THD**;

експериментальна оцінка впливу групування подібних термінів на термінологічне насичення

- (Kosa et al. 2019b): оптимізований метод та алгоритми для обчислення злитих часткових C-value; доведення теореми про тотожність MPCV та C-value; експериментальна перевірка коректності методу
- (Kosa et al. 2020): Експериментальна перевірка незалежності від домену та результативності методу для колекції великого промислового обсягу
- (Kosa and Ermolayev 2020): формулювання ключових тверджень формального фреймворку про метричні властивості функції *thd* та достатні умови існування термінологічного насичення

Апробація матеріалів дисертації. Результати досліджень доповідались (Додаток А) і були схвалені на:

- PhD симпозиумі 13ї міжнародної конференції ICT in Education, Research, and Industrial Applications (ICTERI 2017), Київ, 2017 р. – доповідь відзначена як найкраща
- 14їй міжнародній конференції ICT in Education, Research, and Industrial Applications (ICTERI 2018), Київ, 2018 р.
- 15їй міжнародній конференції «ICT in Education, Research, and Industrial Applications (ICTERI 2019), Херсон, 2019 р.
- Обласному конкурсі молодих науковців Запорізької обласної державної адміністрації «Молода наука», Запоріжжя, 2019 – визнана переможцем
- 1му симпозиумі Masters Symposium on Advances in Data Mining, Machine Learning, and Computer Vision (MS-AMLV 2019), Львів, 2019 р.

Публікація результатів дисертації. Основні наукові результати досліджень опубліковані в 7 роботах, серед яких: 7 статей у закордонних періодичних виданнях, включених до міжнародної наукометричної бази Scopus (додаток Ж).

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, 5ти розділів, висновків, списку використаних джерел та додатків. Загальний обсяг дисертації складає 244 сторінок, з них: 14 алгоритмів; 43 рисунки по тексту; 22 таблиці по тексту; список використаних джерел зі 124 найменувань на 15 сторінках;

7 додатків на 35 сторінках. Основна частина дисертації викладена на 167 сторінках, що складає приблизно 6.9 авторських аркушів.

1 СУЧАСНИЙ СТАН ДОСЛІДЖЕНЬ У КОНТЕКСТІ РОБОТИ

Проблема здобуття насичених наборів термінів з професійних текстів для вивчення онтологій, що описують конкретний домен, була недостатньо досліджена. Тому було проведено більш широкий систематичний огляд відповідних робіт для збору відомостей про сучасний стан досліджень, що є релевантними, у різних галузях знань. Вибір галузей варіювався від розробки онтологій (Ontology Engineering), здобуття онтології з текстів (Ontology Learning from Texts) та науки про інформацію (Information Science), до якісних досліджень у соціальних та медичних науках (Qualitative Research in Social and Medical Sciences).

1.1 Методологія пошуку та відбору релевантних публікацій

Для нашого систематичного огляду літератури ми використовували методологію та конвеєр програмного забезпечення (Dobrovolskyi and Keberle 2018) для збору та забезпечення релевантності вибірки статей для подальшого аналізу. Їх підхід поєднує вибірку методом снігової кулі, імовірнісне моделювання тем та метод аналізу мереж цитування для пошуку по назвам, ключовим словам та анотаціям наукових робіт проіндексованих Microsoft Academic⁴.

Dobrovolskyi and Keberle (2018) запропонували, перед початком вибору статей їх методом, вибрати вручну:

- або невеликий набір дуже відповідних та добре цитованих оглядових робіт на тему, що цікавить;
- або набір ключових фраз, які найкраще характеризують тему, що цікавить.

Оскільки нами не було виявлено оглядових публікацій про використання термінологічного насичення у здобутті онтологій, ми пішли другим шляхом і запропонували кілька релевантних ключових фраз: автоматизоване здобуття термінів; автоматизоване розпізнавання термінів; теоретичне насичення; насиченість термінологією; термінологічне насичення; повнота корпусу; вивчення онтологій з тексту; міра насичення; міра насичення термінології.

⁴ <https://academic.microsoft.com/>

Результатом вибірки літератури стали два кластери статей⁵, що не перетинаються по складу: (i) навчання онтологій з текстів та АЗТ (474 статті); та (ii) теоретичне насичення (насичення даними) у якісних дослідженнях (512 статей). Перегляд назв та анотацій статей у межах першого кластера виявило: “насичення” – 0 статей; “повнота” – 4 статті, 3 з яких були не досить релевантними, а 1 (Ferrari et al. 2014) виявилась релевантною. У другому кластері ми знайшли: “насичення” – 12 статей; “повнота” – 4 статті. У нашому огляді (розділ 2.6) ми використали 2 із 12 статей, в яких згадувалось “насичення” (Aldiabat and Le Navenec 2018; Glaser and Strauss 1967).

1.2 Розробка онтологій домену та здобуття вимог

Онтологія як артефакт, за визначенням (Guarino et al. 2009), є "формальною, експліцитною специфікацією спільної концептуалізації" (Studer et al. 1998). Основна інтерпретація цього терміну в розробці онтологій доменів полягає в тому, що це є формальна описова теорія предметної області (домену). Особливою властивістю онтології домену, в межах нашої уваги, є те, що вона повинна бути спільно використаною специфікацією. Загально прийнятою оцінкою спільного використання є ступінь, в якій онтологія підтримує інтерпретацію, або думки про домен, фахівців – стейкхолдерів знань у домені. Чим більше думок, які далі називаються вимогами, підтримуються онтологією, тим вищим є прийняття стейкхолдерами цієї онтології. Тому, спільнота стейкхолдерів більш готова використовувати цю онтологію.

Щоб розробити онтологію, яка адекватно відповідає вимогам відповідної спільноти стейкхолдерів, важливо бути достатньо поінформованими про їхні думки. Це створює виклик, оскільки важко безпосередньо отримати інтерпретації домену від зацікавлених сторін знань у явній формі (Tatarintseva et al. 2013). Існуючі методології розробки онтології вирішують цю проблему дещо по-різному,

⁵ Списки статей, включаючи кількість цитувань, назви, анотації, та посилання на Microsoft Academic є загальнодоступними за адресою: <https://github.com/OntoElect/LR>. Назви файлів: для кластера (i) [Publist-OLT-474.xlsx](#); для кластера (ii) [Publist-TS-551.xlsx](#).

але усі ті з них, що є найчастіше використаними (і найбільш часто цитованими), базуються на організації систематичних інтерв'ю або проведення мозкових штурмів з експертами, відібраними зі стейкхолдерів знань (Sure et al. 2003; Gómez-Pérez et al. 2004; Pinto et al. 2004; Schreiber et al. 1999; Suárez-Figueroa et al. 2012). Однак, завжди існує ризик того, що відібрана група експертів та їх вимоги не відображають адекватно думки усієї спільноти спеціалістів.

Крім того, в літературі по розробці онтології не було вказівок щодо того, як об'єктивно виміряти репрезентативність групи експертів і, отже, повноту вимог, висунутих до них. Насправді, оскільки робота експертів коштує досить дорого, компроміс між повнотою та ціною часто робиться на користь зниження ціни.

Краудсорсингові (crowd-sourcing) підходи до виявлення вимог у розробці онтологій (Sarasua et al. 2015) набули поширення як дешевий спосіб “розподілити виконання завдання серед потенційно великої групи учасників”. Незважаючи на кілька повідомлень про історії успіху у використанні краудсорсингу в області Семантичного Вебу (с.f. Sarasua et al. 2015), питання щодо повноти вимог, специфікованих командою краудсорсерів, та відсутності їх упередженості в інтерпретації домену на сьогодні залишаються без відповіді.

1.3 Вивчення онтологій з текстів та консенсус спільноти

Для подолання труднощів у забезпеченні репрезентативності (повноти) та зниження рівня упередженості, (Maedche and Staab 2001) було запропоновано вивчати онтології, або вимоги до розробки онтології (які також називаються ознаками), не за допомогою групи відібраних експертів, а з артефактів, розроблених стейкхолдерами знань у домені. Обґрунтування полягало в тому, що здобуття ознак з доступних артефактів є менш затратним у часі та враховує різноманіття думок. Це можна зробити за допомогою часткової автоматизації. Крім того, збір репрезентативної вибірки артефактів є більш здійсненним, ніж залучення репрезентативної групи експертів.

Одним із релевантних видів цих артефактів є професійні текстові документи. Вивчення онтологій з тексту є помітним підрозділом в навчанні онтології за допомогою розроблених методологій та конвеєрів обробки (Buitelaar et al. 2005;

Wong et al. 2012; Maynard et al. 2017). Щоб дізнатися вимоги для розробки онтології домену, потрібно зібрати репрезентативний набір текстових документів. Ця колекція документів повинна:

- Містити релевантні тексти достатньо хорошої якості
- Бути достатньо повною, тобто репрезентативною, щоб відображати консенсус спільноти стейкхолдерів

Не дивлячись на те, що існує безліч підходів для збору відповідних документів (розділ 1.4), проблема перевірки, чи достатньо повною є колекція документів, наскільки нам відомо, досі не була вирішена належним чином.

Можливо, причиною недооцінки важливості забезпечення репрезентативності текстової колекції є те, що текстових ресурсів забагато. Отже, можна завжди розраховувати, що таких ресурсів достатньо, навіть для вузько сфокусованого домену. Можливо зібрати високоякісні наукові публікації за певною темою або у межах певної галузі знань, як, наприклад, це було зроблено для менеджменту знань (Knowledge Management (KM)) (Kosa et al. 2017). Ця KM колекція містить понад 7 400 журнальних повнотекстових статей, що можна вважати достатнім обсягом, враховуючи рекомендації експертів з лінгвістичних корпусів (Corpas Pastor and Seghiri Domínguez 2010). Однак, навіть якщо хтось зібрав те, що він вважає за достатнє, це все ж є його особистою (упередженою) думкою. Для забезпечення неупередженості, повинен існувати спосіб виміряти репрезентативність цього тестового корпусу щодо вивчення онтологій.

Останнім часом було зроблено декілька спроб автоматичного здобуття онтологій із академічних даних що є науковими публікаціями. Одним з актуальних прикладів (Salatino et al. 2018) є здобуття онтології Комп'ютерних наук (Computer Science Ontology (CSO)). CSO було здобуто з використанням алгоритма Klink-2 (Osborne and Motta 2015). Klink-2 виявився досить масштабним, щоб сформувати повну CSO використовуючи колекцію метаданих 16 000 000 наукових

публікацій. Однак, використання лише метаданих, навіть у поєднанні із зовнішніми ресурсами, як DBpedia⁶, дозволяє зробити лише поверхневий аналіз документів.

На відміну від підходу Klink-2, у цій роботі ми обробляємо повні тексти наукових публікацій і часто відфільтровуємо значні частини метаданих, такі як назва журналу або конференції та ім'я автора, що виявляються накопичуваним регулярним шумом, як це можна подивитися у результатах нашого експерименту з колекцією DAC Naturelle (розділ 4.5.3).

1.4 Збір релевантних документів доброї якості

Надійним способом збору якісних документів є перегляд рецензованих наукових публікацій. Рецензування гарантує як релевантність щодо домену так і професійну якість тексту. Проблема полягає в тому, що обсяг наукових праць є величезним. Отже, ручний пошук та вибір репрезентативної колекції статей, що стосуються домену або досліджуваної теми в домені, практично не є можливим.

З літератури відомо декілька підходів до збору відповідних статей ручним способом (Waddington et al. 2012) або (частково) автоматизованим (Lecu and Beatty 2012; Ahad et al. 2016; Medlar et al. 2016; Varela et al. 2018). Методами є: вибірка методом обмеженої снігової кулі з подальшим аналізом мережі цитування (Lecu and Beatty 2012; Ahad et al. 2016; Varela et al. 2018); систематичний пошук на основі імовірнісного моделювання теми (Medlar et al. 2016); або велика систематична робота експертів (Waddington et al. 2012), що об'єднує вибірку за методом снігової кулі, пошук за ключовими словами та читання текстів публікацій.

Як зазначено у розділі 1.1, щоб вибрати відповідні джерела для огляду літератури в цій роботі, ми використали метод та інструментальний конвеєр (Dobrovolskyi and Keberle 2018). Метод полягає у поєднанні автоматизованого імовірнісного моделювання тем, вибірки сніжної кулі і аналізу мережі цитування. Ми також використовували ручний підхід, заснований на судженнях експертів, для відбору колекцій документів, що використовуються в наших експериментах (розділ 4.2).

⁶ <https://wiki.dbpedia.org/>

1.5 Термінологічне насичення та репрезентативність

Як показав результат вибірки літератури (розділ 1.1), використання явища насичення недостатньо досліджено в галузі комп'ютерних наук чи науки про інформацію, зокрема у здобутті знань та вивченні онтологій з текстів. Riazanov and Voronkov (2001) використовували насичення тверджень для оптимізації автоматизованого доведення теорем. Chernyak and Berenstein (2006) використовували насичення термінів для побудови графіків близькості термінів (term proximity) у кластеризації документів (document clustering) та виконанні запитів (query answering). Doerre et al. (2002) використали явище насичення, щоб правильно відібрати підмножину документів для подальшого ієрархічного кластерного аналізу для побудови таксономії теми. Nan et al. (2014) запропонували використовувати міру насичення для кластеризації патентних текстів та класифікації їх за темами.

Ferrari et al. (2014) запропонували дві метрики для оцінки повноти специфікацій вимог до програмного забезпечення, що враховують релевантність термінів та взаємозв'язок між ними. У їх роботі терміни та взаємозв'язки були здобуті із специфікацій вимог, які були надані у формі текстів природньою мовою. Підхід Ferrari et al. (2014) допоміг оцінити повноту зазначених вимог, порівнявши їх з тими, що були здобуті з одного документу або невеликої кількості документів. Однак, це не допомогло з'ясувати, чи використовуваний набір документів є достатньо повним, щоб належним чином відповідати думці більшості в межах спільноти стейкхолдерів знань. Незважаючи на це, підхід Ferrari et al. (2014) є близьким до нашої роботи у аспекті здобуття термінології.

На відміну від Ferrari et al. (2014), наш підхід (див. розділи 2 та 3) дозволяє не тільки вимірювати та виявляти термінологічне насичення, але і здобувати підколекцію з термінологічним ядром, якщо насичення було виявлене. Отже, це дозволяє конструктивно здобувати компактні репрезентативні набори значущих термінів із відповідних підколекцій з термінологічним ядром.

1.6 Теоретичне насичення та вивчення онтологій

У нашому огляді літератури були розглянуті публікації у галузі якісних досліджень, оскільки їхня ціль є дещо узгодженою з вивченням онтологій. Дійсно, методологія якісного дослідження базується на обробці інтерв'ю з суб'єктами. Метою цієї обробки є побудова (описової) теорії для підтримки (або спростування) гіпотези дослідження в досліджуваному контексті. Складність якісного аналізу також схожа на ускладнення інженерів-онтологів (див. розділ 1.2) – інтерв'ю дорого коштують. Ось чому, на відміну від інженерів-онтологів, якісні дослідники шукали спосіб обмежити предметну вибірку, яка вже представляла усі подальші відповіді інших суб'єктів, які ще не були опитані.

Першопрохідцями, які запровадили насичення даними в якісних дослідженнях, були Glaser and Strauss (1967). Вони запропонували метод обґрунтованої теорії для обробки даних інтерв'ю і позначили цей метод як "побудова теорії на основі даних, систематично отриманих в результаті соціальних досліджень" (Glaser and Strauss 1967). На їх думку, обґрунтованість теорії означала, що кожне твердження в ній підтверджується даними. Glaser and Strauss (1967) стверджували, що теорія стає обґрунтованою, якщо її будувати систематично. Однак, окрім надання кількох неформальних евристичних пропозицій, вони не розкрили, як зробити цю побудову систематичною.

В літературі з якісних досліджень були зроблені та широко обговорювались спроби ввести в дію виявлення насичення даними в контексті методу обґрунтованої теорії. Однак аналіз цих пропозицій виходить за рамки цієї роботи. Натомість ми посилаємось на висновок однієї з останніх публікацій (Aldiabat and Le Navenec 2018). Вони заявляють, що виявлення насичення даними все ще є "таємничим кроком". Крім того, слід зазначити, що причиною відсутності прогресу може бути те, що виявлення факторів, які сприяють або перешкоджають насиченню даних, здійснюється неформально. Що стосується термінологічного насичення, його формальна оцінка, з формально доведеною обґрунтованістю, досі не доступна в літературі. Однією з ранніх робіт, в якій намагалися запропонувати алгоритмічний підхід для вимірювання термінологічного насичення, була (Tatarintseva et al. 2013).

Метод обґрунтованої теорії в якісних дослідженнях передбачає, що перевірку насичення слід проводити щоразу, коли враховується нова порція даних. Отже, метод є ітераційним за своєю природою. Питання, на яке соціологи не отримали повноцінної та ретельної відповіді, полягає в наступному: які додаткові елементи даних (документи) необхідно взяти для аналізу, щоб забезпечити якомога компактнішу та стабільнішу⁷ насичену вибірку? Рекомендації, надані Aldiabat and Le Navenec (2018) включають: точне визначення обсягу дослідницького питання; використання методу тріангуляції щодо колекції даних; використання досвідчених дослідників, знайомих з методом дослідження; розуміння філософської основи методу; використання сенсibiliзуючих концепцій.

Однак, на відміну від нашої роботи, ці рекомендації не пропонують жодного об'єктивного правила впорядкування вхідних документів для включення до майбутньої частини вхідних даних.

1.7 Впорядкування документів для обробки: відмітки часу та вплив

Єдиною згадкою про порядок обробки текстів під час формування наборів даних, як ми виявили, було (Doerre et al. 2002), що було поєднанням випадкового та хронологічного відбору. Однак, у їхньому звіті не розкрито, як обраний порядок відбору впливав на насичення даними. У нашій роботі ми експериментуємо з порядками, використаними у (Doerre et al. 2002) і вважаємо, що це не найкращий вибір.

На відміну від (Doerre et al. 2002), ми експериментуємо ще з трьома можливими порядками та рекомендуємо найкращий вибір після їх порівнювального оцінювання. Наш набір порядків враховує не лише час публікації, а й їх вплив.

Декілька публікацій у галузі науки про інформацію досліджували як вимірюваний вплив публікації може вплинути на думки спільноти спеціалістів у домені. Потенційний вплив публікації найчастіше вимірюється використовуючи

⁷ Під стабільністю ми розуміємо найменшу можливу ймовірність зникнення насичення (десатурації) вибірки після додавання нового інкременту елементів даних.

аналіз цитування чи спів-цитування, при цьому в якості показників використовуються кількість цитувань, бали або частота. Слід відмітити декілька цікавих удосконалень цього загальноприйнятого підходу, що стосуються нашої роботи. Pohl and Mottelson (2019) запровадили міри читабельності, новизни або видалення імен шляхом визначення простих правил на основі входження слів. Вони також продемонстрували взаємозв'язок цих показників із кількістю цитувань у часі.

Деякі з цих робіт пропонували уточнення аналізу цитування для вимірювання потенційного впливу публікації з точки зору інноваційності. Schneider and Costas (2017) використовували підхід на базі характеристичних балів та шкал (Characteristic Scores and Scales (CSS)), що був запропонований Glänzel and Schubert (1988), для кластеризації статей на основі партицій розподілу цитувань. На основі цієї кластеризації, вони відфільтрували потенційно багато цитованих “послідовників” (помилково позитивних) від реальних кандидатів “проривних” статей (дійсно позитивних). Savov et al. (2020) запропонували оцінку інновацій, виходячи з того, “наскільки передбачуваний рік публікації випереджає або відстає від фактичного року публікації, що відображає, чи охоплює стаття більше тем, досліджених статтями, які були опубліковані в минулому, чи більша кількість її тем буде охоплена майбутніми статтями”. Вони використовували класифікатор SVM (Support Vector Machine) та приховані розподіли тем для прогнозування відміток у часі.

Будучи обізнаними про цікаві удосконалень аналізу цитування щодо обліку для вимірювання інноваційності публікації, ми зазначаємо, що ці удосконалень вимагають значних додаткових витрат у часі обчислень. Тому, у цій роботі, ми вирішили дотримуватися використання основного показника – частоти цитування – для оцінки впливу документу.

Крім того, у релевантних публікаціях не вистачає обґрунтованої рекомендації щодо того, як впорядковувати документи в ітераціях для здобуття компактних і стабільно насичених наборів термінів, які надалі використовуватимуться як ознаки для систематичного виявлення вимог у розробці онтологій.

1.8 Автоматизовані методи здобуття термінів

Ключовою складовою технології нашого конвеєру є АЗТ. Тому обґрунтований вибір методу для виконання АЗТ є суттєво важливим. Цей вибір було зроблено на базі огляду та порівнювального оцінювання методів та інструментів для АЗТ у (Kosa et al. 2018a).

У більшості підходів до АЗТ (Zhang et al 2008) обробка здійснюється у два послідовні етапи: лінгвістична обробка та статистична обробка. Конвеєри лінгвістичної обробки (Maunard et al. 2017) включають тагери частин мови (part of speech, POS) або чанкери фраз (phrase chunking); відфільтровують стоп-слова; і обмежують терміни-кандидати до послідовностей n-gram (n-gram sequences), таких як іменники, іменникові словосполучення, словосполучення прикметник-іменник, або іменник-прийменник-іменник. Потім застосовується статистична обробка для вимірювання рангу (оцінки) термінів-кандидатів (Zhang et al 2008). Цими мірами є (Zhang et al 2008; Korkontzelos et al. 2008): або міри «юнітхуду», які зосереджені на колокативній силі одиниць (слів), що складають певний термін, або міри «термхуду», що вказують на міцність асоціації термінів з концептами домену.

Для «юнітхуду» використовуються показники, такі як: взаємна інформація (Church and Hanks 1990; Wermter and Hahn 2005; Daille 1996), логарифмічна подібність (Dunning 1993; Wermter and Hahn 2005; Daille 1996), t-тест (t-оцінка) (Wermter and Hahn 2005), варіанти ентропії (Manning and Schutze 1999), парадигматична модифікованість та її варіанти (Fahmi et al. 2007). Міри «термхуду» базуються або на частоті (неконтрольовані підходи), або на використанні референтних корпусів (напівконтрольовані підходи). Найбільш часто використовуваними показниками на основі частоти є частота терміну (term frequency, TF) (Medelyan and Witten 2006), середня або загальна частота терміну (average TF, ATF, або total TF, TTF) (Medelyan and Witten 2006), TTF у поєднанні з оберненою частотою документу (TF with inverse document frequency, TF-IDF) (Evans and Lefferts 1995; Ahmad et al. 1999), залишковий IDF (residual IDF, RIDF) (Church and Gale 1995; Zhang et al 2016). Мірами на основі доменних корпусів є (Astrakhantsev 2016): дивність (weirdness) (Ahmad et al. 1999), відповідність домену

(domain pertinence) (Sclano and Velardi 2007) та релевантність щодо домену (domain relevance) (Peñas et al. 2001). Пізніше було запропоновано гібридні підходи, що поєднують вимірювання «юнітхуду» та «термхуду» у єдиному значенні. Репрезентативною мірою є C/NC-value (Frantzi and Ananiadou 1999). Підходи до АЗТ, які засновані на C/NC-value, отримали подальшу еволюцію в багатьох роботах: підтримка однослівних термінів (Lossio-Ventura et al. 2013), використання відомих термінів (Fahmi et al. 2007), впевненість у терміні GlossEx (GlossEx term confidence) (Kozakov et al. 2004), якщо згадати кілька з них, що є найбільш відомими.

В усіх методах АЗТ лінгвістична обробка організована та реалізована однаково, за винятком того, що деякі з них також включають фільтрування стоп-слів. Стоп-слова (терміни) також можна відфільтрувати і на окремому етапі відсікання (cut off) після статистичної обробки. Статистична обробка іноді додатково розподіляється на дві послідовні підфази: оцінювання термінів-кандидатів та їх ранжування. Оцінювання термінів-кандидатів відображає ймовірність того, що кандидат дійсно є терміном. Відомі способи вимірювання оцінки можна поділити на такі, що спираються на (Astrakhantsev 2016) вимірювання частот появ (включаючи словосполучення), оцінку попадань у контексти, а також такі, що використовують: референтні корпуси, наприклад Вікіпедію (міра PU-ATR) (Astrakhantsev 2015), моделювання тем та доменів (Bordea et al. 2013; Badenes-Olmedo et al. 2017; Dobrovolskyi and Keberle 2018).

Процедура відсікання приймає найбільш вагомих кандидатів на підставі оцінок, і тим самим відрізняє значущі терміни від незначущих (тобто не термінів). Багато методів відсікання спираються на оцінки, отримані з одного алгоритму підрахунку оцінок, і встановлюють поріг відсікання в той чи інший спосіб. Деякі інші, що збирають оцінки з декількох алгоритмів підрахунку оцінок, використовують (зважені) лінійні комбінації (Park et al. 2002), голосування (Zhang et al 2008; Tatarintseva et al. 2013), або (напів-)контрольоване навчання (Nokel and Loukachevitch 2013). У нашому експерименті ми дотримуємося (Tatarintseva et al. 2013; Ermolayev 2018) і робимо відсікання після здобуття термінів, на основі

здобуття простої більшості голосів. Таким чином, алгоритми (та програми) АЗТ, які виконують відсікання разом з оцінюванням, не є релевантними у наших дослідженнях.

На підставі оцінок (Astrakhantsev 2016; Zhang et al 2008), найпоширеніші алгоритми АЗТ, оцінки ефективності яких були опубліковані, порівнюються у Таблиці 1.1.

Таблиця 1.1. Порівняння продуктивності найбільш широко використовуваних методів АЗТ на основі еталонних реалізацій та експериментів, опублікованих у (Astrakhantsev 2016; Zhang et al 2008)

Метод	Незалежність від домену (+/-)	Контроль (U/SS)	Метрики	Значимість терміну	Відсікання (+/-)	Точність (GENIA; середнє)	Тривалість виконання (у відношенні до методу C-value)	Інструмент
TTF	+	U	Загальна Частота Термінів (TTF)	+	-	0.70; 0.35	0.34	ATR4S JATE
ATF	+	U	Середня Частота Термінів	+	-	0.71; 0.33 0.75; 0.32	0.37 0.35	ATR4S JATE
TTF-IDF	+	U	TTF+Зворотня Частота Документів (IDF)	+	-	0.82; 0.51	0.35	ATR4S JATE
RIDF	+	U	Залишкова IDF	-	-	0.71; 0.32 0.80; 0.49	0.53 0.37	ATR4S JATE
C-value	+	U	C-value, NC-value	+	-	0.73; 0.53 0.77; 0.56	1.00 1.00	ATR4S JATE
Weirdness	+/-	SS	Дивність	-	-	0.77; 0.47 0.82; 0.48	0.41 1.67	ATR4S JATE
GlossEx	+	SS	Лексична когезія терміну (cohesion), Специфічність Домену	-	-	0.70; 0.41	0.42	ATR4S JATE
Term Extractor	+	SS	Достовірність Домену, Консенсус Домену, Лексична когезія,	-	+	0.87; 0.46	0.52	ATR4S JATE

		Структурна Відповідність						
PU-ATR	-	SS	NC-value, Специфічність Домену	-	+	0.78; 0.57	809.21	ATR4S JATE

Таблиця також містить оцінку аспектів, які ми використовуємо для вибору найкращого методу для нашого дослідження.

Коментарі до Таблиці 1.1:

Незалежність від домену: “+” – означає незалежний від домену метод; “-” – вказує на те, що цей метод є (або є визначеним його авторами як) специфічний для домену, або оцінюється лише в одному конкретному домені. Ми шукаємо метод незалежний від домену.

Контроль: "U" – неконтрольований; "SS" – напівконтрольований. Ми шукаємо неконтрольований метод.

Значимість терміну: "+" – метод повертає значення для кожного збереженого терміну, яке також може бути використане, як міра його значимості в порівнянні з іншими термінами. "-" – означає, що таке значення не повертається або сам метод робить відсікання. Ми прагнемо отримати значення, щоб пізніше розпочати відсікання.

Відсікання: "+" – метод сам робить відсікання і повертає лише значимі терміни; "-" – метод не робить відсікання. Ми шукаємо "-".

Точність і тривалість виконання: Ці значення базуються на порівнянні двох експериментів з крос-оцінювання, описаних у (Astrakhandsev 2016; Zhang et al 2008). Пусті клітинки в таблиці означають, що в цьому конкретному експерименті не було використано даних для оцінювання цього конкретного методу. Огляд (Astrakhandsev 2016) використовував ATR4S (Astrakhandsev 2016) – програмне забезпечення з відкритим кодом для АРТ написане в Scala (4S). Автором (Astrakhandsev 2016) було перевірено 13 різних методів, на 5ти різних наборах даних, включаючи набір даних GENIA (Kim et al. 2003). Огляд (Zhang et al 2008) використовувало JATE 2.0 (Zhang et al 2008) – вільнодоступне програмне забезпечення для АЗТ, написане на Java. Оцінювалися 9 різних методів,

впроваджених у JATE, на 2ох різних наборах даних, включаючи GENIA. Отже, результати на GENIA є базовими для порівняння точності. Для кожного референтного експерименту даються два значення: точність на GENIA та середня точність. Обидва огляди (Astrakhantsev 2016) та (Zhang et al 2008) експериментували з методом C-value, який був у середньому найповільнішим за (Zhang et al 2008). Таким чином, час виконання для методу C-value використовувався як базовий показник для нормалізації решти значень у стовпчику **Тривалість виконання.**

На основі аналізу Таблиці 1.1, ми підтримуємо висновок (Zhang et al 2008) що C-value є найбільш надійним методом. Цей метод послідовно дає хороші результати в плані точності здобуття термінів на двох різних сумішах (mixes) наборів даних (Astrakhantsev 2016; Zhang et al 2008). Також слід зазначити, що C-value є одним з найповільніших методів у групі неконтрольованих і незалежних від домену, однак його продуктивність порівнянна з найшвидшими. Також, C-value перевершує залежні від домену методи, іноді значно – як у випадку з PU-ATR. Тому, для наших експериментів ми обрали метод C-value.

1.9 Програмні реалізації методів АЗТ

Для вибору програмних інструментів, що реалізують метод C-value для АЗТ ми розглянули описи програмного забезпечення для здобуття термінів на декількох веб-ресурсах, наприклад <http://inmyownterms.com/terminology-extraction-tools/> або https://en.wikipedia.org/wiki/Terminology_extraction. На додаток до згадуваних раніше еталонних реалізацій, ATR4S та JATE 2.0, ми визначили наступні вільно доступні програмні засоби як зазначено у Таблиці 1.2.

Було вирішено не розглядати ATR4S та JATE 2.0 оскільки не було повністю зрозуміло, як здобути реалізацію методу C-value з цих наборів. З вільно доступного програмного забезпечення (див. Таблицю 1.2), два відповідні інструменти потрапили до короткого списку – NaCTeM TerMine (Frantzi and Ananiadou 1999) та UPM Term Extractor (Corcho et al. 2015). Ці інструменти реалізують метод C-value і є нейтральними до домена. Для вибору найкращого відповідного інструменту для

вимірювань термінологічного насичення з цих двох було проведено їх крос-оцінку (Kosa et al. 2018a), як повідомляється у розділі 4.4.

Таблиця 1.2. Вільнодоступні програмні засоби АЗТ (перелічені в алфавітному порядку)

Назва / Власник	Посилання	Короткий опис	Алгоритм / Міра	Домен	Обмеження
BioTex / LIRMM	http://tubo.lirmm.fr/biotex/	Здобуває біомедичні терміни із вільного тексту		Біомедичний	Залежність від домену
FiveFilters / Medialab-Prado	http://fivefilters.org/terms-extraction/	Здобуває терміни через веб-сервіс; окладається на PHP-порт здобуття термінів Toria; проста альтернатива сервісу Yahoo Term Extraction	Поява (TTF) і кількість слів у терміні	Незалежний	Веб-служба, розмір тексту обмежений
TaaS (TaaS EU Project)	https://term.tilde.com/	Визначає кандидатів у терміни в документах та автоматично здобуває їх. Використовує CollTerm (лінгвістичні) або Kilgray (статистичні) сервіси	На основі частоти	Незалежний	Не надає оцінки значущості терміну
TerMine / NaCTeM	http://www.nactem.ac.uk/software/terminer/	Здобуває терміни з простих текстів англійською мовою, забезпечує пакетний режим (доступ для академічних користувачів, що не походять з Великої Британії, треба запросити)	<i>C-value</i>	Незалежний	Сервіс просить уникати важкої масової обробки
TermFinder / Translated.net	https://labs.translated.net/terminology-extraction/	Веб-додаток, який здобуває терміни із вставленого тексту. Порівнює частоту слів у даному документі з їх частотою у мові (загальний корпус)	Статистика Пуассона, оцінка максимально і вірогідності та IDF	Вимагається мовний корпус	Повертає оцінку терміну як числове значення (%)
TBXTools (Oliver and V`azquez 2015) / Universitat Oberta de Catalunya	https://sourceforge.net/projects/tbxtools/	Набір інструментів Python з використанням NLTK (набір інструментів для природної мови)	TTF	Незалежний, вимагається мовний корпус	Видаляє n-грами зі стоп-багатомовними словами
UPM Term Extractor (Corcho et al. 2015) / Dr Inventor EU project	https://github.com/ontologylearning/oeg-epnoi-legacy	Програмне забезпечення Java для здобуття термінів та їх відносин з наукових статей	<i>C-value</i>	Незалежний	Приймає текстові дані розміром не більше 15 Мб

1.10 Вимірювання подібності текстів

В останніх дослідженнях підходів до вимірювання подібності тексту, наприклад, (Goma and Fahmy 2013; Yu et al. 2016), методи (або вимірювання)⁸ згруповані на основі аналізу: (i) символів та їх послідовностей; (ii) токенів; (iii) термінів; (iv) корпусів текстів; або (v) синсетів (наборів синонімів). У (Yu et al. 2016) також згадуються гібридні вимірювання, що дозволяють нечітке співставлення між токенами. Нижче вказані короткі характеристики груп. Індивідуальні методи, що належать до груп, докладно описані у Таблиці 1.3.

Вимірювання на основі символів, або символічних послідовностей, порівнюють символи і їх послідовності в строках, також беручи до уваги порядок символів. До них належать вимірювання співпадаючих символічних послідовностей, наприклад, підстрок; відстані редагування; кількість і порядок співпадаючих знаків між двома строками.

Методи, засновані на токенах, моделюють строку як набір токенів. Окремі символи, символічні n -грами або окремі слова можна розглядати як токени. Кількісна оцінка здійснюється шляхом обчислення розміру перетину, нормалізованого за вимірюванням довжини строки.

Вимірювання на основі термінів аналогічні вимірюванням на основі токенів, але токени відрізняються. Це не символічні n -грами, а терміни, які є n -грамми зі слів з можливим варіюванням n . Більш того, враховується вага термінів, наприклад, їх частота появ. Ці вимірювання більшою мірою застосовуються до довгих символічних строк або документів, тому вони краще підходять для вимірювання подібності документів або текстових наборів даних.

Методи, засновані на референтних корпусах текстів та корпусах, що базуються на синсетах (або на знаннях) є майже не релевантними для наших цілей у цій роботі. Підходи, засновані на корпусах, визначають подібність між словами на підставі (статистичної) інформації, отриманої з великих текстових

⁸ У цьому контексті ми не розрізняємо метод і міру. Метод розуміється як спосіб реалізації відповідної функції вимірювання.

корпусів. Для отримання семантичної подібності між словами, подібні підходи, наприклад, ті що використовують WordNet (Miller et al. 1990), спираються на семантичні мережі. Тому вони є надто громіздкими для обчислень, але можуть бути застосовані до АЗТ, наприклад, для прийняття рішення про відсікання. Групування термінів, техніка, про яку ми також звітуємо у цій роботі, виконується вже після того, як терміни були здобуті. Отже, ми не розглядаємо вимірювання, що базуються на корпусах текстів або синсетах.

Огляд найбільш популярних вимірювань текстової/строкової подібності, згрупованих за типом методів, наведено у Таблиці 1.3. Цей огляд на сьогодні не є повним, оскільки в літературі є багато інших варіантів МСП. Проте ті, котрі ми опускаємо, наскільки нам відомо, ґрунтуються на тих самих принципах, що і вказані у Таблиці 1.3.

Таблиця 1.3. Огляд вимірювань текстової подібності/відстані

Найменування, джерело	Опис	Особливості	Релевантність	
			Подібність терміну	<i>thd</i> ⁹
Символи і вимірювання на основі символічних послідовностей				
Найдовша спільна підстрока (Arnold and Ohlebusch 2011)	Вимірювання на основі спільних символічних послідовностей	повертає ціле число (довжину найдовшої спільної підстроки); може бути нормалізована по загальній довжині	помірна	не релевантна
Відстань Левенштейна (Levenshtein 1966)	Вимірює відстань редагування	повертає ціле число необхідних змін	слабка	не релевантна
Відстань Хеммінга (Hamming 1959)	Вимірює відстань редагування	строки повинні мати однакову довжину	слабка	не релевантна
Відстань Монгер-Елкана (Monger and Elkan 1996)	Вимірює відстань редагування	повертає ціле число необхідних змін	слабка	не релевантна
Відстань Жаро (Jaro 1989)	Підраховує мінімальну кількість перетворень одного символу в одній строці для отримання іншої строки	повертає нормалізоване дійсне число з [0, 1]	добра	не релевантна

⁹ Як представлено у розділах 2 та 3, *thd* є метрикою термінологічної різниці. Позначка «релевантна» у цьому стовпчику означає, що ця міра підходить для вимірювання термінологічної різниці між текстовими документами або наборами даних.

Відстань Жаро-Вінклера (Winkler 1990)	уточнює вимірювання Жаро за допомогою значення шкали префіксів - визначає пріоритетність строк, які співпадають на початку	повертає нормалізоване дійсне число з [0, 1]	добра	не релевантна
Вимірювання на основі токенів				
Коефіцієнт Соренсена-Дайса (Dice 1945; Sørensen 1948)	Підраховує співвідношення ідентичних бі-грамів символів до загальної кількості бі-грамів в обох строках	повертає нормалізоване дійсне число з [0, 1]	добра	не релевантна
Подібність Жакара (Jaccard 1912)	підраховує співвідношення між розмірами перетину та об'єднання наборів символів (юні-грамів) в строках	повертає нормалізоване дійсне число з [0, 1]	добра	не релевантна
Косинус подібність (Yu et al. 2016)	Розмір перетинів символів юні-грам поділений на квадратний корінь суми квадрату загальної кількості юні-грамів в обох строках	повертає нормалізоване додатне дійсне число	слабко(складно обчислити)	не релевантна
Вимірювання на основі термінів				
Євклідова Відстань (Huang 2008)	Вимірювання традиційної євклідової відстані в n -мірному метричному просторі (додатних чисел)	працює для документів; повертає додатне дійсне число	не релевантна	релевантна
Косинус подібність (Singhal 2001)	Визначає косинус між двома векторами в просторі термінів; вектори визначаються вагою термінів (наприклад TF of C-value)	працює для документів; повертає нормалізоване додатне дійсне число	не релевантна	слабка
Кореляція Пірсона (Huang 2008)	Обчислює кореляцію Пірсона для пари векторів у векторному просторі термінів	працює для документів; повертає нормалізоване дійсне число, яке коливається від +1 до -1; коли вектори повністю ідентичні це 1	не релевантна	слабка
Манхеттен (блок) відстань (Goma and Fahmy 2013)	відстань, яку треба подолати, щоб перейти від однієї точки даних до іншої, якщо траєкторія слідує за периметром блоків	працює для документів; тотожна вимірюванню <i>thd</i> (Tatarintseva et al. 2013)	не релевантна	релевантна

Автори Lu et al. (2013) представляють підхід, що базується на розширенні, для ефективного вимірювання строкової подібності при розгляді синонімів. Цей результат також має відношення до нашої роботи, оскільки синонім є однією з категорій термінів кандидатів, які, можливо, доведеться розглядати для групування в наших налаштуваннях. У роботі (Lu et al. 2013), також визнано, що в літературі є

багатий набір заходів вимірювання строкової подібності, серед яких символна подібність n -грамів (Lee et al. 2009), відстань Левенштейна (Levenshtein 1966), вимірювання Жаро-Вінклера (Winkler 1990), подібність Жакара (Jaccard 1912), TF/IDF на основі косинус-подібності (Tsuruoka et al. 2007), і вимірювання на основі прихованої моделі Маркова (Qin et al. 2011).

1.11 Ефективне співставлення строк для пошуку вкладених термінів

Chowdhury та Farrell у своїй нещодавній роботі (Chowdhury and Farrell 2019) розглянули декілька алгоритмів співставлення текстових строк, які мали відношення до вкладеності та пошуку вкладених термінів, та запропонували свій власний алгоритм (ТВІ). Результати їх порівняння зведені у Таблиці 1.4, до якої у останньому рядку ми додали оцінку ефективності нашої базової реалізації (Tatarintseva et al. 2013).

Таблиця 1.4. Порівняння продуктивності алгоритмів, релевантних для пошуку вкладених термінів, за (Chowdhury and Farrell 2019). Для порівняння було використано асимптотичне позначення $O(\cdot)$.

Імплементація, Джерело	Структура даних, Джерело	Складність, Час			Параметри
		Індексація	Пошук	Загальне	
(Zang et al., 2008)	сховище ключових-значень, також відоме як хеш-таблиця (Knuth 1998)	$O(v^2)$	$O(m^2)$	$O(v^2 + v * m^2)$	v – кількість строк-кандидатів у терміни; s – середня кількість вкладених строк термінів m – кількість токенів у вхідному (вкладеному) терміні, $v \gg m$
(Lu and Browne, 2012)	префіксне дерево (trie) (De La Briandais 1959), де слова є вузлами	$O(v * t)$	$O(m^2)$	$O(v * t + v * m^2)$	v – кількість строк-кандидатів у терміни; t – середня кількість слів для терміну, $v \gg t$
(Aho and Corasick 1975)	префіксне дерево (trie) (De La Briandais 1959), де слова є вузлами	$O(v * t)$	$O(m + L)$	$O(v * t + m + L)$	v – кількість строк-кандидатів у терміни; t – середня кількість слів для терміну, $v \gg t$ m – кількість токенів у вхідному (вкладеному) терміні, $v \gg m$ L – загальна довжина всіх термінів
ТВІ, (Chowdhury and Farrell 2019)	сховище термінів, що складається з двох хеш-таблиць (Chowdhury and Farrell 2019)	$O(v * \log(v))$	$O(m^2)$	$O(v * \log(v) + m^2)$	v – кількість строк-кандидатів у терміни; m – кількість токенів у вхідному (вкладеному) терміні, $v \gg m$
Вичерпний пошук,	масив строк-термінів	0	$O(v^2 * m)$	$O(v^2 * m)$	v – кількість строк-кандидатів у терміни; m – кількість токенів у

У роботі (Chowdhury and Farrell 2019) визнано, що їх огляд не є повним. Зазначається, що інші алгоритми співставлення строк, наприклад Karp and Rabin (1987), мають подібну або гіршу ефективність пошуку. Отже, методи, перелічені у Таблиці 1.4, складають репрезентативну вибірку. Базове програмне забезпечення АЗТ у нашому конвеєрі (UPM Term Extractor) використовує модифікацію (Zang et al., 2016).

На основі порівняльного аналізу теоретичних оцінок обчислювальної складності алгоритмів пошуку вкладених термінів (Таблиця 1.4), можна було побачити, що підхід на основі сховища ключових значень (Zang et al., 2008) не є оптимальним вибором. Дійсно, алгоритм (Aho and Corasick 1975) дає кращу теоретичну оцінку продуктивності. Розроблений нами оптимізований конвеєр та алгоритм АЗТ (розділ 3) використовують алгоритм Aho-Corasick для швидшого співставлення строк при обчисленні C-value (див. розділ 3.3.1).

1.12 Прогалини у наявних результатах досліджень та мотивація

У розділах 1.2–1.7, 1.10–1.11 було виявлено декілька прогалин у наявних результатах (НП) досліджень інших авторів, що мотивувало нашу роботу наступним чином.

- **НП1: потенційне упередження в інтерпретації домену.** В основному напрямі розробки онтологій, бракує об'єктивних показників щодо репрезентативності групи експертів щодо виявлення вимог та повноти вимог, висунутих наявними експертами.

Ця прогалина у дослідженнях спонукала нас: розглянути вивчення онтологій аби обійти використання експертів (людей) на етапі виявлення вимог; замінити інтерв'ю з експертами здобуванням термінів з релевантних текстових колекцій; дослідити методології якісного дослідження для розуміння того, як забезпечити репрезентативність вибірки текстів.

¹⁰ Вичерпний пошук було додано як базовий для порівнюваної ефективності алгоритмів.

· **НП2: відсутність міри репрезентативності колекції документів.** У вивченні онтологій з текстів бракує показників репрезентативності колекції документів для достатнього охоплення інтерпретацій стейкхолдерами знань у домені. Отже, немає жодних гарантій того, що використана колекція текстових документів є репрезентативною, навіть якщо вона дуже велика за обсягом. Більш того, може статися так, що навіть невелика частина дуже великої колекції текстів буде достатньо повною, щоб задовільно охопити знання про домен.

Цей розрив у дослідженні спонукав нас запропонувати міру (*thd*) та довести її метричні властивості. У нашому підході ця метрика використовується для виявлення термінологічного насичення та побудови термінологічного ядра колекції, як представлено у розділі 3. Обґрунтуванням зосередження уваги на явищі насичення є те, що наявність методу зменшення об'єму термінологічного ядра колекції, порівняно з об'ємом повної колекції, допомогло б значно зменшити зусилля, необхідні для вивчення онтології з цих текстів.

· **НП3: відсутність масштабованості АЗТ для обробки повнотекстових колекцій промислових об'ємів.** У дослідженнях вивчення онтологій, заснованих на наукових даних, масштабовані підходи, такі як Klink-2, обробляють лише метадані документів. Масштабований підхід, який обробляє повні тексти з колекцій промислового розміру, що містить від десятків тисяч до мільйонів статей, наскільки нам відомо, недоступний. Масштабування АЗТ не є тривіальним, оскільки відомі методи є затратними у часі у порівнянні з об'ємом вхідного тексту. Метод C-value, який ми обрали для нашого конвеєру вимірювання термінологічного насичення (розділи 4.3 та 4.4), є дорожчим ніж інші некеровані та нейтральні до домену методи. Крім того, реалізації АЗТ часто обмежені¹¹ об'ємом вхідного тексту.

¹¹ Наприклад, програмне забезпечення UPM Term Extractor (Corcho et al. 2015) використаний у наших експериментах, який базується на методі C-value, не обробляє тексти обсягом більше 15 Мб.

Щоб зменшити цю прогалину у дослідженнях, ми були зацікавлені у зменшенні об'єму даних, що обробляються, зберігаючи при цьому їх репрезентативність та збільшенні ефективності методу C-value для АЗТ. Зменшення об'єму було досягнуто шляхом: розробки методу здобуття термінологічного ядра колекції (розділ 2.1); налаштування вихідних даних для цього методу таким чином, щоб зробити результати більш компактними та стабільними (розділи 4.2 та 4.5). Для покращення ефективності методу C-value, ми шукали підхід, який: правильно розбиває підколекції документів, які поступово збільшуються, щоб забезпечити паралельну обробку невеликих фрагментів даних (розділ 2.6); і використовує алгоритм швидкого співставлення строк у статистичній частині конвеєру АЗТ (розділ 3.4.1).

- **НП4: відсутність методу та критерію для виявлення термінологічного насичення.** У якісному дослідженні, хоча й використовується насичення даних, не існує об'єктивного формального методу та критерію для з'ясування, чи існує насичення та його виявлення.

Відсутність методу спонукала нас запропонувати використання процесу послідовного наближення та відповідного критерія насичення (розділ 2.1) та довести теорему існування у розділі 2.5.

- **НП5: відсутність знань про найбільш придатний метод та програмне забезпечення АЗТ для виявлення термінологічного насичення.** В АЗТ відомо багато методів, що порівнюються на основі їх якості здобуття (влучність, повнота, F-міра). Однак, не існує інформації про ті методи, які б допомагали судити про їх якість у контексті виявлення термінологічного насичення.

Через наявність багатьох методів АЗТ та їх реалізацій, у нас не було мотиву винаходити та впроваджувати ще один із них. Замість цього, ми були зацікавлені у експериментальному крос-оцінюванні та виборі найкращих(щого) відповідних(ого) кандидатів(у) для нашого конвеєру виявлення термінологічного насичення, як зазначено у розділах 4.3 та 4.4.

- **НП6: відсутність знань про оптимальний порядок додавання документів для обробки.** У якісних дослідженнях також не існує правила для встановлення

порядку обробки документів інтерв'ю, такого, що підвищує ефективність та оперативність їх обробки.

Відсутність правила впорядкування документів спонукала нас переглянути та крос-оцінити різні можливі порядки додавання документів, засновані на часових мітках та вимірному впливі документів на насичення. Ця частина роботи описана у розділі 4.5.

· **НП7: відсутність знань про те, чи дає групування термінів більш компактні та стабільно насичені набори термінів.** Відомі методи АЗТ не групують між собою терміни, які повністю або частково схожі. Часткова схожість використовується у деяких підходах (див. Tuarob et al. 2012), для формування таксономій, що відбувається у конвеєрі вивчення онтологій пізніше ніж АЗТ (див. Wong et al. 2012).

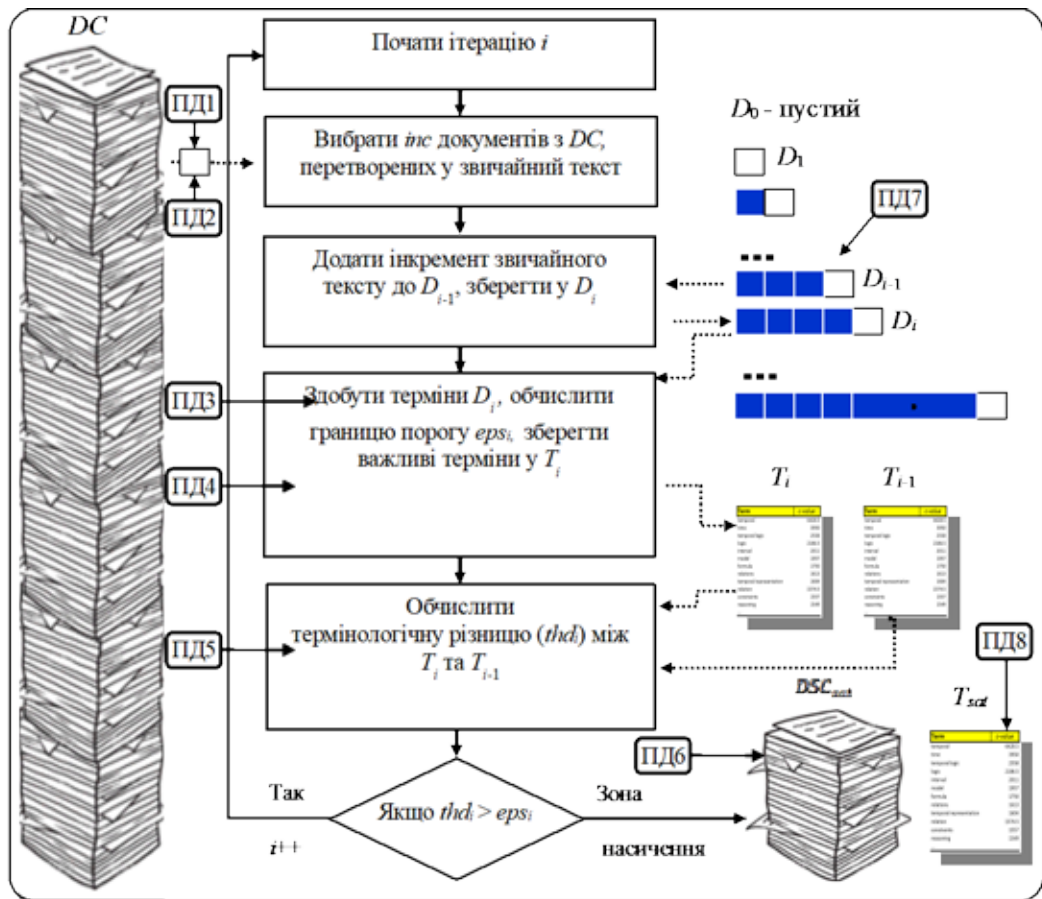
Цей розрив у дослідженні спонукав нас розглянути та крос-оцінити декілька показників схожості коротких строк, визначити пороги подібності термінів, які допомагають покращити термінологічне насичення, коли повністю та / або частково подібні терміни згруповані. Результати представлені у розділах 3.6 та 4.6.

1.13 Питання та завдання дослідження

У цьому розділі, на основі аналізу попередніх робіт та виявлених прогалин у сучасному стані досліджень (НП1 – НП7, розділ 1.12) ми прагнемо сформулювати питання та завдання нашого дослідження. У розділі 1.13.1 ми підходимо до цієї мети, представляючи, наше бачення методу, який може зменшити вищезазначені прогалини в дослідженнях. У цій стислій, але дуже поверхневій презентації, ми ставимо декілька дослідницьких питань у контексті запропонованого підходу. Ці питання та передбачувані шляхи відповідей детально представлені у розділі 1.13.2. Завдання нашого дослідження сформульовані у розділі 1.13.3.

1.13.1 Візія методу виявлення та вимірювання термінологічного насичення

Ми шукаємо рішення для автоматизованого здобуття репрезентативних наборів термінів з використанням підходу ітеративного послідовного наближення. Ітерація цього методу зображена на Рис. 1.1.



- Легенда:
- DC – колекція документів, що релевантна до опису домену
 - inc – кількість документів, відібраних з DC , що додаються до набору даних на будь-якій ітерації
 - D_i – набір даних шлового тексту для i -тої ітерації
 - T_i – набір значущих термінів, збережених з D_i
 - eps_i – границя порогу, який відокремлює значущі від незначущих термінів
 - thd_i – міра термінологічної різниці між двома довільними наборами термінів
 - DSC_{core} – термінологічне ядро колекції документів
 - T_{sat} – насичений набір значущих термінів, збережений з відповідного набору даних $TCSC$
 - – потік управління
 -→ – потік даних
 - ПД... – вказівник на контекст питання дослідження (Розділ 2.13.2)

Рис. 1.1. Візія методу здобуття насиченого набору термінів з колекції документів, що є релевантними для опису домену.

Ідея запозичена у Glaser and Strauss (1967) з методу побудови обґрунтованої теорії у якісних дослідженнях. Ми застосовуємо цей метод для ітеративної побудови термінологічного ядра колекції документів, яке містить насичений набір термінів, що є репрезентативним для домену, що описується документами колекції. Далі ми вдосконалюємо евристику Glaser and Strauss (1967) пропонуючи елементи, яких не вистачає в методі побудови обґрунтованої теорії, і доводячи їх валідність.

Цими елементами є: формальний критерій досягнення термінологічного насичення, включаючи міру, використання якої дозволяє завершити процес; умови, за яких існує термінологічне насичення.

Ми пропонуємо почати процес послідовного наближення з пустого набору документів – набору даних D_0 . На кожній ітерації методу, наприклад i -тій, декілька (*inc*) нових документів беруться із вихідної колекції (DC), попередньо перетворюються у звичайний текстовий файл і додаються до набору даних (D_{i-1}), який було оброблено на попередній ітерації ($i-1$). Це формує набір даних (D_i) для i -тої ітерації. Завдяки цій процедурі набори даних поступово зростають за ітераціями. Ми припускаємо, що, зростаючи, набори даних послідовно наближаються до набору даних (D_{sat}) що відповідає підколекції документів термінологічного ядра (DSC_{sat}). Нарешті, з D_{sat} здобувається насичений набір термінів (T_{sat}).

Щоб оцінити, наскільки ми близькі до D_{sat} , ми покладаємось на вимірювання термінологічної різниці між наборами значущих термінів, збережених із термінів-кандидатів, здобутих із наборів даних у послідовних ітераціях. На i -ій ітерації термінологічна різниця (thd_i) вимірюється між наборами збережених значущих термінів T_i та T_{i-1} .

Для генерації T_i на i -ій ітерації, ми пропонуємо такі кроки:

- Здобути набір термінів-кандидатів з D_i з їх значеннями важливості
- Обчислити поріг відсікання eps_i щоб відокремити важливі від неважливих термінів
- Зберегти терміни-кандидати, які мають значення важливості вище eps_i у T_i як важливі терміни

Для виявлення термінологічного насичення ми пропонуємо спостерігати за thd_i у порівнянні з eps_i і зупиняти процес, коли thd_i надійно опускається нижче eps_i .

1.13.2 Питання дослідження та підходи до відповідей

Як зазначено на Рис. 1.1, відповіді на кілька запитань, від ПД1 до ПД8, потрібні для того, щоб описаний підхід став методом. Ці питання дослідження сформульовані нижче. Крім того, передбачувані способи отримання відповідей

викладені в контексті окремих питань. Розділ узагальнено у Таблиці 1.4, що пропонує посилання на відповідні розділи, підрозділи та артефакти в рамках дисертації.

ПД1 (впорядкування документів): *Які документи в необробленій частині колекції слід відібрати для формування наступного інкременту для обробки?*

Документи в колекції відрізняються один від одного термінологічним «слідом» у домені. Відмінності викликані наступним:

- Зміни термінологічної спрямованості спільноти стейкхолдерів у домені з плином часу – так би мовити, їх термінологічний дрейф у часі
- Вплив документів на термінологію, якою користується спільнота стейкхолдерів у домені

У роботі відповідь на це питання дається шляхом експериментальної крос-оцінки п'яти можливих впорядкувань документів. Ці впорядкування базуються на різниці в часі публікації документів та оцінці їх потенційного впливу на термінологію в домені. Огляд стану відповідних досліджень з цього питання наведено у розділі 1.7. Результати крос-оцінки та наші рекомендації щодо найбільш збалансованого впорядкування представлені у розділі 4.5.

ПД2 (розмір інкременту): *Чи має значення розмір інкременту (inc)? Чи існує якась оптимальна кількість документів для інкременту, що робить термінологічну насиченість досяжною і стабільною за найменшої кількості ітерацій?*

Набори даних можуть генеруватися із використанням менших або більших інкрементів. Здається, що чим менша кількість документів в інкременті тим більшою є кількість необхідних ітерацій. З іншого боку, чим менший інкремент, тим вищою може бути точність здобуття термінологічного ядра підколекції. Отже, було би раціональним знайти оптимальний розмір інкременту для збалансування продуктивності та точності. У представленій роботі відповідь на це запитання формально дається шляхом доведення Наслідку 2.6 до Теорема 2.5 про рівність sv та $trsv$ у розділі 2.7.

ПДЗ (метод АЗТ): *Який метод АЗТ є найбільш придатним до запропонованого підходу послідовного наближення?*

У відомих дослідженнях, методи АЗТ та їх програмні реалізації порівнюються шляхом вимірювання якості здобуття термінів на різних наборах даних (див. розділ 1.8). Цей критерій, хоч і важливий, але не є повністю релевантним для нашого контексту, оскільки термінологічна різниця вимірюється між наборами термінів, здобутими одним і тим самим методом. Тому ці набори термінів характеризуються як однакові за якістю здобуття. Отже, для вибору найкращого методу АЗТ в роботі необхідно враховувати додаткові аспекти якості. Тому відповідь на це запитання дається шляхом крос-оцінювання кількох програмних реалізацій, що потрапили до короткого-листа завдяки їх якості на поєднанні різних наборів даних. Перехресна оцінка проводиться експериментально, у конвеєрі обробки, відображеному на Рис. 1.1, використовуючи програмне забезпечення, яке входить до короткого-листа (розділ 1.9). Порівняння результатів наведено у розділах 4.3 та 4.4.

ПД4 (поріг відсікання): *Яким чином слід обрати поріг відсікання неважливих термінів (ϵ)? Яким є обґрунтування цього вибору?*

Обґрунтування відсікання тих термінів зі здобутого набору, що є неважливими, було запропоновано Tatarintseva et al. (2013), яка робила це базуючись на альянсі публічних виборів. Дійсно, переможцем на публічних виборах є той, хто отримує просту більшість голосів виборців, що становить, як мінімум, п'ятдесят відсотків, плюс один голос. У цій роботі ми дотримуємось цього підходу та детально розкриваємо його формально у розділі 2.

ПД5 (термінологічна різниця): *Яким чином слід обчислювати термінологічну різницю (thd) між двома наборами термінів, у яких терміни мають значення їх важливості? Які властивості має ця функція?*

Очевидно, у цьому питанні є два технічні під-питання.

Перше – як порівняти терміни, які є короткими текстовими строками? З огляду стану досліджень (див. розділ 1.10) відомо декілька мір подібності строк. Крім того, доступно кілька алгоритмів швидкого співставлення строк (див. розділ

1.11). У цій роботі ми досліджуємо декілька методів для повного або часткового співставлення строк у двох контекстах: групування термінів (розділи 2.7, 3.4, та 4.6) та обчислення *thd* (розділ 3.5).

Однак, результати сучасних досліджень у цій сфері не дають формальної відповіді про те, як виміряти різницю між наборами збережених важливих термінів, враховуючи, що кожен термін подається з дійсним числом – його значенням важливості. Tatarintseva et al. (2013) запропонували алгоритм для цього вимірювання. На основі їх роботи ми формально визначаємо міру *thd* у розділі 2.3 і досліджуємо її метричні властивості в розділі 2.4.

ПД6 (існування термінологічного насичення): *Як перевірити чи досягне термінологічне насичення для будь якої колекції документів? Що, якщо *thd* опуститься нижче *eps* на деякій ітерації і та підніметься вище *eps* на ітерації $j > i$? Як переконатися, що насичення є стабільним після його виявлення на ітерації i ?*

Відповіді на цю групу запитань даються формально шляхом доведення Теорема 2.3 про існування термінологічного насичення у розділі 2.6.

ПД7 (оптимізація та масштабованість АЗТ): *Чи можна оптимізувати метод АЗТ, який використовується у пропонованому рішенні для виявлення термінологічного насичення, щоб він був масштабованим до об'ємів реальних промислових колекцій наукових публікацій?*

Як відомо, метод C-value для АЗТ є затратним за часом обчислення (див. розділ 1.8). Крім того, його вимоги до використання оперативної пам'яті швидко зростають зі зростанням об'єму тексту, що обробляється. Отже, програмна реалізація програмного забезпечення АЗТ цього методу працює повільно та обмежена щодо об'єму вихідного тексту. Ця ознака наявного програмного забезпечення АЗТ особливо обмежує, коли набори даних зростають інкрементально і обробляються ітеративно, як у запронованому методі для здобуття насиченої термінології. Корисною ідеєю для подолання цього недоліку масштабованості може бути зміна порядку операцій у конвеєрі, як показано на Рис. 1.2.

Дійсно, якщо використаний метод АЗТ дозволяє адекватно об'єднати набори збережених значущих термінів, то він може застосовуватись тільки до інкрементів (Рис. 1.2(б)), але не до повних інкрементально зростаючих наборів даних (Рис. 1.2(а)). Оскільки частини колекції, що використовуються як прирости наборів даних невеликі, це знімає обидва обмеження програмної реалізації обраного методу АЗТ. Використовуючи цю ідею, ми підходимо до відповідей на цю групу питань шляхом:

- Доведення того, що метод C-value, що обраний для нашого конвеєру обробки, дозволяє здобувати часткові набори термінів з подальшим їх злиттям, у розділі 2.6
- Розробки відповідних алгоритмів у розділі 3.4
- Експериментальної перевірки та оцінювання валідності, продуктивності, незалежності від домену та масштабованості оптимізованого конвеєру обробки для здобуття насиченої термінології у розділі 4.7

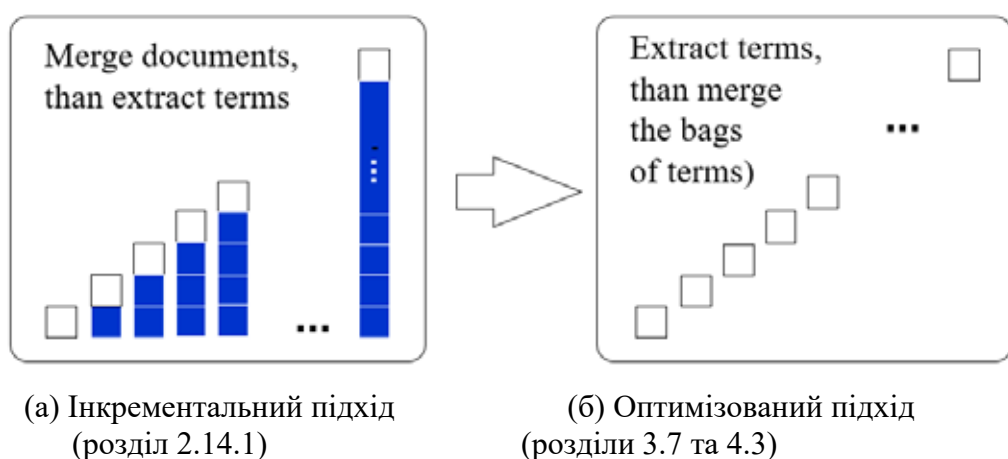


Рис. 1.2. Зміна інкрементального збільшення наборів даних на злиття наборів термінів для кращої масштабованості та підвищення продуктивності АЗТ

ПД8 (групування термінів): *Чи зможе покращити термінологічне насичення використання порогів подібності для групування частково схожих термінів при вимірюванні термінологічної різниці з точки зору меншої кількості ітерацій та кращої стабільності?*

Наше обговорення цього питання полягає в тому, що точне співпадіння текстових рядків дає більш вузький і неповний результат, ніж семантичний збіг у парі термінів. Дійсно, семантично однакові терміни можуть бути представлені в

тексті дещо різними рядками. Отже, спосіб оцінки семантичної подібності може бути більш переважним при обчисленні термінологічних відмінностей, ніж використання точного співпадіння рядків.

У цій роботі (розділ 3.6.2), ми шукаємо спосіб раціонального вибору порогових значень щодо схожості рядків як їх семантичної подібності (або подібності термінів). Порогові значення обираються для чотирьох відібраних мір подібності рядків (розділ 3.6.1). Потім, ми розробляємо алгоритм для групування подібних термінів, який використовує обрані міри подібності рядків та пороги схожості термінів (розділ 3.6.3). На основі його використання, ми вдосконалюємо базовий алгоритм **TND** (Tatarintseva et al. 2013) для вимірювання термінологічної різниці між двома наборами збережених значущих термінів (розділ 3.6.4). Нарешті, ми експериментально оцінюємо вдосконалений алгоритм **TND** та вимірювання подібності рядків у розділі 4.6.

На додаток до пошуку відповідей на сформульовані та розглянуті вище наукові та технічні питання, нам також хотілося б прагнути до практичного використання запропонованого методу здобуття насиченої термінології. У цьому контексті ми додатково формулюємо наступні три питання дослідження.

ПД9 (правильність розробленого конвеєру): *Чи дають правильні результати розроблений метод, алгоритми та програмне забезпечення для вимірювання термінологічного насичення, у граничних випадках колекцій документів: (i) такої, у якій стійке насичення повинно бути виявлено дуже швидко; та (ii) такої, у якій насичення не існує і не повинно бути виявленим?*

Щоб відповісти на це питання експериментально, було зібрано дві синтетичні колекції документів для окреслених граничних випадків: 1DOC та RAW (розділ 4.2.2). До цих колекцій застосовується конвеєр, розроблений у розділах 2 та 3, а результати перевірки на правильність представлені у розділі 4.3.

ПД10 (кейси використання): *Враховуючи те, що метод розроблений та довів свою ефективність, якими є пов'язані промислові та академічні кейси використання розроблених методу та програмної реалізації?*

Щоб відповісти на це питання, ми повідомляємо у розділах 5.1 та 5.2 про два кейси використання результатів дослідження, які було реалізовано: один у промисловому, та один у академічному проектах.

ПД11 (практичні наслідки): *Якими будуть практичні наслідки використання методу як частини інформаційної технології? Якими є потенційні промислові сценарії для трансферу технології?*

Практичні переваги для потенційних впроваджувачів результатів дослідження проаналізовані у розділі 5.3. Потенційні бізнес сценарії щодо застосування отриманих результатів у галузях наукового видавництва та цифрових бібліотек викладені в розділі 5.4.

1.13.3 Завдання дослідження

Завдання дослідження у роботі сформульовані таким чином, щоб доцільно згрупувати очікувані результати відповідей на наші питання дослідження (розділ 1.14.2) методологічно та структурно. Вони і є такими.

ЗД1 (формальний фреймворк): розробити формальний фреймворк методу виявлення та вимірювання термінологічного насичення, що охоплює визначення, формалізми та докази всіх його компонентів для відповіді на наші питання дослідження, необхідних для побудови якісного, ефективного та результативного обчислювального методу. Про цей фреймворк йдеться у розділі 2.

ЗД2 (алгоритмізація): спираючись на розроблений формальний фреймворк, розробити алгоритми, що матеріалізують обчислювальний метод здобуття насиченої термінології; реалізувати ці алгоритми у програмному забезпеченні та надати програмне забезпечення для академічного та промислового використання. Результати виконання цього завдання представлені у розділі 3.

ЗД3 (експериментальна оцінка та перевірка): використовуючи розроблене програмне забезпечення, експериментально оцінити та перевірити розроблений метод виявлення та вимірювання термінологічного насичення на синтетичних та реальних колекціях наукових публікацій, що належать до різних доменів. Про виконання цього завдання повідомляється у розділі 4.

ЗД4 (практичне використання): окреслити, якими є потенційні практичні наслідки використання розробленого методу на основі досвіду реалізації кейсів практичного використання в промисловості та академічній сфері; проаналізувати потенційні переваги впровадження науково-технічного доробку та потенційні бізнес-сценарії у цільовій промисловій галузі наукового видавництва. Виконання цього завдання детально описане у розділі 5.

1.14 Висновок

У цьому розділі ми розглянули та проаналізували сучасний стан наявних досліджень у напрямку роботи. Систематичний огляд відповідних робіт було проведено для збору релевантного досвіду та результатів у різних галузях досліджень. Аналіз цього стану досліджень та наявних результатів допоміг нам краще зрозуміти прогалини у нашій галузі дослідження (розділ 1.12).

З метою зменшення цих прогалин ми запропонували візію підходу до виявлення та вимірювання термінологічного насичення у розділі 1.13.1. Ця візія дозволила сформулювати питання дослідження на які потрібно отримати відповіді, щоб перетворити цей візіонарний підхід у метод, матеріалізувати його у відповідних алгоритмах, імплементувати у програмному забезпеченні, систематично оцінити розроблений метод, та отримати досвід його реального використання у промисловості та академії. Для цього питання дослідження ПД1-ПД11 (розділ 1.13.2) були згруповані у завдання дослідження ЗД1-ЗД4 (розділ 1.13.3).

Групування питань дослідження у завдання показані у Таблиці 1.5. У таблиці також наведено ключ для пошуку у дисертаційній роботі відповідей на питання дослідження.

Таблиця 1.5. Завдання дослідження, питання дослідження та відповіді

Завдання Дослідження	Питання Дослідження	Відповідь	
		Розділ	Результат
ЗД1 (формальний фреймворк)	ПД2 (оптимальний розмір інкременту)	2.6	Наслідок 2.6. Розмір часткової колекції.
	ПД4 (поріг відсікання)	2.1	Визначення 2.4: Індивідуальний поріг значущості терміну (<i>eps</i>)

	ПД5 (обчислення термінологічної різниці)	2.3, 2.4	Метрична функція Термінологічної різниці (<i>thd</i>)
	ПД6 (існування термінологічної насиченості)	2.5	Теорема 2.3. Достатні умови існування термінологічного насичення.
	ПД7 (оптимізація та масштабованість АЗТ)	2.6	Теорема 2.5. Рівність <i>cv</i> та <i>trcv</i> .
ЗД2 (алгоритмізація)	ПД1 (впорядкування документів)	3.3.2	Конфігурація та генерація набору даних (GDS)
	ПД4 (пори́г відсікання)	3.5, 3.6.4	Базовий алгоритм (THD). Вдосконалений алгоритм (R-THD).
	ПД5 (обчислення термінологічної різниці)	3.5, 3.6.4	Базовий алгоритм (THD). Вдосконалений алгоритм (R-THD).
	ПД7 (оптимізація та масштабованість АЗТ)	3.4	Ефективний алгоритм обчислення (часткових) C-Value (AC-CV). Алгоритм злиття часткових C-value (MPCV).
	ПД8 (групування термінів)	3.6	Відбір мір подібності рядків. Порогові значення подібності термінів. Алгоритми групування термінів (STG, M-JR, M-JW, M-JA, M-SD). Вдосконалений алгоритм вимірювання термінологічної різниці (R-THD).
ЗД3 (експериментальна оцінка та перевірка)	ПД3 (метод та програмне забезпечення АЗТ)	3.3, 4.4	Метод C-value реалізований у програмі UPM Term Extractor
	ПД9 (правильність конвеєру)	4.3	Розроблений обчислювальний конвеєр (метод, алгоритми та програмне забезпечення) дає правильні результати у граничних випадках
	ПД1 (впорядкування документів)	4.5	Впорядкування за зменшенням частоти цитування (dcf)
	ПД7 (оптимізація та масштабованість АЗТ)	4.7	Експериментальне доведення гіпотези <i>h1</i> Теорема 2.5. Експериментальне підтвердження ефективності, незалежності від домену та масштабованості оптимізованого обчислювального конвеєру.
	ПД8 (групування термінів)	4.6	Експериментальна оцінка групування термінів
ЗД4 (практичне використання)	ПД10 (кейси використання)	5.1, 5.2	Використання розробленого методу і програмного забезпечення у промисловому проекті (розділ 5.1).

		Використання розробленого метода і програмного забезпечення у академічній програмі (розділ 5.2).
ПД11 (практичні наслідки)	5.3, 5.4	Практичні переваги використання розробленого метода і програмного забезпечення у промисловості (розділ 5.3). Потенційні бізнес-сценарії в науковому видавництві (розділ 5.4).

2 ФОРМАЛЬНИЙ ФРЕЙМВОРК МЕТОДУ ВИЯВЛЕННЯ ТА ВИМІРЮВАННЯ ТЕРМІНОЛОГІЧНОГО НАСИЧЕННЯ

У цьому розділі ми розробляємо формальний фреймворк для методу виявлення та вимірювання термінологічного насичення як явища, що спостерігається у колекціях професійних текстів, обмежених доменами. Ідея цього методу та фреймворку натхнена наступними двома факторами.

- Насичення даними використовується якісними дослідниками, щоб вказати, коли ті, кого інтерв'юють, перестають наводити нові факти, які розширюють теорію, що будується. Якщо це трапляється, теорія вважається насиченою даними інтерв'ю і процес зупиняється.
- У цьому способі побудови теорій є помітна схожість із підходом, який використовується у загальноприйнятих методологіях побудови онтологій (див. розділ 1.6).

Отже, використання явища насичення у здобутті вимог щодо онтології може бути раціональним.

Розділ починається з базових відомостей (розділ 2.1), які містять визначення, що допомагають нам визначити фокус розроблюваного формального фреймворку. Далі ми представляємо гіпотези дослідження, які необхідно перевірити щодо цього фреймворку. Ці гіпотези формулюються на основі питань дослідження (розділ 1.13). У розділах 2.3 та 2.4, ми зосереджуємося на формальному введенні функції термінологічної різниці (*thd*) та доведенні її метричних властивостей у просторі всіх можливих колекцій документів для домену. У розділі 2.5, ми формулюємо і доводимо теорему, про достатні умови існування термінологічного насичення.

2.1 Базові відомості

У цій роботі ми шукаємо набір термінів, який репрезентативно описує довільний домен (*Dom*), для якого потрібно розробити або вдосконалити онтологію домену. Наше припущення полягає в тому, що, якщо ми маємо досить добре обмежений *Dom*, набір термінів, що використовується для його опису, є кінцевим

і не дуже великим за обсягом¹². Ці терміни можуть бути здобуті з документів (*Doc*), що належать до колекції документів ($DC = \{Doc\}$), що описуєть *Dom*. Гіпотетично можна зібрати всі існуючі документи, що описують *Dom* – повну колекцію документів.

Визначення 2.1: *Повна колекція документів для Dom. DC що містить усі документи, що описують Dom є повною DC (CDC).*

Для будь-якого реалістичного *Dom*, його *CDC* може бути дуже великою за обсягом. Отже, здобути з нього репрезентативний набір термінів було б трудомістким завданням. Ось чому ми розглядаємо підколекції документів *CDC* для *Dom*.

Визначення 2.2: *Підколекція документів. Підколекція документів (DSC) для Dom є піднабором повної колекції документів CDC: $DSC \subset CDC$.*

Нам цікаво знайти *DSC* мінімально можливого розміру, який містить практично той самий набір термінів, що і *CDC*. Набори термінів вважаються практично однаковими, якщо термінологічна різниця між наборами термінів, здобутих з цих *DSC* та *CDC* незначна. Отже, така *DSC* мінімально можливого розміру описує *Dom* так само повно, як *CDC* для цього *Dom*.

Наша посилка полягає в тому, що для добре обмеженого домену можна спостерігати термінологічне насичення у інкрементально розширюваній послідовності *DSC*. Термінологічно насичена *DSC* міститиме практично той самий набір термінів, що і *CDC*.

Перш ніж пояснити, як визначається та вимірюється термінологічна насиченість, слід визначити кілька базових понять та положень.

Термінологічний базис домену. Це поняття уточнює, що таке *DC* описуюча *Dom*. У цій роботі ми розуміємо цей опис як такий, що містить підмножину дійсних та суттєвих ознак, що характеризують *Dom*. Ці ознаки можуть бути надалі

¹² Наприклад, порівняно із загальною кількістю слів або значущих фраз у доступних документах, що описують *Dom*.

використані як вимоги до побудови онтології для Dom і позначені, використовуючи терміни, що містяться в документах DC , що описують домен.

Визначення 2.3: *Термінологічний базис.* Кінцевий набір термінів t_i , що ідентифікує всі ознаки, що характеризують Dom , утворюють термінологічний базис $TB = \{t_i\}$, $i = 1, \dots, dim \text{ of } Dom$.

Без втрати загальності, ми зосередимося на спрощеному випадку, в якому всі ці терміни t_i є ортогональними – тобто абсолютно різними у своєму сенсі. Як пояснено у розділі 3.4, більш складні випадки, в яких розглядаються частково подібні терміни, можна спростити, використовуючи техніку групування подібних термінів (Chugunenko et al. 2018).

За визначенням 2.3, TB містить усі терміни, що стосуються домену. Ці терміни можуть бути не однаково важливими для опису домену, як це відображено в документах, що містяться у CDC . Нехай дійсне додатне значення (*score*) пов'язується з кожним терміном, що використовується в документах CDC . Чим більш значущим є термін для опису домену, тим вище *score*. Отже, модель векторного простору (VSM) може бути підходящим формальним поданням простору документів для Dom . У цій моделі будь-який документ або DC , включаючи CDC , є точкою у векторному просторі з базисом $TB = \{t_i\}$ що має розмірність dim . У цьому контексті, корисно мати спосіб обчислити положення документа (колекції) у просторі документів та відстань між різними документами або колекціями.

Однак, створити TB для будь-якого реалістичного домену досить не легко. Ми повинні мати техніку яка здобуває t_i з документів, що описують Dom , переконатися, що здобуті $t_i \in TB$, і переконатися, що всі такі t_i були здобуті.

Здобуті терміни. Припустимо, що існує відображення яке перетворює DC в набір термінів (B), здобутих з документів цієї DC . У нашій роботі це відображення матеріалізується у конвеєрі АЗТ. Кожен елемент b в B це пара $\langle t, score \rangle$, де t – це кандидат у терміни та *score* – оцінка ймовірності того, що t є релевантним терміном для Dom : чим вище *score*, тим більше ймовірність того, що $t \in TB$. Терміни-кандидати, які мають високі *scores* позначаються як **значущі терміни**.

Збережені значущі терміни. Припустимо, що поріг значущості термінів (*eps*) раціонально обраний (або оцінений) для *scores* окремих термінів у *B*. Цей поріг вказує на межу, над якою терміни вважаються релевантними та дійсними, отже, належать до *TV*. Він також використовується багатьма статистичними методами АЗТ для відсікання тих $b \in B$ які, ймовірно, не є дійсними термінами у *Dom*. Побудувавши *B*, збережемо bs з $score > eps$ у відповідний набір збережених значущих термінів (*T*) – нижче див. Визначення 3.5.

У багатьох методах АЗТ, поріг значущості терміну обирається емпірично, або вибирається на основі міркувань здорового глузду. Ми пропонуємо обґрунтування вибору *eps*.

Проста більшість голосів. Згідно (Tatarintseva et al. 2013), підмножина термінів-кандидатів вважається такою, що містить усі важливі терміни, якщо терміни в ній відображають просту більшість голосів стейкхолдерів знань у домені. Наше припущення полягає в тому, що ці сентименти стейкхолдерів містяться в документах, що описують цей домен та авторами яких і є стейкхолдери знань у даному домені. Вказівники на ці сентименти – це є терміни, що використовуються для їх опису. Якщо термін використовується авторами більш інтенсивно, то його вплив на опис домену є більш значним. Чим більш значущим є термін, тим більше сентиментів стейкхолдерів знань цей термін побічно відображає. Отже, *score* може бути інтерпретована як сума **голосів** за цей термін стейкхолдерів знань у домені. Проста більшість виборців досягається, коли сума їх голосів перевищує половину суми всіх голосів.

Визначення 2.4: *Індивідуальний поріг значущості терміну.* Нехай

$$B_{desc} = \{b_i = \langle t_i, score_i \rangle\} \quad (2.1)$$

є набором здобутих термінів *B*, відсортованим у порядку зменшення їх *scores*. Тоді індивідуальний поріг значущості терміну для набору здобутих термінів *B* обчислюється наступним чином:

$$eps = score_i : \sum_{j=1}^i score_j > 1/2 \sum_{j=1}^{|B|} score_j, \quad (2.2)$$

де *i* – мінімальна кількість, така, що, виконується умова після двокрапки.

Визначення 2.4 вказує, що для обчислення eps для B , знайдено мінімальну підмножину bs у верхній частині B_{desc} , голоси, подані за які, становлять просту більшість. Отже, сума їх $scores$ перевищує $\frac{1}{2}$ загальної суми $scores$ в усьому наборі термінів. Поріг відсікання нижче цього мінімального піднабору bs визначається шляхом встановлення eps рівного $score$ значущості першого терміна¹³ нижче лінії відсікання.

Визначення 2.5: *Набір збережених значущих термінів.* Нехай $T \subset B_{desc} = \{b_i = \langle t_i, score_i \rangle\}$ такий, що виконується наступна умова (2.3):

$$\forall i, score_i > eps, \quad (2.3)$$

де eps індивідуальний поріг значущості терміна у B . Тоді T набір збережених значущих термінів з B .

Послідовне наближення до побудови TB . Якщо доступне відображення $DC \rightarrow T$, одним (наївним) способом побудови TB може бути здобуття та збереження всіх важливих термінів з CDC для Dom . Терміни у наборі збережених термінів T_{CDC} фактично становитимуть TB для Dom . Однак, це навряд чи можна виконати для будь-якого реалістичного Dom через, принаймні, дві причини:

(i) Потрібно переконатись, що колекція документів, яка є у наявності – це CDC , що складно; та

(ii) Обробка DC , яка кваліфікується як CDC для реалістичного домену, може бути дуже затратною у часі через її об'єм

Більш реальним способом є використання статистично репрезентативної DSC замість CDC . Тоді релевантні і валідні терміни, здобуті з цієї DSC , сформулюють базис дуже схожий на TB , із незначною різницею. Отже, нам потрібен спосіб з'ясувати чи є DSC репрезентативною. Одним з варіантів може бути перевірка, чи буде набір релевантних та валідних термінів, здобутий з DSC , статистично таким самим як і TB . Це важко перевірити за допомогою прямого порівняння, оскільки TB складно

¹³ Це також може бути групою термінів, що мають рівні оцінки (бали). Якщо це так, лінія відсікання проводиться під цією групою термінів.

побудувати. Тому доводиться застосовувати непрямий метод. У зв'язку з цим може бути використаний підхід на базі послідовного наближення.

Термінологічне Насичення. Нехай $DSC_1, \dots, DSC_i, \dots$ є послідовністю підколекцій документів такою, що $DSC_1 \subset \dots \subset DSC_i \subset \dots \subset CDC$; T_1, \dots, T_i, \dots є наборами збережених значущих термінів здобутих з $DSC_1, \dots, DSC_i, \dots$. Припустимо, що існує функція (*thd*) яка порівнює набори термінів T_i, T_{i+1} збережених з послідовних DSC_i, DSC_{i+1} і повертає різницю як дійсне додатне значення. Якщо, при деякому i :

(i) *thd* опускається нижче порогу статистичної похибки ε ; та

(ii) є переконливі докази того, що вона ніколи не перевищить цей поріг

то різниця (відстань) між T_i та T_{CDC} є не вище ніж ε . Набір термінів у такому T_i може використовуватися як ε – наближення TB . Можна стверджувати, що такий T_i , позначений далі як T_{sat} , є **насиченим набором термінів**, B_{sat} є набором термінів з якого зберігається T_{sat} , та DSC_{sat} є насиченою DSC для Dom . Різниця (*thd*) між T_{sat} та будь-яким послідовним T , включаючи T_{CDC} , знаходиться в межах допустимої та незначної похибки: $thd(T_{sat}, T_{CDC}) < \varepsilon$.

Поріг Термінологічного Насичення. Припущення для раціонального вибору порогу (ε) для виявлення термінологічного насичення полягає в тому, що набір термінів стає насиченим якщо він вже містить усі терміни із TB . Отже, які б терміни не були додані в наступних T , вони не є важливими. Встановимо $\varepsilon = eps_{B_{sat}}$ – індивідуальний поріг значущості терміна, обчислений для B_{sat} . Тоді T_{sat} буде містити статистично той самий набір термінів, що і TB .

Визначення 2.6: *Підколекція термінологічного ядра.* Підколекція термінологічного ядра (DSC_{sat}) – це підколекція DSC_i з якої було збережено насичений набір значущих термінів T_{sat} .

Умова Термінологічного Насичення. В OntoElect, індикатором термінологічного насичення для T_i є (Tatarintseva et al. 2013):

$$\begin{aligned} & \text{(i) } thd(T_i T_{i-1}) < eps_i; \text{ та} & (2.4) \\ & \text{(ii) } \forall j > i, thd(T_j T_{j-1}) < eps_j. \end{aligned}$$

2.2 Гіпотези дослідження

На базі попередніх відомостей, що формують базу формальної моделі процесу термінологічного насичення (розділ 2.1), ми висуваємо гіпотези, які необхідно довести, щоб сформувавши обґрунтований формальний фреймворк. Гіпотези сформовані таким чином, щоб довести можливість вирішення питань, пов'язаних з волатильністю значень thd (див. розділ 4.2.3), що спричиняє можливу нестабільність термінологічного насичення. Ці твердження формують логіку розробленого формального фреймворку.

Як зазначено в умові термінологічного насичення (2.4), насичення досягається, якщо певна поведінка функцій thd та eps : (i) спостерігається – необхідна умова; та (ii) доведено, що зберігається – достатня умова.

Отже, необхідно вивчити формальні властивості функцій thd та eps і знайти умови, за яких існування термінологічного насичення є доведеним.

Наступні гіпотези висуваються у вищезазначеному контексті для подальшого підтвердження у нашому формальному фреймворку. Якщо їх буде доведено, вони, принаймні частково, дадуть відповідь на деякі з наших питань дослідження, пов'язаних з завданням дослідження ЗД1 (Таблиця 1.5).

Гіпотеза Н2.1: thd є метричною функцією. Нехай простір документів CDS є векторним простором, що є сформованим з VSM репрезентацій усіх можливих документів та підколекцій документів CDC у Dom . Якщо thd введено як функцію відстані Манхеттен (Goma and Fahmy 2013), тоді $thd(T_i, T_j)$ є метричною функцією та CDS є метричним простором.

Для підтвердження цього твердження необхідно перевірити метричні умови для $thd(T_i, T_j)$: (i) невід'ємність; (ii) нерівність трикутника; (iii) симетрія; та (iv) ідентичність нерозрізних. Крім того, слід довести, що CDS це метричний простір з метрикою відстані $thd(T_i, T_j)$.

Гіпотеза Н3.2: Існування термінологічного насичення. Термінологічне насичення існує, якщо існує: (i) монотонно неспадаюча функція $eps_{min}(i)$; монотонно незростаюча функція $thds_{max}(i)$; перетин цих функцій на деякому i .

Для вимірювання термінологічного насичення нас цікавить обчислення thd не для довільних T_i, T_j , а для послідовних пар T_i, T_{i+1} , $i = 1, 2, \dots$. Отже практично цікавою для нас є функція $thds(i) = thd(T_i, T_{i+1})$. Зокрема нас цікавить коли $thds(i)$ опускається нижче $eps_{B_{sat}}$. Щоб це з'ясувати, ми маємо проаналізувати:

- Значення індивідуального порогу значущості терміну eps_i , що використовується для збереження термінів у T_i , які можна розглядати як функцію $eps(i) = eps_i, i = 1, 2, \dots: B_i \rightarrow \mathfrak{R}^+$

- Значення $thds(i), i = 1, 2, \dots: \{T_i, T_{i+1}\} \rightarrow \mathfrak{R}^+$

Як виявлено в наших експериментах (розділ 4), $eps(i)$ не обов'язково є монотонно неспадаючою функцією. Однак, можна було б побудувати її наближення $eps_{min}(i)$, як нижню огибаючу функцію, яка є монотонно неспадаючою функцією.

Аналогічно $eps(i)$, $thds(i)$ не обов'язково є монотонно незростаючою функцією. Її наближення також можливо побудувати, $thds_{max}(i)$ як верхню огибаючу функцію, що є монотонно незростаючою функцією.

Гіпотеза Н2.3: *Еквівалентність MPCV та C-value.* Згідно з визначенням 2.9 (розділ 2.7), злиті часткові C-value (Merged partial C-values (MPCV)) є сумами, за усіма партиціям колекції документів, часткових C-value (PCV, див. визначення 2.8, розділ 2.7) конкретних термінів. Гіпотеза полягає у тому що, для будь-якого терміну, його MPCV є таким-самим як і C-value, обчислене традиційним шляхом (Frantzi and Ananiadou 1999).

Якщо гіпотеза є вірною, це відкриває шлях для обчислення значень C-value використовуючи тільки партиції колекції документів, а не інкрементально збільшувані наближення до повної колекції. Такий підхід може дати суттєвий вигравш з точки зору на час обчислень та масштабованість методу у порівнянні з базовим методом.

2.3 Функція термінологічної різниці (thd)

Виходячи з базових відомостей, представлених у розділі 2.1, визначимо функцію thd для вимірювання відстані між наборами термінів. У якості вхідних даних, thd бере пару наборів термінів і повертає дійсне додатне значення відстані

між ними. Отже, це відображення з множини всіх можливих пар наборів (збережених значущих) термінів у множину додатних дійсних чисел:

$$thd: \{ \langle T_i, T_j \rangle \} \rightarrow \mathfrak{R}^+ \quad (2.5)$$

При побудові *thd* слід враховувати, що в нашому підході колекції документів з яких здобуваються набори термінів інкрементально зростають таким чином, що $DSC_1 \subset DSC_2 \dots \subset DSC_i \dots \subset CDC$. Тому,

$$DSC_{i+1} = DSC_i \cup INC, \quad (2.6)$$

де *INC* це набір документів, описуючих *Dom*, які додаються для розширення підколекції, щоб зробити її потенційно більш репрезентативною.

Через інкрементальний характер підколекцій DSC_i , зростатиме не лише кількість здобутих термінів у B_{i+k} в порівнянні з B_i , але також і *scores* термінів. Отже, для порівняння *scores* в парах, необхідно використовувати нормалізовані значення *score*. Нехай *maxscore* це *score* терміну, що має максимальне значення. У B_{desc} (2.1), *maxscore* це *score* першого елемента. Тоді нормалізований *score* (*ns*) можна обчислити як:

$$ns = score / maxscore \quad (2.7)$$

Тепер ми представимо *thd* як функцію Манхеттен відстані. За визначенням (Goma and Fahmy 2013), відстань Манхеттен (або таксікеб) це:

$$dman(v_i, v_j) = \sum_{k=1}^n |v_i^k - v_j^k|, \quad (2.8)$$

де $v_i = (v_i^1, v_i^2, \dots, v_i^n)$, $v_j = (v_j^1, v_j^2, \dots, v_j^n)$ є векторами у n -мірному дійсному векторному просторі із фіксованою декартовою системою координат.

Дотримуючись підходу VSM, розглянемо $TB = \{t^1, t^2, \dots, t^n\}$ як базис для простору документів, описуючих *Dom*. Тоді, будь-яка підколекція документів DSC_i , що має відповідний набір збережених значущих термінів T_i , може бути відображена у простір документів *CDS* з базисом *TB* як вектор $V_i = (ns_i^1, ns_i^2, \dots, ns_i^n)$, де *ns* – це нормалізовані показники значущості (2.7). T_i , який є набором збережених значущих термінів підколекції DSC_i , може не містити усіх термінів *TB*. Отже, деякі ns_i у V_i будуть дорівнювати нулю.

Нехай DSC_i та DSC_j це різні підколекції документів однієї CDC . Нехай також DSC_i та DSC_j будуються інкрементально (2.6). Тоді наборами їх збережених значущих термінів є T_i та T_j . Подібно до здобутих термінів B , з яких зберігається T , кожен елемент b в T_i є парою $b = \langle t_i^k, ns_i^k \rangle$, де $t_i^k \in TB_i \subseteq TB$ та ns_i^k є його нормалізованими показниками значущості.

Через інкрементальний характер DSC_i та DSC_j з одного боку та різні індивідуальні пороги значущості термінів з іншого боку, щодо T_i та T_j справедливо наступне:

$$\begin{aligned} int(TB_i, TB_j) &= int(TB_j, TB_i) = TB_i \cap TB_j \neq \emptyset, \\ dif(TB_i, TB_j) &= TB_i \setminus int(TB_i, TB_j) \neq \\ &\neq dif(TB_j, TB_i) = TB_j \setminus int(TB_j, TB_i), \\ int(TB_i, TB_j) \cap dif(TB_i, TB_j) \cap dif(TB_j, TB_i) &= \emptyset \end{aligned} \quad (2.9)$$

Отже, відстань між T_i та T_j можна обчислити як суму трьох часткових відстаней (2.10 – 2.12):

$$thd_{int(TB_i, TB_j)} = \sum_{k=1}^{\|int(TB_i, TB_j)\|} |ns_i^k - ns_j^k|, \quad (2.10)$$

де: $\|int(TB_i, TB_j)\|$ кількість елементів у $int(TB_i, TB_j)$; ns_i^k, ns_j^k індивідуальні нормалізовані показники значущості того ж самого терміну t^k у T_i та T_j ; $t^k \in int(TB_i, TB_j)$.

$$thd_{dif(TB_i, TB_j)} = \sum_{k=1}^{\|dif(TB_i, TB_j)\|} |ns_i^k - 0| = \sum_{k=1}^{\|dif(TB_i, TB_j)\|} ns_i^k, \quad (2.11)$$

де: ns_i^k індивідуальні нормалізовані показники значущості терміну t^k у T_i ; $t^k \in dif(TB_i, TB_j)$.

$$thd_{dif(TB_j, TB_i)} = \sum_{k=1}^{\|dif(TB_j, TB_i)\|} |ns_j^k - 0| = \sum_{k=1}^{\|dif(TB_j, TB_i)\|} ns_j^k, \quad (2.12)$$

де: ns_j^k індивідуальні нормалізовані показники значущості терміна t^k у T_j ; $t^k \in dif(TB_j, TB_i)$.

Усі три часткові відстані (2.10 – 2.12) є Манхеттен відстанями (Goma and Fahmy 2013) для різних підпросторів CDS .

Далі, відстань між T_i та T_j можна обчислити як:

$$\begin{aligned} thd(T_i, T_j) &= thd_{int(TB_i, TB_j)} + thd_{dif(TB_i, TB_j)} + thd_{dif(TB_j, TB_i)} = \\ &= \sum_{k=1}^n \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| + \sum_{k=1}^n \|dif(TB_i, TB_j)\| ns_i^k + \sum_{k=1}^n \|dif(TB_j, TB_i)\| ns_j^k \end{aligned} \quad (2.13)$$

Отже, $thd(T_i, T_j)$ це також Манхеттен відстань у CDS з базисом $TB = \{t^1, t^2, \dots, t^n\}$. Щоб переконатися, що $\forall k = 1, \dots, \|int(TB_i, TB_j)\| t_i^k = t_j^k$, T_i, T_j повинні бути попередньо відсортовані з використанням їх TB_i, TB_j як наборів ключів для сортування.

Отже, для обчислення thd згідно з (2.13), нам потрібно:

- Здобути B_i з DSC_i та B_j з DSC_j
- Обчислити індивідуальні пороги значущості термінів eps_i для B_i та eps_j для B_j використовуючи (2.2)
- Зберегти значущі терміни у T_i та T_j
- Обчислити ns використовуючи (2.7) для T_i та T_j
- Застосувати (2.13) для обчислення відстані T_i та T_j

2.4 Метричні властивості функції thd

У розділі 2.3 було введено функцію $thd(T_i, T_j)$ як Манхеттен відстань між наборами збережених значущих термінів T_i та T_j з нормалізованими показниками значущості термінів (ns). Отже, thd має усі властивості Манхеттен відстані (Goma and Fahmy 2013), яка задовольняє усі аксіоми Гільберта для евклідових просторів з декартовим базисом координат, за виключенням аксіоми про дві сторони кута (side-angle-side). Доведемо тепер, що $thd(T_i, T_j) \in$ **метричною функцією** для CDS . Для цього ми спочатку доводимо, що CDS , з мірою відстані $thd(T_i, T_j)$, є простором у наступній Лемі 2.1.

Лема 2.1. CDS , з мірою відстані $thd(T_i, T_j) \in$ простором.

Доведення. Оскільки $thd(T_i, T_j)$ це Манхеттен відстань, вона задовільняє (Goma and Fahmy 2013) усі аксіоми Гільбертового простору. Тоді, CDS , з мірою

відстані $thd(T_i, T_j)$, є Гільбертовим простором¹⁴. Отже, CDS , з мірою відстані $thd(T_i, T_j)$, є простором. ■

Доведемо тепер, що $thd(T_i, T_j)$ є метричною функцією в CDS у наступній Теоремі 2.2.

Теорема 2.2. $thd(T_i, T_j)$ є метричною функцією та CDS є метричним простором.

Доведення. За визначенням метричної функції (Moore and Cloud 2007), thd має бути функцією, що вимірює відстань між елементами множини, яка задовольняє наступним умовам:

- (i) Невід'ємність: $\forall T_i, T_j \in CDS \ thd(T_i, T_j) \geq 0$
- (ii) Нерівність трикутника: $thd(T_i, T_j) + thd(T_j, T_m) \geq thd(T_i, T_m)$
- (iii) Симетрія: $thd(T_i, T_j) = thd(T_j, T_i)$
- (iv) Ідентичність нерозрізних: $(thd(T_i, T_j) = 0) \equiv (T_i = T_j)$

Якщо умова (iv) не виконується, але замінюється на $thd(T_i, T_i) = 0$ тоді $thd(T_i, T_j)$ є псевдо-метричною.

(i) Невід'ємність. $thd(T_i, T_j)$ є невід'ємною функцією, оскільки жоден із доданків у (2.13) не може бути від'ємним. $|ns_i^k - ns_j^k|$ щонайменше дорівнює 0 (якщо $ns_i^k == ns_j^k$), або іншим чином додатне.

(ii) Нерівність трикутника.

$$\begin{aligned} thd(T_i, T_j) + thd(T_j, T_m) &= \\ &= \sum_{k=1} \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| + \sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k + \sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k + \\ &+ \sum_{k=1} \|int(TB_j, TB_m)\| |ns_j^k - ns_m^k| + \sum_{k=1} \|dif(TB_j, TB_m)\| ns_j^k + \sum_{k=1} \|dif(TB_m, TB_j)\| ns_m^k = \end{aligned}$$

¹⁴ CDS може бути розглянутий як простір тільки тоді, коли він містить нескінченне число елементів. Однак, для реалістичних доменів, кількість документів може вимірюватися сотнями тисяч, або більшими значеннями. З цієї причини, кількість усіх можливих комбінацій цих документів та підмножин документів буде дуже великою. Таким чином, розглядати CDS як множину, що містить нескінченну кількість елементів, є реалістичною апроксимацією.

$$\begin{aligned}
&= \left(\sum_{k=1} \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| + \sum_{k=1} \|int(TB_j, TB_m)\| |ns_j^k - ns_m^k| \right) + \\
&\quad + \left(\sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k + \sum_{k=1} \|dif(TB_j, TB_m)\| ns_j^k \right) + \\
&\quad + \left(\sum_{k=1} \|dif(TB_m, TB_j)\| ns_m^k + \sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k \right)
\end{aligned}$$

Щоб довести, що *thd* задовольняє умову нерівності трикутника (ii), достатньо довести, що усі (2.14 – 2.16) виконуються:

$$\begin{aligned}
\left(\sum_{k=1} \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| + \sum_{k=1} \|int(TB_j, TB_m)\| |ns_j^k - ns_m^k| \right) &\geq \\
&\geq \sum_{k=1} \|int(TB_i, TB_m)\| |ns_i^k - ns_m^k|
\end{aligned} \tag{2.14}$$

$$\left(\sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k + \sum_{k=1} \|dif(TB_j, TB_m)\| ns_j^k \right) \geq \sum_{k=1} \|dif(TB_i, TB_m)\| ns_i^k \tag{2.15}$$

$$\left(\sum_{k=1} \|dif(TB_m, TB_j)\| ns_m^k + \sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k \right) \geq \sum_{k=1} \|dif(TB_m, TB_i)\| ns_m^k \tag{2.16}$$

Згадаймо, що $TB_i \subseteq TB, TB_j \subseteq TB, TB_m \subseteq TB$ завдяки інкрементальному характеру підколекцій документів DSC_i . Отже, без будь-якої втрати значення відстані, (2.14 – 2.16) можуть бути перетворені, як показано нижче в (2.17 – 2.19). Ліву частину (2.14) можна переписати таким чином:

$$\begin{aligned}
&\left(\sum_{k=1} \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| + \sum_{k=1} \|int(TB_j, TB_m)\| |ns_j^k - ns_m^k| \right) = \\
&= \left(\sum_{k=1} \|TB\| |ns_i^k - ns_j^k| + \sum_{k=1} \|TB\| |ns_j^k - ns_m^k| \right) = \\
&= \left(\sum_{k=1} \|TB\| (|ns_i^k - ns_j^k| + |ns_j^k - ns_m^k|) \right),
\end{aligned} \tag{2.17}$$

де: ns_i^k, ns_j^k, ns_m^k відповідають тим $t^k \in TB$, які також належать до TB_i, TB_j, TB_m ; та $(|ns_i^k - ns_j^k| + |ns_j^k - ns_m^k|) \geq |ns_i^k - ns_m^k|$ оскільки виконується нерівність трикутника для відстаней між дійсними числами. Це доводить (2.14).

Подібно, ліву частину (2.15) можна переписати наступним чином:

$$\begin{aligned}
\left(\sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k + \sum_{k=1} \|dif(TB_j, TB_m)\| ns_j^k \right) &= \sum_{k=1} \|TB\| (ns_i^k + ns_j^k) \geq \\
&\geq \sum_{k=1} \|TB\| ns_i^k,
\end{aligned} \tag{2.18}$$

як будь яке $ns_j^k > eps > 0$. Це доводить (2.15).

Отже, переписавши ліву частину (2.16) аналогічно (2.18) маємо:

$$\begin{aligned}
\left(\sum_{k=1} \|dif(TB_m, TB_j)\| ns_m^k + \sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k \right) &= \sum_{k=1} \|dif(TB)\| (ns_m^k + \\
&ns_j^k) \geq \\
&\geq \sum_{k=1} \|C_{CTS}\| ns_m^k .
\end{aligned} \tag{2.19}$$

Це доводить (2.16) і, нарешті, доводить, що для *thd* виконується нерівність трикутника.

(iii) Симетрія. Симетрія *thd* доводиться використовуючи (2.13) і той факт, що операція об'єднання для множин є симетричною, тобто $int(TB_i, TB_j) = TB_i \cap TB_j = TB_j \cap TB_i = int(TB_j, TB_i)$:

$$\begin{aligned}
thd(T_i, T_j) &= \\
&= \sum_{k=1} \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| + \sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k + \sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k = \\
&= \sum_{k=1} \|int(TB_j, TB_i)\| |ns_j^k - ns_i^k| + \sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k + \sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k = \\
&= thd(T_j, T_i).
\end{aligned}$$

(iv) Ідентичність нерозрізних. З (2.13) випливає, що:

$$\begin{aligned}
thd(T_i, T_i) &= thd_{int(TB_i, TB_i)} + thd_{dif(TB_i, TB_i)} + thd_{dif(TB_i, TB_i)} = \\
&= \sum_{k=1} \|int(TB_i, TB_i)\| |ns_i^k - ns_i^k| + 0 + 0 = 0 + 0 + 0 = 0.
\end{aligned} \tag{2.20}$$

Тому, $thd(T_i, T_i) = 0$.

З іншого боку, відповідно до (3.13), $(thd(T_i, T_j) = 0)$ означає, що

$$\begin{aligned}
\left(\sum_{k=1} \|int(TB_i, TB_j)\| |ns_i^k - ns_j^k| = 0 \right) \wedge \left(\sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k = 0 \right) \wedge \\
\wedge \left(\sum_{k=1} \|dif(TB_j, TB_i)\| ns_j^k = 0 \right).
\end{aligned}$$

$\sum_{k=1} \|dif(TB_i, TB_j)\| ns_i^k = 0$ означає, що $dif(TB_i, TB_j) = \emptyset$ тому, що будь-яке $ns_i^k > eps_i > 0$. Те саме вірно також для $dif(TB_i, TB_j)$. Отже, $TB_i = TB_j = int(TB_i, TB_j)$, що означає, що T_i та T_j містить однаковий набір термінів.

$\sum_{k=1}^{\|int(TB_i, TB_j)\|} |ns_i^k - ns_j^k| = 0$ означає, що $\forall t^k \in int(TB_i, TB_j) ns_i^k = ns_j^k$. Це може статися, лише якщо $i = j$ оскільки різні T_i та T_j формуються за допомогою документів з різних DSC_i та DSC_j . Навіть якщо документи в цих колекціях всі однакові¹⁵ (тому всі терміни у T_i та T_j однакові) значення важливості термінів розрізняються.

Отже, $(thd(T_i, T_j) = 0) \rightarrow (T_i = T_j)$ як $i = j$ та $T_i = T_j$ за визначенням T .

Таким чином, $thd(T_i, T_j)$ є метричною функцією у CDS . Отже, CDS є метричним простором, оскільки $thd(T_i, T_j)$ є метричною функцією у ньому. ■

Доведення Теорема 2.2 формально валідує нашу гіпотезу дослідження **H2.1** (розділ 2.2) як дійсну.

2.5 Умови існування термінологічного насичення

Перевіримо тепер нашу гіпотезу **H2.2**, яка стосується достатніх умов існування термінологічного насичення. Для цього ми переформулюємо цю гіпотезу у вигляді теореми існування та доведемо цю теорему.

Теорема 2.3. *Достатні умови існування термінологічного насичення.* Нехай:

- (i) $DSC_1 \subset DSC_2 \dots \subset DSC_i \dots$ є послідовністю підколекцій документів, кожна з яких описує той самий довільний домен Dom
- (ii) $V_1, V_2, \dots, V_i, \dots$ є послідовністю наборів термінів, здобутих з $DSC_1, DSC_2, \dots, DSC_i \dots$ та $eps(i)$ – функцією індивідуальних порогів значущості термінів для $V_i, i = 1, 2, \dots$
- (i) $T_1, T_2, \dots, T_i, T_{i+1}, \dots$ є послідовністю наборів збережених значущих термінів, для яких попарна послідовна термінологічна різниця обчислюється за допомогою функції $thds(i)$.

Тоді послідовність DSC_i є термінологічно насиченою, i є точкою насичення, та $TB_i = TB_{sat} = TB$, якщо:

¹⁵ Це може бути так, якщо документи копіюються для формування синтетичних підколекцій, як це було зроблено у (Kosa et al. 2018a).

- (i) Існує неспадаюча нижня огинаюча функція $eps_{min}(i)$ для $eps(i)$
- (ii) Існує незростаюча верхня огинаюча функція $thds_{max}(i)$ для $thds(i)$
- (iii) $eps_{min}(i) \geq thds_{max}(i)$

Доведення.

Припустимо, що $eps(i)$ та $thds(i)$ є у наявності – тобто їх значення для $i = 1, 2, \dots$ можна обчислити. Виходячи з цього, побудуємо шукані огинаючі функції $eps_{min}(i)$ та $thds_{max}(i)$.

Нехай $eps_{min}(1) = eps(1)$. $eps_{min}(i)$ буде неспадаючою якщо:

$$\forall i > 1 \quad eps(i) \geq eps_{min}(1) \quad (2.21)$$

Припустимо, що $\exists j > 1: \forall i > 1 \quad eps(j) = \min(eps(i))$. Тоді $eps_{min}(j) = eps(j)$ та:

$$\forall i = 2, \dots, j - 1 \quad eps_{min}(i) = eps_{min}(1) + (i - 1) \frac{eps_{min}(j) - eps_{min}(1)}{j - 1} \quad (2.22)$$

Далі припустимо, що $eps_{min}(k)$ обчислено. Потім ми повторюємо попередню ітерацію наступним чином:

Ми шукаємо мінімальний $j > k$ такий, що $\forall i > k \quad eps(j) = \min(eps(i))$. Для цього ми призначаємо $eps_{min}(j) = eps(j)$ і значення $eps_{min}(i)$ для $i = k + 1, \dots, j - 1$ обчислюються як:

$$\forall i = k + 1, \dots, j - 1 \quad eps_{min}(i) = eps_{min}(k) + (i - k) \frac{eps_{min}(j) - eps_{min}(k)}{j - k} \quad (2.23)$$

Тепер побудуємо $thds_{max}(i)$ подібно до побудови $eps_{min}(i)$. Нехай спочатку $thds_{max}(1) = thds(1)$. $thds_{max}(i)$ буде незростаючою, якщо:

$$\forall i > 1 \quad thds(i) \leq thds_{max}(1) \quad (2.24)$$

Далі припустимо, що $thds_{max}(k)$ для деякого $k = 1, \dots$ було обчислено. Ми шукаємо мінімальний $j > k$ такий, що $\forall i > k \quad thds(j) = \max(thds(i))$.

Для цього ми призначаємо $thds_{max}(j) = thds(j)$ і значення $thds_{max}(i)$ для $i = k + 1, \dots, j - 1$ обчислюються як:

$$\forall i = k + 1, \dots, j - 1$$

$$thds_{max}(i) = thds_{max}(k) + (i - k) \frac{thds_{max}(j) - thds_{max}(k)}{j - k} \quad (2.25)$$

Ми довели існування $eps_{min}(i)$ та $thds_{max}(i)$ побудувавши їх за умов (2.21) та (2.24) відповідно. Отже умови (i) та (ii) теореми виконуються, якщо виконуються (2.21) та (2.24).

Припустимо тепер, що $\exists i > 1: eps_{min}(i) \geq thds_{max}(i)$. Тоді,

$$\forall j > i, eps_{min}(j) \geq eps_{min}(i) \geq thds_{max}(i) \geq thds_{max}(j) \quad (2.26)$$

через (2.23) та (2.25). Отже, умова (iii) теореми виконується, $eps_{min}(i) = \varepsilon = eps_{B_{sat}}$ є порогом термінологічного насичення, i є точкою насичення, та $TB_i = TB_{sat} = TB$.

■

Теорема 2.3 забезпечує підхід до виявлення термінологічного насичення у послідовності $DSC_1 \subset DSC_2 \dots \subset DSC_i \dots$, якщо воно може бути досягнуто. Однак це не допомагає впевнено довести, що термінологічне насичення у будь-якій послідовності підколекцій документів дійсно є досяжним, перш ніж всі підколекції будуть опрацьовані. Тому, було б корисним з'ясувати чи можна передбачити термінологічне насичення виходячи з невеликої кількості початкових ітерацій процесу послідовного наближення. Цей напрямок досліджень заплановано на майбутню роботу.

2.6 Масштабованість та оптимізація

Метод C-value (Frantzi and Ananiadou 1999), який у нашій роботі був обраний для здобуття термінів (див. розділ 1.8), є гібридним, оскільки він використовує міри юнітхуду (unithood) та термхуду (termhood) в одній формулі ранжування (2.28). Він також поєднує лінгвістичні та статистичні етапи, що застосовуються до всієї колекції документів (текстового корпусу). Метод починається з лінгвістичного конвеєру, який видає список строк-кандидатів у терміни. Потім він переходить до статистичної частини, яка обчислює показники значущості для цих строк-кандидатів у терміни як C-values. Діаграма витраченого часу на виконання в залежності від обсягу вхідного тексту, представлена у розділі 4 (Рис. 4.29) у

випадку цього конвеєру ілюструє, за значеннями часу виконання, значну обчислювальну складність методу C-value. Завданням цього розділу є модифікувати метод C-value (Frantzi and Ananiadou 1999) таким чином, щоб значно знизити пов'язані з ним обчислювальні витрати.

Розглянемо колекцію документів DC , як композицію її частин, що не перетинаються.

Визначення 2.7. Часткова колекція та партиція колекції документів. $DC_i, i = 1, \dots, n$ є частковими колекціями документів DC та $\{DC_i\} = \{DC_i\}_{i=1}^n$ є партицією DC , якщо виконуються наступні умови:

$$\text{Умова 1: } DC = \bigcup_{i=1}^n DC_i, \quad (2.27)$$

$$\text{Умова 2: } \bigcap_{i=1}^n DC_i = \emptyset.$$

Лінгвістична частина методу C-value обробляє окремі речення. Тому: (i) її обчислювальна складність є функцією кількості речень у колекції документів; та (ii) часткові колекції $DC_i, i = 1, \dots, n$ (Визначення 2.7) можуть оброблятися незалежно та вихідні дані потім об'єднуватися. Отже, застосування лінгвістичного кроку до партиції DC могло б бути, принаймні, розпаралеленим, що призводить до зменшення часу виконання в n разів.

У нашому конвеєрі, лінгвістичний крок ітеративно застосовується до поступово збільшених наборів даних (див. розділ 2.1). Тому однакові фрагменти тексту обробляються багато разів. Припустимо, що DC містить k документів та $inc = k/n$ інкремент для збільшення наборів даних. Тоді, кількість документів, яку потрібно обробити:

- У випадку інкрементально збільшених наборів даних:
 $inc + 2 \cdot inc + \dots + n \cdot inc = k \cdot \frac{1+2+\dots+n}{n} \approx (n+1)/2 \cdot k$, що значно більше ніж k якщо $n > 1$

- У випадку часткових колекцій: k

Таким чином, обробка часткових колекцій замість інкрементально збільшених наборів даних дає істотний вигравш в часі виконання, що складає $(\frac{n+1}{2} - 1) \cdot k$ разів.

У статистичній частині методу, ранг C-value (Frantzi and Ananiadou 1999), далі позначений як cv у формулах і рівняннях, обчислюється для кожної строки-кандидата у терміни, що є здобутою лінгвістичною частиною конвеєру. C-value формула (2.28) використовує декілька статистичних характеристик відповідної строки-кандидата у терміни. Цими характеристиками є: загальна частота (кількість) появ строки кандидата у корпусі документів; частота (кількість) появ строки-кандидата як частини іншої, більш довгої строки-кандидата у терміни; кількість таких довших строк-кандидатів у терміни; довжина строки-кандидата (кількість слів). Нехай: s буде строкою кандидатом у терміни; $|s|$ – довжина s в словах; ls – довша строка кандидат у терміни, в якій s міститься як підстрока; $f(\cdot)$ – частота (кількість) появ строки термінів кандидатів у DC ; T^s – набір здобутих строк-кандидатів ls , що містять s ; та $P(T^s)$ – кількість цих ls . Тоді (повний) cv для s визначається (Frantzi and Ananiadou 1999) наступним чином:

$$cv(s) = \begin{cases} \log_2(|s|) \cdot f(s) & \text{якщо } s \text{ не міститься в будь-якій } ls \text{ здобутій з } DC \\ \log_2(|s|) \cdot \left(f(s) - \frac{1}{P(T^s)} \sum_{ls \in T^s} f(ls) \right) & \text{в іншому випадку} \end{cases} \quad (2.28)$$

Отже, обчислювальна складність $cv(s)$ залежить від $P(T^s)$, яка може суттєво зростати із зростанням набору текстових даних, що обробляється. Тому може бути доцільним застосувати статистичний крок конвеєру не до інкрементально збільшених наборів даних, а до часткових колекцій.

Проте, не є очевидним що: (i) обчислення C-values для термінів здобутих з часткових колекцій; та (ii) подальше злиття цих наборів термінів з їх показниками значущості – дасть той самий результат, що і застосування статистичного кроку до інкрементально збільшених наборів даних. У решті цього розділу, ми доводимо, що поділ обчислення C-value з подальшим злиттям обчислень дає правильні результати з точністю до вірності гіпотези **h1** у (2.34). Цю гіпотезу валідовано в нашому експериментальному дослідженні, яке представлено у розділі 4.7.

Визначення 2.8. *Часткове C-value.* Часткове C-value строки термінів кандидатів s здобутих з часткової колекції документів DC_i обчислюється як:

$$pcv_i(s) = \begin{cases} \log_2(|s|) \cdot f_i(s) & \text{якщо } s \text{ не міститься у будь-якій } ls \text{ здобутих з } DC_i \\ \log_2(|s|) \cdot \left(f_i(s) - \frac{1}{P(T_i^s)} \sum_{ls \in T_i^s} f_i(ls) \right) & \text{в іншому випадку} \end{cases}, (2.29)$$

де: T_i^s набір строк термінів кандидатів ls , що міститься у s , здобутих з DC_i ; $f_i(\cdot)$ Кількість появ s або ls у DC_i .

Лема 2.4. *Загальна частота вкладених появ.* Загальне значення частоти вкладених появ, у DC , строки-кандидата у терміни s у довшій строки-кандидаті у терміни ls є сумою значень загальної частоти вкладених появ s в ls у всіх часткових колекціях DC_i з DC :

$$tnest(s) = \sum_{ls \in T^s} f(ls) = \sum_{i=1}^n \left(\sum_{ls \in T_i^s} f_i(ls) \right) = \sum_{i=1}^n (tnest_i(s)). \quad (2.30)$$

Доведення.

З Визначення 2.8 (часткове C-value) випливає, що $tnest_i(s)$ це загальна кількість появ строки кандидата у терміни s у всіх довших строках кандидатів у терміни ls здобутих з часткової колекції DC_i . Кількість цих довших строк кандидатів у терміни дорівнює $P(T_i^s)$. Внаслідок того, що часткові колекції DC_i непересічні (Умова 2 Визначення 2.7 (2.27)), $f(ls) = \sum_{i=1}^n f_i(ls)$. Отже, в силу Умови 1 Визначення 2.7 (2.27), загальна кількість появ s у всіх ls здобутих з DC буде:

$$\begin{aligned} tnest(s) &= \sum_{ls \in T^s} f(ls) = \sum_{ls \in T^s} \sum_{i=1}^n f_i(ls) = \sum_{ls \in \cup_{i=1}^n (T_i^s)} \left(\sum_{i=1}^n f_i(ls) \right) = \\ &= \sum_{i=1}^n \left(\sum_{ls \in T_i^s} f_i(ls) \right) = \sum_{i=1}^n (tnest_i(s)). \end{aligned}$$

■

Визначення 2.9. *Злиті часткові C-value.* Злиті часткові C-value строк-кандидатів у терміни (s) обчислюються як:

$$trpcv(s) = \sum_{i=1}^n pcv_i(s). \quad (2.31)$$

Наступна Теорема 2.5 дозволяє обчислити $cv(s)$ для всієї колекції DC на основі злиття відомих часткових C-values $pcv_i(s), i = 1, \dots, n$ для часткових колекцій $DC_i, i = 1, \dots, n$ of DC .

Теорема 2.5. *Рівність cv та $trpcv$.* Якщо колекція документів DC поділена як $\{DC_i\} = \{DC_i\}_{i=1}^n$, що означає виконання Умов 1 та 2 (2.27), то

$$cv(s) = mpcv(s) \quad (2.32)$$

Доведення.

Доведення є структурованим на три випадки: (i) s ніколи не міститься у ls ; (ii) $\forall DC_i$, s міститься принаймні один раз і принаймні в одній ls ; та (iii) s міститься в ls для деяких DC_i .

Випадок (i): не міститься. Якщо, $\forall i = 1, \dots, n$, s здобутих з DC_i не міститься в жодній ls здобутих з DC_i , тоді s не міститься в жодній ls здобутих з DC . Тому, для кожної з таких s :

$$\begin{aligned} mpcv(s) &= \sum_{i=1}^n pcv_i(s) = \sum_{i=1}^n \log_2(|s|) \cdot f_i(s) = \\ &= \log_2(|s|) \cdot \sum_{i=1}^n f_i(s) = \log_2(|s|) \cdot f(s) = cv(s), \end{aligned} \quad (2.33)$$

внаслідок Умов 1 та 2 (2.27) та за визначенням $f(\cdot)$.

Випадок (ii): міститься у всіх. Якщо, $\forall i = 1, \dots, n$, s здобутої з DC_j міститься у ls здобутої з DC_j , тоді: (а) ця s (здобута з DC) міститься у цій ls (здобутої з DC); та (б) $ls \in T_j^s \subset T^s$ – тому що $DC_i \subset DC$ внаслідок Умови 1 (2.27). Тоді:

$$\begin{aligned} mpcv(s) &= \sum_{i=1}^n pcv_i(s) = \\ &= \sum_{i=1}^n \left(\log_2(|s|) \cdot \left(f_i(s) - \frac{1}{P(T_i^s)} \sum_{ls \in T_i^s} f_i(ls) \right) \right) = \\ &= \log_2(|s|) \cdot \sum_{i=1}^n \left(f_i(s) - \frac{1}{P(T_i^s)} \sum_{ls \in T_i^s} f_i(ls) \right) = \\ &= \log_2(|s|) \cdot \left(\sum_{i=1}^n f_i(s) - \sum_{i=1}^n \left(\frac{1}{P(T_i^s)} \sum_{ls \in T_i^s} f_i(ls) \right) \right) = \\ &= \log_2(|s|) \cdot \left(f(s) - \sum_{i=1}^n \left(\frac{1}{P(T_i^s)} \sum_{ls \in T_i^s} f_i(ls) \right) \right) \approx_{|h1} \\ &\approx_{|h1} \log_2(|s|) \cdot \left(f(s) - \frac{1}{P(T^s)} \sum_{i=1}^n \left(\sum_{ls \in T_i^s} f_i(ls) \right) \right) = \\ &= \log_2(|s|) \cdot \left(f(s) - \frac{1}{P(T^s)} \sum_{ls \in T^s} (f(ls)) \right) = cv(s) \end{aligned} \quad (2.34)$$

Тут “ $\approx_{|h1}$ ” означає гіпотетично приблизно рівний. Гіпотеза $h1$ про приблизну рівність означає $\sum_{i=1}^n \left(\frac{1}{P(T_i^s)} \right) \approx \frac{1}{P(T^s)}$. Формально, $\sum_{i=1}^n \left(\frac{1}{P(T_i^s)} \right) > \frac{1}{P(T^s)}$. Однак, асимптотично, $P(T_i^s) = o(tnest_i(s))$ та $P(T^s) = o(tnest(s))$ внаслідок: (а) збігів у T_i^s ; та (б) можливих вкладень у декількох випадках ls . Отже, вплив цих знаменників

у (2.34) стає меншим зі зростанням об'єму DC та її часткових колекцій DC_i . Це дає надію, що $h1$ може бути вірною.

Випадок (iii): s інколи міститься у ls . Існує декілька часткових колекцій DC_j , для яких застосовується Випадок (ii). Для решти часткових колекцій DC_k застосовується Випадок (i). В цій ситуації розраховуються дві часткові суми – $trpcv_1(s)$ та $trpcv_2(s)$ – для цих піднаборів поділу DC . Подібно до Випадку (ii), $cv(s) \approx_{|h1} trpcv_1(s) + trpcv_2(s)$.

Отже, якщо виконується гіпотеза $h1$, Випадки (i)-(iii) доводять Теорему 2.5. ■

Прямий наслідок з Теорема 2.5 полягає в тому, що C-value не залежать від поділу колекції документів на частини.

Наслідок 2.6. *Розмір часткової колекції.* Нехай $\{DC_i\}_{i=1}^n; \{DC_j\}_{j=1}^m$; $n \neq m$ дві різні партиції колекції документів DC . Тоді:

$$\forall s, trpcv(s)|_{\{DC_i\}} = trpcv(s)|_{\{DC_j\}} \approx_{|h1} cv(s), \quad (2.35)$$

де: s є строка-кандидат у терміни, здобута з колекції документів DC ; $trpcv(s)|_{\{DC_i\}}$ злите часткове C-value строки-кандидата у терміни s обчислене для партиції $\{DC_i\}$ колекції DC ; $trpcv(s)|_{\{DC_j\}}$ злите часткове C-value строки-кандидата у терміни s обчислене для партиції $\{DC_j\}$ з DC .

Базуючись на Наслідку 2.6, розмір часткової колекції $DC_i \in \{DC_i\}$ може бути раціонально обраний в залежності від специфіки проблеми та доступних апаратних ресурсів, зокрема, оперативної пам'яті. Одним з можливих сценаріїв може бути здобуття термінів з потоку текстових документів, наприклад, публікацій у блогах або твітах. У цьому випадку розмір часткової колекції має бути меншим за розмір вікна потоку.

2.7 Висновок

У цьому розділі ми розробили формальний фреймворк для виявлення та вимірювання термінологічного насичення у відповідь на наші питання дослідження

ПД4, 5, 6, 7 та 2. Фактично, ці питання стосувались трьох важливих складових формального фреймворку: (i) функції вимірювання термінологічної

різниці (*thd*) та її формальних властивостей; (ii) достатніх умов існування термінологічного насичення; та (iii) формального підходу для оптимізації обчислень у процесі здобуття термінів. Отже, щоб відповісти на питання дослідження ясно і структуровано, було сформульовано та перевірено гіпотези дослідження **H2.1**, **H2.2** та **H2.3**.

H2.1 було формально перевірено у розділах 2.3 та 2.4. Функція вимірювання термінологічної різниці (*thd*) між наборами термінів була формально визначена у розділі 2.3 як різновид функції Манхеттен відстані. У визначенні *thd* опосередковано було використано визначення індивідуального порогу значущості (*eps*), що було введено у розділі 2.1. Метричні властивості функції *thd* формально доведені доказами Лема 2.1 та Теореми 2.2 у розділі 2.4.

H2.2 було формально підтверджено у розділі 2.5 доказом Теореми 2.3 (існування). Ця відповідь, однак, не дозволила обчислювати прогнози точок насичення у довільних колекціях документів на основі кількох початкових ітерацій. Такі прогнози виходять за рамки цієї роботи і залишаються для подальшої роботи, як пропонується у (Kosa and Ermolayev, 2020).

H2.3 було формально протестовано у розділі 2.6. Для перевірки гіпотези були введені поняття партиції колекції документів, часткового C-value, та часткового злитого C-value. Крім того, Лема 2.3 та Теорема 2.4 формально довели, що звичайні C-value еквівалентні злитим частковим C-value з точністю до справедливості гіпотези *h1* про приблизну рівність у $\sum_{i=1}^n \left(\frac{1}{P(T_i^S)} \right) \approx \frac{1}{P(T^S)}$.

На запитання **ПД2** про оптимальний розмір інкременту набору даних відповів доказ Наслідку 2.6 до Теореми 2.5. Цей наслідок визначив, що значення злитих часткових C-value не залежать від розміру інкременту (часткової колекції у партиції). Отже, розмір часткових колекцій у партиції може бути розумно обраний виходячи із особливостей проблеми. Цей результат фактично зняв обмеження програмних реалізацій методу C-value і дозволив ефективно використовувати метод MPCV до реальних колекцій документів промислового розміру.

Підсумовуючи, можна сказати, що на всі питання дослідження, що були висунуті у цьому розділі було дано позитивні відповіді, що підтвердили справедливність трьох гіпотез дослідження. Таким чином, опрацювавши всі необхідні компоненти формального фреймворку, ми виконали завдання дослідження ЗД1.

3 АЛГОРИТМИ ДЛЯ ВИЯВЛЕННЯ ТА ВИМІРЮВАННЯ ТЕРМІНОЛОГІЧНОГО НАСИЧЕННЯ

У цьому розділі ми розробляємо нові або вдосконалюємо раніше розроблені алгоритми, що матеріалізують розроблений формальний фреймворк (розділ 3) для обчислень.

Ми починаємо з розробки обчислювального конвеєру у розділі 3.1. Для цього ми пропонуємо робочий процес для виявлення та вимірювання термінологічного насичення (Рис. 3.1). Робочий процес детально описаний у функціональній блок-схемі, яка розкриває модульну структуру нашого алгоритмічного набору та залежності між його модулями. Опис алгоритмів, що представляють модулі, більш розлого надається наступним чином. У розділі 3.2 представлені алгоритми розроблені для інструментування підготовки даних для подальшої обробки. Сюди входить генерація каталогу колекції документів та завантаження повнотекстових документів (у PDF) на основі каталогізованих метаданих. У розділі 3.3 представлена фаза перед-обробки, що включає перетворення PDF у звичайний (плоский) текст, конфігурація ходу обчислень та генерація наборів даних із документів плоского тексту. Оптимізований алгоритм для статистичної частини конвеєру здобуття термінів та алгоритм для обчислення злитих часткових C-value детально описано у розділі 3.4. Базовий алгоритм для обчислення термінологічної різниці між двома наборами термінів представлено у розділі 3.5. У розділі 3.6 вдосконалено цей базовий алгоритм, шляхом включення розробленої техніки групування термінів, включаючи алгоритми вимірювання подібності строк та вдосконалений алгоритм для вимірювання термінологічної різниці. Алгоритм для видалення накопиченого регулярного шуму з наборів термінів представлено у розділі 3.7. У розділі 3.8 викладена реалізація алгоритмічного набору у програмному забезпеченні та пропонуються посилання на це програмне забезпечення задля забезпечення загальної доступності розроблених програмних інструментів та відтворюваності отриманих результатів. Наприкінці, у розділі 3.9 зроблено висновок про результати, які були представлені у розділі 3.

3.1 Потік обчислень для виявлення та вимірювання термінологічного насичення

Як зазначено у формальному фреймворці, для здобуття насиченої термінології (розділ 2), мета цієї дії полягає в тому, щоб отримати підколекцію документів термінологічного ядра з більш великої колекції документів. Якщо досягається термінологічне насичення, то підколекція термінологічного ядра містить набір термінів, який незначно відрізняється від набору термінів, який можна було б здобути зі всієї колекції. Для цього було запропоновано та розроблено формально процес ітеративного наближення до набору термінів підколекції термінологічного ядра (розділ 2). Кожна ітерація (i) у цьому процесі базується на вимірюванні термінологічної різниці ($thd(T_i, T_{i+1})$) між двома наборами збережених значущих термінів (T_i and T_{i+1}) здобутих з двох підколекцій документів, DSC_i та DSC_{i+1} , де $DSC_{i+1} = DSC_i \cup INC$ (2.6) та INC це набір документів, які додаються до DSC_i щоб розширити цю підколекцію і зробити її потенційно більш статистично репрезентативною. Наступні кроки запропоновані у розділі 2.3 для вимірювання термінологічної різниці (ітерація 0, ...):

- Сформуувати підколекцію документів DSC_{i+1}
- Здобути набір термінів B_{i+1} з DSC_{i+1} (B_i було здобуто з DSC_i на попередній ітерації ($i - 1$))
 - Обчислити індивідуальний поріг значущості терміну eps_{i+1} для B_{i+1}
 - Зберегти значущі терміни у T_{i+1}
 - Обчислити ns для T_{i+1}
 - Обчислити термінологічну різницю $thd(T_i, T_{i+1})$

У цьому розділі ми прагнемо реалізувати цей послідовний процес наближення, шляхом розробки робочого процесу, представленого на Рис. 3.1, використовуючи Business Process Model та Notation (BPMN)¹⁶ як формальну мета-модель.

¹⁶ Специфікацію використовуваного BPMN 2.0 отримано з <https://www.omg.org/spec/BPMN/2.0/PDF>

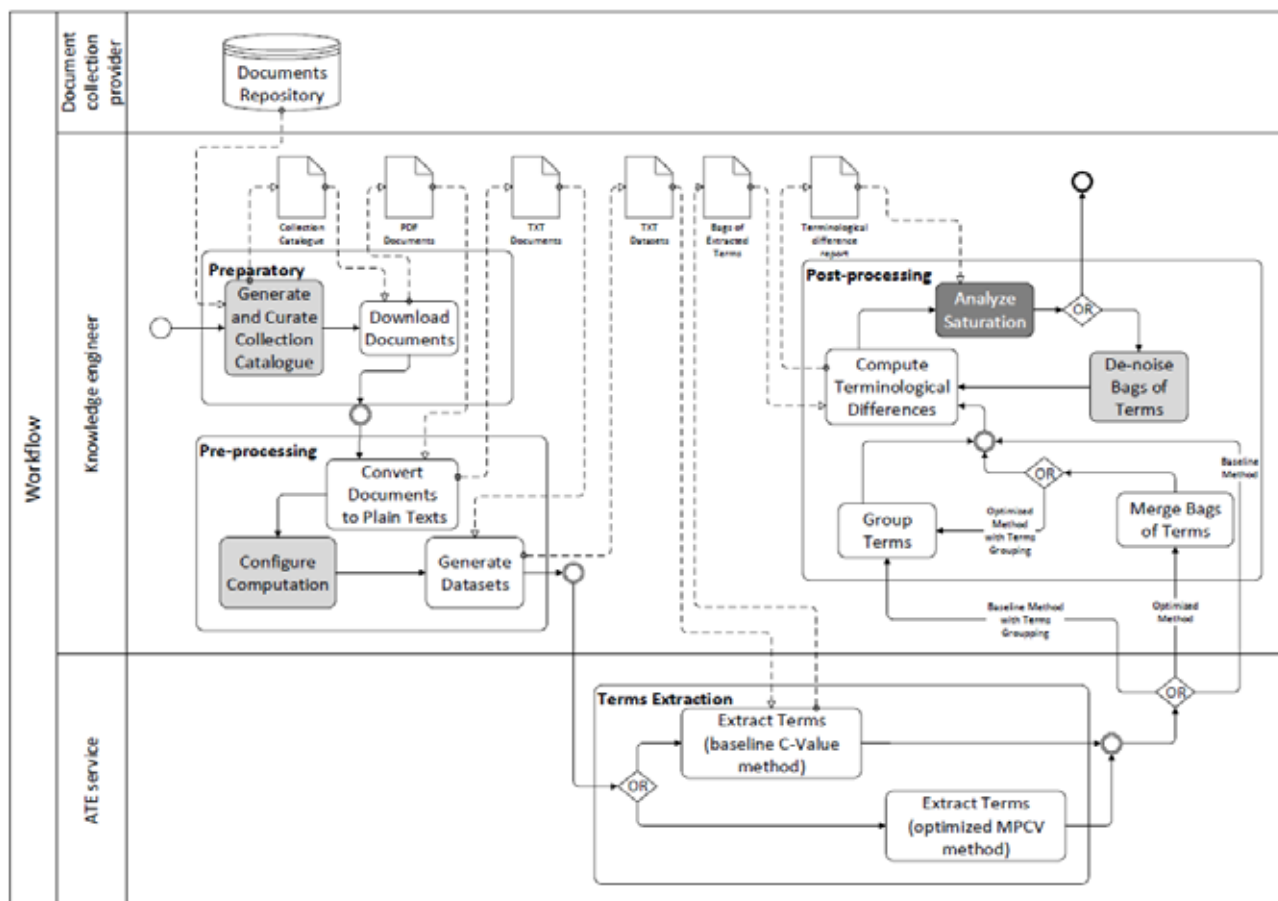


Рис. 3.1. Робочий процес для виявлення та вимірювання термінологічного насичення. Активності пофарбовані у білий колір повністю автоматизовані, світло-сірі вимагають взаємодії з користувачем, темно-сірі виконуються вручну.

Структурно робочий процес містить наступні чотири етапи: (i) підготовчий; (ii) перед-обробка; (iii) здобуття термінів; та (iv) пост-обробка. Завдання цих етапів є наступними:

- **Підготовчий:** зібрати колекцію з якої буде здобуто підколекцію документів термінологічного ядра. Це включає відбір документів, розробку каталогу колекції, та здобуття повних текстів документів.
- **Перед-обробка:** підготувати дані для обчислення. Це включає перетворення отриманих документів у плоский текст, налаштування процесу обчислень та генерацію наборів даних.
- **Здобуття термінів:** здобути набори термінів з наборів даних. У нашій роботі здобуття можна зробити будь-яким методом, реалізованим у програмному забезпеченні, що повертає набори термінів (символьних строк) з їх обчисленими

показниками значущості. На цьому етапі не передбачено, що відсікаються незначущі кандидати у терміни. У нашій роботі використовуються дві різні реалізації методу C-value (базова та оптимізована). Вибір базового методу аргументується у розділах 1.8, 1.9 та 4.4. Оптимізований метод (MPCV) було розроблено у розділі 2.6. Використання алгоритму співставлення строк Aho-Corasick для оптимізації обчислення часткових C-value аргументується у розділі 1.11.

- **Пост-обробка:** обчислити (виміряти) термінологічні різниці та виявити термінологічне насичення. Сюди входить об'єднання здобутих наборів термінів (для методу MPCV, розділ 3.3), групування термінів (розділ 3.5), обчислення термінологічних різниць, аналіз насичення та видалення накопиченого регулярного шуму (якщо потрібно). Рішення про використання групування термінів приймається під час налаштування потоку обчислень. На підставі цього рішення, для обчислення термінологічних різниць використовується або базовий (розділ 3.4) або вдосконалений (розділ 3.5.4) алгоритм **THD**. Аналіз насичення виконується вручну (виділено сірим кольором на Рис. 3.1) на основі звітів про термінологічні різниці (див. Рис. 3.2). У результаті може бути виявлено накопичення регулярного шуму (див. наприклад розділ 4.5.3). У цьому випадку очищення наборів термінів виконується з частковою автоматизацією (розділ 3.6).

Dataset:	Extracted:	Score > 1:	Retained:	eps:	thd:	thdr
D1.txt:	39132:	20609:	1864:	12.0:	62.89:	100
D2.txt:	63676:	33854:	2537:	15.5:	31.50:	43.68
D3.txt:	85596:	46149:	3022:	18.0:	23.38:	32.93
D4.txt:	109076:	58400:	3399:	19.65:	18.03:	27.23
D5.txt:	135006:	71194:	3983:	23.22:	15.80:	21.43
D6.txt:	152777:	80495:	4279:	24.0:	10.06:	13.78
D7.txt:	173655:	91576:	4954:	24.0:	10.67:	13.59
D8.txt:	193998:	102449:	5020:	26.0:	9.13:	11.32
D9.txt:	215213:	113413:	5345:	28.0:	8.68:	10.33
D10.txt:	237382:	124313:	5656:	28.52:	7.56:	8.69
D11.txt:	258522:	135172:	6177:	28.52:	7.30:	8.13
D12.txt:	275178:	144390:	6625:	28.52:	6.53:	6.97
D13.txt:	291482:	152890:	6661:	30.0:	6.43:	6.81
D14.txt:	306387:	161301:	5685:	38.0:	15.46:	15.87
D15.txt:	313506:	164726:	5815:	38.0:	3.52:	3.66

Рис. 3.2 Приклад автоматично зформованого звіту про термінологічну різницю для послідовності інкрементально збільшених наборів даних.

Робочий процес, зображений на Рис. 3.1 і обговорений вище, представляє собою високорівневе уявлення про процес обчислення для здобуття насиченої термінології. Деталі, які необхідні для його алгоритмізації, представлені на структурованій блок-схемі обчислень конвеєру здобуття насиченої термінології на Рис. 3.3.

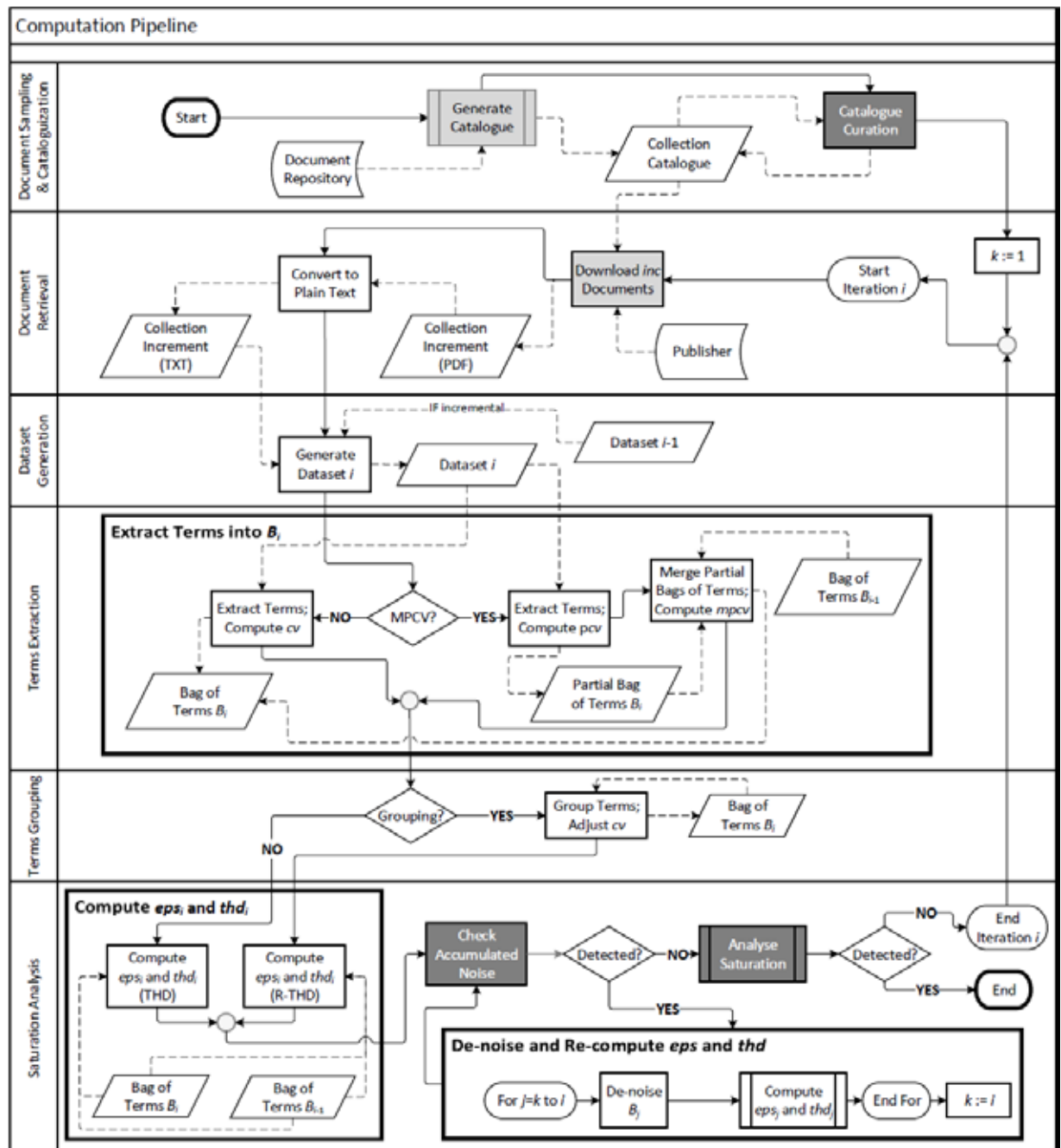


Рис. 3.3. Структурована блок-схема обчислень у процесі здобуття насиченої термінології. Світло-сірі модулі та підпроцеси вимагають взаємодії з користувачем або виконуються частково вручну. Темно-сірі модулі та підпроцеси виконуються вручну.

3.2 Підготовчі кроки та алгоритми

Підготовчий етап включає шаги: відбору та каталогізації документів для формування колекції; здобуття повних текстів цих документів (Рис. 3.3).

3.2.1 Генерація каталогу

Для відбору релевантних документів у роботі використовується метод та програмне забезпечення (Dobrovolskyi and Keberle 2018). Для генерації каталогу відібраних документів потрібно метадані для кожного документу, що входить до колекції, здобути з вихідного репозиторію та додати до каталогу колекції. Окрім метаданих, потрібно отримати інформацію про кількість цитувань для кожного документа. Каталог колекції – це двовимірна таблиця, у якій строки представляють документи, а стовпці – атрибути цих документів. Атрибути пояснюються у Таблиці 3.1. Приклад сформованого каталогу наведено в Додатку Б. Алгоритм створення каталогу для колекції документів представлено в Алг. 3.1 (Додаток В).

Алгоритмічно, створення каталогу колекції документів залежить від репозиторія. Специфіка полягає в тому, як конкретний репозиторій представляє документи та дозволяє отримати доступ до їх метаданих, використовуючи свій конкретний інтерфейс прикладного програмування (API).

Отже функції (**DR-query()**, **parse()**), що використовуються в алгоритмі **GCAT**, є специфічними для їх реалізації у залежності від API репозиторію документів. Аналогічно, формат вхідної строки запиту Q (другий вхідний параметр **GCAT**) залежить від API репозиторію документів. У нашій роботі було здійснено дві різні реалізації: (i) для репозиторію Springer Nature Journals (<https://link.springer.com/journals/>); та (ii) для репозиторію Microsoft Research (<https://academic.microsoft.com/>) – див. також розділ 3.7.

Таблиця 3.1. Атрибути документів у каталозі колекції

Стовпець	Заголовок (Атрибут)	Пояснення
A	Publication Year	Рік публікації документа
B	Type	Одне з наступних значень, яке слід вибрати на основі типу документа: UN - невідомо, JA – стаття журналу, PT - патент, CP – стаття конференції, BC – глава книги, BO - книга, BR - посилання на книгу, DA – набір даних, RE – репозиторій.

C	Venue (Journal / Series / Conference)	Значення: (i) назва журналу, якщо документ є статтею журналу (JA); назва серії книг (напр. LNCS) якщо документ - це глава книги (BC); назва конференції + серія книг (напр. ISWC 2020; LNCS) якщо документ є статтею конференції або семінару (CP)
D	Publisher	Назва видавця (напр. Springer-Nature, Elsevier, ACM, інші.)
E	Volume No	Це номер журналу або серії тому, лише для JA, CP, BC, BO.
F	Issue No	Номер випуску журналу, лише для JA,
G	Pages	Для JA, BC, CP, початкові та кінцеві сторінки у томі. Для електронних публікацій це можуть бути стаття № та кількість сторінок. Для рукописів, таких як дисертації – загальна кількість сторінок.
H	DOI	Цифровий ідентифікатор об'єкта документа (Digital Object Identifier, DOI) (за наявності).
I	DOI Link	URL-адреса, що містить DOI, та вказує на оригінальну публікацію на ресурсі Видавця (за наявності).
J	MSR ID	Ідентифікатор документа у репозиторії MS Research (за наявності).
K	Title	Назва документу, включаючи підзаголовок, якщо такий є.
L	Authors	Список авторів документа, розділений крапкою з комою у форматі MSR.
M	Affiliations	Список організацій, з якими афілійовані автори, розділений крапкою з комою у форматі MSR.
N	Complete Citation	Повне цитування документа в одному з бібліографічних форматів, напр. BibTeX
O	Abstract	Анотація документа (за наявності).
P, Q, R	Category	Ці атрибути повинні використовуватися для категоризації або розділення колекції, що залежить від використання.
T	-- Reserved --	
U	Citation Count (GS)	Кількість цитувань документа, отримана з Google Scholar
V	Citations per Year (GS)	Частота цитування для ранжування / впорядкування документів, обчислюється як формула: Кількість-цитувань-(GS)-(U) / (<поточний рік> - Рік публікації-(A)+1).
W	Document File Name	Ім'я файлу з повним текстом документу, що надалі завантажується та зберігається для обробки. Правило для генерації є наступним: <Рік>+"-+"<Тип>+"-+"<Том>+"("+"<Випуск>+"-("+"<Сторінки>+"-+"<підстрока <DOI> після '/'; вся строка <DOI> якщо в ній немає '/') - Рік: Рік публікації, атрибут А - Тип: Тип, атрибут В - Том: Номер тому, атрибут Е, 1 якщо немає - Випуск: Номер випуску, атрибут F, 1 якщо немає - Сторінки: атрибут G - DOI: атрибут H
X	MSR Entry (URL)	URL адреса для опису документа в MSR.
Y	Full Text Download (URL)	URL-адреса для завантаження повного тексту документа.

3.2.2 Завантаження документів

Етап завантаження повнотекстових документів включає завантаження файлів документів у форматі PDF. Алгоритм цієї обчислювальної активності представлений в Алг. 3.2 (Додаток В).

Алгоритм **DDL** виконує пакетне завантаження файлів. Отже, реалізація функції **download()** для конкретного репозиторію, повинна уникати надсилання занадто великої кількості запитів на завантаження не чекаючи поки черга запитів спорожниться або зменшиться до розміру, допустимого сервісом репозиторію. Процедура подання запиту на масове завантаження, як правило, регулюється умовами використання репозиторію, де часто вказується інтервал часу, на який слід чекати перед подачею наступного масового запиту.

3.3 Кроки та алгоритми перед-обробки

Етап перед-обробки включає перетворення завантажених файлів PDF у плоскі тексти, конфігурацію потоку обчислень та генерацію наборів даних із плоских текстових документів.

3.3.1 Конвертація документів (PDF у плоский текст)

Оскільки екстрактори термінології, включаючи ті, що використовуються в нашій роботі, беруть на вхід лише текстові файли, завантажені повнотекстові документи потрібно конвертувати у плоский текст таким чином, щоб забезпечити максимально можливу якість конвеєру лінгвістичної обробки обраного програмного забезпечення екстрактора. Отже, програмне забезпечення конвертора повинно відповідати таким вимогам:

- Видалити переноси та поєднати зі строками після переносів
- Забезпечити початок кожного речення з нової строки
- Замінити усі лігатури в тексті на відповідні послідовності літер. Таблиця лігатур повинна бути зчитувана з окремого файлу.
- Зберегти вихідний плоский текст у кодуванні, що є припустимим для програмного забезпечення екстрактора

Оскільки один із екстракторів, що використовується в нашій роботі приймає файли в ASCII¹⁷ (ANSIS 1986), а інший в UNICODE¹⁸, вихідне кодування передається як значення параметра в алгоритм **PDF2TXT** – див. Алг. 3.3 (Додаток В). Алгоритм виконує масове перетворення файлів PDF з каталогу вводу та зберігає файли TXT у вихідному каталозі з тими ж самими іменами файлів.

3.3.2 Конфігурація та генерація наборів даних

Обчислювальний конвеєр для здобуття насиченої термінології (див. Рис. 3.3) містить кілька логічних розгалужень у потоці обчислень. Вибір шляхів у цих розгалуженнях і деяких інших обчислювальних параметрів залежать від конфігурації. У Таблиці 3.2 представлено перелік параметрів конфігурації, їх допустимі значення та обґрунтування вибору. Параметри розділені на дві групи, залежно від їх спрямованості:

- Формування потоку обчислень; та
- Групування документів у послідовності наборів даних у вигляді простого тексту для здобуття насиченої термінології

Конфігурація потоку обчислень виконується:

- Або за допомогою простої форми GUI з подальшою перевіркою введених параметрів на їх відповідність допустимим значенням і взаємозв'язкам між параметрами
- Або шляхом введення правильних значень у скрипті, що викликає програмні модулі, які реалізують відповідні алгоритми конвеєру.

Таблиця 3.2. Параметри конфігурації для конвеєру здобуття насиченої термінології

#	Параметр конфігурації	Допустимі значення	Обґрунтування вибору значень
1	Обчислювальний потік		
1.1	UTG використання групування термінів	- <i>true, false</i>	UTG = true обирається, якщо етап групування термінів було включено до конвеєру. Якщо true, то слід також встановити такі параметри: TSM, TST. Вибір цього значення викликає виконання алгоритму STG (розділ 3.5.3) до і виконання

¹⁷ <http://ascii-table.com/ansi-codes.php>

¹⁸ <https://unicode.org/charts/>

			алгоритму R-THD (розділ 3.5.4) при обчисленні термінологічної різниці. UTG = <i>false</i> обирається, якщо групування термінів вважається непотрібним. Якщо вибрано це значення, для обчислення термінологічних різниць виконується базовий алгоритм THD (розділ 3.4), що використовує еквівалентність символічних строк для співставлення термінів.
1.2	TSM – міра подібності термінів	<i>jr, jw, ja, sd</i>	TSM = <i>jr</i> означає, що міра подібності Жаро використовується для групування подібних і співпадаючих термінів. Ця міра реалізована у алгоритмі M-JR . TSM = <i>jw</i> означає використання подібності Жаро-Вінклера (алгоритм M-JW). TSM = <i>ja</i> означає використання подібності Жакара (алгоритм M-JA). TSM = <i>sd</i> означає використання подібності Соренсена-Дайса (алгоритм M-SD). Ці алгоритми представлені у розділі 3.5.3. Вибір цих мір аргументовано у розділі 3.5.1.
1.3	TST – поріг подібності термінів	Значення від [0, 1]	Вибір порогу для обраної міри подібності пояснюється у розділі 3.5.2.
1.4	CVM - метод для обчислення C-values	<i>cv, mpcv</i>	CVM = <i>cv</i> викликає базовий метод та використання інкрементально розширених наборів даних (DP = <i>false</i>). Цей базовий метод істотно обчислювально складніше, як доведено у розділі 4.7. CVM = <i>mpcv</i> викликає оптимізований метод, заснований на використанні часткових колекцій документів (розділ 2.6). Цей оптимізований метод, як додатковий крок, використовує злиття часткових C-values, перед обчисленням термінологічних різниць (розділ 3.3, алгоритм STG). Цей метод вимагає створення наборів даних, як частин колекції документів, що неперетинаються: DP = <i>true</i> .
2	Набори Даних		
2.1	DO – впорядкування документів для формування наборів даних	<i>ch, rch, rn, bd, dcf</i>	DO = <i>ch</i> означає, що документи вибираються до інкременту набору даних, у хронологічному порядку, на основі їх часу публікації. DO = <i>rch</i> означає зворотньо-хронологічний порядок. DO = <i>rn</i> означає випадковий вибір документів. DO = <i>bd</i> означає, що документи беруться у двох напрямках – новіші, старіші, і т.д. DO = <i>dcf</i> означає використання порядку за спадаючою частотою цитувань.
2.2	INC – розмір інкременту для послідовності наборів даних	натуральне число	INC – вказує кількість документів, які формують інкремент до підколекції документів у процесі послідовного наближення побудови колекції термінологічного ядра (див. розділ 2.1). Ці документи беруться із колекції та додаються до наступного набору даних у порядку, визначеному параметром DP .
2.3	DP – розділ колекції документів на частини	<i>true, false</i>	Якщо DP = <i>false</i> , набір даних $D_{i+1} = D_i \cup INC$, де <i>INC</i> інкремент, що містить декілька (INC) текстових документів, вибраних із колекції в порядку, визначеному DO . Отже, D_{i+1} є інкрементальним розширенням D_i . Якщо DP = <i>true</i> , набір даних $D_{i+1} = INC$. Отже D_i та D_{i+1} є різними частинами колекції і їх перетин (у документах) порожній.

Генерація набору даних для поточної ітерації обчислювального потоку виконується за допомогою алгоритму **GDS** представленого в Алг. 3.4 (Додаток В).

3.4 Алгоритми оптимізованого обчислювального конвеєру

У нашій роботі було виявлено два аспекти, які допомагають істотно скоротити час обчислення статистичної частини конвеєру здобуття термінів для виявлення термінологічного насичення. У розділі 1.11, ми з'ясували, що використання методу Aho-Corascik для ефективного співставлення декількох символічних строк, що містять вхідний набір строк, значно знижує обчислювальні витрати на обчислення значень C-value. Потім, у розділі 2.6, ми розробили обчислювально ефективний метод обчислення об'єднаних часткових C-values на основі використання часткових колекцій документів. У цьому розділі, ми розробляємо алгоритми для обох вищезазначених вдосконалень базового обчислювального методу C-value, що використовує інкрементально збільшені набори даних.

3.4.1 Використання алгоритму Aho-Corascik в обчисленні C-value

Aho and Corascik (1975) запропонували створити структуру даних префіксного дерева (trie) та використовувати скінчений автомат (Final State Machine, FSM) для ефективного пошуку символічних підстрок із заданого набору (словника) у довшій текстовій строці. Їх результати доводять підвищення продуктивності в п'ять-десять разів порівняно із звичайним пошуком підстрок. Підхід Aho and Corascik був інтегрований у наш базовий алгоритм для обчислення C-values. Результати представлені в Алг. 3.5 (Додаток В).

3.4.2 Злиття часткових C-value

Злиття часткових C-value (див. розділ 2.6, рівняння 2.29 та 2.31) необхідне для обчислення $mpcv(s)$, що практично не відрізняються від звичайно обчислених C-value, як доведено Теоремою 2.5. Це злиття для послідовності наборів здобутих термінів $B_i, i = 1, \dots, n$, де n кількість часткових колекцій документів, здійснюється шляхом повторення алгоритму **MPCV**, представленого в Алг. 3.6 (Додаток В).

3.5 Базовий алгоритм вимірювання термінологічної різниці

Базовий алгоритм (**THD**) для обчислення термінологічної різниці thd між двома наборами збережених значущих термінів T_i та T_{i+1} (розділ 2.3) був запропонований Tatarintseva et al. (2013). У цьому розділі, для повноти викладу,

міститься опис цього алгоритму у вдосконаленій та розширеній формі та іншій нотації (Алг. 3.7, Додаток В). На додаток до (Tatarintseva et al. 2013), ми включили до **ТНД** обчислення порогу значущості термінів (*eps*) для набору здобутих термінів V_{i+1} та, таким чином, здобуття T_{i+1} . Передбачається, що T_i береться з попередньої ітерації обчислювального потоку. Після здобуття T_{i+1} , базовий алгоритм **ТНД** накопичує різниці *n-score*, у значенні *thd*, для набору T_{i+1} , якщо в T_i та T_{i+1} були однакові терміни. Для порівнювання термінів використовується посимвольне порівняння відповідних строк символів (див. рядок 7 в Алг. 3.7). Якщо в T_i не було того самого терміну, алгоритм додає *n-score* нового терміну до значення *thd* в T_{i+1} . Після обчислення *thd*, відносна термінологічна різниця *thdr* отримує своє значення як *thd* поділене на суму *n-scores* в T_{i+1} .

Для подальшого аналізу використовуються абсолютні (*thd*) та відносні (*thdr*) термінологічні різниці. Якщо T_{i+1} відрізняється від T_i більше, ніж індивідуальний поріг значущості терміну *eps*, вважається, що на цій ітерації термінологічне насичення не досягнуто. Якщо ні, це означає, що додавання інкременту документів до D_i для створення D_{i+1} не додало помітної кількості нової термінології. Отже піднабір D_{i+1} всієї колекції документів міг стати термінологічно насиченим. Однак для отримання більшої впевненості щодо насиченості пропонується оцінити більше наступних пар T_i і T_{i+1} . Якщо спостерігається стабільне насичення, то процес пошуку мінімального насиченого піднабору можна зупинити.

3.6 Алгоритми групування термінів

На одне питання дослідження (**ПД8**, див. розділ 1.13) не було дано відповіді у формальному фреймворці, розробленому у розділі 2. Питання полягало в тому, чи може групування подібних термінів сприяти швидшій та менш волатильній збіжності до термінологічного насичення. У цьому розділі ми наступним чином формуємо гіпотезу дослідження, щоб відповісти на це питання.

Гіпотеза НЗ.1: *Групування подібних термінів призводить до швидшого термінологічного насичення та більш компактних наборів термінів.*

Нехай набір здобутих термінів містить піднабір строк-кандидатів подібних за їх передбачуваним сенсом. Тоді кожен з таких піднаборів можна згрупувати в одну

строку-кандидат та скоригувати його C-value, щоб відобразити консолідовану значущість піднабора. Для цього: (i) слід обрати відповідні міри подібності термінів; (ii) повинні бути раціонально запропоновані пороги для розрізнення подібних та несхожих кандидатів; та (iii) повинні бути розроблені алгоритми для вимірювання подібності, групування термінів та вдосконаленого вимірювання термінологічної різниці. Обґрунтованість та корисність цього підходу необхідно додатково оцінити експериментально. Спираючись на схему підходу до перевірки **НЗ.1**, ми детально розробимо цей підхід в даному розділі. Експериментальна оцінка проводитиметься в розділі 4.6.

Для розробки підходу спочатку ми обираємо чотири міри подібності строк із тих, що були розглянуті в розділі 1.10. Потім ми обговорюємо можливі випадки того, як часткову подібність строк можна розглядати як відповідну подібність сенсу термінів. На основі цього обговорення та зібраного вручну тестового набору пар термінів ми обґрунтовуємо пороги подібності для всіх чотирьох вибраних мір строкової подібності. Використовуючи формальне представлення цих мір та порогів, ми розробляємо алгоритми для: обчислень мір (**M-JR**, **M-JW**, **M-JA**, **M-SD**); групування подібних термінів (**STG**). Нарешті, ми розробляємо вдосконалений алгоритм, **R-THD**, для вимірювання термінологічної різниці, який використовує міри строкової подібності та відповідні пороги подібності термінів.

3.6.1 Вибір мір строкової подібності

З множини згаданих вище мір, через специфіку нашого завдання приблизного порівняння коротких строк, що містять кілька слів, ми відкинули ті з них, що: (i) вимагають довгих строк або наборів строк доволі великого розміру; (ii) є затратними у обчисленнях. Також, наскільки це можливо, ми намагалися зберегти представників усіх видів строкових мір у нашому списку кандидатів. У результаті ми склали такий перелік мір, які слід розглянути для подальшого використання:

- Міри на основі символів: відстань Левенштейна (Levenshtein 1966), відстань Хеммінга (Hamming 1959), подібність Жаро (Jaro 1989), та подібність Жаро-Вінклера (Winkler 1990)

· Міри на основі токенів: подібність Жакара для порівняння *юні*-грамів (Jaccard 1912), косинус подібність для порівняння *юні*-грамів (Yu et al. 2016), та коефіцієнт Соренсена-Дайса для порівняння *бі*-грамів (Dice 1945; Sørensen 1948)

Серед них, найменш релевантними є відстані Левенштейна та Хеммінга у силу їх обмеженої відповідності нашому контексту. Левенштейн повертає ціле число необхідних змін, а решта вимірювань повертають нормалізоване дійсне число. Таким чином, незрозуміло, чи нормалізація значень, що повертає Левенштейн, дійсно дасть результат порівнюваний з іншими мірами, з тим, щоб використовувати один і той самий поріг подібності. Міра Хеммінга застосовується лише до строк однакової довжини. Тому додавання пробілів у коротшу строку значно знизить точність вимірювань. Косинус подібність базується на тих самих принципах, що і подібність Жакара, але є більш обчислювально складною через наявність квадратного коріння в знаменнику. Тому ми вирішили використати міри Жаро, Жаро-Вінклера, Жакара, та Соренсена-Дайса для реалізації та експериментальної оцінки в нашій роботі. Далі коротко пояснюється, як обчислюються обрані міри.

Подібність Жаро sim_j між двома строками S_1 і S_2 обчислюється як мінімальна кількість односимвольних перетворень першої строки, що потрібні для отримання другої строки у порівнюваній парі:

$$sim_j = \begin{cases} 0 & \text{якщо } m = 0 \\ \frac{1}{3} * \left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m} \right) & \text{інакше} \end{cases}, \quad (3.1)$$

де: $|S_1|$, $|S_2|$ довжина строк, які порівнюються; m кількість співпадаючих символів; t половина кількості переставлених символів. Символи співпадають, якщо вони однакові і їх відстань від початку строки відрізняється не більше ніж на $\lfloor \max(|S_1|, |S_2|)/2 \rfloor - 1$. Кількість переставлених символів це кількість співпадаючих символів, що мають різний порядок послідовності.

Міра подібності Жаро-Вінклера sim_{j-w} вдосконалює міру подібності Жаро використовуючи значення ваги префіксу p , яке присвоює кращі рейтинги для строк, які співпадають з їх початку і до довжини префіксу l . Отже, для двох строк S_1 і S_2 ця міра обчислюється як:

$$sim_{j-w} = sim_j + l * p * (1 - sim_j), \quad (3.2)$$

де: l довжина спільного префіксу (максимум до 4 символів); p постійний коефіцієнт зважування, що визначає наскільки величина подібності скоригована вгору за наявності спільного префіксу (до 0.25, в іншому випадку міра може стати більшою за 1; Winkler (1990) радить, що $p = 0.1$).

Іноді бонус префіксу Вінклера $l * p * (1 - sim_j)$ надається лише тим парам строк, що мають значення подібності Жаро вище, ніж певний поріг. Цей поріг пропонується (Winkler 1990) рівним 0.7.

Індекс подібності Жакара sim_{ja} є мірою подібності для кінцевих множин, у нашому випадку, символів. Для двох строк S_1 та S_2 , він обчислюється як відношення між потужностями перетину та об'єднання наборів символів у S_1 і S_2 :

$$sim_{ja} = (|S_1| \cap |S_2|) / (|S_1| \cup |S_2|). \quad (3.1)$$

Нарешті, коефіцієнт Соренсена-Дайса обчислюється шляхом підрахунку ідентичних бі-грамів символів в S_1 і S_2 і відносить їх до загальної кількості бі-грамів в обох строках:

$$sim_{sd} = 2n_{\equiv} / (n_{S_1} + n_{S_2}), \quad (3.4)$$

де: n_{\equiv} кількість бі-грамів у S_1 і також у S_2 ; n_{S_1}, n_{S_2} є кількостями усіх бі-грамів у S_1 та S_2 відповідно.

3.6.2 Випадки подібності термінів та порогові значення

Для належного використання реалізованих функцій мір строкової подібності (МСП) у контексті порівняння та групування термінів, необхідно визначити, яким буде розумний поріг для розрізнення (семантично) подібних і не подібних термінів. Для визначення цього враховуються такі випадки в порівнянні строк.

Повністю позитивний випадок (Full Positives, FP). У цьому випадку, порівнювані строки символів повністю однакові, що однозначно дає подібні (однакові) терміни.

Повністю негативний випадок (Full Negatives, FN). У цьому випадку, порівнювані строки символів дуже різні та терміни у цих строках мають різну

семантику. Цей випадок також є зрозумілим і може бути позначений низькими значеннями мір подібності.

Частково позитивний випадок (Partial Positives, PP). У цьому випадку, порівнювані строки символів частково однакові і терміни в цих строках мають однакову чи подібну семантику. Терміни в кожній строці подібні, хоча це не дуже легко визначити. Наведемо різні категорії термінів, які приводять до цього випадку: слова в термінах мають різні закінчення (наприклад форми однини/множини); використовуються різні роздільники (наприклад “-”, чи “_”, чи “ - ”); відсутній символ, помилково додано, або неправильно написано (помилка); один термін є підстрокою іншого (наприклад є частковим випадком до другого); одна зі строк містить зайві додаткові символи (наприклад два або три пробіла замість одного).

Частково негативний випадок (Partial Negatives, PN). У цьому випадку, порівнювані строки символів частково однакові, але терміни в цих строках мають різну семантику. Терміни в таких строках різні, хоча це не дуже легко визначити. Наведемо категорії, які приводять до даного випадку: терміни, що відносяться до порівнюваних строк, відрізняються кількома символами, але мають різні значення (наприклад “deprecate” та “depreciate”); порівнювані терміни мають загальне слово (слова), але повністю відрізняються за своїми значеннями (наприклад “affect them” порівнюється з “effect them”). Частково негативний випадок – найважчий випадок для виявлення.

Тестовий набір термінів, що входять до описаних вище випадків, був розроблений вручну¹⁹.

Для кожної пари термінів у цьому тестовому наборі були обчислені всі чотири вимірювання строкової подібності. Витяг з цього результату наведений в Таблиці 3.3. Ми обчислили середні значення для всіх чотирьох вимірювань подібності для кожної категорії, використовуючи всі пари термінів тестового набору, що потрапляють до цієї категорії. Ці значення представлені в Таблиці 3.4.

¹⁹ Тестовий набір та обчислені значення подібності термінів доступні за посиланням <https://github.com/OntoElect/Data/blob/master/STG/Test-Set.xls>

Пороги подібності термінів слід обирати таким чином, щоб повністю та частково негативні випадки розглядалися як не схожі, але повні та частково позитивні випадки вважалися подібними. Отже, для випадку частково позитивних порогов слід вибирати як мінімальні для всіх категорій, а для частково негативних – як максимальні для всіх категорій. Порогові значення показані в Таблиці 3.5.

Таблиця 3.3. Міри подібності для різних тестових випадків

Випадок	Категорія	Пара термінів	Соренсен-Дайс	Жакар	Жаро	Жаро-Вінклер
Різні строки (FN)		whirled world	0.0	0.5	0.790	0.811
		traces creta	0.0	0.833	0.588	0.588
		time domain ontology lifecycle	0.0	0.428	0.445	0.445
Ідентичні строки (FP)		identical strings identical strings	1.0	1.0	1.0	1.0
Подібна семантика (FP)	Зайві (додаткові) символи	*system?problems system problems	0.814	0.769	0.936	0.936
		sad data mining sqr data mining	0.769	0.818	0.859	0.873
	Спільні частини (слова)	marcov chain monte carlo methods monte carlo methods	0.782	0.766	0.629	0.666
		data mining algorithm data mining	0.642	0.666	0.842	0.904
		cation error error	0.533	0.333	0.427	0.427
	Помилки	fraud detection froud ditection	0.714	0.916	0.859	0.887
		monte carlo monte ??rlo	0.7	0.727	0.878	0.927
		data mining data minin	0.941	0.875	0.969	0.981
	Різні роздільники	computer science computerscience	0.896	0.916	0.979	0.987
		serial episodes serial&&episodes	0.827	0.818	0.936	0.961
		data cube data_cube	0.75	0.777	0.925	0.955
	Різні закінчення	network structure network structures	0.969	1.0	0.981	0.988
		time complexity time complexities	0.896	0.833	0.981	0.951
		value values	0.888	0.833	0.918	0.951
	Різна семантика (FN)	Спільні частини (слова)	database military base	0.400	0.500	0.410
brainstorm stormy weather			0.363	0.428	0.509	0.509
iron clad iron maiden			0.444	0.636	0.804	0.882
jellyfish fish tank			0.352	0.307	0.614	0.614
four delegates delegated authority			0.451	0.666	0.557	0.557
string theory string format			0.583	0.571	0.812	0.887
Дуже мало символічних відмінностей		deprecate against depreciate against	0.909	1.0	0.903	0.941
		alternately move alternatively move	0.933	0.916	0.9	0.94
		affect them effect them	0.9	0.758	0.906	0.906

Ці пороги надалі використовуються як границі для відповідних порогових інтервалів у наших експериментах. Ці інтервали були рівномірно розподілені по чотирьох точках, наведених у Таблиці 3.5. Вимоги до частково позитивних і негативних, на жаль, суперечать один одному. Наприклад, якщо порогове значення вибрано, щоб відфільтрувати частково негативні, деякі з частково позитивних елементів також будуть відфільтровані. Отже, якщо вважати, що частково негативні зустрічаються рідко, було вирішено використовувати пороги для частково позитивних.

Таблиця 3.4. Середні значення вимірювань подібності для різних категорій пар термінів з тестового набору

Випадок / Категорія	Ел-тів у тест. наборі	Соренсен-Дайс	Жакар	Жаро	Жаро-Вінклер
Різні строки (FN)	6	0.03	0.45	0.55	0.55
Ідентичні строки (FP)	3	1.00	1.00	1.00	1.00
Подібна семантика (PP)	32	0.71	0.72	0.63	0.70
- Зайві (додаткові) символи	7	0.8401	0.8820	0.8714	0.8784
- Спільні частини (слова)	6	0.7122	0.7280	0.6375	0.7043
- Помилки	6	0.7797	0.8637	0.8863	0.9220
- Різні роздільники	6	0.7860	0.8473	0.9125	0.9442
- Різні закінчення	7	0.8911	0.9135	0.9410	0.9590
Різна семантика (PN)	18	0.89	0.89	0.89	0.91
- Спільні частини (слова)	11	0.4336	0.5221	0.6161	0.6408
- Дуже мало символічних відмінностей	7	0.8826	0.8845	0.8914	0.9059
Загальне:	59				

Таблиця 3.5. Пороги подібності термінів, що вибрані для експериментальної оцінки

Метод	Поріг подібності термінів			
	Min	Ave-1	Ave-2	Max
Соренсен-Дайс	0.71	0.76	0.83	0.89
Жакар	0.72	0.77	0.83	0.89
Жаро	0.63	0.72	0.80	0.89
Жаро-Вінклер	0.70	0.77	0.84	0.91

3.6.3 Групування термінів та вимірювання подібності

Наше завдання – вдосконалити базовий алгоритм **TND** (розділ 3.4) таким чином, щоб дозволити знаходити не зовсім однакові, але достатньо схожі терміни, застосовуючи міри строкової подібності (розділ 3.6.1) з відповідними пороговими значеннями, як пояснювалось у попередньому розділі 3.6.2. Для цього вводиться попередній етап групування термінів, щоб уникнути дублювання виявлення подібності. Для кожного з порівнюваних наборів термінів T_i та T_{i+1} на цьому попередньому етапі застосовується алгоритм групування подібних термінів (**STG**) – див. Алг. 3.8 (Додаток В). Алгоритми для обчислення обраних мір строкової подібності представлено як Алг. 3.9 – 3.12 (Додаток В).

M-JR (Алг. 3.9, Додаток В) обчислює подібність Жаро для двох вхідних символічних строк, S_1 і S_2 , на основі рівняння (3.1). Ідея використовувати бітові карти для позначення позицій співпадаючих символів запозичена з подібного

алгоритму опублікованого на RosettaCode²⁰. На відміну від алгоритму RosettaCode, **M-JR**: (i) розрізняє більш коротшу та більш довшу вхідну символну строку; та (ii) повертає не значення подібності Жаро, а результат його порівняння з порогом подібності (*th*): *false* якщо подібність нижче порогу або *true* в іншому випадку. **M-JW** (Алг. 3.10, Додаток В) обчислює подібність Жаро-Вінклера для двох вхідних символних строк, S_1 і S_2 на основі рівняння (3.2). Це включає в себе обчислення подібності Жаро та його позитивне коригування на основі довжини спільного префіксу в S_1 і S_2 . Подібність Жаро обчислюється ідентично **M-JR** представленому в Алг. 3.9. **M-JA** (Алг. 3.11, Додаток В), що обчислює подібність Жакара для двох вхідних символних строк, S_1 та S_2 , на основі рівняння (3.3). **M-SD** (Алг. 3.12, Додаток В) обчислює подібність Соренсена-Дайса для двох вхідних символних строк, S_1 та S_2 , на основі рівняння (3.4). Це включає генерацію наборів *bi*-грам для S_1 та S_2 і обчислення подібності Жакара для цих наборів *bi*-грам. Подібність Жакара обчислюється ідентично **M-JA** представленому в Алг. 3.11.

3.6.4 Вдосконалений алгоритм вимірювання термінологічної різниці

Після того, як виконано групування термінів для обох наборів зі збереженими значущими термінами T_i та T_{i+1} , виконується вдосконалений алгоритм THD (**R-THD**, Алг. 3.13, Додаток В) для обчислення термінологічної різниці між цими наборами термінів. На відміну від **THD** (Алг. 3.7), **R-THD** має ще два вхідні параметри: M та th . Він використовує M – метод (**M-JR**, **M-JW**, **M-JA**, або **M-SD**) для співставлення подібних термінів (рядок 24). Поріг th передається в M щоб розрізнити подібні і неподібні строки.

3.7 Алгоритм видалення накопиченого регулярного шуму

Через інкрементальний характер наборів даних, що використовуються в нашому базовому обчислювальному конвеєрі, ймовірно, що деякі строки-кандидати у терміни, які насправді не є реальними термінами, матимуть дуже високі частоти появи в оброблюваних наборах даних. Починаючи з деякої ітерації (i) у процесі обчислення, ці строки можуть займати верхню частину набору

²⁰ https://rosettacode.org/wiki/Jaro_distance

збережених значущих термінів T_i та, таким чином, витіснити дійсні терміни з нижчими C-value. У нашій роботі ситуація називається появою накопиченого регулярного шуму (ARN). Очевидно, слід уникати ARN, оскільки це суттєво знижує якість здобуття термінів та вимірювання насичення, наприклад для помірно зашумлених колекцій документів. В експериментальній частині нашої роботи (розділ 5) ми визначили показники наявності ARN. Ними є: (i) значне зменшення кількості збережених значущих термінів; та (ii) істотне збільшення індивідуального порогу значущості терміну. Якщо ці показники спостерігаються, здобутий T_i повинен бути досліджений, а набір строк, які є ARN, зібрати і потім видалити з усіх T_i обчислених раніше. Хоча перевірку T_i та ідентифікацію набору строк ARN потрібно робити вручну, видалення цього шуму автоматизується шляхом застосування алгоритму **ARNR** представленого як Алг. 3.14 (Додаток В).

3.8 Реалізація у програмному забезпеченні

Алгоритми, представлені в розділах 3.2 – 3.7, реалізовані в наборі програм для обчислення термінологічного насичення. Модулі цього програмного комплексу реалізують усі автоматизовані та підтримують частково автоматизовані завдання в робочому процесі (Рис. 3.1). Реалізовані функції відповідають потоку обчислень (Рис. 3.3). Програмні модулі набору є загальнодоступними для наукових досліджень²¹. Опис програмного забезпечення та посилання на модулі наведені в Додатку Г.

3.9 Висновок

У цьому розділі ми зосередилися на виконанні завдання дослідження ЗД2 (див. розділ 1.13.3). Для цього ми розробили набір алгоритмів для матеріалізації розробленого формального фреймворку для виявлення та вимірювання термінологічного насичення (розділ 2) як обчислювального методу. Зокрема, алгоритми, розроблені в цьому розділі та реалізовані в програмному забезпеченні (розділ 3.8), використовуються для експериментальної оцінки розробленого

²¹ Програмне забезпечення доступно за умовами ліцензії Apache License, Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

обчислювального методу в розділі 4. Крім того, програмне забезпечення зроблене загальнодоступним для дослідницьких цілей з метою сприяння поширенню та перевірки відтворюваності наших результатів.

Набір алгоритмів розроблений для автоматизації або часткової автоматизації розробленого робочого процесу для виявлення та вимірювання термінологічного насичення (див. розділ 3.1, Рис. 3.1) на рівні деталізації завдань. Завдання були згруповані по етапах. З метою розробки алгоритмічного набору робочий процес був детально описаний як потік обчислень (див. розділ 3.1, Рис. 3.3), включаючи специфікації модулів (функцій), потоків даних та їх перетворення, ітерацій та критеріїв завершення. Алгоритми для всіх модулів, що вимагають використання програмного забезпечення, були розроблені в розділах 3.2 – 3.7. У розділі 3.8, ми повідомили про імплементацію алгоритмів в програмному забезпеченні. Розробляючи та впроваджуючи алгоритми, ми матеріалізували програмні інструменти, необхідні для повних відповідей на питання дослідження ПД1, 4, 5, та 7 (див. розділ 1.14, Таблиця 1.5).

Крім того, у розділі 3.6, ми розробили формальний та алгоритмічний підхід, щоб відповісти на наше питання дослідження ПД8 щодо корисності групування термінів для вимірювання термінологічного насичення. На основі гіпотези дослідження НЗ.1, висунутої в розділі 3.6, ми обрали чотири відповідні міри строкової подібності, обґрунтували вибір порогів подібності термінів для обраних мір строкової подібності, та розробили необхідні алгоритми для вимірювання подібності, групування термінів, та вимірювання термінологічної різниці. Розробивши усі згадані алгоритми ми виконали завдання дослідження ЗД2.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА РОЗРОБЛЕНОГО МЕТОДУ

У цьому розділі ми представляємо результати експериментальної оцінки розробленого методу та алгоритмів, реалізованих у програмному забезпеченні. Ми починаємо з опису завдань експериментів у розділі 4.1. Пояснення загальної постановки експериментів наведено у розділі 4.2, що містить: схему експериментального робочого процесу та набору використаних програмних інструментів; опис колекцій документів, використаних у якості вихідних даних для експериментів; презентацію аспектів вимірювання та мір, зроблених в експериментах; і дані про експериментальне середовище. Решта розділу 4 зосереджена на презентації конкретних експериментів для оцінки або валідації різних аспектів розробленого методу для виявлення та вимірювання термінологічного насичення. У розділі 4.3, ми представляємо та обговорюємо результати експериментів, що доводять правильність роботи методу та алгоритмів у граничних випадках: (i) у випадку, у якому повинно спостерігатися швидке та стійке насичення; та (ii) у випадку, у якому повинна спостерігатися відсутність насичення. У розділі 4.4, ми відповідаємо на наше питання дослідження ПДЗ щодо вибору програмного інструменту для АЗТ. Це робиться шляхом крос-оцінювання двох інструментів, що потрапили до короткого списку кандидатів –TerMine та Term Extractor (розділ 1.9). Потім, у розділі 4.5, ми вивчаємо вплив п'яти можливих впорядкувань додавання документів до наборів даних на результат вимірювання термінологічного насичення. Це робиться шляхом подачі наборів даних, сформованих з використанням різних впорядкувань документів, у конвеєр обробки та порівняння результатів із використанням аспектів вимірювання (розділ 4.2.3).

У розділі 4.6, ми зосереджуємось на експериментальній оцінці корисності групування термінів для розроблених методу та алгоритмів виявлення та вимірювання термінологічного насичення. Ця оцінка виконується з використанням різних вимірювань подібності строк та порогів подібності термінів. Нарешті, у розділі 4.7, ми проводимо експериментальну перевірку валідності, ефективності, незалежності від домену, та масштабованості оптимізованого конвеєру обробки. У

розділі 4.8 зроблено висновок за результатами експериментального оцінювання доробку роботи.

4.1 Завдання експериментів

Завдання експериментів поставлені задля отримання відповідей на питання дослідження ПД9, 3, 1, 7, та 8, що в сукупності складають завдання дослідження ЗДЗ (див. Таблицю 1.5). Ці експериментальні завдання полягають у наступному.

1. Перевірити, чи дає обчислювальний конвеєр для виявлення та вимірювання термінологічного насичення, розроблений у розділах 2 та 3, **правильні результати** у граничних випадках: як тоді, коли насичення слід очікувати дуже швидко, так і тоді, коли цього не можна очікувати. Для цієї перевірки ми готуємо дві синтетичні колекції, 1DOC та RAW, у розділі 4.2.2 та застосовуємо розроблений конвеєр до цих колекцій у розділі 4.3.

2. Рекомендувати найбільш відповідну зовнішню програмну реалізацію програмного забезпечення методу C-value, яка буде використана для виявлення та вимірювання термінологічного насичення. Для цього, ми проводимо крос-оцінку двох програмних інструментів NaCTeM TerMine та UPM Term Extractor (див. розділ 2.9) шляхом використання цих інструментів у нашому базовому конвеєрі стосовно трьох реальних колекцій документів у розділі 4.4.

3. Дізнатися яке з можливих впорядкувань додавання документів для обробки найкраще гарантує, що: набір термінів, на якому досягається насичення, є компактним та репрезентативним; насичення досягається швидко, є стабільним після моменту досягнення та не є насиченням накопленого регулярного шуму. Для виконання цього завдання ми проводимо крос-оцінку п'яти можливих впорядкувань документів та ранжування результатів у розділі 4.5. Ранжування виконується з використанням вимірюваних аспектів, представлених у розділі 4.2.3.

4. Виявити чи використання техніки групування термінів, на основі вимірювання подібності термінів **позитивно впливає на ефективність та якість термінологічного насичення** з точки зору швидшого виявлення насичення та збереження більшої кількості значущих термінів. Для досягнення цієї мети ми проводимо крос-оцінку: (i) базового обчислювального конвеєру, що використовує

базовий алгоритм **TND**; проти (ii) розширеного конвеєру з групуванням термінів за допомогою алгоритмів **STG**, **M-JR**, **M-JW**, **M-JA**, **M-SD**, та **R-TND** (розділ 3.6).

5. Валідувати оптимізований метод та обчислювальний конвеєр для виявлення та вимірювання термінологічного насичення. Для цього ми, у розділі 4.7, експериментально доводимо гіпотезу *h1* Теорема 2.5. Ми також **оцінюємо ефективність, незалежність від домену, та масштабованість** розробленого оптимізованого конвеєру обчислень (розділ 2.6), застосовуючи його алгоритми (розділ 3.4) реалізовані у програмному забезпеченні (розділ 3.8) до трьох різних реальних колекцій документів.

4.2 Загальні налаштування експериментів

У цьому розділі пояснюється загальна постановка експериментів. Особливості окремих експериментальних серій, якщо такі є, детально описані нижче у відповідних розділах з 4.3 до 4.7. Експериментальний робочий процес та набір інструментів, підготовлений на базі розробленого програмного забезпечення (розділ 3), описані у розділі 4.2.1; колекції документів та набори даних – у розділі 4.2.2; міри, використані в експериментах – у розділі 4.2.3; та експериментальне середовище – у розділі 4.2.4.

4.2.1 Експериментальний робочий процес та інструментальне програмне забезпечення

Загальний робочий процес, який використовується в наших оціночних експериментах, базується на робочому процесі для виявлення та вимірювання термінологічного насичення (Рис. 3.1). Він використовує ті самі модулі, що формують потік обчислень (Рис. 3.3) і, якщо потрібно, ітерації, як описано у розділі 3.1. Інструментальне програмне забезпечення для виконання експериментального робочого процесу представлено у розділі 3.8. Для ілюстрації, експериментальний робочий процес представлено на Рис. 4.1.

4.2.2 Колекції документів та набори даних

У цьому розділі ми описуємо дані, які були використані в наших експериментах. Ці дані надходять із двох синтетичних та чотирьох реальних колекцій документів.

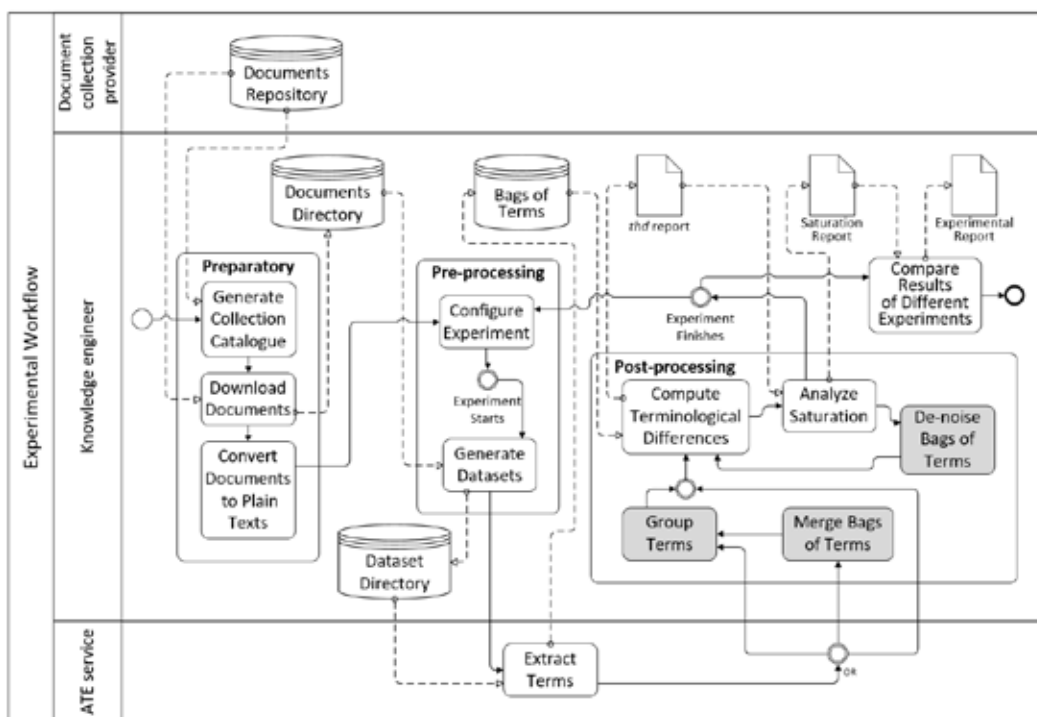


Рис. 4.1. Експериментальний робочий процес. Необов’язкові завдання виділено сірим кольором.

Синтетичні дані використовувались для перевірки правильності нашого методу виявлення та вимірювання термінологічного насичення та реалізованого програмного забезпечення. Тому колекції, 1DOC та RAW, були підготовлені для оцінки граничних випадків: того, в якому термінологічне насичення має відбуватися негайно; та іншого, в якому не повинно відбуватися термінологічне насичення.

1DOC – це колекція документів, що містить лише один документ. Як вихідний документ для цієї колекції ми використали повний текст (Ermolayev et al. 2014). Колекція була вручну перетворена в плоский текстовий формат ANSI. З плоского тексту було створено набори даних D_1, D_2, \dots, D_{20} , як описано у розділі 3, а інкрементом для кожного наступного набору даних був текст цього документа. Отже, D_1 містив одну копію тексту цього документа, D_2 – дві копії цього ж тексту, \dots, D_{20} – 20 копій цього самого тексту. Зрозуміло, що, якщо наш підхід до вимірювання насичення правильний, насичення в даному випадку слід спостерігати досить швидко з thd близьким до 0, оскільки всі інкременти однакові.

Ідея побудови колекції випадково обраних статей з Вікіпедії (Random Articles from Wikipedia (RAW)) протилежна попередньому кейсу. Щоб уникнути насичення, потрібна колекція, у якому всі інкременти суттєво відрізняються між собою термінологічно. Щоб зробити це, ми повинні скласти документи, що стосуються різних тем з різних областей і, отже, використовують дуже різну термінологію. Для створення такої колекції RAW, ми випадково вибрали 80 статей з англійської Вікіпедії таким чином, що жодна з них не схожа з іншими, а розмір статті не надто малий. Статті завантажено у форматі PDF в 1 стовпчик. Далі ми обробили ці PDF-файли для перетворення в плоскі ASCII тексти, використовуючи наш конвертор з PDF в плоский текст. Тексти не були очищені, щоб зберегти можливість перевірки того, як шум, який додається Вікіпедією у роздруківки PDF, впливає на насиченість. З цих текстів, ми створили 20 наборів даних, D_1, D_2, \dots, D_{20} , з інкрементами, що складаються з 4 випадкових документів з цієї колекції.

Для оціночних експериментів із реальними науковими документами було зібрано наступні колекції рецензованих наукових статей: TIME, DMKD, DAC, та KM. Ці колекції були попередньо оброблені та відповідні набори даних сформовані Kosa et al. (2017a). Кожна з цих чотирьох колекції складаються із статей, опублікованих на одному, або декількох різних міжнародних наукових майданчиках, але таких, що належать до одного домену. Колекція DMKD містить підмножину повнотекстових статей з журналу Springer «Data Mining and Knowledge Discovery»²². Домен цієї колекції – управління знаннями (Knowledge Management, KM). Колекція TIME містить повні тексти статей серії симпозіумів TIME²³. Доменом є подання та вивід знань про час (Time Representation and Reasoning). Колекція DAC містить підмножину повнотекстових статей з праць конференцій «Design Automation Conference»²⁴. Доменом колекції DAC є автоматизація інженерного дизайну (Engineering Design Automation).

²² <https://link.springer.com/journal/10618>

²³ http://time.di.unimi.it/TIME_Home.html

²⁴ <http://dac.com/>

Для генерації наборів даних із колекцій TIME, DMKD та DAC, було вибрано інкремент в 20 статей²⁵.

До колекції документів²⁶ DMKD, ми включили 300 статей опублікованих між 1997 і 2010 роками. Усі документи у повному тексті були автоматично оброблені за допомогою нашого інструментального конвеєру (розділ 4.1). На основі доступних документів ми створили 15 інкрементально збільшених наборів даних D_1, D_2, \dots, D_{15} .²⁷

Колекція TIME була зібрана у ході нашого попереднього дослідження (Ermolayev et al. 2014). Вона складається з повних текстів статей, опублікованих у матеріалах серії симпозіумів TIME в період з 1994 по 2013 і нараховує 437 документів. Документи цієї колекції оброблялися вручну, включаючи їх перетворення у плоскі тексти та очищення цих текстів. Тому створені набори даних були помірно зашумлені (moderately noisy). На основі доступних текстів ми створили 22 інкрементально збільшені набори даних D_1, D_2, \dots, D_{22} .²⁸

Для колекції DAC було відібрано 506 статей, опублікованих в період з 2004 по 2010 роки. Ці документи були автоматично перетворені у плоскі тексти за допомогою нашого інструментального програмного забезпечення (розділ 4.1). Було

²⁵ Інкремент в 20 статей було обрано як відповідний до форми діаграм та довжини таблиць. Відповідно до Висновку 3.6, див. також (Kosa et al. 2020), розмір інкременту не впливає на результат.

²⁶ Ця колекція надана Springer на основі їх політики забезпечення повними текстами для цілей інтелектуального аналізу даних: <https://www.springer.com/gp/rights-permissions/springer-s-text-and-data-mining-policy/29056>

²⁷ Колекція **DMKD** у плоскому тексті: <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-637dc34c-fa29-4587-9f63-df0e602d6e86>; інкрементально збільшені набори даних, створені з цих текстів: <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-9043e879-5f77-486f-bc56-cb6af3cdd306>

²⁸ Колекція **TIME** у плоскому тексті: <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-d1e5f2b6-c51e-4572-b10d-0e2ebcceed02>; інкрементально збільшені набори даних, створені з цих текстів: <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-06cc93de-de58-43b9-a378-743b61ef9093>

створено 26 інкрементально збільшені набори даних D_1, D_2, \dots, D_{26} ²⁹. Усунення шуму в тексті було навмисно виключено, щоб можна було порівнювати результати між наборами термінів, що мають багато шуму (без видалення стоп-термінів) з помірно зашумленими наборами термінів (після видалення стоп-термінів) відповідно до підходу, запропонованого у (Kosa et al. 2018a).

Розміри зібраних колекцій та сформованих наборів даних можуть показатися замалими, щоб забезпечити достатню кількість експериментальних свідчень. Однак це не так. Усі три колекції документів були відібрані з більших колекцій таким чином, щоб зробити їх якомога малими, але достатньо великими щоб спостерігати термінологічне насичення. У представленому дослідженні, чим менше колекція, тим краще для обговорення результатів, оскільки відмінності у вимірюваних аспектах (розділи 4.4 – 4.6) краще помітні при менших наборах даних та інкрементах.

Масштабованість та обчислювальна ефективність обчислювального конвеєру для виявлення та вимірювання термінологічної насиченості було перевірено, використовуючи впорядкування документів **dcf** (Kosa et al. 2020), на повній колекції документів КМ у домені управління знаннями (Kosa et al. 2017a). Загалом колекція документів КМ містить близько 9 000 статей, з 15 журналів Springer, що стосуються різних аспектів управління знаннями. Склад цієї колекції документів схематично показаний на Рис. 4.2.

Відфільтрувавши статті, які не були надто інформативними для здобуття знань, наприклад, редакційні статті, і статті, для яких не були доступні повні тексти, ми зібрали колекцію з 7 445 повнотекстових журнальних статей³⁰. Ці

²⁹ Колекція **DAC** у плоскому тексті: <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-010b1add-cd5c-4b33-b6ce-8c93301d880b>; інкрементально збільшені набори даних, створені з цих текстів: <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-e07d55cc-6830-4d9e-9b90-d69010050508>

³⁰ Повні тексти були надані Springer на основі їх політики щодо надання повнотекстових матеріалів для цілей інтелектуального аналізу даних: <https://www.springer.com/gp/rights-permissions/springer-s-text-and-data-mining-policy/29056>. Обсяг колекції КМ після перетворення на звичайні тексти становить 413.66 Мб.

документи були автоматично попередньо перетворені у плоскі тексти (Kosa et al. 2018a) і не були очищені. Тому отримані набори даних були помірно зашумлені, подібно до колекції DMKD. Ми обрали інкремент (*inc*) для створення наборів даних у $inc = 100$ статей³¹. Отже, ми сформували 75 датасетів розміру *inc*, утворюючих партицію $\{D_i = \{d_j\}_{j=1}^{100}\}_{i=1}^{75}$ колекції КМ³² для оптимізованого конвеєру. Інкрементально збільшені набори даних не створювалися.



Рис. 4.2. Розподіл статей у журналах колекції КМ (Kosa et al. 2017a). Вісь Y представляє роки публікації, а вісь X відповідає журналам (скорочення). Цифри в стовпцях: кількість томів, кількість випусків, та загальна кількість статей.

Характеристики всіх колекцій документів та наборів даних, що використовувались в наших експериментах зведені в Таблиці 4.1.

4.2.3 Вимірювані аспекти та міри

У своїх експериментах ми вивчаємо вплив різних аспектів використовуваних інструментів та даних на: (i) термінологічне насичення; (ii) виявлення накопичення регулярного шуму; (iii) продуктивність методу, включаючи його масштабованість. У таблиці 4.2 представлені вимірювані аспекти, що використовуються при аналізі експериментальних результатів, по відношенню до наших цілей (розділ 4.1).

³¹ Інкремент в 100 статей було обрано як відповідний до форми діаграм на рисунках.

³² Партиція колекції КМ не була зроблена загальнодоступною, оскільки для цього потрібні додаткові дозволи Springer.

Таблиця 4.1. Особливості використовуваних колекцій документів
та наборів даних

Колекція	Тип	Тип та Макет документів	Кіл-ть док-ів	Шум	Інкремент (кіл-ть статей)	Кіл-ть наборів даних
IDOC	синтетична	журнал, ACM 1-колонка	1	очищена вручну	1 стаття	20
RAW	синтетична	Вікіпедія 1-колонка	80	не очищена, помірно зашумлена	4 статей	20
DMKD	реальна	журнал, Springer 1-колонка	300	не очищена, незначно зашумлена, регулярний шум не накопичується	20 статей	15
TIME	реальна	конференція, IEEE 2-колонки	437	ручна обробка, помірно зашумлена, регулярний шум не накопичується	20 статей	22
DAC	реальна	конференція, IEEE 2-колонки	506	не очищена, значно зашумлена, накопичується регулярний шум	20 статей	26
KM	реальна	журнал, Springer 1-колонка	7 445	не очищена, незначно зашумлена регулярний шум не накопичується	100 статей	75

Таблиця 4.2. Вимірювані аспекти, використані в експериментах

Аспект	Означення	Що вимірюється	Обґрунтування	Правило Ранжування
Вивчення впливу на термінологічне насичення (DMKD, TIME, DAC очищені набори даних)				
(i) Компактність	<i>Seps</i>	Індивідуальний поріг значущості терміну (<i>eps</i>) у точці насичення	З визначення <i>eps</i> у розділі 2 (Визначення 2.4) впливає, що вищі значення призводять до більш компактних наборів значущих збережених термінів (за принципом простої більшості).	Чим вище – тим краще
(ii) Продуктивність	<i>SthdSP</i>	Точка входу в зону насичення (<i>thd</i> нижче, ніж <i>eps</i>) – точка насичення	Чим раніше виявлено насичення, тим менше вимірювань потрібно проводити на наборах даних менших розмірів.	Чим раніше – тим краще
(iii) Репрезентативність	<i>SPrtSP</i>	Середня частка кількостей збережених значущих термінів до всіх здобутих термінів протягом усього діапазону вимірювань	Більша частка збережених значущих термінів прямо вказує на більшу повноту (<i>recall</i>), отже, призводить до більш репрезентативного набору термінів. Середнє значення протягом всього інтервалу вимірювання використовується для обходу флуктуацій в окремих точках.	Чим вище – тим краще
(iv) Стабільність	<i>SthdV</i>	Інтегральна волатильність значень <i>thd</i> у зоні насичення (від інтегрально, тим менш імовірно, що	Чим нижчою спостерігається волатильність значень <i>thd</i>	Чим нижче – тим краще

	точки входу до останнього вимірювання, на яку не впливає регулярне накопичення шуму)	насичення порушується в наступних точках вимірювання - отже, воно є більш стабільним .	
Вивчення чутливості до надмірного регулярного шуму (Колекція DAC Naturelle)			
(v) Чутливість до накопиченого регулярного шуму	<i>NepsPH</i> Висота піку <i>eps</i>	Якщо спостерігається різкий і високий пік <i>eps</i> , то можна припустити, що накопичився регулярний шум (з високими C-value). Оскільки ці регулярні елементи шуму накопичуються у верхній частині набору термінів, для реальних кандидатів у терміни більше немає місця. Отже, будь-які подальші здобуття марні, оскільки це здобуття регулярного шуму. Порядок був би більш чутливим до регулярного шуму, якщо він призводить до вищого (<i>NepsPH</i>) та більш раннього (<i>NepsPP</i>) піку <i>eps</i> .	Чим вище – тим краще
	<i>NepsPP</i> Точка вимірювання піку <i>eps</i>		Чим раніше – тим краще
	<i>NthdPP</i> Точка вимірювання піку <i>thd</i>	Виходячи з визначення <i>thd</i> як міри термінологічної різниці, можна очікувати, що за наявності накопиченого регулярного шуму вона: (а) різко коливатиметься з великою амплітудою; та (б) незабаром після цього впаде до значень, близьких до нуля.	Чим раніше – тим краще

Для оцінки цих аспектів у різних експериментах проводяться наступні вимірювання:

- NET – кількість здобутих кандидатів у терміни (у B_i)
- *eps* – індивідуальний поріг значущості терміна
- NRT – кількість збережених значущих термінів (у T_i)
- PRT – частка кількостей збережених значущих (NRT) до всіх здобутих (NET) термінів у відсотках
- *thd* – значення термінологічної різниці між T_i та T_{i-1}
- *thdr* – нормалізоване значення термінологічної різниці між T_i та T_{i-1}
- ThdV – дискретний аналог 1^{oi} похідної значень *thd*; обчислюється як різниця між поточними значеннями $thd(T_i T_{i-1})$ та попередніми значеннями $thd(T_{i-1} T_{i-2})$

4.2.4 Експериментальне середовище

Усі розрахунки, крім здобуття термінів, виконані на 64-розрядному PC Windows 7 з процесором Intel® Core™ i5, M520 @ 2.40 ГГц; 8.0 Гб вбудованої пам'яті; графічний процесор NVIDIA Geforce GT330M GPU. Для здобуття термінів використаний ПК Ubuntu 16.04LTS, з процесором Intel® Xeon® E5-2603 v3 @ 1.60ГГц 12 ядер, 128 Гб вбудованої пам'яті, та і жорстким диском 3.5Тб.

4.3 Перевірка коректності методу на синтетичних колекціях

Метою цієї експериментальної серії було перевірити розроблені метод та інструментальний конвеєр програмного забезпечення, включаючи обидва інструменти, що потрапили до короткого списку для АЗТ (NaCTeM TerMine та UPM Term Extractor, розділ 1.9) щодо граничних випадків: (i) колекція 1DOC; та (ii) колекція RAW. Завдяки дизайну цих синтетичних колекцій (розділ 4.2.2), очікується, що результати на 1DOC продемонструють швидку та стабільну термінологічну насиченість, тоді як результати на RAW не повинні давати термінологічну насиченість.

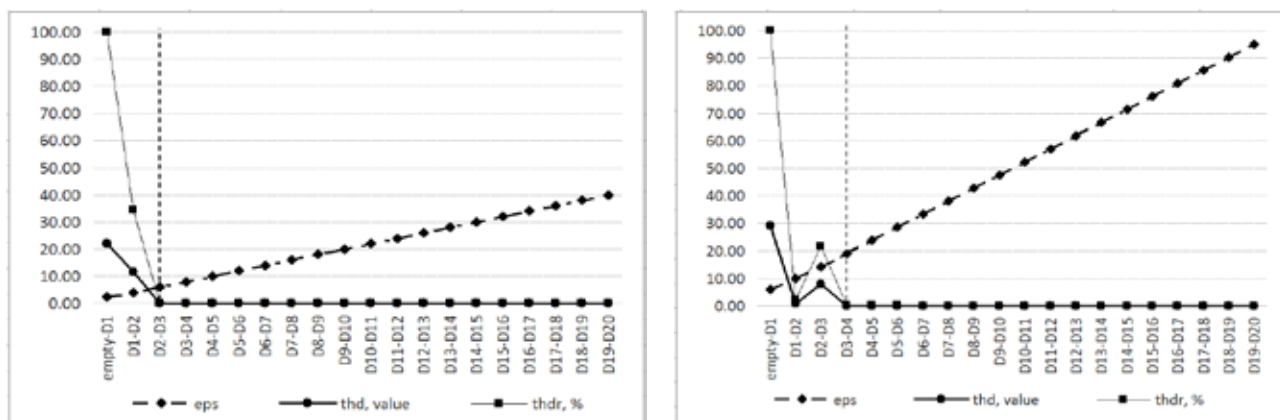
5.3.1 Результати експериментів та їх обговорення

Для наборів термінів, здобутих з 1DOC, результати вимірювання насичення виглядають наступним чином. Спочатку ми обробили набори термінів, здобуті за допомогою TerMine. Результати вимірювання індивідуальних порогів значущості термінів (*eps*) та термінологічних різниць (*thd*, *thdr*) наведено на Рис. 4.3(а)³³. Потім ми виміряли термінологічні різниці між наборами термінів, здобутих за допомогою UPM Extractor. Результати вимірювання індивідуальних порогів значущості термінів (*eps*) та термінологічних різниць (*thd*, *thdr*) представлено на Рис. 4.3(б).

Пунктирна вертикальна лінія на Рис. 4.3(а) вказує на набір термінів (здобутий з D_3) в якому вперше спостерігався показник насичення, коли *thd* опустився нижче *eps*. Насправді, як і очікувалося, пізніше ми також спостерігаємо стійке насичення

³³ Значення, виміряні у всіх експериментах, про які повідомляється, хоча іноді згадуються в тексті, не представлені в дисертації для економії місця. Усі ці експериментальні дані та результати представлені в деталях у супроводжуючому технічному звіті (Kosa et al. 2017b), який є загальнодоступним в Інтернеті.

з тією ж кількістю здобутих термінів та збільшенням індивідуального порогу значущості терміну *eps*.



(а) Набори термінів, здобуті TerMine

(б) Набори термінів, здобуті UPM Extractor

Рис. 4.3. Візуалізація вимірювань насичення на наборах даних 1DOC

Значення *thd* та *thdr* опускаються, щоб стати статистично рівними нулю, починаючи з T_2 - T_3 . Пунктирна вертикальна лінія на Рис. 4.3(б) вказує на набір термінів (здобутий з D_4) в якому вперше спостерігався показник насичення, коли *thd* опустився нижче *eps*. Дуже подібно до випадку TerMine, як і очікувалось, пізніше ми також спостерігали дуже стабільну насиченість з такою ж кількістю здобутих термінів та збільшенням індивідуального порогу значущості терміну *eps*. Значення *thd* та *thdr* опускаються, щоб стати статистично рівними нулю, починаючи з T_3 - T_4 .

Різниця у вимірюваннях насичення для наборів термінів, здобутих за допомогою TerMine та UPM Extractor наступна: (i) UPM Extractor генерує більші набори термінів з $C\text{-value} > 1$: 3 019 термінів проти 1 208 у випадку TerMine; (ii) індивідуальний поріг значущості термінів (*eps*) був приблизно в 2.5 рази вище для UPM Extractor; (iii) кількість збережених термінів з $C\text{-value} > eps$ було у ~ 2 рази більше у випадку UPM Extractor

Загалом, порівняно з результатами UPM Extractor, результати TerMine показали дещо швидшу збіжність до насичення. З іншого боку: (i) кількість збережених значущих термінів із насиченої підколекції; та (ii) точка відсікання індивідуального порогу значущості терміну – були вищими в результатах UPM Extractor. На основі спостереження за цими різницями, можна зробити висновок,

що з лінгвістичної точки зору, TerMine був приблизно у 3 рази більш вибірковою щодо здобуття кандидатів у терміни. Отже, перед-обробка в TerMine є більш складною. З іншого боку, відсікання на виходах UPM Extractor відбувались для термінів які були приблизно вдвічі більш значущими. Таким чином, частина статистичної обробки в UPM Extractor здобуває більш компактні, але значущі набори термінів. Це вказує на те, що завдяки фазі статистичної обробки, UPM Extractor є більш вибірковою інструментом.

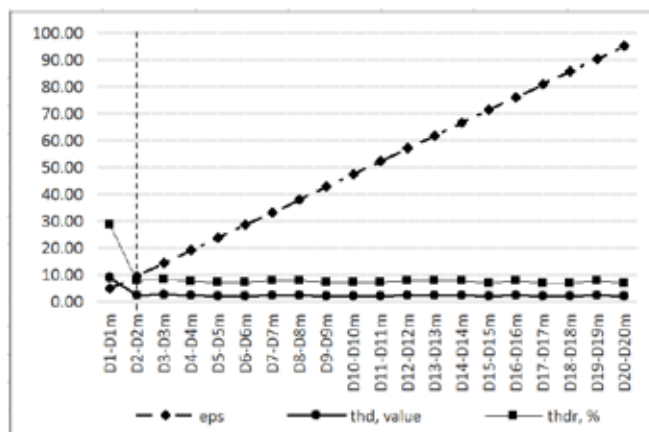


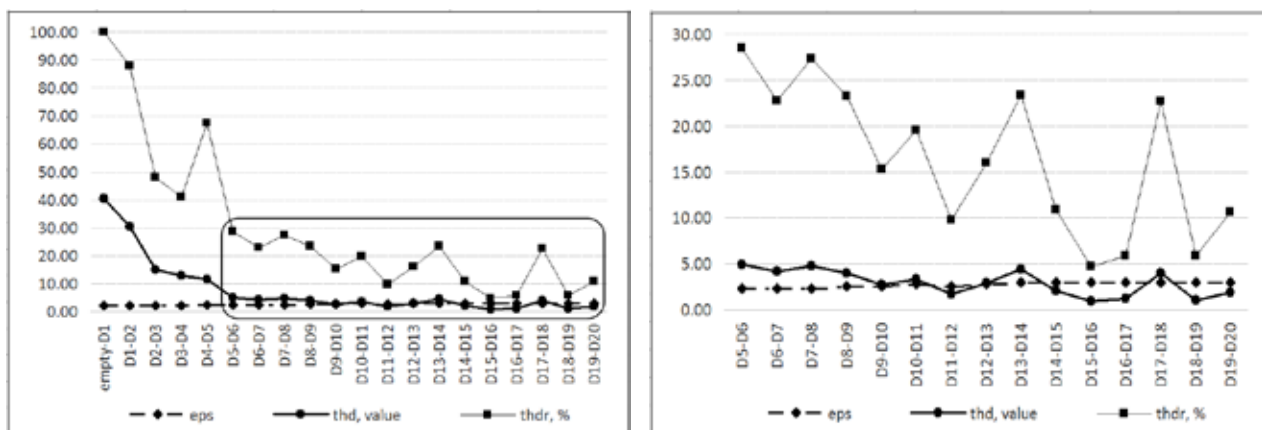
Рис. 4.4. Порівняння збережених наборів термінів, здобутих з колекції 1DOC програмами UPM Extractor та TerMine

Далі було перевірено, чи здобували обидва інструменти статистично схожі набори термінів з колекції 1DOC. Ці вимірювання відображені на Рис. 4.4. З рисунка видно, що обидва інструменти здобували статистично однакові набори термінів незважаючи на те, що кількість збережених термінів суттєво відрізнялася в окремих випадках (про які повідомлялося вище). Термінологічна різниця стала статистично незначною у другій точці вимірювання, де значення *thd* (2.291409) значно опустилося нижче *eps* (9.509775). Ця ситуація була стабільною, оскільки значення *thd* коливались близько 2.1, а значення *eps* стабільно зростали до 95.

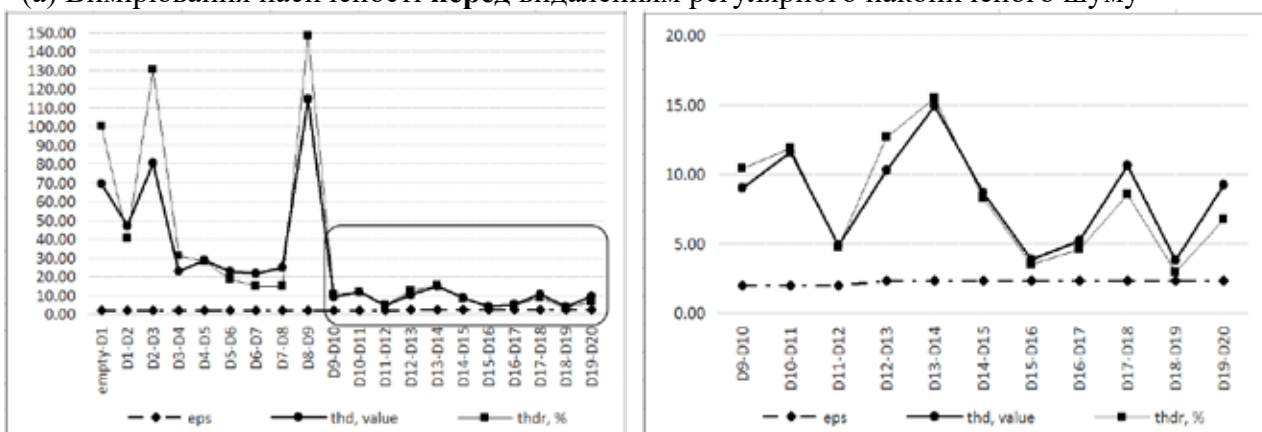
Для наборів термінів, здобутих з RAW результати вимірювання насичення виглядають наступним чином. Спочатку ми обробили набори термінів, здобуті за допомогою TerMine. Результати вимірювання індивідуальних порогів значущості термінів (*eps*) та термінологічних різниць (*thd*, *thdr*) представлені на Рис. 4.5(a).

Потім ми проаналізували B_{20} , здобутий за допомогою TerMine, починаючи з вершини списку до термінів, що мають C-value більше 40. На основі цього

сканування ми здобули список з приблизно 200 стоп-термінів. Ці стоп-терміни були видалені з наборів термінів B_1, \dots, B_{20} і аналіз насичення було повторено. Результати вимірювання індивідуальних порогів значущості термінів (eps) та термінологічних різниць ($thd, thdr$) для очищених наборів термінів представлені на Рис. 4.5(б).



(а) Вимірювання насиченості **перед** видаленням регулярного накопиченого шуму



(б) Вимірювання насичення **після** видалення регулярного накопиченого шуму

Рис. 4.5. Візуалізація вимірювань насичення на наборах термінів колекції RAW здобутих за допомогою TerMine. Діаграма праворуч представляє більш детальний вигляд закругленого прямокутника на діаграмі зліва.

Дивлячись на Рис. 4.5(а) і, особливо, на Рис. 4.5(б), ми спостерігаємо, що, як і очікувалось, колекція RAW не є термінологічно насиченою. Далі, розглядаючи відмінності між Рис. 4.5 (а) та (б), ми спостерігаємо деякі показники наявності шуму в текстових документах колекції. Дійсно, значення $thdr$ на Рис. 4.5(а) набагато вищі за відповідні значення thd . Хоча значення thd натякають на те, що набори термінів можуть бути близькими до насичення, значення $thdr$ набагато

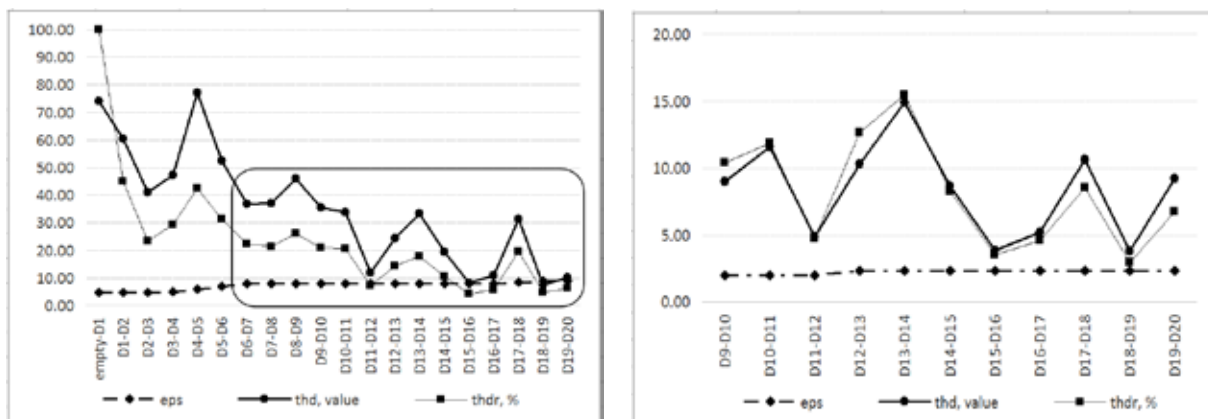
перевищують *eps*. Дуже цікаво, що значення *thd* виміряні після видалення стоп-термінів стають подібними до значень *thdr*. У той же час криві *thd* та *thdr* на Рис. 4.5(б) дуже нагадують криві *thdr* на Рис. 4.5(а). Отже, суттєві відмінності між значеннями *thd* та *thdr* можуть сигналізувати про можливу необхідність видалення регулярного накопиченого шуму з наборів термінів.

Потім той самий експеримент був повторений для наборів термінів, здобутих за допомогою UPM Term Extractor. Результати вимірювання насичення виглядають наступним чином.

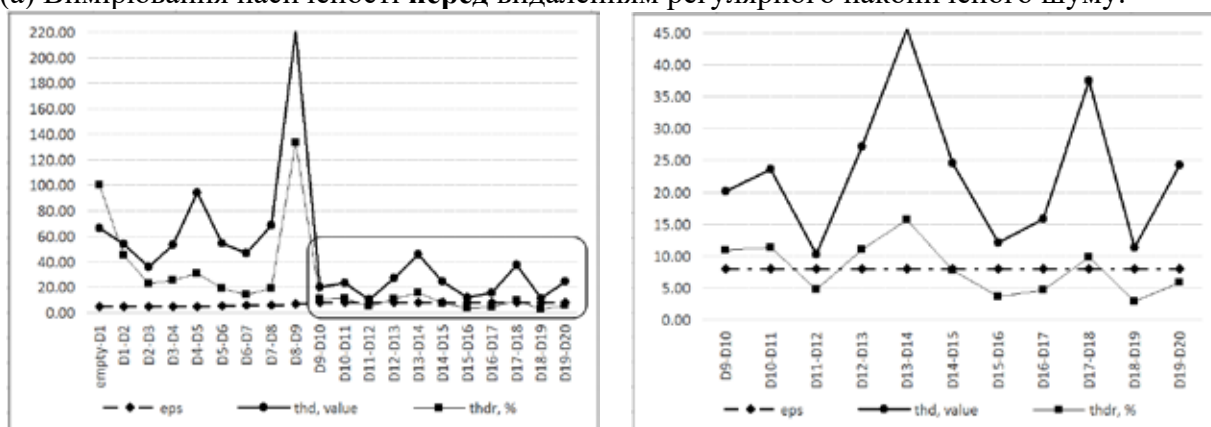
Значення індивідуальних порогів значущості термінів (*eps*) та термінологічних різниць (*thd*, *thdr*) представлено на Рис. 4.6(а). Потім ми проаналізували B_{20} , здобутий за допомогою UPM Extractor, починаючи з вершини списку до термінів, що мають C-value більше 40. На основі цього сканування ми здобули список з приблизно 220 стоп-термінів. Ці стоп-терміни були видалені з наборів термінів B_1, \dots, B_{20} і аналіз насичення було повторено. Значення індивідуальних порогів значущості термінів (*eps*) та термінологічних відмінностей (*thd*, *thdr*) для очищених наборів термінів показано на Рис. 4.6(б).

Порівняно з вимірюваннями насичення для наборів термінів здобутих за допомогою TerMine, значення *thd* для наборів термінів, здобутих за допомогою UPM Extractor формують більш чітку картину відсутності насичення. Фактично, значення *thd* виміряні в результатах UPM Extractor перед видаленням стоп-термінів в 2.5-3 рази перевищують показники результатів TerMine після видалення стоп-термінів.

Отже, результати UPM Extractor є більш контрастними порівняно з результатами TerMine з точки зору виявлення відсутності насичення. З іншого боку, значення *thdr* виміряні за допомогою TerMine є чіткішим показником необхідності очищення наборів термінів. Значення *thdr* виміряні за результатами UPM Extractor не відрізняються від відповідних значень *thd*. Якщо UPM Extractor використовується для виявлення відсутності насичення, немає реальної потреби аналізувати, чи вказують значення *thdr* на наявність шуму. Отже, у даному випадку використання UPM Extractor є кращим, оскільки це більш вибірковий інструмент.



(а) Вимірювання насиченості **перед** видаленням регулярного накопиченого шуму.



(б) Вимірювання насиченості **після** видалення регулярного накопиченого шуму.

Рис. 4.6. Візуалізація вимірювань насичення на наборах термінів колекції RAW здобутих за допомогою UPM Extractor.

Для цієї колекції не було виміряно, чи здобувають обидва інструменти статистично подібні набори термінів. Це вимірювання не мало б значення за відсутності насичення.

4.3.2 Рекомендація інструмента для АЗТ

На підставі результатів наших експериментів з перевірки правильності (розділ 4.3.1), можна стверджувати, що: (і) обидва оцінені інструменти працюють правильно і дають очікувані результати; та (ii) використання UPM Term Extractor є кращим, ніж NaСТeM TerMine, як для термінологічно насичених колекцій документів, так і для виявлення того, що насиченість навряд можна очікувати.

4.4 Вибір програмного забезпечення для АЗТ

У цьому розділі ми повідомляємо результати крос-оцінки UPM Term Extractor та NaСТeM TerMine з використанням реальних колекцій документів, DMKD, TIME, та DAC (розділ 4.2.2).

4.4.1 Результати експериментів

Для наборів даних, здобутих з DMKD, результати виглядають наступним чином. Набори термінів, здобуті TerMine були оброблені першими. Результати вимірювання індивідуальних порогів значущості термінів (*eps*) та термінологічних різниць (*thd*, *thdr*) представлені на Рис. 4.7.

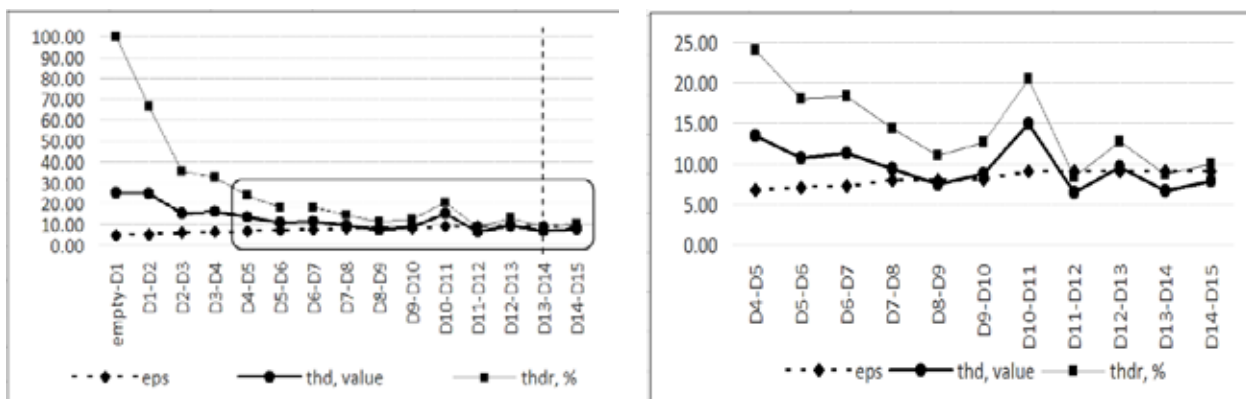


Рис. 4.7. Вимірювання насиченості в наборах даних DMKD на основі наборів термінів, здобутих за допомогою TerMine

Діаграма зліва візуалізує усі вимірювання. Округлений прямокутник обмежує область на діаграмі зліва, яка більш детально представлена на діаграмі справа. Пунктирна вертикальна лінія вказує на набір термінів (здобутий з D_{14}) в якому вперше спостерігався показник насичення, коли *thd* опустилася нижче *eps*. Аналіз цих результатів вказує на те, що існує тенденція щодо досягнення термінологічної насиченості, можливо, для більших наборів даних. Значення *eps* мають тенденцію до зростання, а значення *thd*, *thdr* знижуються зі збільшенням номерів набору даних. Зростання кількості збережених термінів також зменшується. Є три термінологічних піки в області, що нас найбільше цікавить D_{10} - D_{11} , D_{12} - D_{13} , та D_{14} - D_{15} . Однак, внесок цих піків не є дуже значним, оскільки значення *thd* зростає не дуже сильно порівняно з сусідніми вимірами – див. результати DAC нижче, де спостерігаються значно вищі піки. Загалом, зарано вважати DMKD насиченим на основі результатів, здобутих за допомогою TerMine.

Результати вимірювання насичення на основі наборів термінів, здобутих UPM Extractor зображені на Рис. 4.8. Можна відзначити, що стабільна насиченість досягається при D_5 - D_6 . Кількість збережених термінів (з B_6) становить 4 113, що

істотно менше, ніж 5 009 у першій точці потенційного насичення у випадку TerMine. Цікаво, що значення *thd* та *thdr* виміряні за результатами UPM Extractor поводяться досить схоже на значення, виміряні за результатами TerMine, також натякаючи на термінологічні піки в тих самих точках. Кількість збережених термінів, за результатами, отриманими за допомогою UPM Extractor, нижча, хоча і не суттєво. Насиченість досягається завдяки набагато вищим значенням індивідуального порогу значущості терміну *eps*.

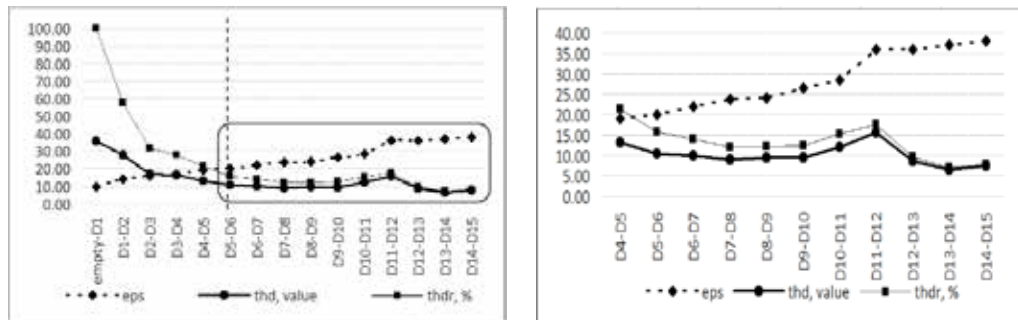


Рис. 4.8. Вимірювання насиченості в наборах даних DMKD на основі наборів термінів, здобутих за допомогою UPM Extractor

Отже, для цієї колекції документів, **UPM Extractor** дає **більш компактні** набори **значущих термінів** і відсікання відбувається при значно вищих значеннях індивідуального порогу значущості терміну.

Однією з гіпотез про причину кращої роботи UPM Extractor може бути те, що він здобуває не всі терміни з документів, які приймає, і TerMine досягає значно більшої повноти. Щоб перевірити це, ми виміряли термінологічні відмінності між наборами термінів, що здобуваються, з тих самих наборів даних UPM Extractor та TerMine. Результат зображений на Рис. 4.9. Рис. 4.9 показує, що обидва інструменти здобувають дещо схожі набори термінів. Ця подібність збільшується із зростанням набору даних. Ці спостереження натякають на те, що здобуті набори термінів схожі, і значення повноти окремих інструментів відрізняються не надто сильно, що є прийнятним.

Цікаво, що термінологічна різниця (*thd*) на Рис. 4.9 опускається нижче *eps* саме в той момент, коли результати TerMine показують найвищий термінологічний пік (див. Рис. 4.7). Отже, схоже на те, що обидва інструменти здобувають подібні

набори термінів але TerMine досягає рівня насиченості трохи пізніше, коли отримує вхідні дані з інкременту, що сприяє найвищому піку термінології. Але що цікаво, значення *thd* виходять за межі *eps* після D_{11} . Однією з можливих причин цього може бути зростаючий вплив накопичення регулярного шуму в наборах даних, що сприймається окремими інструментами по-різному.

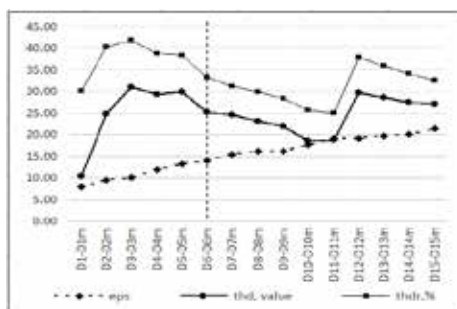


Рис. 4.9. Порівняння збережених наборів термінів, здобутих з колекції DMKD за допомогою UPM Extractor та TerMine

Результати вимірювань насиченості для TIME зображені на Рис. 4.10. Вимірювання насиченості на основі наборів термінів, здобутих TerMine не показали насичення – Рис. 4.10(a). Значення *thd* не опустилися нижче *eps*.

Тенденція подібна до експерименту DMKD – тренд щодо досягнення термінологічного насичення, яке можливе для більших наборів даних. Значення *eps* зростають зі збільшенням кількості наборів даних, хоча значно повільніше, ніж у випадку DMKD. Максимальне спостережуване значення *eps* становить 5 для TIME проти 9 для DMKD. Значення *thd* та *thdr* знижуються зі збільшенням номерів наборів даних, але недостатньо швидко, щоб опуститися нижче *eps*. Як наслідок, максимальна кількість збережених значущих термінів значно вища, ніж у випадку DMKD: 8 343 проти 5 438, хоча різниця у кількості здобутих термінів не така значна: ~287К проти ~253К. Термінологічні піки в колекції TIME спостерігаються у D_3 - D_4 , D_{10} - D_{11} , D_{17} - D_{18} , та D_{19} - D_{20} . Найвищий пік у D_{10} - D_{11} . Подібно до DMKD, внесок цих піків є не дуже значним, оскільки значення *thd* зростає не дуже сильно порівняно з сусідніми вимірюваннями.

Вимірювання насиченості на наборах термінів, здобутих UPM Extractor виявляють **стабільне насичення**, починаючи з D_{11} - D_{12} – як показано на Рис. 4.10(б) вертикальною штриховою лінією. Значення *thd* та *thdr* нагадують значення

випадку TerMine, тому крива насичення має термінологічні піки майже в одних і тих самих точках. Однак, висота цих піків нижча.

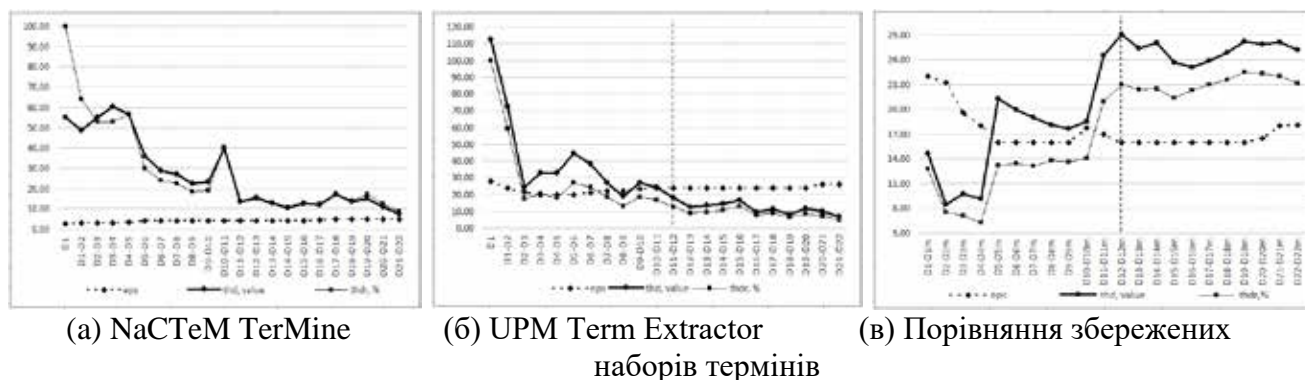


Рис. 4.10. Вимірювання насичення для колекції TIME

Значення індивідуальних порогів значущості термінів *eps* є набагато вищими – подібно до експерименту з DMKD. Насичення виявляється при *eps* рівному 23.774, тоді як значення *eps* у випадку TerMine не збільшуються понад 5.000. Кількість значущих термінів, збережених з B_{12} становить 7 110, що є лише 2.47% від загальної кількості термінів, здобутих з B_{12} . Тому ми можемо зробити подібний висновок і для цього експерименту. Насиченість досягається завдяки набагато вищим значенням індивідуального порогу значущості терміну *eps*. Для TIME, **UPM Term Extractor** дає **більше компактні набори значущих термінів** і відсікання відбувається при набагато вищих значеннях індивідуального порогу значущості термінів.

Ми також перевірили, чи обидва інструменти здобувають подібні набори термінів із колекції TIME. Результати вимірювались за тим самим підходом, що і у випадку з DMKD і зображені на Рис. 4.10(в). Можна побачити, що термінологічна різниця (*thd*) між наборами збережених значущих термінів в точці насичення D_{12} - $D_{12}m^{34}$ дорівнює ~ 29 , тоді як *eps* дорівнює 16. Отже, *thd* в 1.81 рази перевищує *eps*. У випадку DMKD різниця між *thd* та *eps* в точці насичення дещо нижча – у 1.80 рази. Дуже подібно до випадку DMKD, різниця зростає після точки насичення, що, як ми вважаємо, можна пояснити тією ж причиною – впливом накопичення

³⁴ D_{12} – це набір даних, з якого здобувається B_{12} за допомогою UPM Extractor і $B_{12}m$ за допомогою TerMine. $B_{12}m$ перетворено у формат UPM Extractor і пара $(B_{12}, B_{12}m)$ подана в модуль THD.

регулярного шуму в наборах даних за межами точки насичення. Отже, ручне очищення документів TIME (розділ 4.2.2) насправді не дуже допомогло, оскільки результати дуже нагадують випадок DMKD, для якої файли документів не очищались вручну.

Результати вимірювань насичення для DAC наведені на Рис. 4.11. Колекція DAC більше зашумлена, ніж DMKD і TIME. Результати також відрізняються за значеннями, але не в загальній картині.

Вимірювання насичення на наборах термінів, здобутих TerMine, виявили потенційну точку насичення лише в останньому вимірі при $D_{25}-D_{26}$ – як зображено на Рис. 4.11(a). Однак термінологічний пік на $D_{24}-D_{25}$, при якому *thd* дорівнює 135.49, натякає на подальшу нестабільність. Отже, говорити про тенденцію до досягнення стабільного насичення пізніше у даному випадку є спекулятивним. Щоб судити про це, потрібні додаткові виміри.

Цікаво також порівняти поведінку насичення в DAC та в TIME, оскільки обидві колекції походять від одного видавця, тому мають однаковий макет і представляють статті однакового розміру. Різниця полягає в тому, що TIME було очищено вручну, а DAC ні. Рис. 4.10(a) для TIME і Рис. 4.11(a) для DAC, якщо порівнювати, показують різницю у значеннях вимірювань для пар даних приблизно однакових розмірів.

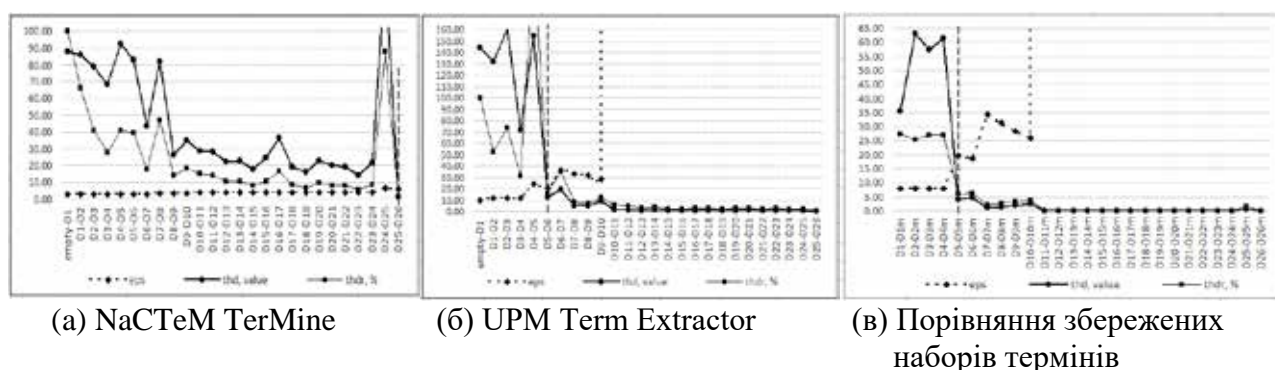


Рис. 4.11. Вимірювання насичення для колекції DAC

Порівняння вимірювань для TIME та DAC, на наборах термінів здобутих за допомогою TerMine, показує, що: (i) значення *eps* зростають швидше для TIME ніж для DAC; (ii) кількість здобутих та збережених термінів для DAC значно більша, ніж для TIME; (iii) кількість збережених термінів для TIME монотонно зростає, і це

зростання сповільнюється – показник можливого насичення в майбутніх вимірюваннях; (iv) кількість збережених значущих термінів для DAC опускається суттєво нижче попереднього значення на D_{24} - D_{25} і thd різко зростає з 21.51 до 135.49. Ми знову вважаємо, що причиною цього падіння та піку є вплив накопичення регулярного шуму. Однак, TerMine сигналізує про проблему доволі пізно.

Вимірювання насиченості на наборах термінів, здобутих за допомогою UPM Extractor **виявляють стійку насиченість**, починаючи з D_5 - D_6 з eps приблизно рівним 20, як показано на Рис. 4.11(б) вертикальною штриховою лінією. Однак значення eps досягають піка 18 294 при D_{10} - D_{11} , а кількість збережених членів знижується до 34, що більше ніж у 100 разів менше попереднього значення. Більш уважне вивчення показало, що ці 34 терміни є нічим іншим, як накопиченим регулярним шумом, про що вже було отримано сигнал набагато раніше у випадку з UPM Extractor. Тому, у випадку зашумленої колекції документів, UPM Extractor набагато чутливіше виявляє надмірний шум у порівнянні з TerMine. Отже, ситуація, зображена на Рис. 4.11(б) може бути використана як показник необхідності видалення накопиченого регулярного шуму зі здобутих наборів термінів.

Ми також порівняли, чи були набори термінів, здобуті обома інструментами, статистично однаковими. Результат зображений на Рис. 4.11(в). Порівняння показало, що, починаючи з D_5 , де thd дорівнює 3.97 та eps до 19.65, обидва інструменти успішно здобувають дуже подібні набори накопиченого шуму.

4.4.2 Обговорення та рекомендація

Щодо DMKD, яка була попередньо оброблена автоматично (розділ 4.2.2), на основі результатів здобуття TerMine, не можна достовірно судити про те, що колекція DMKD насичена. На відміну від цього, вимірювання насичення на наборах термінів, здобутих UPM Extractor, досить швидко виявили стійке насичення. Обидва інструменти здобували статистично подібні набори термінів.

Для TIME, яка була очищена вручну, вимірювання насичення на наборах термінів, здобутих TerMine, не виявили насичення. Дуже подібно до випадку

DMKD, вимірювання насичення на наборах термінів, здобутих UPM Extractor, виявило стійку насиченість досить швидко, а також із значно вищими індивідуальними порогами значущості термінів *eps*. Це призвело до здобуття більш компактних наборів збережених значущих термінів.

Випадки DMKD та TIME продемонстрували подібні переваги UPM Extractor перед TerMine у виявленні насичення та збереженні значущих термінів, надавши більш компактні набори значущих термінів. Зменшення шуму вручну в колекції TIME не допомогло покращити результати вимірювань насичення, тому не було необхідним.

Для DAC, яка була навмисно залишена з великою кількістю шуму, UPM Extractor продемонстрував здатність виявляти накопичення надмірного регулярного шуму в наборах термінів значно раніше, ніж TerMine. Крива насичення, побудована для вимірювань за допомогою результатів UPM Extractor, сигналізує про цей шум досить різко, при цьому кількість збережених значущих термінів падає на два порядки, а індивідуальні пороги значущості термінів зростають до максимуму на три порядки.

Отже, у випадку з зашумленими наборами даних та через відсутність надмірної невибіркової у здобутті кандидатів у терміни, UPM Extractor є набагато більш чутливим у виявленні накопичення регулярного шуму, порівняно з TerMine.

На основі узагальнених спостережень, використання UPM Extractor є кращим порівняно з TerMine для виявлення термінологічного насичення або надмірного шуму. Цей висновок не обмежується доменом і не залежить від ручного видалення шуму у вихідних даних колекції.

4.5 Вплив впорядкування документів

У цьому розділі ми представляємо та обговорюємо результати наших експериментальних досліджень, спрямованих на з'ясування, яке впорядкування документів при побудові наборів даних є найкращим. Ця презентація розпочинається з опису додаткових умов експерименту у контексті цієї серії експериментів (розділ 4.5.1). Дискусія продовжується результатами порівняльного

дослідження впливу різних можливих впорядкувань на термінологічне насичення у розділі 4.5.2. Далі, результати дослідження чутливості до накопичення регулярного шуму обговорюються у розділі 4.5.3. У розділі 4.5.4, наведено ранжування впорядкувань за результатами цих двох досліджень та запропоновано нашу рекомендацію.

Усі експериментальні результати детально представлені, як таблиці та діаграми, у супроводжувальному технічному звіті (Kosa et al. 2018b)³⁵ та доступні як частина нашого загальнодоступного набору даних OntoElect³⁶.

5.5.1 Особливості експериментальних налаштувань

Як було представлено у розділі 2.1, підхід послідовного наближення для побудови послідовності підколекцій документів $DSC_1, \dots, DSC_i, \dots$, є інкрементальним. Отже, $DSC_1 = INC_1$ та $\forall i \geq 2, DSC_i = DSC_{i-1} \cup INC_i$, де INC_i є підмножиною певного числа (*inc*) документів з *CDC*. Щодо цього ітеративного процесу необхідно висловити кілька застережень.

(i) Компактність. Додавання нових документів до обробленої підколекції означає, що кількість вилучених термінів зростає із збільшенням кількості ітерацій. Аналогічно, кількість збережених значущих термінів помітно зростає, якщо не досить швидко зростає індивідуальний поріг значущості терміну eps_i . Чим більша кількість збережених термінів, тим більш обчислювально затратною є подальша частина конвеєру вивчення онтології. Отже, наша мета полягає в тому, щоб знайти спосіб максимізувати зростання eps , щоб мати більш компактні, але репрезентативні набори, що зберігають важливі терміни.

(ii) Продуктивність. Інкрементальне зростання підколекцій означає, що чим більше ітерацій проводиться, тим більш обчислювально затратним є такий конвеєр.

³⁵ Технічний звіт (Kosa et al. 2018b) є загальнодоступним за посиланням

<https://github.com/OntoElect/Docs/blob/master/Reports/TS-RTDC-TR-2018-2-v2.pdf>

³⁶ <http://dx.doi.org/10.17632/knb8fgyr8n.1#folder-348f6201-7a55-4c96-a67a-47ec54c9d558>

Отже, однією з цілей нашого дослідження є пошук способу якнайшвидшого досягнення насичення, щоб зменшити понесені обчислювальні витрати³⁷.

(iii) Репрезентативність. По суті, підхід (розділ 2.1) заснований на балансуванні на межі простої більшості голосів щодо обчислення *eps* на кожній ітерації. Отже, набори збережених значущих термінів не можуть мати високу повноту. Знаходження способу збільшити пропорції збережених значущих до всіх здобутих термінів, зберігаючи при цьому компактність збережених наборів термінів, може підвищити їх репрезентативність та якість здобуття термінів. Покращення репрезентативності - ще одне завдання нашого дослідження.

(iv) Стабільність. Як було продемонстровано в наших попередніх дослідженнях, наприклад (Ermolayev et al. 2014), умова насичення (2.4) може бути порушена після її виконання на попередніх ітераціях. Причиною цього може бути отримання значної кількості нових значущих термінів у наступних ітераціях. У таких випадках ми кваліфікуємо насичення як нестабільне. Отже, ще одна проблема, яку ми вирішуємо – знайти спосіб мінімізувати нерівномірність у розподіленні збережених значущих термінів у послідовності інкрементально зростаючих підколекцій для покращення стабільності термінологічного насичення.

(v) Чутливість до накопичення регулярного шуму. Через інкрементальний та ітеративний характер нашого конвеєру, фрагменти текстів, які є елементами регулярного шуму (наприклад, часто згадуване місце публікації або строка-тире, що представляє лінію), можуть накопичуватися з високими значеннями *score*. Ці стоп-терміни можуть займати верхні частини наборів збережених значущих термінів. Отже, виявлене насичення насправді є насиченням стоп термінів. Щоб уникнути цих помилкових неправдиво-позитивних кейсів, відшукується спосіб підвищення чутливості конвеєру щодо накопичення регулярного шуму. Це ще одне завдання нашого дослідження.

³⁷ Альтернативним способом підвищення продуктивності є використання партиції колекції документів та обчислення об'єднаних часткових C-value (Kosa et al. 2020) замість C-value (Frantzi and Ananiadou 1999). У будь-якому випадку, підхід (Kosa et al. 2020) також є ітеративним. Отже, чим раніше виявляється насичення, тим кращою є продуктивність.

Аспекти (i - v) були фактично об'єднані у питанні дослідження (ПД1 – впорядкування документів, розділ 1.13.2). Далі ми розширюємо формулювання цього питання, щоб краще висвітлити, як збалансовано враховуються ці аспекти:

ПД1 (розширене): *Як впорядкувати документи з CDC для формування INC_i щоб забезпечити, що насичення швидко досягається на компактному, хоча і репрезентативному наборі збережених значущих термінів, є стабільним після цього моменту, та не є насиченням накопиченого регулярного шуму?*

Щоб відповісти на це питання, ми припускаємо, що колекції складаються з релевантних документів, опублікованих у різний час. Отже, можливо, що для доменів, які еволюціонують у часі, документи, що складають колекцію, містять різні терміни і розподіл цих термінів не є рівномірним по всій колекції. Ми позначаємо це явище як **часове зміщення** у термінології. Деякі терміни можуть пройти перевірку часом і довго залишатись значущими для опису домену. Отже, ці терміни розподілені досить рівномірно по всій колекції. Інші терміни можуть стати незначними доволі швидко після їх введення. Отже, їх розподіл може бути нерівномірним. Тому, часове зміщення є ускладнюючим фактором для формування підколекцій документів. У цьому контексті, ми визначаємо (Kosa et al. 2017a) чотири різні можливі впорядкування документів, які потрібно додати до підколекцій, виходячи з їх міток на шкалі часу:

- Хронологічне впорядкування (**ch**) – значення часових міток збільшуються. Першими беруться найдавніші документи. Найновіші документи беруться в останню чергу. Використання цього порядку може мати сенс, якщо в колекції переважають документи, що пройшли перевірку часом, а домен є добре усталеним, тому з часом він суттєво не змінюється.

- Зворотнє хронологічне впорядкування (**rch**) – значення часових міток зменшуються. Спочатку беруться новіші документи. Найдавніші документи беруться останніми. Використання цього порядку може мати сенс, якщо домен з недавнього часу перебуває у стані швидких змін.

- Двоспрямоване впорядкування (**bd**) – документи беруться по обидва боки періоду часових міток, по черзі. Спочатку береться найстаріший, потім

найновіший, потім найстаріший із решти і т.д. Використання цього порядку може мати сенс, якщо існує приблизний баланс між документами, які пройшли перевірку часом та документами, які нещодавно стали значущими. Можливим недоліком такого шляху може бути те, що документи, опубліковані в середині часового інтервалу колекції ніколи не відбираються.

- Випадкове впорядкування (**rn**) – документи беруться випадковим чином. Використання цього порядку може мати сенс, якщо немає інформації про вплив різних документів та швидкість термінологічних змін у домені.

На додаток до часу публікації, ще одним аспектом, важливим з точки зору значущості та строку життя терміну, є вплив документа, що містить цей термін, на опис домену. Дійсно, якщо документ впливає на спільноту та читачів у домені, тоді терміни, використані в цьому документі, можуть бути важливими для спільноти. У наукових доменах, де документи є науковими статтями, релевантним способом вимірювання такого впливу є кількість цитувань. Однак, ця кількість повинна бути збалансована часом, що минув з моменту публікації. Отже, частота цитування документів (наприклад, кількість цитувань на рік) може бути відповідною мірою, оскільки вона врівноважує позицію документа на шкалі часу та його вплив на опис домену. Отже, ще одним потенційно цікавим впорядкуванням документів для підколекцій є:

- Впорядкування за зменшенням частоти цитування (**dcf**) – документи не беруться в порядку їх часових позначок. Першими обираються документи, що мають найвищу частоту цитування³⁸ і вибір продовжується до найнижчої частоти цитування.

Гіпотеза, яку ми висунули та перевірили в нашому експериментальному дослідженні, така:

Гіпотеза Н4.1: *впорядкування dcf*. Використання впорядкування **dcf** є **найбільш збалансованим та ефективним** з точки зору компактності,

³⁸ Частота цитування у нашому підході обчислюється як кількість цитат протягом років після публікації, поділена на кількість років після публікації мінус один.

продуктивності, репрезентативності, стабільності та чутливості до ARN (аспекти і - v), порівняно з усіма впорядкуваннями на основі часових позначок.

Ми вивчаємо вплив порядку додавання документів до наборів даних на:

- (i) Термінологічне насичення; та
- (ii) Виявлення накопичення регулярного шуму

Цей вплив оцінюється експериментально за допомогою трьох колекцій документів, DMKD, TIME, та DAC, описаних у розділі 4.2.2.

Щодо дослідження термінологічного насичення, наші конкретні експериментальні завдання полягають у тому, щоб з'ясувати, чи існує певний порядок, який приводить до збалансування аспектів (і – iv) компактності, продуктивності, репрезентативності та стабільності (розділ 4.5.2). Це дослідження проводиться з використанням наборів даних DMKD, TIME, та DAC. В експерименті з використанням наборів даних DAC, здобуті набори термінів очищуються від шуму на фазі пост-обробки, використовуючи алгоритм **ARNR** (розділ 3.7), які далі позначені як набори даних DAC Cleaned.

При дослідженні чутливості до накопичення регулярного шуму (аспект (v)), наша експериментальне завдання полягає в тому, щоб з'ясувати, чи існує певне впорядкування, яке є найбільш чутливим до появи такого шуму в даних, а отже і виявляє його швидше. Це дослідження проводиться на наборах даних колекції DAC. У цьому експерименті здобуті набори термінів не очищуються від шуму – далі позначені як набори даних DAC Naturelle.

У аналізі результатів експерименту ми використовуємо кілька вимірюваних аспектів для ранжування впорядкування (Таблиця 4.2). Ранжування проводиться за шкалою від 1 (найкраще) до 5 (найгірше) для кожної комбінації окремого аспекту та колекції документів. Ці окремі ранги підсумовуються, щоб визначити загально найкраще впорядкування (Таблиця 4.5).

Використовується експериментальний робочий процес та інструментальне програмне забезпечення, які описані у розділі 4.2.1. Загалом, **Підготовчий** етап виконується тричі – по одному на колекцію документів. Етап **Перед-обробки** виконується п'ятнадцять разів – по одному на впорядкування додавання документів

до колекції. Етапи **Здобуття Термінів** та **Пост-обробки** виконуються двадцять разів – п'ятнадцять разів для різних порядків, і, крім того, п'ять разів для DAC Cleaned, після видалення накопиченого регулярного шуму.

4.5.2 Результати дослідження термінологічного насичення

Вимірювання в точках насичення для усіх п'яти оцінюваних порядків (розділ 4.5.1) наведені у Таблиці 4.3 для порівняння.

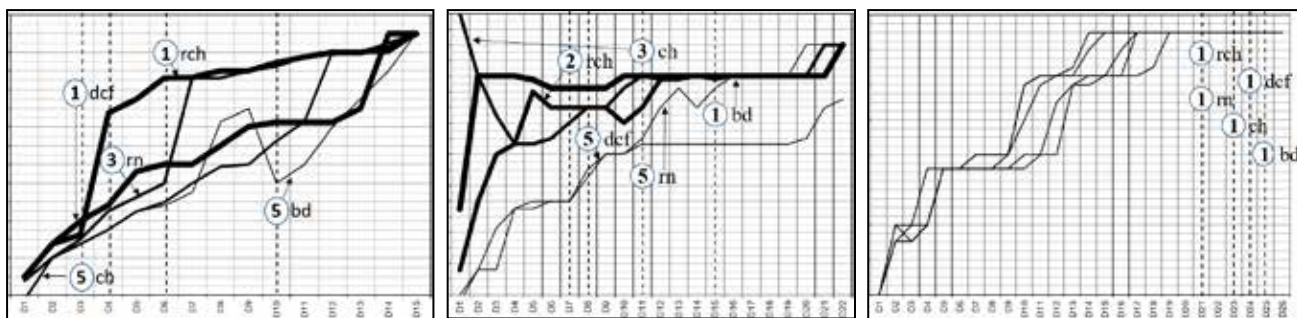
Таблиця 4.3. Порівняння вимірювань насичення у їх точках насичення для всіх порядків

Поря- док	Пара наборів даних (D_i, D_{i-1})	NET (B_i)	eps (B_i)	NRT (T_i) ($C\text{-value} > eps$)	PRT, %	thd (T_i, T_{i-1})	ThdV
<i>DMKD набори даних</i>							
ch	D_4, D_3	89 617	17.0000	3 242	3.6176	16.2008	-0.8587
rn	D_5, D_4	124 331	20.5000	3 739	3.0073	17.9221	-5.9448
rch	D_7, D_6	182 768	33.2842	3 774	2.0649	18.9340	-18.0829
bd	D_{11}, D_{10}	319 441	24.0000	8 054	2.5213	13.9341	-31.7709
dcf	D_4, D_3	109 076	19.6515	3 399	3.1162	18.0366	-5.3458
<i>TIME набори даних</i>							
ch	D_{12}, D_{11}	287 887	23.7744	7 110	2.4697	18.1096	-6.0550
rn	D_{12}, D_{11}	252 767	22.0000	7 129	2.8204	10.4441	-15.4888
rch	D_8, D_7	155 715	22.0000	4 550	2.9220	20.1549	-29.9347
bd	D_{15}, D_{14}	329 240	24.0000	8 723	2.6494	11.4488	-15.8609
dcf	D_9, D_8	174 103	19.0000	5 493	2.5692	15.4377	-4.1768
<i>DAC Cleaned набори даних</i>							
ch	D_{24}, D_{23}	514 364	15.5098	20 558	3.9968	13.6261	-2.2046
rn	D_{22}, D_{21}	481 885	15.5098	18 516	3.8424	10.8970	-6.8867
rch	D_{22}, D_{21}	486 649	15.5098	18 865	3.8765	10.9433	-7.6712
bd	D_{26}, D_{25}	551 165	15.5098	21 468	3.8950	3.7072	-12.8841
dcf	D_{25}, D_{24}	518 765	15.5098	19 991	3.8536	3.1238	-8.0866

Індивідуальні пороги значущості термінів ($Seps$). Індивідуальні пороги значущості термінів (eps) у точках виявлення термінологічного насичення порівнюються в контексті цього вимірюваного аспекту. Результати зображені на Рис. 4.12 (а)–(в).

Як видно з Рис. 4.12(а), найвищі значення eps досягаються в точках насичення, якщо для колекції DMKD використовуються впорядкування **dcf** або **rch**. Однак, точка насичення для **rch** (D_7, D_6) досягається пізніше, ніж для **dcf** (D_4, D_3). Використання впорядкування **rn** оцінюється як середнє за шкалою – 3.

Впорядкування **ch** та **bd** кваліфікуються як найгірші, оскільки вони дають найнижчі значення *eps* у відповідних точках насичення.



(а) Набори даних DMKD (б) Набори даних TIME (в) Набори даних DAC Cleaned

Рис. 4.12. Індивідуальні пороги значущості термінів (*eps*) для різних впорядкувань додавання документів до наборів даних³⁹.

Ранжування впорядкувань відносно колекції TIME (Рис. 4.12(б)) є відмінним від DMKD. Найвищий ранг присвоюється впорядкуванню **bd**, хоча його точка насичення є найпізнішою. Другий ранг має **rch**, третій – **ch**, тоді як **dcf** та **rn** мають найнижчий ранг – 5.

У експерименті з наборами даних DAC Cleaned (Рис. 4.12(в)), насичення досягається доволі пізно. При цьому, всі впорядкування мають однакові значення *eps* у точках насичення. Отже, всі порядки ранжуються однаково – 1.

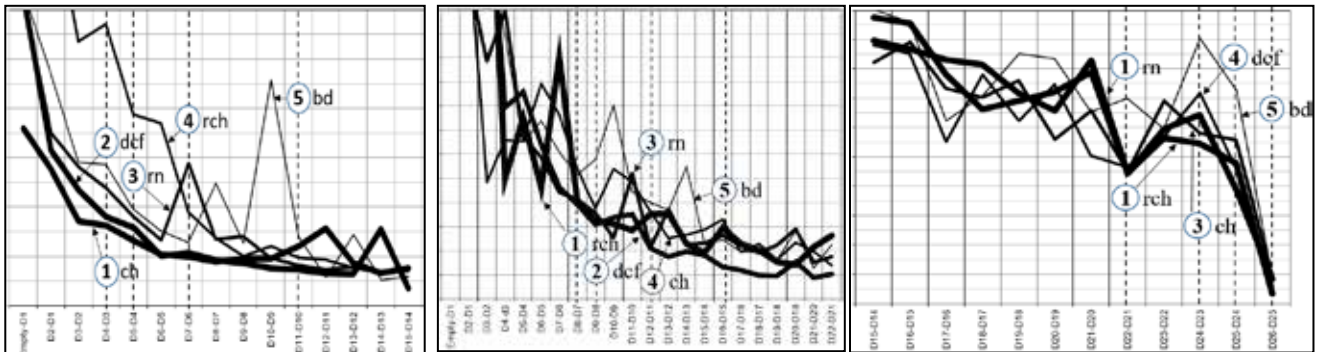
Підсумкові ранги щодо цього вимірюваного аспекту для трьох колекцій: 4 – **rch**; 7 – **bd, dcf**; 9 – **ch, rn**.

Точки входу в зони насичення (*SthdSP*). Сенс цього вимірюваного аспекту полягає в тому, щоб з'ясувати, який порядок додавання документів дозволяє швидше досягти термінологічного насичення. Насичення виявляється, як описано у розділі 2.1, перевіркою умови (2.4). Графічно ця умова означає, що крива *thd* опускається нижче кривої *eps* у точці вимірювання (D_i, D_{i-1}) – яка є точкою

³⁹ На цьому рисунку і далі на Рис. 4.13 – 4.15, 4.17 у цьому підрозділі впорядкування вказуються легендами, що сполучені до кривих стрілками. Точки насичення позначені вертикальними пунктирними лініями. Легенди також включають ранги відповідних впорядкувань, подані у колах. Діаграми відображають криві із різною товщиною. Чим товще лінія, тим краще ранг впорядкування.

насичення – і вона ніколи не виходить за межі кривої *eps*, принаймні до кінця вимірювань.

Для цього вимірюваного аспекту ми порівнюємо номери ітерацій *i* для різних впорядкувань. У випадках з однаковими *i* ми також порівнюємо значення *thd*. Нижче значення у випадку з однаковими *i* дає кращий ранг. Результати зображені на Рис. 4.13 (а)–(в).



(а) Набори даних DMKD

(б) Набори даних TIME

(в) Набори даних DAC Cleaned

Рис. 4.13. Термінологічні відмінності (*thd*) для різних впорядкувань додавання документів до наборів даних.

У випадку DMKD, Рис. 4.13(а), спостерігається, що впорядкування **ch** та **dcf** мають однаковий номер ітерації *i* виявлення насичення - (D_4, D_3). Впорядкування **ch** кваліфікується як найкраще, оскільки при ньому досягається менше значення *thd* у цій точці вимірювання. Впорядкування **dcf** посідає друге місце серед найкращих. Впорядкування **rn** посідає третє місце за рангом на (D_5, D_4). Впорядкування **rch** кваліфікується як четверте на (D_7, D_6). Впорядкування **bd** посідає п'яте місце на (D_{11}, D_{10}).

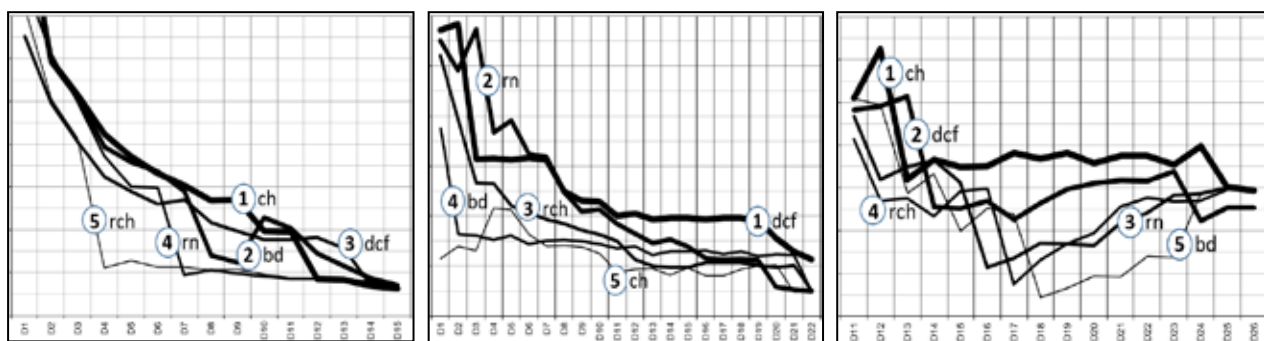
У випадку TIME, Рис. 4.13(б), ранги виглядають наступним чином. Впорядкування **rch** кваліфікується як перше найкраще на (D_8, D_7). Впорядкування **dcf** посідає друге місце серед найкращих на (D_9, D_8). Нічия на (D_{12}, D_{11}) вирішується на користь **rn** (третє) яке має нижче значення *thd* порівняно з **ch** (четверте). Нарешті, впорядкування **bd** оцінюється як п'яте на (D_{16}, D_{15}).

У випадку DAC Cleaned, Рис. 4.13(в), ранги виглядають наступним чином. Нічия на (D_{22}, D_{21}) вирішується на користь **rn** (перше) яке має нижче значення *thd* (10.8970, Таблиця 4.4) ніж **rch** (друге) з трохи вищим значенням *thd* (10.9433).

Впорядкування **ch** кваліфікується як третє на (D_{24}, D_{23}) . Впорядкування **dcf** кваліфікується як четверте на (D_{25}, D_{24}) . Впорядкування **bd** посідає п'яте місце на (D_{26}, D_{25}) .

Підсумкові ранги щодо цього вимірюваного аспекту для трьох колекцій: 6 – **ch**, **rch**; 7 – **rn**; 8 – **dcf**; 15 – **bd**.

Середня частка збережених значущих до всіх здобутих термінів ($SPrtSP$). Сенс цього вимірюваного аспекту полягає у з'ясуванні, яке впорядкування додавання документів призводить до більш репрезентативних наборів збережених значущих термінів. Пропорції обчислюються для всіх ітерацій у діапазоні вимірювань, а потім осереднюються. Результати зображені на Рис. 4.14 (а)–(в).



(а) Набори даних DMKD (б) Набори даних TIME (в) Набори даних DAC Cleaned

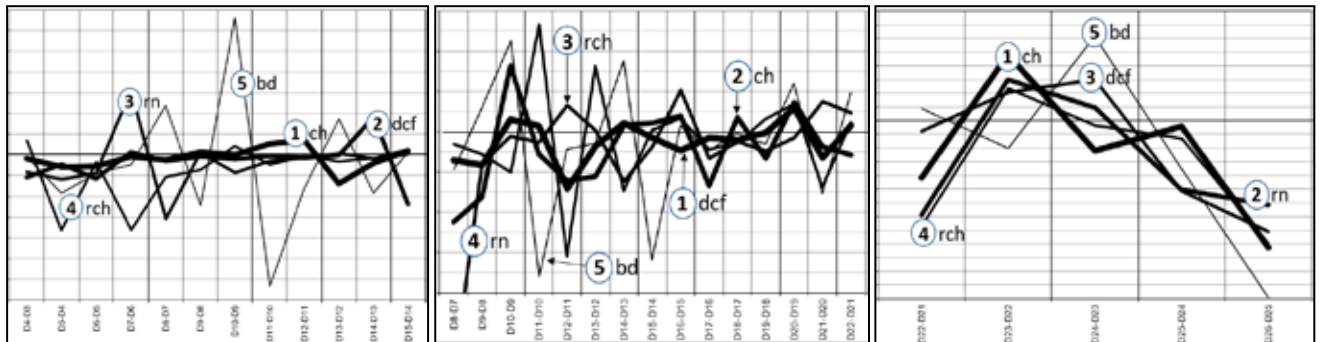
Рис. 4.14. Середня частка збережених до всіх здобутих термінів для різних впорядкувань додавання документів до наборів даних.

У випадку DMKD, Рис. 4.14(а), **ch** призводить до найбільшої частки, що вказує на найкращу репрезентативність відповідних збережених наборів термінів. Найменш репрезентативний результат був отриманий з використанням **rch**. У випадку TIME, Рис. 4.14(б), результат істотно відрізняється, причому **dcf** отримав найвищий ранг, а **ch** – найнижчий. У випадку DAC Cleaned, Рис. 4.14(в), **ch** знову займає найвище місце. Подібні нестабільності спостерігаються для всіх впорядкувань, крім **dcf**, який посідає найвище місце у сумі для всіх трьох колекцій.

Підсумкові ранги щодо цього вимірюваного аспекту для трьох колекцій: 6 – **dcf**; 7 – **ch**, 9 – **rn**; 12 – **rch**; 13 – **bd**.

Інтегральна волатильність thd у зонах насичення ($SthdV$). Результати зображені на Рис. 4.15 (а)–(в). Сенс цього вимірюваного аспекту полягає в тому,

щоб з'ясувати, яке впорядкування додавання документів призводить до більш стабільного процесу насичення. Передбачається, що чим більш плавною є лінія волатильності, тим меншою є вірогідність того, що *thd* виходить за межі *eps* у вимірюваннях після точки насичення (у зони насичення). Значення точкової волатильності (*ThdV*) обчислюються, як пояснено в розділі 4.2.3. Абсолютні значення підсумовуються для всіх вимірювань у зонах насичення для різних впорядкувань.



(а) Набори даних DMKD (б) Набори даних TIME (в) Набори даних DAC Cleaned

Рис. 4.15. Інтегральні волатильності *thd* у зонах насичення для різних впорядкувань додавання документів до наборів даних. Ці ранги базуються на сумах значень точкової волатильності у відповідних зонах насичення. Найменш волатильне впорядкування є найкращим.

Підсумкові ранги щодо цього вимірюваного аспекту для трьох колекцій є: 4 – **ch**; 6 – **dcf**; 9 – **rn**; 11 – **rch**; 15 – **bd**.

4.5.3 Результати дослідження чутливості до регулярного шуму

У цьому експерименті ми зосереджуємося на з'ясуванні того, яке з впорядкувань є більш чутливим до накопичення регулярного шуму. Для цього ми розглядаємо, наскільки різко і рано виникають піки *eps* (*NepsPH*, *NepsPP*) та *thd* (*NthdPP*), що вказують на шум, у наших вимірах, на наборах даних DAC Naturelle, які є суттєво зашумленими. Ці вимірювання представлені Таблиці 4.4 в точках появи піків, які виділені жирним шрифтом.

Таблиця 4.4. DAC Naturelle. Порівняння показників шуму для всіх порядків

Порядок	Набори даних	NET (B_i)	eps (B_i)	NRT (T_i) (C -value > eps)	PRT, %	thd (T_i, T_{i-1})	ThdV
(а) піки eps							
ch	D_{11}	230 726	18 294.0372	34	0,0147	1,6057	-7,3833
rn	D_{12}	286 161	19 616.9148	36	0.0126	0.5045	-51.0163
rch	D_{16}	393 997	3 107.6393	51	0.0129	3.3400	-0.3046
bd	D_{21}	463 660	352.0000	291	0.0628	18.4497	10.0066
dcf	D_{11}	218 764	19 338.4980	33	0.0151	1.4475	-5.7119
(б) піки thd							
ch	D_5, D_4	277 775	32.0000	3191	1.1488	154.2663	82.8619
rn	D_{11}, D_{10}	265 540	48.0000	1934	0.7283	51.5208	87.8267
rch	D_{11}, D_{10}	277 775	32.0000	3191	1.1488	107.9931	86.4081
bd	D_{10}, D_9	234 634	23.7744	4378	1.8659	81.0028	62.2701
dcf	D_5, D_4	106 814	24.0000	1906	1.7844	149.3096	83.8976

Приклад спостережуваного накопичення регулярного шуму у цьому експерименті зображено на Рис. 4.16. Цей рисунок ілюструє, що пік thd є більш точним показником потенційного накопичення регулярного шуму оскільки він виникає саме в точці вимірювання, де це накопичення стає помітним (Рис. 4.16 (а) та (б)). Тому амплітуда піку thd не така велика порівняно з амплітудою піку eps що спостерігається через кілька точок вимірювання пізніше. Пік eps вказує на, так би мовити, сталу основну частину регулярного шуму яка переважає або навіть повністю заміщує збережені значущі терміни. Це спостереження підтверджується: (і) набагато більшою амплітудою піку, порівняно з thd ; та (ii) дуже суттєвим зменшенням кількості збережених значущих термінів.

Рис. 4.17 зображує результати оцінювання та ранжування чутливості різних впорядкувань до накопичення регулярного шуму. Вони оцінюються за допомогою вимірюваних аспектів, представлених у розділі 4.2.3 (Таблиця 4.2) на основі вимірювань eps та thd для всіх порядків.

Щодо висоти (амплітуди) піку eps (Рис. 4.17(а)), найбільш чутливим впорядкуванням є **rn**, що посідає перше місце. За ним слідує впорядкування **dcf** (друге) та **ch** (третє). Всі ці три впорядкування демонструють піки відносно однакової висоти в порівнянні з набагато гіршою чутливістю **rch** (четверте) та **bd** (п'яте).

```

design:527.5
circuit:442.0
delay:433.5
vt prr rrr runo oov jlj kkk:429.04981118614774
power:349.5
algorithm:330.5
number:298.0
time:288.5
model:280.0
input:272.0
system:255.0
performance:232.5
circuits:230.5
t:229.0
results:224.5
algorithm embedding:219.65963210934144
cut cut cut cut cut cut:217.1368500605771

```

(а) T_4 здобуте з D_4 , $thd = 65.42$

```

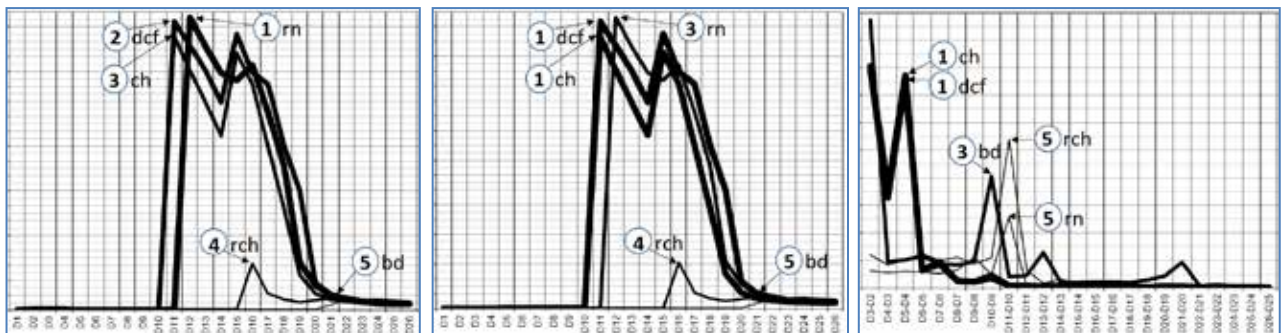
-----:2961.3306319874837
-----:2958.5104260326916
-----:2952.7851388435147
-----:2944.134357365111
-----:2933.074384120042
-----:2918.02162681787
-----:2902.410118409203
-----:2880.0
-----:2861.0166004500015
-----:2829.9051009600926
-----:2809.131974711405
-----:2767.579785640185
-----:2747.009877080575
-----:2692.073503896107
-----:2674.9213164722653
-----:2605.6422007269252
-----:2593.1568569324177

```

(б) T_5 здобуте з D_5 , $thd = 149.31$

Рис. 4.16. Приклад накопичення регулярного шуму в наборах збережених значущих термінів, здобутих з DAC Naturelle з використанням впорядкування **dcf**: (а) верхня частина T_4 , в якій регулярний шум, хоча і присутній (підкреслений), ще не накопичений; (б) верхня частина T_5 в якій накопичився регулярний шум. На (D_5, D_4) спостерігається пік thd – збільшення з 65.42 до 149.31.

Що стосується точок вимірювання піків eps (Рис. 4.17(б)), то найбільш ранніми, отже, найбільш чутливими, є впорядкування **dcf** та **ch**, що обидва мають пік на D_{11} . Тому, обидва займають перше місце. Впорядкування **rn** досягає піку на рівні D_{12} , посівши третє місце. Пік впорядкування **rch** відбувається на D_{16} – четверте місце. Впорядкування **bd** досягає піку на D_{21} , і займає п'яте місце.



(а) Висота піку eps ($NepsPH$) (б) Точка піку eps ($NepsPP$) (в) Точка піку thd ($NthdPP$)

Рис. 4.17. Чутливість до накопичення регулярного шуму з використанням різних порядків додавання документів до наборів даних. Впорядкування вказуються легендами, сполученими до кривих стрілками.

Рейтинг чутливості на основі точок вимірювання піків thd (Рис. 4.17(в)), виявив наступне. Найбільш чутливими є **dcf** та **ch** які обидва досягають піку найпершими, на (D_5, D_4) , з дуже подібними амплітудами. Третє місце отримує **bd** з піком на (D_{10}, D_9) . На п'ятому місці є **rch** та **rn** з піком на (D_{11}, D_{10}) .

Підсумкові ранги щодо чутливості до накопичення регулярного шуму за трьома вимірюваними аспектами: 4 – **dcf**; 5 – **ch**; 9 – **rn**; 13 – **rch**, **bd**.

4.5.4 Загальний ранг та рекомендації

Цей розділ узагальнює та порівнює наші висновки та ранги після аналізу результатів експериментів з крос-оцінювання різних впорядкувань додавання документів до наборів даних для: (i) вимірювання термінологічного насичення; та (ii) виявлення накопичення регулярного шуму. Узагальнення (Таблиця 4.5) структуровано за колекціями документів та вимірюваними аспектами, що використовуються в наших експериментах.

В стовпцях впорядкувань, для окремих впорядкувань ранги вказано для кожного аспекту. Ранги підсумуються для кожної колекції, а ранги впорядкувань присвоюються для кожної колекції, по шкалі: 1 – найвищий до 5 – найнижчий. Нарешті, індивідуальні ранги (за вимірюваними аспектами) та ранги колекції підсумовуються внизу таблиці. Отже, сума індивідуальних рангів вказує на те, наскільки хорошим було впорядкування в цілому по вказаним вимірюваним аспектам. Сума рангів на основі колекцій визначає, наскільки збалансованим було впорядкування у всіх колекціях та експериментах.

Судячи з рангів індивідуальних аспектів, **dcf** демонструє **найкращу загальну відповідність для балансування** всіх необхідних аспектів оцінки (i – v, розділ 4.5.1) з найнижчою сумою індивідуальних рангів – 31. Другим найкращим є **ch** із сумою – 33. Однак, щодо балансу за усіма колекціями, **dcf** та **ch** виявляється мають однакові **найкращі ранги** у всіх експериментах при рівних сумах накопичених рангів – 7. Більш детальне порівняння **ch** з **dcf** показує що: (i) **dcf** (ранг = 4) програє **ch** (ранг = 1) в експерименті з найбільш зашумленою колекцією DAC; однак (ii) **dcf** виявляється найбільш чутливим до накопичення регулярного шуму (ранг = 1), причому **ch** є наступним (ранг = 2). Отже, **dcf** не дуже добре працює на DAC Cleaned через залишки надмірного шуму, який, з іншого боку, краще виявляється за допомогою **dcf**. Виходячи з цього аналізу, **dcf** переважає **ch** в остаточному загальному ранжуванні. Решта порядків є програшними, як це показано у всіх підсумкових рангах у Таблиці 4.5.

Таблиця 4.5. Узагальнення крос-оцінювання різних впорядкувань додавання документів до наборів даних на основі зазначених вимірюваних аспектів.

Вимірюваний аспект	Колекція	Впорядкування				
	Правило ранжування	ch	rch	rn	bd	dcf
Вплив на термінологічне насичення						
DMKD						
<i>Seps</i>	Чим вище – тим краще	5	1	3	5	1
<i>SthdSP</i>	Чим раніше – тим краще	1	4	3	5	2
<i>SPrtSP</i>	Чим вище – тим краще	1	5	4	2	3
<i>SthdV</i>	Чим нижче – тим краще	1	4	3	5	2
Сума:		8	14	13	17	8
Ранг Колекції:		1	4	3	5	1
TIME						
<i>Seps</i>	Чим вище – тим краще	3	2	5	1	5
<i>SthdSP</i>	Чим раніше – тим краще	4	1	3	5	2
<i>SPrtSP</i>	Чим вище – тим краще	5	3	2	4	1
<i>SthdV</i>	Чим нижче – тим краще	2	3	4	5	1
Сума:		14	9	14	15	9
Ранг Колекції:		3	1	3	5	1
DAC Cleaned						
<i>Seps</i>	Чим вище – тим краще	1	1	1	1	1
<i>SthdSP</i>	Чим раніше – тим краще	3	1	1	5	4
<i>SPrtSP</i>	Чим вище – тим краще	1	4	3	5	2
<i>SthdV</i>	Чим нижче – тим краще	1	4	2	5	3
Сума:		6	10	7	16	10
Ранг Колекції:		1	4	2	5	4
Сума Рангів Вимірюваних Аспектів (насичення):		28	33	34	48	27
Сума Рангів Колекцій:		5	9	8	15	6
Чутливість до накопичення регулярного шуму						
DAC Naturelle						
<i>NepsPH</i>	Чим вище – тим краще	3	4	1	5	2
<i>NepsPP</i>	Чим раніше – тим краще	1	4	3	5	1
<i>NthdPP</i>	Чим раніше – тим краще	1	5	5	3	1
Сума Рангів Вимірюваних Аспектів (шум):		5	13	9	13	4
Ранг Колекції:		2	5	3	5	1
Загальна Сума Рангів Вимірюваних Аспектів:		33	46	43	61	31
Загальна Сума Рангів Колекцій:		7	14	11	20	7
Кінцевий Ранг:		2	4	3	5	1

Ранжування на базі зазначених вимірюваних аспектів та результатах наших експериментів з крос-оцінювання доводить, що порядок **dcf** можна обґрунтовано рекомендувати як такий, що дає найкращий баланс між вимогами компактності, продуктивності, репрезентативності, стабільності та чутливості до накопичення регулярного шуму.

4.6 Вплив групування термінів

Цей розділ повідомляє про наше експериментальне дослідження впливу групування термінів на термінологічне насичення. розділ 4.6.1 описує особливості постановки наших експериментів. Результати цих експериментів обговорюються в розділі 4.6.2. Ранжування отриманих результатів та рекомендації, складені на основі загального рейтингу, представлені в розділі 4.6.3.

4.6.1 Особливості Налаштувань Експериментів

Ми оцінюємо вдосконалений алгоритм **R-THD** (розділ 3.6.4) для вимірювання термінологічної різниці у порівнянні з базовим алгоритмом **THD** (розділ 3.5). Ця оцінка виконується з використанням експериментального робочого процесу, представленого в (розділі 4.5.1) та трьох реальних колекцій документів, TIME, DMKD, та DAC Cleaned, представлених у розділі 4.5.2.

Завданням наших експериментів є з'ясувати, чи дає використання вдосконаленого алгоритму **R-THD** швидше та стабільніше термінологічне насичення порівняно з використанням базового алгоритму **THD**. Ми також розглядаємо з'ясування того, які міри термінологічної подібності найкраще підходять до вимірювання термінологічного насичення.

Щоб зробити результати порівнюваними, подаються ті самі набори даних, створені з колекцій документів, як описано в розділі 4.2.2, як до вдосконаленого алгоритму **R-THD** так і до базового алгоритму **THD**. Для кожної колекції документів ми застосовуємо:

- Вдосконалений **R-THD** – шістнадцять разів – по одному на кожну міру строкової подібності (МСП) M^{40} (розділ 3.6.1) та по одному на індивідуальний поріг подібності терміну th (розділ 3.6.2, Таблиця 3.5); та
- Базовий **THD** – один раз, оскільки він не залежить від порогу подібності терміну

⁴⁰ Функції для усіх чотирьох вибраних МСП (розділ 3.8) повертають дійсні значення в межах [0, 1].

Вимірюються наступні значення: (i) кількість збережених значущих термінів (NRT); (ii) абсолютна термінологічна різниця (*thd*); та (iii) час, необхідний для виконання групування подібних термінів за алгоритмом **STG** (*sec*).

Нарешті, щоб перевірити, чи правильні наші реалізації МСП, а отже і **STG** та вдосконалений **R-TND** алгоритми, ми перевіряємо, чи повертає вдосконалений алгоритм **R-TND** результати, які задовільно схожі на результати базового алгоритму **TND**, коли поріг подібності терміну встановлений на 1.00. Це порогове значення означає, що слід розглядати лише еквівалентні строки, як подібні терміни.

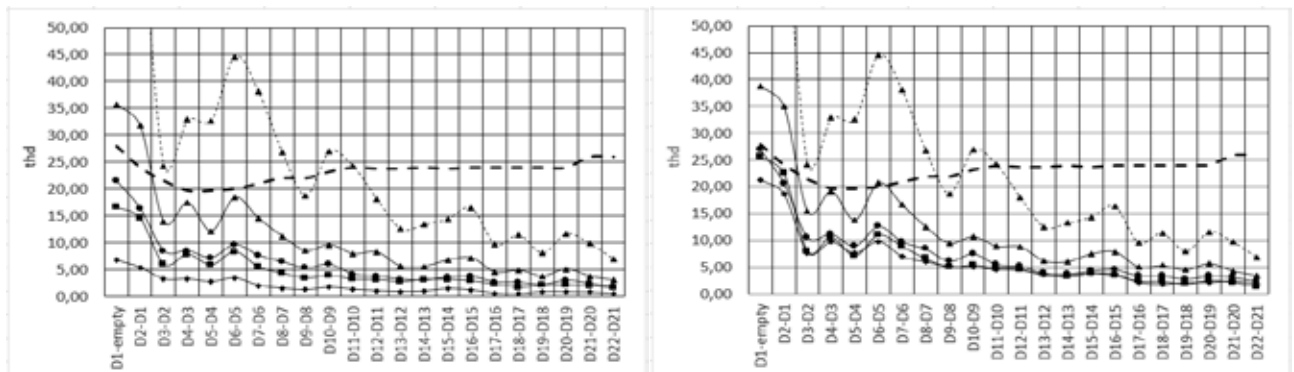
4.6.2 Результати і обговорення

Вимірювання, проведені в наших експериментах для різних колекцій та точки порогів подібності термінів представлені графічно на рисунках нижче та загальнодоступні⁴¹ в повному обсязі, включаючи значення.

Результати наших вимірювань термінологічного насичення (*thd*) зображені на Рис. 4.18 – 4.20. Вимірювання насичення (*thd*) показують, що вдосконалений алгоритм **R-TND** виявляє термінологічне насичення швидше ніж базовий алгоритм **TND**, незалежно від того, якою обрана міра подібності термінів (*M*) або поріг подібності (*th*).

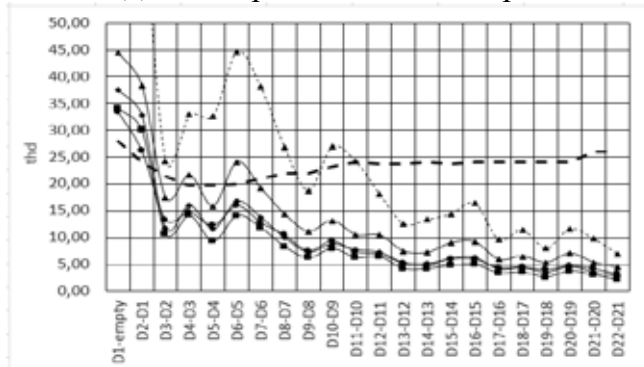
Якщо порівнювати результати для різних вимірювань, то можна зазначити, що відповідні криві насичення поведуться по-різному, залежно від точки порогу подібності. В цілому, як можна побачити на Рис. 4.18-4.20 (а)–(г), використання міри Соренсена-Дайса демонструє найменш волатильну поведінку для усіх порогів подібності термінів. Соренсен-Дайс також призводить до того, що вдосконалений алгоритм **R-TND** виявляє насичення повільніше ніж три інші міри для *Min*, *Ave-1*, та *Ave-2*. Для *Max*, він такий же швидкий, як Жаро і трохи повільніший ніж Жакар і Жаро-Вінклер.

⁴¹ <https://github.com/OntoElect/Experiments/tree/master/STG>. Назви файлів {TIME, DMKD-300, DAC-cleaned}-Results-Alltogether-{min, ave, ave2, max, 1}-th.xlsx.

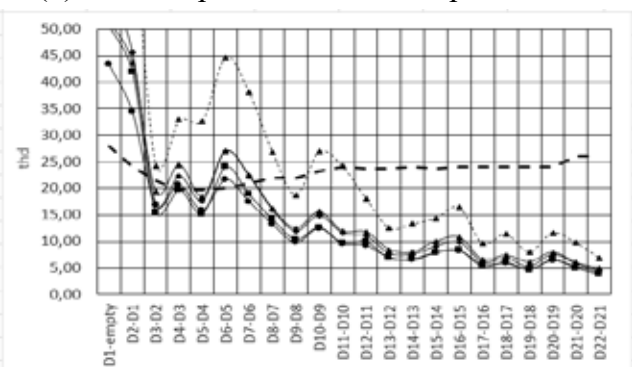


(а) *Min* пороги подібності термінів

(б) *Ave-1* пороги подібності термінів



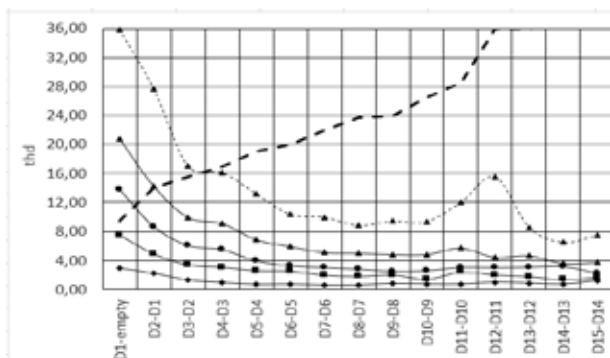
(в) *Ave-2* пороги подібності термінів



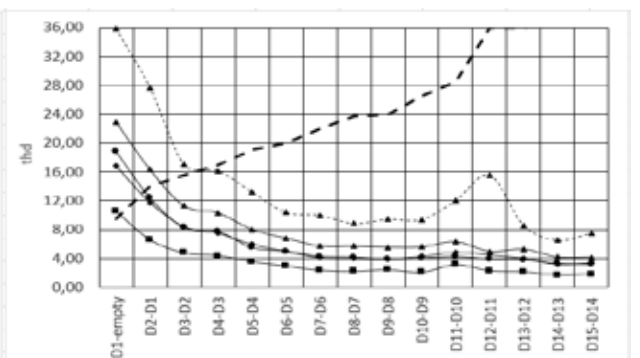
(г) *Max* пороги подібності термінів

Легенда: ▲ Sorensen-Dice ● Jaccard ◆ Jaro ■ Jaro-Winkler ▲ Baseline - - eps

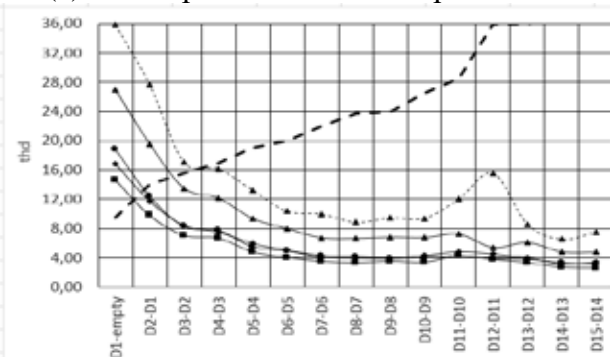
Рис. 4.18. Вимірювання термінологічного насичення на TIME



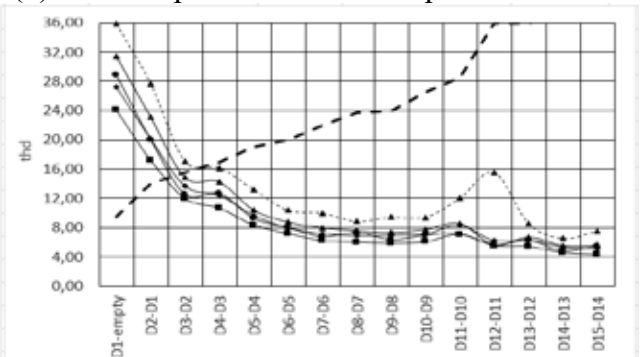
(а) *Min* пороги подібності термінів



(б) *Ave-1* пороги подібності термінів



(в) *Ave-2* пороги подібності термінів

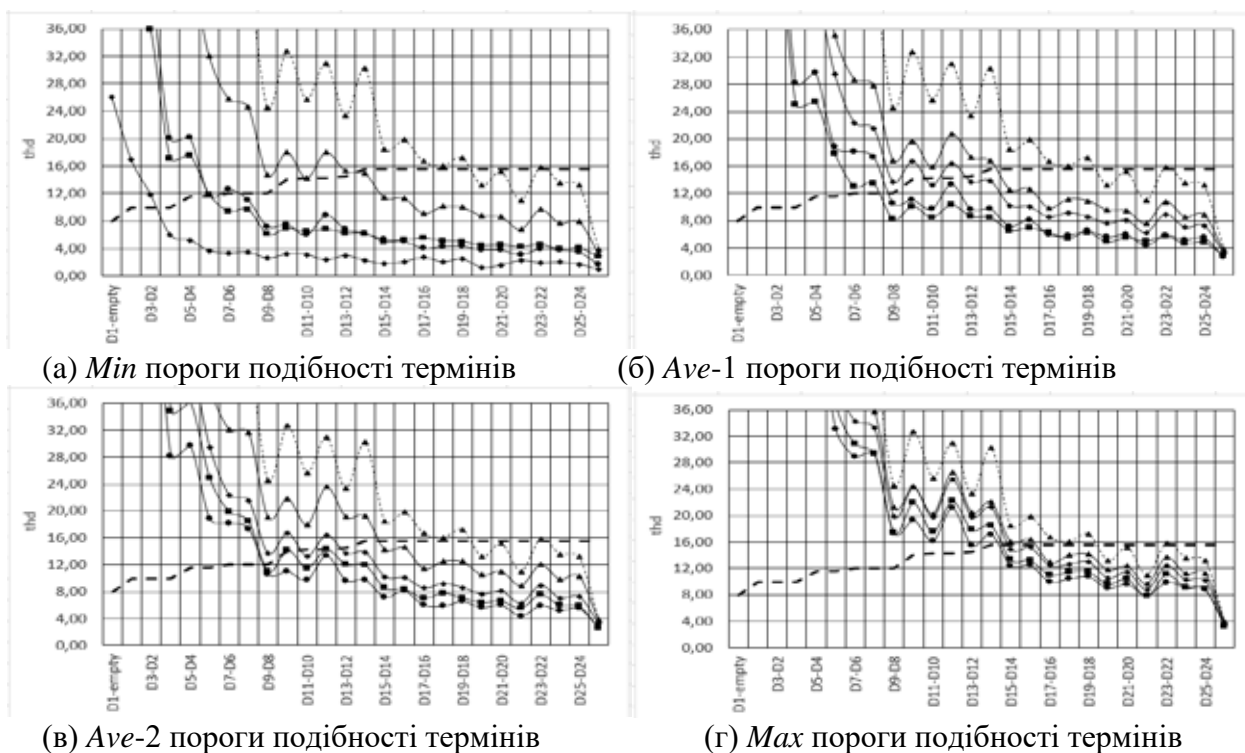


(г) *Max* пороги подібності термінів

Легенда: ▲ Sorensen-Dice ● Jaccard ◆ Jaro ■ Jaro-Winkler ▲ Baseline - - eps

Рис. 4.19. Вимірювання термінологічного насичення на DMKD

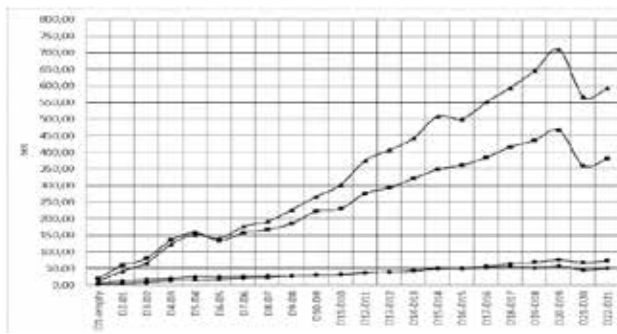
Ще одне спостереження полягає в тому, що інтегрально, всі реалізовані міри подібності термінів добре справлялися зі збереженням значущих термінів з усіх трьох колекцій документів. На це вказують суміщення піків термінологічного внеску на діаграмах (а)–(г) на Рис. 4.18-4.20. Як видно на Рис. 4.18-4.20 (г), для порогової точки *Max*, всі криві методів подібності строк досить точно відповідають формі базової кривої **THD**. Отже, вони мають піки точно в тих самих точках вимірювання *thd* де і базова крива, вказуючи на більшу кількість нових значущих термінів. Найбільш чутливим до піків внеску в термінологію був Соренсен-Дайс.



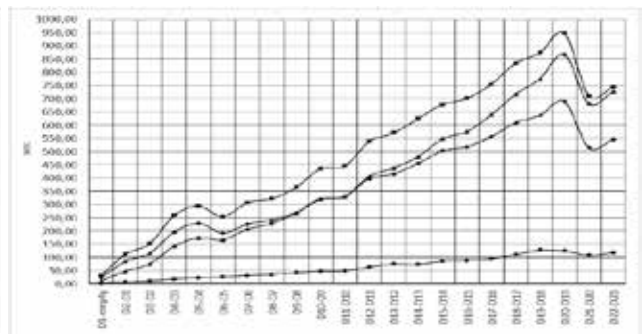
Легенда —▲— Sorensen-Dice —●— Jaccard —◆— Jaro —■— Jaro-Winkler —▲— Baseline — — eps

Рис. 4.20 Вимірювання термінологічного насичення на DAC Cleaned

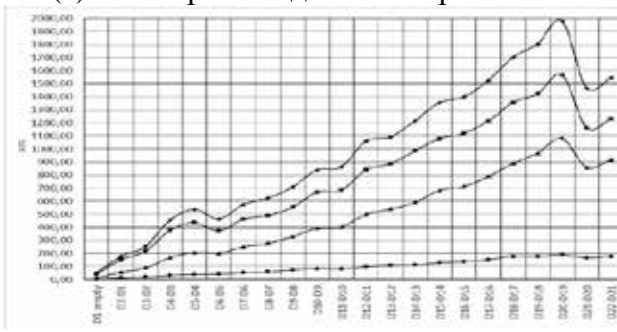
Діаграми на Рис. 4.21-4.23 показують час, витрачений алгоритмом **STG** на виявлення та групування подібних термінів для різних обраних порогів подібності термінів. Одна конкретна діаграма відповідає певному порогу подібності термінів (*Min*, *Ave-1*, *Ave-2*, або *Max*). Слід зазначити, що введення мір подібності строк, при обчисленні термінологічної різниці, суттєво збільшує обчислювальну складність. Як можна помітити на Рис. 4.21–4.23 (а) – (г), час зростає зі значеннями порогу подібності термінів (*th*) і сягає тисяч секунд для порогових значень *Max*.



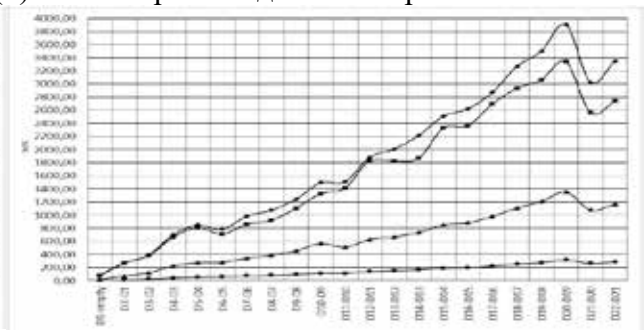
(a) *Min* пороги подібності термінів



(б) *Ave-1* пороги подібності термінів



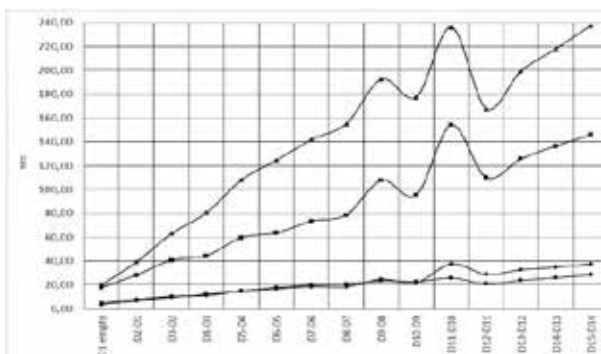
(в) *Ave-2* пороги подібності термінів



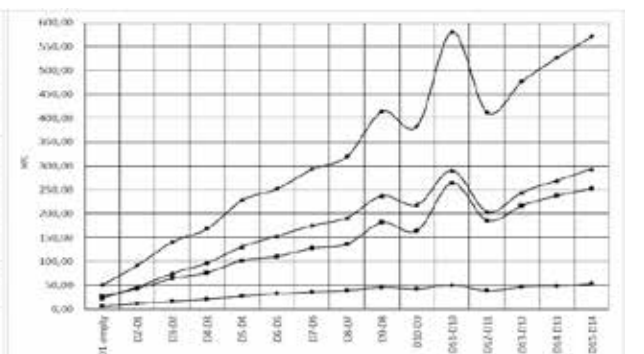
(г) *Max* пороги подібності термінів

Легенда: —▲— Sorensen-Dice —●— Jaccard —◆— Jaro —■— Jaro-Winkler

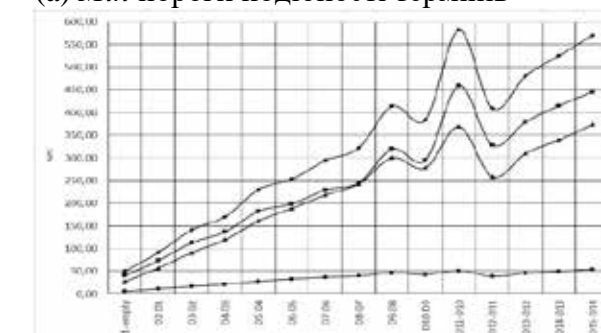
Рис. 4.21. Час (сек) витрачений алгоритмом **STG** для групування подібних термінів на наборах термінів **TIME**



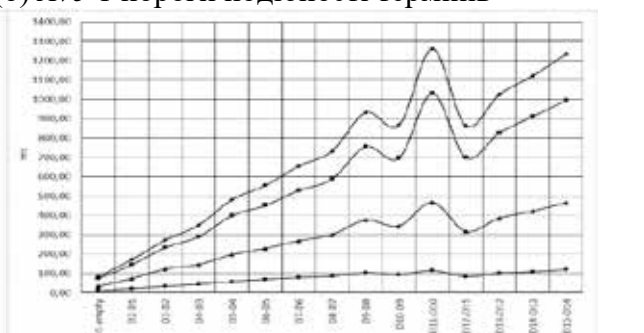
(a) *Min* пороги подібності термінів



(б) *Ave-1* пороги подібності термінів



(в) *Ave-2* пороги подібності термінів



(г) *Max* пороги подібності термінів

Легенда: —▲— Sorensen-Dice —●— Jaccard —◆— Jaro —■— Jaro-Winkler

Рис. 4.22. Час (сек) витрачений алгоритмом **STG** для групування подібних термінів на наборах термінів **DMKD**

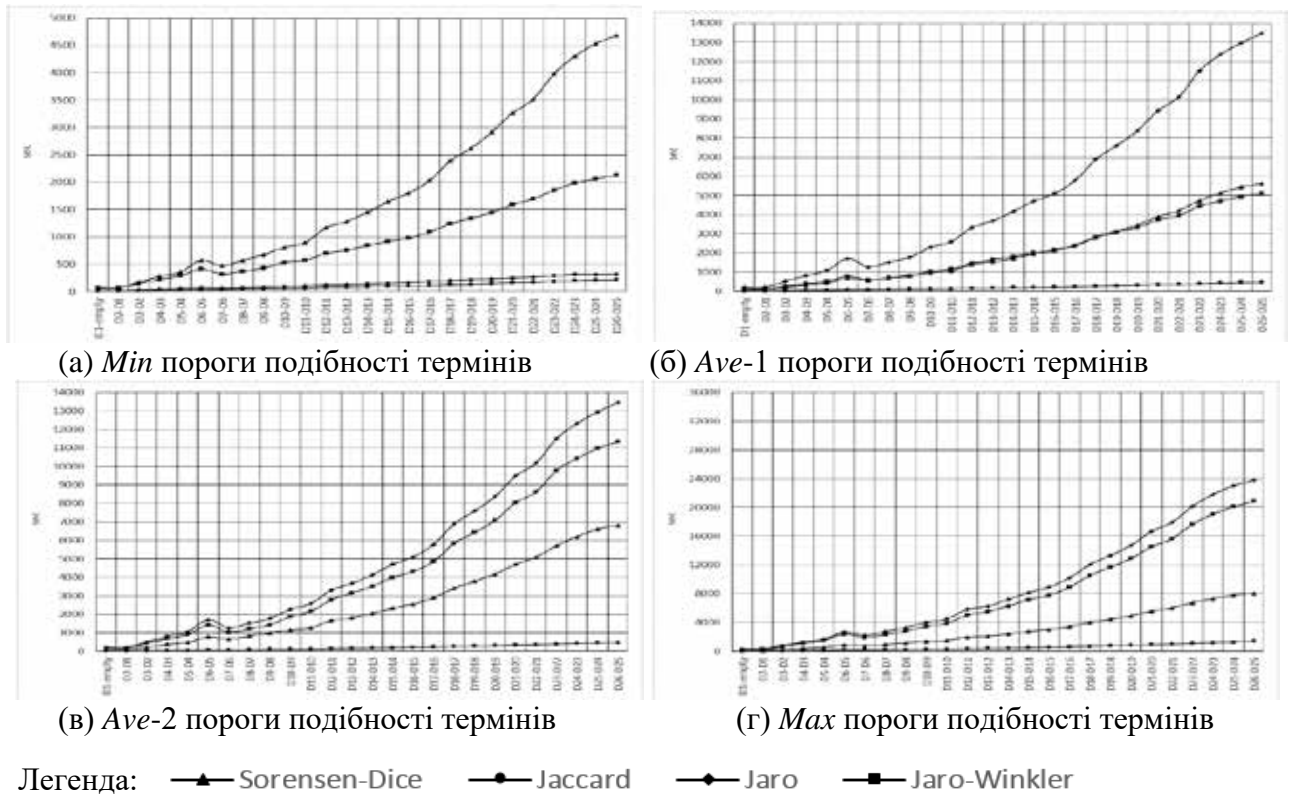


Рис. 4.23. Час (сек) витрачений алгоритмом **STG** для групування подібних термінів на наборах термінів DAC Cleaned

Варто визнати, що Соренсен-Дайс і Жакар є значно стабільнішими до збільшення *th* ніж Жаро та Жаро-Вінклер. Однак, Соренсен-Дайс витрачає в рази більше часу ніж Жакар. З іншого боку, Жакар не дуже чутливий до термінологічних піків і зберігає значно менше термінів, ніж Соренсен-Дайс.

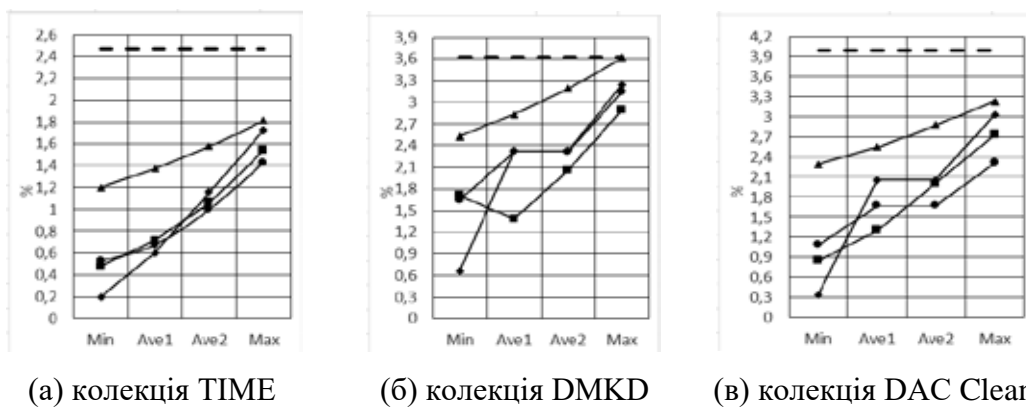


Рис. 4.24. Пропорції збережених значущих до всіх здобутих термінів для різних мір подібності термінів по кожній колекції

Рис. 4.24 зображує пропорції збережених до всіх здобутих термінів при виявленні насичення, обчислені за різними пороговими точками подібності, для

наборів термінів, здобутих з наших трьох колекцій документів. З Рис. 4.24 добре видно, що Соренсен-Дайс дає другу найбільшу пропорцію для всіх колекцій та використаних порогів подібності термінів після базового, що не групує терміни.

У останньому експерименті поріг подібності термінів був встановлений рівним 1.00, а вдосконалена реалізація **R-THD** була оцінена для всіх трьох колекцій, для пар наборів термінів, у кількох ітераціях поблизу точок насичення. Завдання полягало в тому, щоб перевірити, чи вдосконалений **R-THD** з групуванням подібних термінів: (i) виявляє термінологічне насичення в тій самій точці, що і базовий **THD** – тому, вимірюються значення *thd*; (ii) зберігає таку ж кількість значущих термінів, що і базовий **THD** – тому, вимірюється кількість збережених термінів (NRT).

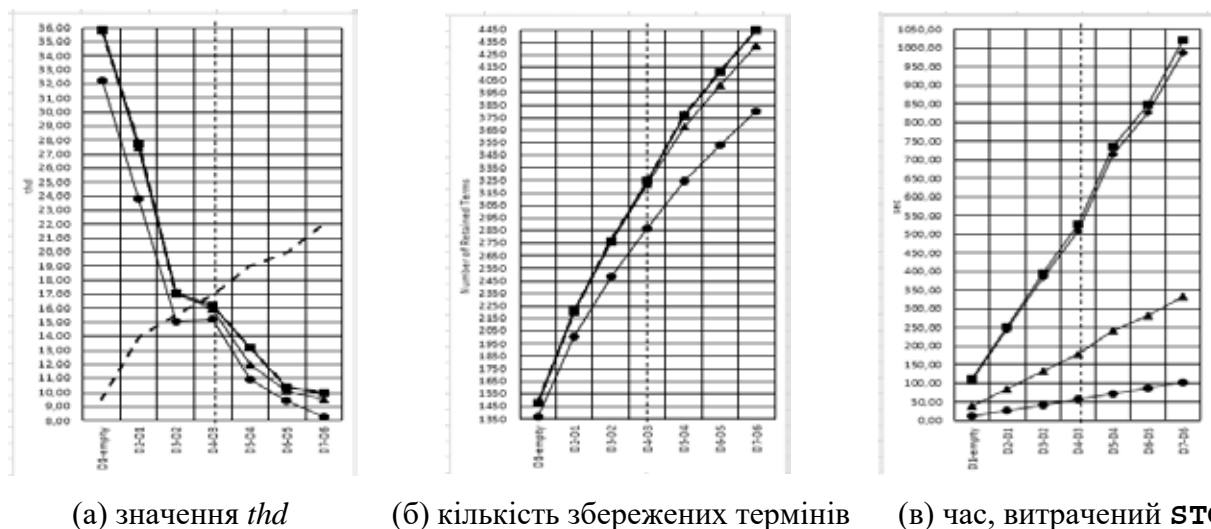
Ми також буди зацікавлені у порівнянні часу, необхідного для виконання групування термінів (**STG**).

Результати для колекції DMKD зображені на Рис. 4.25. Результати для колекцій⁴² TIME та DAC Cleaned дуже схожі на результати для DMKD і не змінюють наш висновок та рекомендацію.

На Рис. 4.25 (а) та (б) видно, що реалізації Жаро і Жаро-Вінклера повністю повторюють результати базового **THD**, як по вимірним значенням *thd* так і по кількості збережених значущих термінів. Соренсен-Дайс поводить себе подібно до Жаро і Жаро-Вінклера аж до точки насичення. Після цього він повертає трохи нижчий *thd* і зберігає трохи менше важливих термінів. Така поведінка є прийнятною, оскільки вимірювання після точки насичення не є суттєво цікавим. Однак, реалізація Жакара повертає значно нижчі значення *thd* і значно меншу кількість збережених термінів у всіх точках вимірювання – до і після виявлення насичення. Жакар також виявляє насичення на одну точку вимірювання раніше, ніж інші МСП, що не є вірним для даного порогу подібності (1.00).

⁴² Доступ до цих результатів можна отримати за посиланням <https://github.com/OntoElect/Experiments/tree/master/STG>. Імена файлів: {TIME, DMKD-300, DAC-cleaned}-Results-Alltogether-1-th.xlsx.

Рис. 4.25 (в) показує, що для того, щоб бути точними у вимірюваннях при дуже високому порозі подібності (1.00), Жаро и Жаро-Вінклер витрачають занадто багато часу на обчислення. Однак, Соренсен-Дайс і Жакар залишаються стабільнішими до зростання th , як і раніше для порогових точок $Ave1$, $Ave2$, та Max .



Легенда: \blacktriangle Sorensen-Dice \bullet Jaccard \blacklozenge Jaro \blacksquare Jaro-Winkler \blacktriangle Baseline $- -$ eps

Рис. 4.25. Оцінка реалізації вдосконаленого **R-TND** з різними МСП при $th = 1.00$ на наборах термінів DMKD. Вертикальні штрихові лінії позначають точку термінологічного насичення.

4.6.3 Загальний рейтинг та рекомендації

Експериментальні результати підсумовано у Таблиці 4.6 у вигляді рейтингу. Ми кваліфікуємо продуктивність усіх оцінених МСП та базового TND за шкалою від 1 (найкраща) до 5 (найгірша) для кожної колекції документів та для кожної порогової точки подібності термінів (Min , $Ave1$, $Ave2$, Max) в кожній колекції. Ми також розглядаємо середній рейтинг для всіх чотирьох порогових точок. Аспекти, які ми розглядаємо в цьому рейтингу, такі: (i) швидкість виявлення термінологічної насиченості, чим швидше – тим краще (Рис. 4.18–4.20); (ii) кількість збережених значущих термінів, чим більше – тим краще (Рис. 4.24); та (iii) час, витрачений методом на виконання обчислень, чим менше – тим краще (Рис. 4.21–4.23). Таблиця 4.7 містить значення показників продуктивності для різних МСП та базового TND відносно чотирьох порогових точок подібності термінів та їх середніх значень. Це

робиться для двох випадків: (i) з урахуванням критерію часу виконання (менше часу на виконання у Таблиці 4.6); та (ii) не враховуючи критерій часу виконання.

Таблиця 4.6. Рейтинг оцінюваних МСП

Критерій	Поріг подібності строк	Рейтинг (1-5)				
		Базовий ТНД	Соренсен-Дайс	Жакар	Жаро	Жаро-Вінклер
Колекція TIME						
Швидше виявлення насичення	Min	5	4	1	1	1
	Ave1	5	4	1	1	1
	Ave2	5	4	1	1	1
	Max	5	3	1	3	1
	Середнє	5	3.75	1	1.5	1
Збережено більше значущих термінів	Min	1	2	3	5	4
	Ave1	1	2	4	5	3
	Ave2	1	2	5	3	4
	Max	1	2	5	3	4
	Середнє	1	2	4.25	4	3.75
Менше часу на виконання	Min	1	5	3	2	4
	Ave1	1	4	2	3	5
	Ave2	1	3	2	5	4
	Max	1	3	2	5	4
	Середнє	1	3.75	2.25	3.75	4.25
Колекція DMKD						
Швидше виявлення насичення	Min	5	4	3	1	1
	Ave1	5	4	1	1	1
	Ave2	5	4	1	1	1
	Max	5	1	1	1	1
	Середнє	5	3.25	1.5	1	1
Збережено більше значущих термінів	Min	1	2	3	5	4
	Ave1	1	2	4	5	3
	Ave2	1	2	5	3	4
	Max	1	2	5	3	4
	Середнє	1	2	4.25	4	3.75
Менше часу на виконання	Min	1	5	2	3	4
	Ave1	1	4	2	5	3
	Ave2	1	3	2	5	4
	Max	1	3	2	5	4
	Середнє	1	3.75	2	4.5	3.75
Колекція DAC Cleaned						
Швидше виявлення насичення	Min	5	4	3	1	2
	Ave1	5	4	1	3	1
	Ave2	5	4	1	3	1
	Max	5	4	1	1	1
	Середнє	5	4	1.5	2	1.25
Збережено більше значущих термінів	Min	1	2	3	5	4
	Ave1	1	2	4	3	5
	Ave2	1	2	5	3	4
	Max	1	2	5	3	4
	Середнє	1	2	4.25	3.5	4.25
Менше часу на виконання	Min	1	5	2	3	4
	Ave1	1	4	2	5	3
	Ave2	1	3	2	5	4
	Max	1	3	2	5	4
	Середнє	1	3.75	2	4.5	3.75

Це було зроблено для аналізу цінності використання МСП у випадку якщо обчислювальні витрати не важливі. Значення розраховувались шляхом підсумовування всіх рангів для різних колекцій та критеріїв, взятих із відповідних строк порогових точок Таблиці 4.6. Ці суми були додатково відняті від найвищого значення рангу⁴³, щоб відобразити на шкалу з критерієм "чим вище – тим краще" у Таблиці 4.7. Показники продуктивності також представлені на Рис. 4.26.

Таблиця 4.7. Показники продуктивності оцінених МСП з урахуванням та без урахування часу виконання

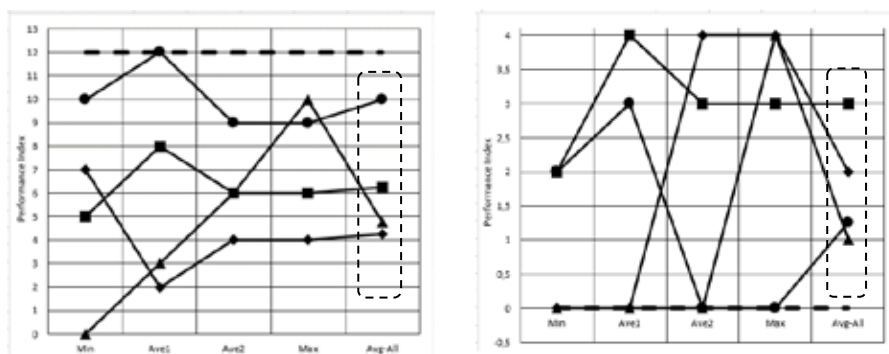
Поріг	Базовий ТНД	Соренсен-Дайс	Жакар	Жаро	Жаро-Вінклер
(а) Критерій часу виконання враховується					
Min	12	0	10	7	5
Ave1	12	3	12	2	8
Ave2	12	6	9	4	6
Max	12	10	9	4	6
Середнє (Таблиця 5.6)	12	4.75	10	4.25	6.25
(б) Критерій часу виконання не враховується					
Min	0	0	2	0	2
Ave1	0	0	3	0	4
Ave2	0	0	0	4	3
Max	0	4	0	4	3
Середнє (Таблиця 5.6)	0	1	1.25	2	3

Щодо оцінки правильності при $th = 1.00$, МСП, що поводить себе як правильно, так і обчислювально ефективно, аж до точки насичення, є Соренсен-Дайс. Жаро и Жаро-Вінклер, хоча і правильні, але витрачають багато часу на виконання при цьому високому значенні th . Жакар не є повністю правильним.

Підведемо підсумок порівняльного аналізу продуктивності всіх МСП в обох випадках, (а) та (б), представлених в Таблиці 4.7 та на Рис. 4.26. У випадку (а), коли у порівнювальному рейтингу враховується час виконання, вивід є наступним. Можливо дивно, що Жакар, яка є обчислювально найлегшою МСП (Рис. 4.21 - 4.23), демонструє найкращу загальну продуктивність. У цьому випадку він, як і

⁴³ Значення рангу – це сума усіх рангів для методу в межах певного порогу, наведеного в Таблиці 4.6. Найвище значення рангу вказує на найнижчу продуктивність. Для випадку (а) воно дорівнює 12, що є для Соренсена-Дайса на порозі *Min*. Для випадку (б) воно дорівнює 4.

раніше, не перевершує базовий **THD** тому що: (i) потребується додатковий час для **STG**; та (ii) він зберігає менше значущих термінів⁴⁴. Жакар є найкращим збалансованим за всіма критеріями оцінки, порівняно з рештою оцінених МСП. Одним з важливих недоліків Жакара є те, що він працює не зовсім коректно при $th = 1.00$. Отже, використання Жакара може бути рекомендовано у випадках з низькими порогами подібності термінів (як *Min* або *Ave1*) та жорсткими обмеженнями на час виконання. Показники продуктивності також хороші для Соренсена-Дайса і Жаро-Вінклера та обидва працюють правильно при $th = 1.00$. Ці дві МСП, здається, взаємодоповнюють одна одну у тому сенсі, що: (i) Жаро-Вінклер краще ніж Соренсен-Дайс при нижчих порогах подібності термінів, як *Min* або *Ave1*; (ii) Соренсен-Дайс перевершує Жаро-Вінклера при більш високих порогах подібності термінів, таких як *Ave2* або *Max*. Жаро у випадку (а) є явним аутсайдером; його використання не рекомендовано.



(а) з урахуванням часу виконання (б) без урахування часу виконання

Легенда: ▲ Sorensen-Dice ● Jaccard ◆ Jaro ■ Jaro-Winkler — Baseline

Рис. 4.26. Показники продуктивності оцінених МСП по пороговим значенням подібності термінів з (а) та без (б) урахування часу виконання. Точки в пунктирних прямокутниках представляють середні значення для всіх порогів.

У випадку (б), коли час виконання не враховується у порівняльному рейтингу, вивід є дещо інакшим. Як чітко видно на Рис. 4.26(б), всі МСП перевершують базовий **THD** в середньому та у пороговій точці *Max*. Жаро-Вінклер

⁴⁴ Так і повинно бути, оскільки базовий **THD** не групує терміни. Отже, будь-який альтернативний метод, який робить подібне групування термінів, зберігає менше значущих термінів.

найкраще підходить для порогів *Min* та *Ave1*, але поступається Жаро на *Ave2* та *Max*. Він також поступається Соренсену-Дайсу на *Max*. Тим не менш, Жаро-Вінклер виглядає найбільш збалансованим по продуктивності щодо всіх чотирьох порогових значень, що виділено значенням *Avg-All*. Жакар у випадку (б) є явним аутсайдером і тому не є рекомендованим для використання.

Якщо поєднати оцінки для випадків (а) та (б) можна дати наступну рекомендацію. За рахунок істотно більшого часу виконання алгоритм **R-TND** з використанням Жаро-Вінклера (на всіх порогах, крім *Max*) або з Соренсеном-Дайсом (на порозі *Max*) є рекомендованим вибором для виявлення та вимірювання термінологічного насичення. Жаро-Вінклер є кращим вибором, оскільки є найбільш збалансованим по продуктивності для всіх оцінених порогів.

4.7 Валідність та масштабованість оптимізованого конвеєру здобуття термінів

Цей розділ повідомляє про наші експерименти з оцінки оптимізованого методу вимірювання термінологічного насичення (розділ 2.6) реалізованого у декількох алгоритмах (розділ 3.6). У розділі 4.7.1 викладені особливості налаштування наших оціночних експериментів. Результати цих експериментів обговорюються в розділі 4.7.2.

4.7.1 Особливості налаштувань експериментів

У наших оціночних експериментах ми перевіряли ефективність, незалежність від домену та результативність оптимізованого конвеєру. Щодо перевірки **ефективності**, завданням нашого першого експерименту було порівняння базового та оптимізованого конвеєрів обробки на основі оцінювання:

- **Правильності.** Чи є об'єднані часткові *C-value*, обчислені за допомогою оптимізованого конвеєру практично такими ж самими, що і *C-value* обчислені за допомогою базового конвеєру?
- **Часу виконання.** Яка різниця у часі обчислень для здобуття тих самих наборів збережених значущих термінів між базовим і оптимізованим конвеєрами?

Перевірка правильності підтвердила гіпотезу *h1* (розділ 2.6) для повного доведення Теорема 2.5. Якщо *h1* є вірною, тоді оптимізований конвеєр обробки

може бути використаний для здобуття термінів для виявлення та вимірювання термінологічного насичення в колекціях документів. Порівняння часу виконання звичайного та оптимізованого конвеєрів обробки дозволило оцінити **ефективність** оптимізованого конвеєру.

Завданням нашого другого експерименту було перевірити **незалежність розробленого оптимізованого конвеєру від домену**. Для цього ми повторили експеримент з ефективністю, не вимірюючи час виконання, для іншої колекції документів, що належить до іншого домену.

Нарешті, у третьому експерименті ми перевірили, чи є **результативним застосуванням** оптимізованого конвеєру до колекції документів великого промислового розміру.

У нашому експерименті з перевірки **ефективності** була використана колекція документів DMKD (розділ 4.2.2). На додаток до наборів даних для звичайного конвеєру, описаного у розділі 4.2.2, ми згенерували часткові набори даних для оптимізованого конвеєру. Щоб забезпечити сумісність даних, для створення цих наборів даних ми обрали один і той самий інкремент (*inc*) що складає 20 статей. Отже, на основі доступних текстів, використовуючи наше програмне забезпечення Dataset Generator (розділ 3.8) ми згенерували 15 наборів даних розміру *inc*, що утворюють партицію $\{D_i = \{d_j\}_{j=1}^{20}\}_{i=1}^{15}$ ⁴⁵ колекції DMKD.

У нашому експерименті з **незалежності від домену** була використана колекція документів TIME (розділ 4.2.2). Для цього експерименту ми також створили 22 набори даних розміром *inc* = 20 (крім останнього) що утворюють партицію $\{D_i = \{d_j\}_{j=1}^{20}\}_{i=1}^{22}$ ⁴⁶ колекції TIME.

У нашому експерименті з **результативності** була використана колекція документів KM (розділ 4.2.2). Для цього експерименту ми вибрали інкремент (*inc*)

⁴⁵ Часткова колекція DMKD: <https://github.com/OntoElect/Data/blob/master/DMKD-300-DCF-Part.zip>

⁴⁶ Часткова колекція TIME: <https://github.com/OntoElect/Data/blob/master/TIME-DCF-Part.zip>

100 статей⁴⁷ для створення наборів даних. Отже, ми сформуваємо 75 наборів даних розміром inc , що утворюють партицію $\{D_i = \{d_j\}_{j=1}^{100}\}_{i=1}^{75}$ колекції КМ.

У всіх випадках для формування часткових колекцій було використано впорядкування документів за зменшенням частоти цитування (**dcf**) (розділ 4.5).

Для нашого першого та другого експериментів щодо оцінки правильності, часу виконання та оцінки незалежності від домену, обчислення було конфігуровано як два паралельних конвеєри – базовий та оптимізований, як зображено на Рис. 4.27.

Базовий конвеєр реалізовував обробку інкрементально збільшених підколекцій документів, як пояснюється у розділі 2.1. Він отримував на вхід файли колекції документів у зазначеному порядку (**dcf**) та створював інкрементально збільшені набори даних (розділ 4.2.2) за допомогою генератора наборів даних (розділ 3.8). На наступному кроці набори даних були завантажені у програмне забезпечення UPM Term Extractor (розділ 3.8) яке видавало набори збережених значущих термінів T_i та вимірювало час виконання $time_i$.

Оптимізований конвеєр реалізовував обробку часткових підколекцій документів, як описано у розділі 2.6. Він отримував на вхід файли колекції документів у тому самому порядку (**dcf**) та створював часткові набори даних за допомогою генератора наборів даних. На наступному кроці набори даних були завантажені у оптимізоване програмне забезпечення здобуття термінів, яке видавало набори збережених значущих термінів T_i та вимірювало час виконання.

На наступному кроці, здобуті набори термінів були подані на вхід модулю злиття (розділ 3.8) який застосовував алгоритм **MPCV** (розділ 3.4.2) до пар $\{T_i, T_{i+1}\}$ як зображено на Рис. 4.27. В результаті були створені злиті набори термінів $T_i^m = \cup_{k=1}^i T_k$. Також було виміряно час виконання операції злиття. Необхідний загальний час виконання ($time_i^m$) було обчислено як суми відповідного часу виконання здобуття термінів та часу їх злиття.

⁴⁷ Інкремент у 100 статей був обраний, як відповідний формі діаграм на рисунках.

Після виконання цих двох паралельних гілок, якщо $h1$ (розділ 2.6) виконується, T_i що надходить із базового конвеєру та T_i^m що надходить із оптимізованого конвеєру повинні містити дуже подібні набори термінів з приблизно однаковими C-value. Це було перевірено шляхом застосування базового алгоритму **THD** (розділ 3.5) реалізованого у модулі THD (розділ 3.8), що був застосований до: (i) пар $\{T_i, T_{i+1}\}$ та $\{T_i^m, T_{i+1}^m\}$ для порівняння кривих насичення для звичайного та оптимізованого випадків; та (ii) пар $\{T_i, T_i^m\}$ для обчислення термінологічної різниці між гіпотетично однаковими наборами термінів. Для перевірки **результативності**, до колекції КМ було застосовано оптимізований конвеєр (права частина на Рис. 4.27).

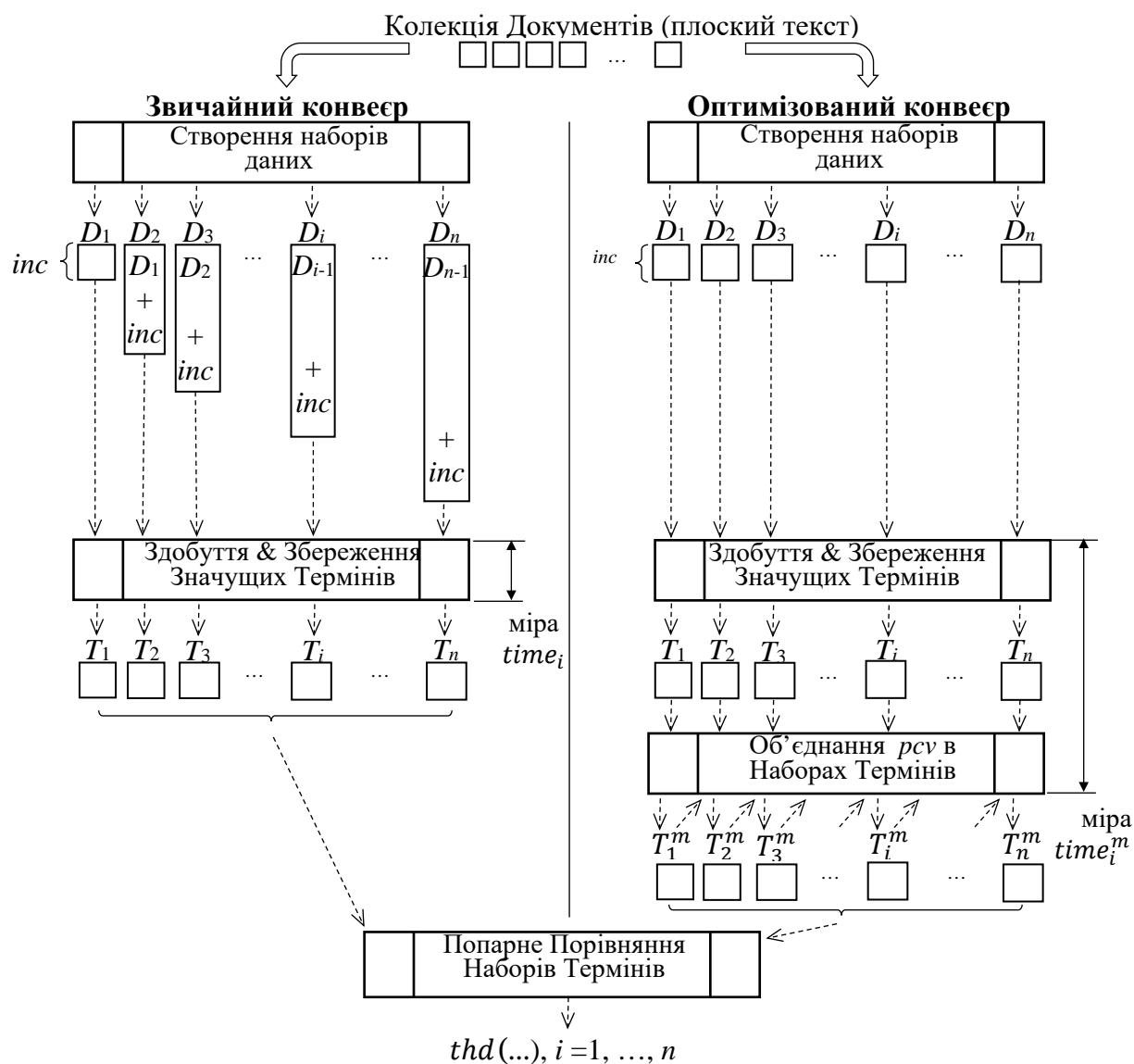


Рис. 4.27. Послідовність виконання оцінюваних експериментів

4.7.2 Результати і обговорення

Результати нашого першого експерименту (**правильність та час виконання**) представлені у табличній формі Таблиця 4.8 та графічно зображені на Рис. 4.28 та Рис. 4.29.

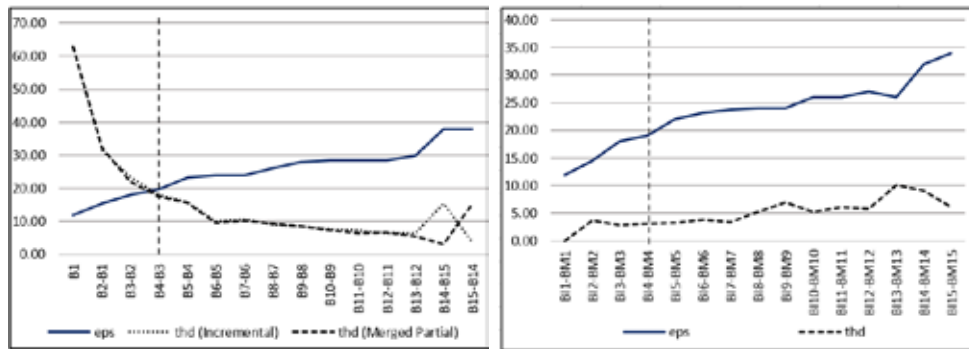
Таблиця 4.8. Вимірювання *thd* та часу виконання для звичайного і оптимізованого конвеєрів (колекція DMKD).

Набір Термінів No (<i>i</i>)	<i>eps</i>	<i>thd</i>			Час виконання (sec)	
		T_{i-1}, T_i	T_{i-1}^m, T_i^m	T_i, T_i^m	T_i	T_i^m
1	12.00	62.89	62.89	0.00	30.41	31.94
2	15.50	31.51	32.22	3.64	60.47	32.79
3	18.00	23.38	22.17	2.85	85.34	31.64
4	19.65	18.04	17.63	3.08	111.72	31.85
5	23.22	15.80	15.57	3.31	147.30	37.11
6	24.00	10.07	9.50	3.86	153.78	32.25
7	24.00	10.67	10.14	3.40	195.27	41.49
8	26.00	9.14	9.19	5.28	217.62	43.28
9	28.00	8.68	8.54	6.95	268.59	47.16
10	28.53	7.56	7.42	5.21	296.49	41.32
11	28.53	7.30	6.44	6.13	324.39	41.58
12	28.53	6.53	6.56	5.88	363.05	45.20
13	30.00	6.43	5.42	10.07	401.94	40.55
14	38.00	15.47	3.13	9.09	401.19	39.81
15	38.00	3.53	15.37	6.14	459.75	50.12

Рис. 4.28(a) показує, що результати вимірювань насичення з використанням наборів термінів отриманих з: оптимізованого конвеєру – стовпець “ T_{i-1}^m, T_i^m ” у Таблиці 4.7 та крива “Merged Partial” на Рис. 5.28(a); і базового конвеєру – стовпець “ T_{i-1}, T_i ” у Таблиці 4.7 та Інкрементальна крива на Рис. 4.28(a) – практично однакові, за винятком останніх двох вимірювань. Відхилення в хвості можна пояснити тим, що в цих двох випадках регулярний шум накопичується по-різному. Приємним побічним результатом у цьому контексті є те, що оптимізований конвеєр із використанням злитих часткових C-value накопичує менше регулярного шуму.

На Рис. 4.28(б) зображено той факт, що різниця між наборами термінів T_i і T_i^m не перевищує приблизно 1/3 індивідуального порогу значущості терміну *eps*, що використовується для відсікання незначних термінів.

У сукупності, ці два спостереження надійно доводять⁴⁸ нашу гіпотезу $h1$ (розділ 2.6), принаймні для колекції DMKD.



(а) Вимірювання насичення

(б) Термінологічні різниці

Рис. 4.28. DMKD: Злиті часткові C-value обчислені оптимізованим конвеєром практично однакові з C-value обчисленими за допомогою базового конвеєру.

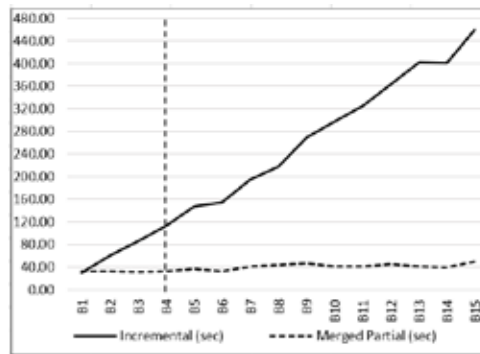


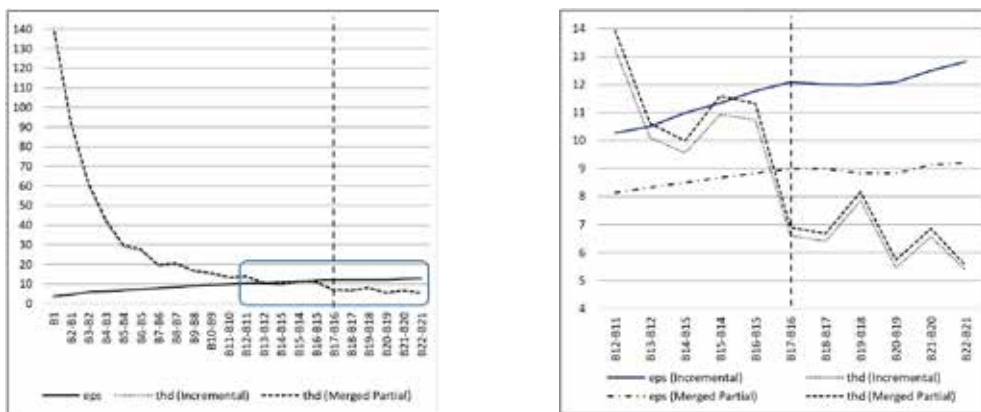
Рис. 4.29. Час виконання базового (Incremental) та оптимізованого (Merged Partial) конвеєрів.

Порівняння часу виконання, що представлено у Таблиці 4.8 та на Рис. 4.29, показує, що оптимізований конвеєр, який продемонстрував майже незмінний час виконання за ітераціями, значно перевершує за ефективністю базовий конвеєр.

Результати нашого другого експерименту (**незалежність від домену**) представлено на Рис. 4.30. Подібно до результатів DMKD, значення *thd* виміряні

⁴⁸ Можна вважати, що представлений експеримент був лише експериментом з однією колекцією документів. Отже, для іншої колекції документів результати можуть відрізнятись в залежності від дійсності $h1$. Нашим контраргументом є те, що обчислення C-value не залежить від колекції та домену. Це доведено в нашому експерименті з незалежності від домену. Результати такі ж самі для іншої колекції з іншого домену.

на колекції TIME за допомогою звичайного та оптимізованого конвеєрів, Ці результати для різних колекцій є дуже схожими між собою. Однак, індивідуальні пороги значущості термінів, обчислені за допомогою звичайного та оптимізованого конвеєрів істотно відрізняються у випадку TIME. Як зображено на Рис. 4.30(б), інкрементальні значення eps на 25 - 30 відсотків вище ніж злиті часткові.



(а) Вимірювання насичення

(б) Область насичення більш детально:

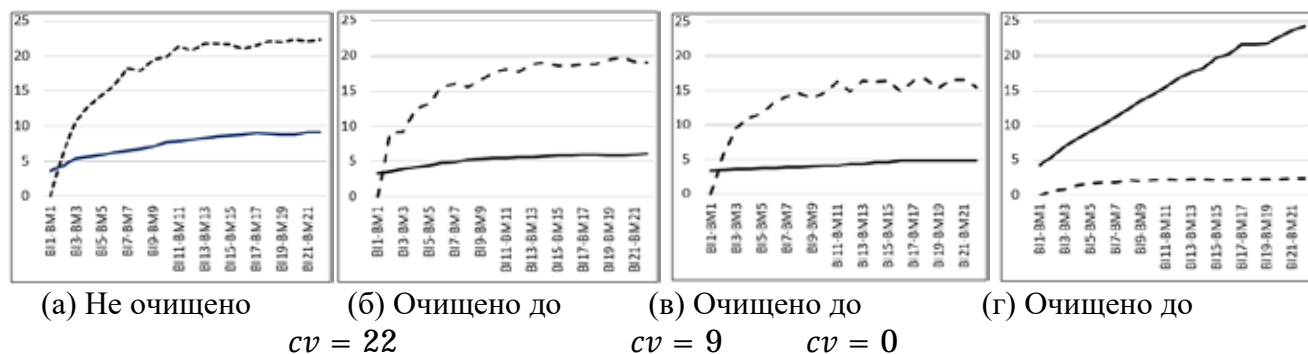
округлий прямокутник у (а)

Рис. 4.30. Криві термінологічної різниці, обчислені для наборів термінів зі звичайно обчисленими C-value (Incremental) та зі злитими частковими C-value (Merged Partial) практично однакові. Результати для колекції TIME.

Крім того, як зображено на Рис. 4.31(а), thd між T_i та T_i^m для колекції TIME виявилася помітно вищою ніж індивідуальний поріг значущості терміну для T_i^m . Це означає, що на відміну від результатів DMKD, звичайний та оптимізований конвеєри, застосовані до колекції TIME повертали різні набори значущих термінів. Аналогічно експерименту з правильністю на колекції DMKD, ми висунули гіпотезу, що ці відмінності були викликані різним обсягом накопиченого регулярного шуму.

Щоб перевірити гіпотезу про різний обсяг накопичення регулярного шуму, ми спочатку вивчили набори термінів B_{22} та B_{22}^m здобуті з наборів даних TIME на останній ітерації звичайного та оптимізованого конвеєрів відповідно. Ця ручна перевірка, зроблена автором (Ermolayev et al. 2014), який є експертом у домені

TIME, показала, що хибно позитивні елементи⁴⁹ накопичувались у верхній частині набору термінів та їх оцінки C-value були часто і в середньому вище, ніж у правдиво позитивних елементів – див. Рис. 4.32. Це свідчило про значне накопичення регулярного шуму.



Легенда: - - - $thd(T_i, T_i^m)$; — eps

Рис. 4.31. Динаміка впливу регулярного шуму на різницю між T_i та T_i^m

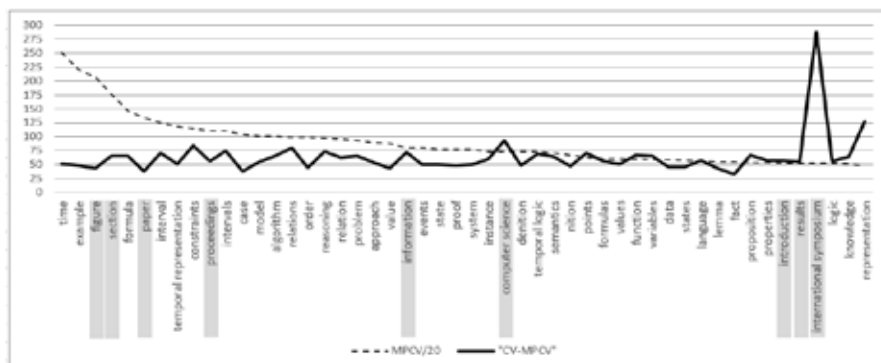


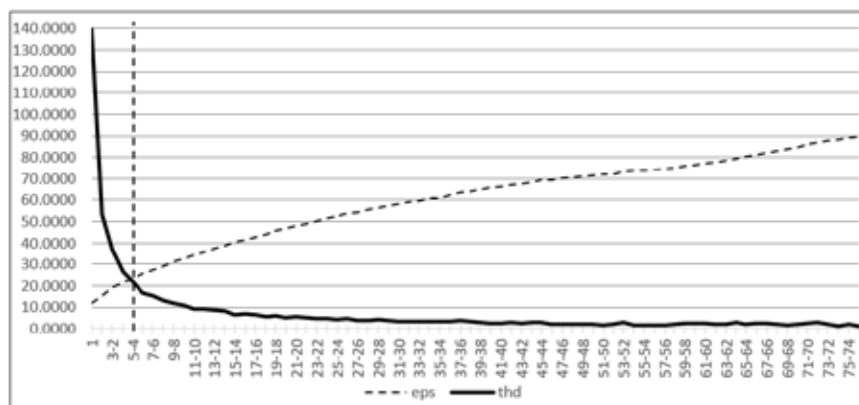
Рис. 4.32. Накопичені хибно позитивні елементи (регулярний шум, виділено сірим кольором) що спостерігаються у найбільшому наборі термінів, здобутому за допомогою оптимізованого конвеєру (B_{22}^m). Різниця ($|cv - trscv|$) для хибно позитивних елементів у середньому вищі, ніж для правдиво позитивних елементів.

Щоб пом'якшити вплив накопиченого регулярного шуму, хибно позитивні елементи було виявлено вручну у B_{22}^m , а потім видалено з усіх наборів термінів за допомогою нашого модуля видалення стоп термінів (розділ 3.8). Ця процедура проводилась у декілька ітерацій для спостереження за змінами в динаміці – до: $cv = 22$; $cv = 9$; $cv = 0$; як зображено на Рис. 4.31(б-г).

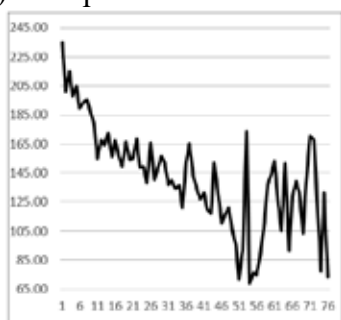
⁴⁹ Хибно позитивний елемент тут і на Рис. 4.32 є строка-кандидат у терміни з високим C-value, що не є терміном для людини експерта (стейкхолдера знань у домені).

На Рис. 4.31(г) видно, що після видалення всіх стоп-термінів (до $sv = 0$) значення термінологічної різниці між наборами збережених значущих термінів здобутих за допомогою базового (T_i) та оптимізованого (T_i^m) конвеєрів стали значно нижчими за індивідуальні пороги значущості термінів. Ситуація для очищених наборів термінів TIME на Рис. 4.31(г) навіть краща, ніж аналогічна для DMKD на Рис. 4.28(б), можливо тому, що набори термінів DMKD не були очищені.

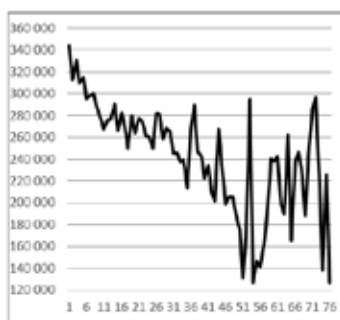
Результати третього експерименту (результативність) з вимірювання термінологічного насичення колекції КМ зображено на Рис. 4.33.



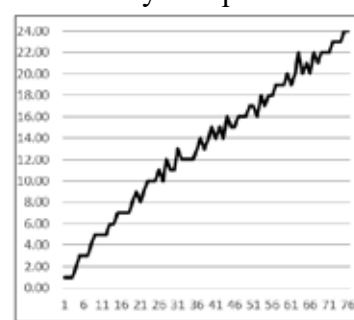
(а) Вимірювання насичення. Точка насичення позначена вертикальною пунктирною лінією.



(б) Час здобуття, sec



(в) Кіль-ть здобутих кандидатів у терміни



(г) Час злиття, sec

Рис. 4.33. Вимірювання насичення для колекції КМ

На підставі результатів експерименту з перевірки результативності можна сказати, що використання оптимізованого конвеєру повністю знімає обмеження на обсяг вхідного текстового набору даних для програмного забезпечення, для здобуття термінів, на основі методу C-value. Це твердження також справедливо для завдань, відмінних від вимірювання термінологічного насичення. Крім того, діаграма часу виконання (Рис. 4.33(б)) показує, що використання паралельної обробки робить такий спосіб здобуття термінології з тексту досить ефективним та економічним. Примітне, що така паралельна обробка дійсно можлива через те, що

частини колекції не перетинаються. Більше того, обчислювальні витрати на злиття часткових C-value (Рис. 4.33(г)) є досить малими, оскільки становлять менше одного відсотка часу виконання, витраченого на здобуття термінів.

4.8 Висновок

У цьому розділі виконано завдання дослідження **ЗДЗ** (див. розділ 1.13.3). Для цього сплановано та виконано декілька серій експериментів, проаналізовано та обговорено результати експериментів і надано рекомендації.

Завдання були поставлені у розділі 4.1 з метою перевірки питань дослідження **ПД9, 3, 1, 8, та 7** (див. Таблицю 1.5). Як частини плану ми: (i) розробили робочий процес; (ii) зібрали необхідні дані; (iii) визначили вимірювані аспекти та міри для оцінки результатів; та (iv) налаштували обчислювальне середовище для виконання експериментів. Постановку експериментів було викладено у розділі 4.2. У розділі 4.2.1 ми розробили експериментальний робочий процес як спеціалізацію обчислювального робочого процесу для виявлення та вимірювання термінологічного насичення (розділ 3.1). Далі, у розділі 4.2.2 ми представили шість колекцій документів, дві синтетичні та чотири реальні, які було використано як дані у запланованих експериментах. У розділі 4.2.3 ми запропонували та обґрунтували вимірювані аспекти та відповідні міри, що було використано для аналізу експериментальних результатів.

У розділі 4.3, ми відповіли на **ПД9** щодо правильності розробленого методу та алгоритмічного набору в граничних випадках. Для цього ми провели експерименти з використанням синтетичних колекцій, 1DOC та RAW, які були зібрані для наступних граничних випадків: швидке та стійке насичення; відсутність насичення. Результати експериментів продемонстрували, що розроблений метод та його алгоритмічна і програмна реалізація є правильними, оскільки дають очікувані результати.

У розділі 4.4, ми відповіли на **ПДЗ** щодо вибору найбільш підходящого програмного засобу АЗТ для вимірювання термінологічного насичення. В результаті було обрано програмне забезпечення UPM Term Extractor на основі

порівнювальної оцінки з NaСТeM TerMine. Таким чином, UPM Term Extractor надалі використовувався як частина базового конвеєру обчислень.

У розділі 4.5, ми відповіли на питання про вплив різних можливих впорядкувань документів на виявлення та вимірювання термінологічного насичення (ПД1). На основі крос-оцінювання п'яти різних впорядкувань щодо вимірюваних аспектів (Таблиця 4.2), ми рекомендували впорядкування за спаданням частоти цитувань (**dcf**) як найбільш збалансоване щодо різних аспектів та доменів.

У розділі 4.6, ми перевірили, чи позитивно впливає використання групування подібних термінів на термінологічне насичення (ПД8). У цій перевірці ми застосували вдосконалений конвеєр обробки із групуванням термінів та вимірюванням подібності термінів до трьох різних реальних колекцій документів у різних доменах. В експериментах ми варіювали методи вимірювання подібності термінів та відповідні пороги подібності, щоб знайти найбільш продуктивну конфігурацію з точки зору якості результату та обчислювальної ефективності. Як результат, ми рекомендували для використання дві конфігурації з найкращим рейтингом.

Наприкінці, у розділі 4.7, ми відповіли на питання щодо ефективності, домен-нейтральності та результативності оптимізованого конвеєру обчислень (ПД7), розробленого у розділах 2.6 та 3.4. Ми експериментально довели гіпотезу $h1$ Теорема 2.5, що завершило доведення цієї теореми. Ми також виявили, що оптимізований конвеєр є обчислювально ефективним, допускає паралельну обробку часткових колекцій, і є результативним за рахунок зняття обмежень на обсяг оброблюваних даних. Було перевірено, що результативність знімає обмеження, притаманне іншим відомим реалізаціям АЗТ. Отже, оптимізований конвеєр можна використовувати для обробки колекцій документів промислового розміру.

Таким чином, ми провели всі оціночні експерименти, необхідні для відповіді на наші питання дослідження ПД3, ПД9, ПД7, ПД1, та ПД8. Отже, було виконане завдання дослідження ЗД3.

5 ПРАКТИЧНЕ ВИКОРИСТАННЯ РОЗРОБЛЕНОГО МЕТОДУ

Розроблені метод та його програмна реалізація виявлення та вимірювання термінологічного насичення у колекціях документів вже використовувались у зовнішніх промислових та академічних проектах, що офіційно підтверджено документами, наведеними у Додатку Д. У цьому розділі ми повідомляємо про одне промислове (розділ 5.1) та одне академічне (розділ 5.2) застосування результатів роботи. Розділ 5.3 представляє наше бачення практичних переваг, що можуть бути очікувані тими, хто буде впроваджувати розроблений метод у промисловість, як технологію. Розділ 5.4 представляє наш прогноз щодо спектру потенційного використання розробленого методу, реалізованого у програмному забезпеченні, в галузі наукового видавництва. Розділ 5.5 містить висновок.

5.1 Перевірка технологічного прогнозу Гартнер за допомогою термінологічного аналізу

У цьому розділі ми повідомляємо про використання нашого методу для термінологічного аналізу в промисловому проекті ⁵⁰, що виконується у співпраці з компанією ТОВ ГРУПБВТ ⁵¹.

Кожен постачальник інформаційних або програмних технологій, будь то починаюча компанія, мале чи середнє підприємство, чи корпорація, уважно спостерігає та відстежує розвиток технологічних та прикладних тенденцій у своїх фокусах діяльності та інтересах. Ця діяльність необхідна для виживання їхнього бізнесу, оскільки безпосередньо впливає на перспективи покращення їх конкурентоспроможності на основних ринках.

Одним із визнаних у всьому світі джерел інформації про технологічні тенденції в цьому промисловому секторі є збірник звітів, які щорічно публікуються Гартнер⁵². Їх методологія полягає в опитуванні ключових спеціалістів у

⁵⁰ SAGOIT-IT: Strategic Analysis of R&D Gaps and Opportunities for Industrial Uptake in Trending IT Fields – проект фінансований ТОВ ГРУПБВТ.

⁵¹ <https://groupbwt.com/>

⁵² <https://www.gartner.com/en>

відповідних секторах і галузях та представленні результатів аналітичної обробки цих інтерв'ю у серії звітів. Кожен з цих звітів є орієнтованим на власну фокусну аудиторію – найвищий рівень менеджменту. Відповідними звітами Гартнер для відстеження технологічних тенденцій є (Gartner 2019; Cearley et al. 2019). На основі інформації, наведеної у цих звітах, технічний та технологічний менеджмент компанії може оцінити поточне становище компанії, наявні можливості та компетенції, а також потенціал слідувати фокусній тенденції і, таким чином, підвищувати свою конкурентоспроможність у середньостроковій перспективі.

Однак результати, якісних досліджень щодо технологічних тенденцій, що проводить Гартнер, є занадто загальними і не містять необхідних деталей для правильного позиціонування, спрямування та вдосконалення програми досліджень і розробок (R&D) на рівні компанії. Отже, маючи бачення тенденції від Гартнер і обравши пріоритетну технологію, компанія повинна більш уважно вивчити обраний контекст з технічної точки зору. Для цього на наступному етапі стратегічного аналізу потрібно більш ґрунтовно розглянути технічні деталі.

У цьому розділі ми представляємо більш низькорівневий погляд на аналіз технологічного потенціалу у контексті однієї обраної перспективної технології серед тих, що викладені Гартнер (Gartner 2019) – Генеративні Змагальні Мережі (GAN)⁵³. Наш підхід базується на термінологічному аналізі зібраного корпусу документів для домену GAN. Щоб з'ясувати, чи існує вікно можливостей у межах обраного фокусу інтересів, ми вимірюємо термінологічні відмінності між академічним, виробничим та спільними піднаборами зібраної колекції документів. Інтуїція, що стоїть за цим розділенням, складається з трьох частин:

- Термінологічні відбитки, здобуті з репрезентативного набору відповідних документів відображають сучасний стан досліджень (SotA) та технології (SotT) у галузі. Отже, якщо у документах домену фокусній галузі існує стійкий, зрілий термінологічний відбиток, ця галузь є стійкою та термінологічно зрілою.

⁵³ Детальна презентація наших результатів у цьому дослідженні доступна як технічний звіт (Ermolayev et al. 2020)

- Документи науковців, очевидно, є на передньому краї досліджень, оскільки їх прийняття промисловістю іде за академічними досягненнями із затримкою у часі. Отже, відмінності між академічними та промисловими термінологічними відбитками вказують на відставання. Крім того, якщо набір термінів, здобутий з промислових статей не є насиченим, то галузь ще не стала зрілою щодо аналізованої технології, принаймні до цього моменту.

- Аналіз термінологічного відбитку спільних статей – це хороший спосіб подивитися на те, наскільки активною є галузь у дослідженнях, пов'язаних з обраною технологією, а також наскільки вона прагне перейняти технологію з академічних кіл.

5.1.2 Обговорення результатів експериментів

Питання та методика експериментального дослідження з перевірки прогнозу Гартнер щодо технології GAN детально представлені в Додатку Е. У цьому розділі наведені та проаналізовані результати експериментів.

Як було встановлено Goodfellow et al. (2014), GAN – це підхід глибокого навчання (deep learning), при якому дві моделі навчаються паралельно та в конкурентній формі. На сьогоднішній день GAN є одним із основних підходів до глибокого навчання з кількома різними застосуваннями у галузях комп'ютерного бачення (computer vision), обробки та розпізнавання мовлення (speech processing and recognition) та обробки природніх мов (natural language processing).

Ми розпочали вибірку статей релевантних для GAN використовуючи для початку два останні огляди Gui et al. (2020) та Pan et al. (2019). На основі цього початкового набору статей, ми змогли отримати 262 документи які були розділені на: (i) академічні – 138 статей; (ii) промислові – 38 статей; та (iii) спільні – 86 статей. Для цих частин колекції ми провели вимірювання та аналіз термінологічного насичення і виявили, що колекція не є репрезентативною. Щоб покращити репрезентативність, ми додали до початкового набору статей перші десять найчастіше цитованих документів із отриманої вибірки. Цими статтями були

(Krizhevsky et al. 2017):17224.75 (пром)⁵⁴, (Simonyan and Zisserman 2014): 6216.71 (акад), (Szegedy et al. 2015): 3971.83 (спіл), (Ioffe and Szegedy 2015): 3427.66 (пром), (Russakovsky et al. 2015): 3011.00 (акад), (Long et al. 2015): 2965.00 (акад), (Salvaris et al. 2018): 2353.00 (акад), (Girshick et al. 2014): 2020.57 (акад), (Suthaharan 2016): 1913.40 (акад), та (Zeiler and Fergus 2014): 1432.71 (акад).

На основі цього розширеного початкового набору, ми змогли зібрати: (i) академічних – 319 статей; (ii) промислових – 100 статей; та (iii) спільних – 198 статей. Загалом отримана колекція містила 617 документів у згаданих трьох частинах. Ці частини були використані під час другої ітерації термінологічного аналізу, що дозволило нам відповісти на питання **П1** та **П2** нашого дослідження.

Щоб відповісти на питання **П1** про зрілість опублікованих знань про GAN, ми виміряли термінологічне насичення для всіх трьох частин колекції. Результати цих вимірювань наведені на Рис. 5.1 та Рис. 5.2.

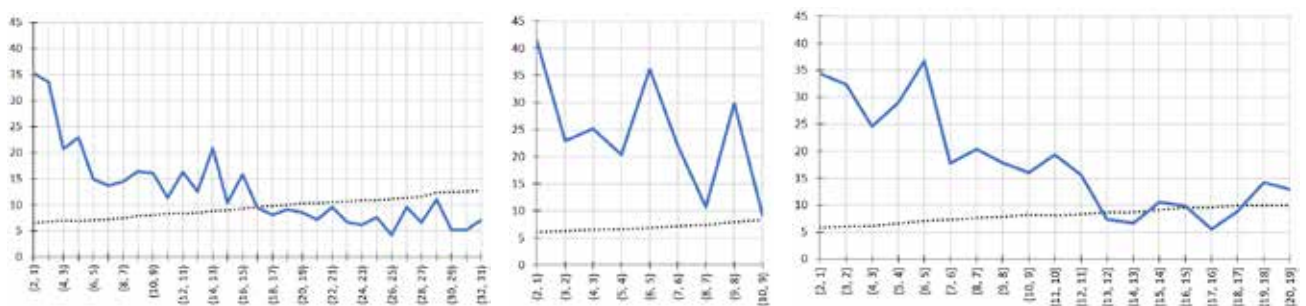
Лише одна частина з трьох – академічна – продемонструвала термінологічне насичення (Рис. 5.1(a)). Цей факт свідчить про зрілість термінологічного набору в цій частині, а отже, і про академічну спільноту в GAN. На основі отриманої нами колекції статей, дві інші частини – промислова та спільна – не продемонстрували термінологічне насичення (Рис. 5.1(б) та (в)).

Волатильність термінологічних різниць, виміряних для академічної частини (Рис. 5.2(a)) підтверджує, що виявлена насиченість є стабільною і, швидше за все, збережеться, якщо до колекції буде додано більше наукових робіт з GAN. Дійсно, діапазон (від мінімального до максимального) значень волатильності в оцінюваній зоні насичення (пара наборів даних (17, 18) і далі) становить $5.2832+5.8326 = 11.1158$, що нижче діапазону індивідуального порогу значущості термінів: з 11.3329 у (27, 26) до 12.7122 у (32, 31) з тенденцією до монотонного зростання.

Щоб відповісти на питання **П2** щодо прогалин у сукупності опублікованих знань про GAN, ми виміряли термінологічні різниці у трьох парах частин колекції:

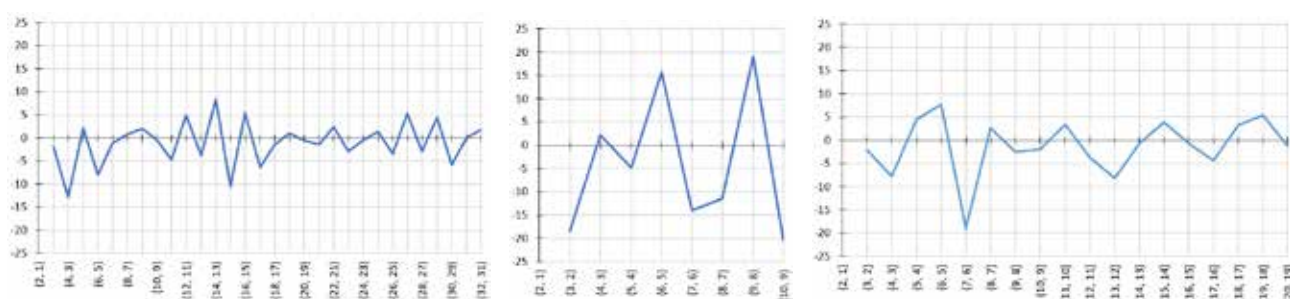
⁵⁴ Тут і нижче в списку початкових статей, число після посилання – це кількість цитат на рік (Google Scholar) за якою йде ідентифікатор частини колекції (пром/акад/спіл).

(i) академічна та промислова; (ii) академічна та спільна; і (iii) спільна та промислова. Результати цих вимірювань представлені на Рис. 5.1.



(а) академічна частина: насичена (б) промислова частина: не насичена (в) спільна частина: не насичена

Рис. 5.1. Вимірювання термінологічного насичення: *thd* (суцільна крива) по відношенню до *eps* (пунктирна крива)



(а) академічна частина (б) промислова частина (в) спільна частина

Рис. 5.2. Вимірювання волатильності термінологічних різниць

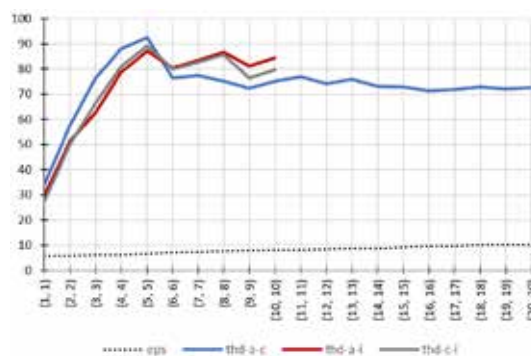


Рис. 5.3. Криві термінологічної різниці для пар частин колекції: **thd-a-c** – для академічної та спільної; **thd-a-i** – для академічної та промислової; **thd-c-i** – для спільної та промислової; **eps** – крива максимального індивідуального порогу значущості терміну.

Криві на Рис. 5.3 вказують на те, що термінологічні різниці в опублікованих роботах щодо GAN між науковцями та промисловцями, навіть враховуючи їх спільну роботу, є дуже значними. Отже, прогалини в опублікованих знаннях також

є значними. Вони вказують на те, що SotA та SotT відрізняються і для тих, хто впроваджує GAN є широкий простір для об'єднання зусиль щодо передачі технологій GAN у промисловість.

5.2 Інструментальне забезпечення виконання магістрами огляду літератури

У цьому розділі ми повідомляємо про використання нашого методу та програмного забезпечення для термінологічного аналізу в Міжнародній магістерській програмі з інформатики та науки про дані Українського католицького університету⁵⁵. Метод був використаний як частина інструментального конвеєру запропонованого студентам, щоб допомогти їм зібрати літературні джерела для оглядів стану досліджень в доменах, відповідних до їх магістерських робіт. Ця робота була запропонована як курсове завдання за дисципліною «Академічне письмо» в осінньому семестрі їх першого року навчання.

5.2.1 Завдання для студентів

На лекціях з дисципліни «Академічне письмо» студенти були поінформовані про те, що хороший огляд відповідних робіт є важливим компонентом будь-якої дослідницької роботи, щоб зрозуміти прогалини в результатах, опублікованих іншими, і, отже, отримати мотивацію щодо доробку у вибраній галузі досліджень. Це завдання курсу було спрямоване на те, щоб дати студентам можливість (і настанову) виконувати цей компонент їх магістерського проекту систематично, отже, ефективно та продуктивно.

Завдання було запропоновано після завершення першої частини курсу. Це завдання стосувалось пошуку та відбору літератури в рамках обраної теми магістерського проекту. Після відбору вибрані джерела літератури були вивчені, прочитані, розглянуті та проаналізовані з метою знайти прогалину у результатах попередніх досліджень, яку бажано усунути або зменшити. Результати повинні були бути задокументовані у технічному звіті.

Звіт оцінювався на основі аспектів, які нагадують критерії, що використовуються для рецензування оглядових статей. Цими аспектами та

⁵⁵ <https://ucu.edu.ua/en/>

відповідними шкалами оцінок були: методологія (0–5); репрезентативність (0–10); релевантність (0–10); структура (0–5); аналіз прогалін (0–10); стиль написання (0–5).

Студентам було повідомлено, що у якості попередньої умови для обґрунтування та об'єктивності їх огляду та аналізу колекція джерел літератури повинна бути: **репрезентативною** – такою, що є достатньо повною, для висвітлення всіх значних досліджень та розробок у вибраній галузі; **релевантною** – такою, що містить лише джерела, які за своєю суттю, але не за назвою, повідомляють про результати, які можуть бути корисними для або впливати на роботу, заплановану у проекті.

Що стосується репрезентативності зібраної літератури, студентам було запропоновано об'єктивно довести, що принаймні, всі основні релевантні джерела, на час виконання завдання, були включені в колекцію.

Що стосується впевненої відповіді на питання про релевантність, студент повинен був мати чітко сформоване уявлення про те, що планується зробити в проекті. Цю візію пропонувалося уточнювати у процесі ітерацій відбору релевантних джерел у відповідності до підходу активного навчання (Settles 2009). Студентам також було надано інструментальну підтримку (програмне забезпечення) для цієї ітеративної діяльності з відбору джерел літератури. Надання інструментального конвеєру мало на меті допомогти їм відповісти на питання репрезентативності та релевантності наступним чином.

Релевантність. Студентові пропонується надати невеликий “початковий” набір дослідницьких статей, проіндексованих Microsoft Research⁵⁶. Метадані цих робіт надалі подаються на вхід інструменту контрольованого відбору за методом снігової кулі (Dobrovolskyi and Keberle 2020). Інструмент генерує каталог вибраних релевантних публікацій за допомогою аналізу мережі цитувань, імовірнісного моделювання тем, та вибірки снігової кулі. Повні тексти статей, які є загальнодоступними, завантажуються для подальшого читання та аналізу. Ітерації

⁵⁶ <https://academic.microsoft.com/>

можуть бути виконані шляхом уточнення “початкового” набору дослідницьких робіт, на основі результатів попередньої ітерації.

Репрезентативність. Студентові пропонується використовувати каталог колекції та завантажені повні тексти публікацій для проведення аналізу (тобто виявлення та вимірювання) термінологічного насичення (Kosa et al. 2021) у межах зібраного набору статей. Термінологічне насичення в піднаборі статей, якщо воно спостерігається, може бути використано як доказ репрезентативності цього піднабору. Крім того, цей піднабір менше, ніж весь зібраний набір статей і вимагає менше зусиль для читання та аналізу.

Обидва інструменти включені в конвеєр обчислень, як представлено в розділі 3.1 (Рис. 3.3). Студенти не були зобов’язані використовувати запропонований інструментальний набір програм. Натомість вони могли вільно обирати будь-які інші засоби для збору репрезентативного та відповідного набору публікацій для аналізу літератури. Однак їм було повідомлено, що аргументи на підтвердження репрезентативності та релевантності все ж повинні бути наведені в технічному звіті для отримання відповідних оцінок. Їм також було запропоновано представити обрану методологію збору літератури таким чином і в деталях, які дозволили б відтворити цю роботу.

5.2.2 Результати використання

Дев’ятнадцять із двадцяти трьох студентів з групи подали звіти про курсові завдання. Серед цих дев’ятнадцяти:

- Шістнадцять студентів частково використали запропонований інструментальний конвеєр (для відбору літератури), хоча це не було обов’язковим
- Четверо студентів повністю використали запропонований інструментальний конвеєр (для відбору літературних джерел та аналізу термінологічного насичення). Усі ці четверо опинилися у чверті з найвищими оцінками після того як лектор оцінив усі звіти.

На основі цих випробувань можна сказати, що метод та програмне забезпечення для аналізу термінологічного насичення легко приймаються студентами магістерського рівня і сприймаються ними як корисні.

5.3 Практичні переваги

З практичної точки зору, корисність наявності компактного та стабільного насиченого набору термінів і відповідного термінологічного ядра підколекції бачиться у комбінації двох важливих аспектів.

Перший аспект полягає в тому, що аналіз тексту та даних повнотекстових статей у комерційних цілях часто тягне суттєві витрати на придбання права використання цих текстів⁵⁷. Очевидно, що придбання доступу до термінологічного ядра підколекції в разі ефективніше ніж до повної колекції. По-друге, метод послідовного наближення, представлений у розділі 2, дозволяє здобувати доступ частинами, до інкрементів, доки не буде зібрано підколекцію термінологічного ядра. Це дозволяє оплачувати у точності мінімально результативний набір статей.

Крім того, у порівнянні з результатами SotA (див. розділ 1), розроблений метод для здобуття термінологічного ядра підколекцій та відповідних насичених наборів термінів має такі практичні переваги:

Незалежність від домену. На відміну від кількох рішень на основі машинного навчання для АЗТ, наш метод не потребує ані мічених даних (labeled data) для тренування методу, ані повторного тренування при переході на інший домен. Отже, розроблений метод є повністю домен-нейтральним, при тому він дає результати АЗТ доволі хорошої якості. Таким чином, потенційний впроваджувач нашого методу, реалізованого у програмному забезпеченні, може безперешкодно застосовувати ці доробки на будь-яких текстових колекціях, у будь-якому домені, що його цікавить, без витрат на адаптацію.

Нейтральність до методу АЗТ. Як представлено в цій роботі, метод (розділ 2) та конвеєр програмного забезпечення (розділ 3) використовують метод C-value для АЗТ. Однак, це не обмежує технологію, прив'язуючи її до методу C-value. Метод АЗТ можна легко замінити бажаною альтернативою, яка повинна бути реалізована

⁵⁷ Наприклад, інформацію про політики використання API інтелектуального аналізу тексту і даних Springer-Nature можна отримати за посиланням: <https://www.springernature.com/gp/researchers/text-and-data-mining>

таким чином, щоб виводити результати у тій самій формі, що описана у розділі 2.1: кожен термін у здобутому наборі термінів повинен бути парою $\langle t_i, score_i \rangle$, де t_i – це символічна строка, що представляє термін та $score_i$ це – значення важливості цього терміну. Нейтральність до методу АЗТ також відкриває шлях до обробки текстів на мовах, відмінних від англійської.

Обробка повного тексту. SotA підходи до вивчення онтологій з текстів, що масштабуються на колекції промислових розмірів, обробляють лише анотації документів (див. розділ 1.3). Це пояснюється тим, що: (i) у SotA не розглядається мінімальний репрезентативний піднабір документів; та (ii) використовувані методи АЗТ погано масштабуються зі зростанням обсягу текстів, що оброблюються. Наш метод долає ці два недоліки SotA, що дозволяє йому масштабовано справлятися з обробкою повного тексту. Крім того, якість здобуття ознак з повних текстів значно краща, ніж з анотацій. Отже, наш метод забезпечує кращу якість результатів для подальшого вивчення онтологій.

Зняття обмеження на розмір колекції. Як детально описано у розділах 2.6, 3.4, та 4.7, наш оптимізований метод АЗТ не обмежений обсягом тексту, що підлягає обробці. Тому, на відміну від попередніх методів, він цілком пристосований до колекцій документів промислового розміру.

Легке розпаралелювання. На відміну від інших конвеєрів АЗТ, які вимагають, щоб весь текст (колекція) був присутній в одному вузлі обробки, наша оптимізована реалізація (розділи 2.6 та 3.4) вимагає тільки одного розділу колекції (*inc* документів) за раз. Отже, як доведено в розділі 2.6, можна використовувати стільки паралельних процесорів, скільки доступно для одночасної обробки частин колекції. Як було продемонстровано в нашому експерименті з масштабованістю (розділ 4.7), використання цього підходу істотно скорочує час виконання.

Збалансоване використання впливу та дати публікації. Використання рекомендованого порядку **dcf** (розділ. 4.2) гарантує, що зібрана підколекція містить усі тексти, що дають домінуючий акумульований вплив на формування думки більшості у спільноті спеціалістів даного домену.

5.4 Потенційні сценарії застосувань у галузі наукового видавництва

Бізнес-застосування розробленого методу для обчислення насичених термінологічних відбитків текстових колекцій у галузі наукового видавництва передбачаються у розробці програмних засобів для інструментарію, на базі розробленого методу, та підтримки: уточнення таксономії предметних дескрипторів; управління портфоліо журналів; рекомендації видань для публікацій потенційних авторів; рекомендації потенційних рецензентів, членів редакційної колегії, або членів програмного комітету

Таксономії предметних дескрипторів, як відомо, корисні, та використовуються для структурування контенту в багатьох електронних бібліотеках наукових публікацій та служб індексації, таких як ACM DL⁵⁸, Microsoft Academic⁵⁹, Scopus⁶⁰, Web of Science⁶¹. Відомо також, що ці таксономії розробляються та оновлюються вручну і, отже, є неточними⁶², неповними, не розвиваються за темою так саме швидко, як відповідний науковий контент. Таким чином, використання оновлень наукового контенту для регулярного покращення таксономії предметних дескрипторів може бути корисним у галузі цифрових бібліотек.

Управління портфоліо журналів, на рівні менеджерів програм (program management level), часто стикається з проблемою значного дублювання тематики у портфоліо різних журналів. Наприклад, журнали з дослідження операцій можна побачити у портфоліо з Математики або Економіки. Для керівників програм, які здійснюють нагляд та координацію різних портфоліо з метою покращення їх впливу та ефективності публікацій, інструмент для порівняння семантичних “відбитків” різних портфоліо може представляти значний інтерес. Для такого

⁵⁸ <https://dl.acm.org/>

⁵⁹ <https://academic.microsoft.com/>

⁶⁰ <https://www.scopus.com/>

⁶¹ <https://webofknowledge.com/>

⁶² Наприклад, тема “насичення” пов’язана з хімією, але не з комп’ютерними науками або якісними дослідженнями.

порівняння, ефективним рішенням може бути співставлення насичених наборів термінів, здобутих з різних портфоліо колекцій документів або похідних з цих наборів термінів графів знань.

Рекомендації видань для публікації надаються потенційним авторам багатьма науковими видавцями через онлайн-служби рекомендацій видань, релевантних до змісту статті. Ці служби, наприклад Springer Journal Suggester⁶³ або Elsevier JournalFinder⁶⁴, базуються на обробці природньої мови та аналізі поданого тексту і знаходженні збігів з предметними дескрипторами портфоліо журналу. Результати, отримані цими сервісами, можуть бути покращені шляхом вдосконалення предметних дескрипторів – як зазначено вище.

Рекомендації потенційних рецензентів. Ще однією цікавою практичною проблемою для редакторів або керівників програм видавництва є надійне співставлення статті, що подано на рецензування, компетенціям та інтересам потенційних рецензентів. Рецензент може бути обґрунтовано обраний на основі термінологічного порівняння змісту публікацій кандидатів із семантичним відбитком конкретного подання.

У згаданих вище прикладних контекстах, використання розробленого у цій роботі методу здобуття термінологічних ядер та відповідних насичених наборів термінів може бути економічно ефективним та надійним способом інструментування відповідної діяльності.

5.5 Висновок

У цьому розділі ми виконали завдання дослідження ЗД4 (див. розділ 1.13.3). Для цього ми представили наші відповіді на питання дослідження ПД10 (розділи 5.1 та 5.2) та ПД11 (розділи 5.3 та 5.4).

У розділі 5.1 ми повідомили про наш досвід використання розробленого програмного забезпечення в рамках промислового проекту, спрямованого на перевірку прогнозу Гартнер щодо тенденцій впровадження інформаційних

⁶³ <https://journalsuggester.springer.com/>

⁶⁴ <https://journalfinder.elsevier.com/>

технологій, у контексті технології GAN. Цей кейс використання продемонстрував, що розроблений метод і конвеєр програмного забезпечення можуть бути використані не тільки для здобуття термінологічних ядер колекцій, але і виявляти відсутність термінологічного насичення для аналізу тенденцій впровадження технологій.

У розділі 5.2 ми повідомили про використання нашого методу та програмного забезпечення у навчанні для проведення пошукових досліджень літератури магістрами з метою написання оглядів стану досліджень для їх магістерських робіт. Цей кейс використання продемонстрував, що розроблений метод виявився зручним та результативним у цих академічних умовах. Він також був добре прийнятий та оцінений студентами як корисний.

У розділі 5.3 ми узагальнили наш досвід щодо кейсів використання, виклавши практичні переваги впровадження розробленого методу та програмного забезпечення для промисловості. У розділі 5.4, ми представили наш погляд на потенційні промислові-сценарії у одній з фокусних галузей для цієї роботи – галузі наукового видавництва.

Досвід використання результатів роботи, аналіз переваг доробку у порівнянні з SotA, викладення візії можливих сценаріїв промислового застосування допомогли нам чітко зрозуміти, які способи застосування та промислові галузі є фокусними для потенційних майбутніх впроваджень наукового доробку цього дослідження в промисловість та академічну сферу. Таким чином, завдання дослідження ЗД4 було виконано.

6 ВИСНОВКИ

Проблема забезпечення повноти при здобутті, з колекцій професійних текстів, потреб до подання знань в онтологіях доменів є складною і такою, що до сих пір не мала об'єктивного та строгого вирішення. У цій галузі, потреби, здобути з текстів, що описують домен, є інтерпретаціями думок спільноти стейкхолдерів знань у домені. Репрезентативність у цьому контексті – це ступінь врахування всіх цих інтерпретацій для формування вимог щодо побудови обґрунтованої та повної описової теорії домену.

У дисертації вирішено задачу виявлення та вимірювання ступеню репрезентативності текстових колекцій для здобуття знань, що є суттєвим внеском задля забезпечення вищезазначеної повноти здобутих потреб до подання знань. Для цього було вперше розроблено комплексний обчислювальний метод для виявлення та вимірювання термінологічного насичення у інкрементально зростаючій послідовності підколекцій наявної колекції текстових документів. Розроблений метод дозволяє автоматично здобувати репрезентативний набір термінів мінімального розміру та сформувати термінологічне ядро підколекції документів мінімального розміру, що містить усі ці терміни. Для побудови методу було введено поняття термінологічного насичення як процесу послідовного наближення, що веде до екстракції термінологічного ядра підколекції для довільного домену.

На основі систематичного огляду відповідних робіт було виявлено прогалини в сучасному стані досліджень, які було потрібно усунути. На основі аналізу цих прогалин у дослідженнях було запропоновано підхід до здобуття насиченої термінології. З метою втілення цього підходу в комплексний метод, ми сформулювали питання нашого дослідження в розділі 1. Ці питання дослідження були методологічно згруповані в чотири наступні завдання дослідження: **(ЗД1)** розробка формального фреймворку (розділ 2); **(ЗД2)** розробка набору алгоритмів (розділ 3); **(ЗД3)** експериментальна оцінка та перевірка (розділ 4); та **(ЗД4)** практичне використання (розділ 5). Усі завдання дослідження було повністю виконано, як викладено у відповідних розділах роботи та висновках до них.

У розділі 6.1, ми підсумовуємо ці висновки про результати дисертаційної роботи, показуючи, що розроблений комплексний метод для виявлення та вимірювання термінологічного насичення в колекціях текстових документів дійсно є ефективним та результативним, як того вимагала мета дослідження. Ми підсумовуємо проблеми, які досі залишаються невирішеними, і представляємо плани майбутньої роботи у розділі 6.2.

6.1 Підсумки науково-технічного доробку

Відповідно до мети та завдань дослідження, науково-технічний доробок дисертації направлений на підвищення репрезентативності, ефективності та результативності автоматизованого здобуття термінологічних наборів з колекцій професійних документів. Нижче підсумовано, яке підвищення було досягнуто завдяки результатам.

В дисертації було розроблено формальний фреймворк для цього комплексного методу, що включає формалізацію процесу термінологічного насичення, як процесу послідовних наближень, метрику термінологічної різниці між наборами значущих збережених термінів, достатні умови існування термінологічного насичення, оптимізований метод на базі використання партицій колекції документів.

Розроблений формальний фреймворк було матеріалізовано в наборі відповідних алгоритмів. Алгоритми було реалізовано у модулях програмного забезпечення, що були агреговані в обчислювальний конвеєр для виявлення та вимірювання термінологічного насичення.

Реалізований обчислювальний конвеєр було систематично перевірено та валідовано у декількох серіях обчислювальних експериментів. На основі результатів експериментів було рекомендовано найбільш збалансовану конфігурацію обчислень в конвеєрі. Експерименти також продемонстрували, що розроблений комплексний метод є коректним, ефективним та результативним для обробки колекцій професійних документів (публікацій) з будь-якого домену.

Ефективність розробленого комплексного методу полягає в тому, що:

(i) **Компактність.** У випадку існування термінологічного насичення, метод здобуває суттєво більш компактні підколекції термінологічного ядра, ніж вихідна колекція документів. Насичені набори збережених значущих термінів є також суттєво більш компактними, ніж набори термінів, що здобувалися з вихідної колекції документів. Експерименти довели, що цей аспект ефективності спостерігається незалежно від домену колекції. Так, для реальних колекцій, на яких проводилися експерименти були отримані суттєві зменшення обсягів (розділ 4), що підсумовано у Таблиці 6.1. У роботі також доведено експериментально (розділ 4), що використання групування термінів дає ще більш компактні набори збережених значущих термінів незалежно від домену. Але, цей здобуток отримується за рахунок значного (у рази) підвищення часу обчислень.

Таблиця 6.1. Підсумок зменшення обсягів колекцій документів та здобутих термінів

Колекція	Кількість документів		Кількість здобутих термінів		
	Повна колекція	Термінологічне ядро	Повна колекція	Термінологічне ядро	З групуванням
DMKD	300	80 (26.7%)	313 506	3 399 (1.08%)	2 135 (0.68%)
TIME	437	180 (41.2%)	315 474	5 493 (1.74%)	3 456 (1.10%)
DAC Cleaned	506	220 (43.5%)	518 765	16 703 (3.22%)	7 496 (1.44%)
КМ	7 500	500 (6.6%)	4 035 760	13 579 (0.34%)	---

(ii) **Час виконання.** Розроблений оптимізований конвеєр виявлення та вимірювання термінологічного насичення, за рахунок використання партицій колекцій та префіксних дерев для пошуку співпадаючих строк у статистичній частині реалізації методу C-value, витрачає суттєво менший час на обчислення, ніж базовий конвеєр. Як показано у розділі 4.7.2 стосовно експерименту з колекцією DMKD, оптимізований конвеєр, який продемонстрував майже незмінний час виконання за ітераціями (від 32 до 51 сек), значно перевершує за ефективністю базовий конвеєр, що витрачав на обробку тих самих даних від 30 сек. на 1ї ітерації до 460 сек на 15й ітерації.

Результативність розробленого комплексного методу полягає в тому, що:

(i) **Репрезентативність.** У випадку існування термінологічного насичення, метод здобуває підколекції термінологічного ядра, що є гарантовано репрезентативними.

(ii) **Масштабованість.** Завдяки розробці оптимізованого обчислювального методу та конвеєру, метод не є обмеженим на кількість документів в наявній колекції. Завдяки практично незмінного часу обробки однієї часткової колекції та очевидної можливості їх паралельної обробки, оптимізований конвеєр є добре масштабованим щодо зростання обсягів колекцій до реальних промислових значень, як продемонстровано в експерименті з колекцією КМ в розділі 4.7.2.

Практичну застосованість та значущість розробленого комплексного методу та реалізованого обчислювального конвеєру доведено у розділі 5. Практичне застосування доробку дисертації та його добре сприйняття користувачами доведено впровадженнями у промисловість та академічну практику на рівні магістерської програми. Досвід практичного використання проаналізовано та узагальнено шляхом наведення потенційних переваг щодо впровадження розробленого методу. Також, наведено потенційно релевантні бізнес сценарії для подальшого впровадження доробку дисертаційного дослідження у галузь наукового видавництва. Більш детальну інформацію про результати роботи в публікаціях та використанні наведено в Додатку Ж.

6.2 Напрямки подальшої роботи

Подальші дослідження в контексті представленої дисертаційної роботи можливі і плануються за наступними двома напрямками:

- Подальший розвиток методу, розробка інформаційної технології
- Розробка сценаріїв використання та впровадження інформаційної технології

Щодо подальшого розвитку методу та розробки інформаційної технології для виявлення, вимірювання та аналізу термінологічного насичення, можна окреслити кілька відкритих питань R&D, які потребують подальшої роботи за межами цього PhD проекту.

Прогнозування термінологічного насичення. Як визначено у розділі 2.7, доведення Теорема 2.3 про достатні умови існування термінологічного насичення не дозволяє використовувати ці умови для прогнозування термінологічного насичення на основі декількох початкових ітерацій вимірювань. Вдосконалення розробленого формального фреймворку у цьому напрямку планується шляхом аналізу статистичних властивостей розподілу термінів у корпусах документів, обмежених доменом.

Крос-мовні колекції документів. Одним із поточних обмежень нашого методу та його програмної реалізації є те, що вони застосовується лише до документів, написаних англійською мовою. Однак, бажано розширити сферу застосування розробленого рішення, охопивши інші мови, крім англійської, та колекції, що містять документи, написані кількома мовами. Оскільки розроблений метод є нейтральним, щодо мови (мов) документів колекції, за винятком його частини для здобуття термінів (див. розділ 3.8), рішення може бути поширене на інші мови шляхом додавання модулів здобуття термінів з текстів для цих мов або крос-мовних екстракторів термінів до програмного забезпечення. Це розширення також планується на майбутню роботу.

Розробка платформи-як-сервісу (Platform-as-a-Service). В даний час реалізоване інструментальне програмне забезпечення є загальнодоступним у вигляді дослідницького прототипу. У прозвітованих кейсах використання (розділи 5.1 та 5.2) прототип був розгорнутий спеціально для потреб цих конкретних кейсів. Наступним кроком є розробка та розгортання інструментальної програмної платформи, у хмарі, яка може бути використана тим, хто потребує аналізу термінологічної насиченості у своїх дослідженнях чи бізнесі. Для цієї роботи шукається партнер у галузі промислових програмних технологій. Розробка буде здійснюватися в майбутньому та у співпраці з таким партнером.

Діяльність з трансферу технології плануватиметься на базі виявлення потенційних сценаріїв бізнес-застосувань, що представлені у розділі 5.4.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- (Ahad et al. 2016) Ahad, A., Fayaz, M., Shah, A.S.: Navigation through citation network based on content similarity using cosine similarity algorithm. *International Journal of Database Theory and Applications* 9(5), 9–20 (2016). doi: 0.14257/ijdta.2016.9.5.02
- (Ahmad et al. 1999) Ahmad, K., Gillam, L., Tostevin, L.: University of surrey participation in trec8: weirdness indexing for logical document extrapolation and retrieval (wilder). In: 8th Text Retrieval Conf, TREC-8 (1999)
- (Aho and Corasick 1975) Aho, A. V., Corasick, M. J.: Efficient string matching: an aid to bibliographic search. *Communications of the ACM* 18(6), 333–340 (1975)
- (Aldiabat and Le Navenec 2018) Aldiabat, K.M., Le Navenec C.-L.: Data saturation: The mysterious step in grounded theory method. *The Qualitative Report* 23(1), 245–261 (2018)
- (ANSIS 1986) American National Standard for Information Systems – Coded Character Sets – 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII), ANSI X3.4-1986". American National Standards Institute (ANSI). 1986-03-26
- (Arnold and Ohlebusch 2011) Arnold, M., Ohlebusch, E.: Linear Time Algorithms for Generalizations of the Longest Common Substring Problem. *Algorithmica* 60(4), 806–818 (2011)
- (Astrakhantsev 2015) Astrakhantsev, N.: Methods and software for terminology extraction from domain-specific text collection. PhD thesis. Institute for System Programming of Russian Academy of Sciences (2015)
- (Astrakhantsev 2016) Astrakhantsev, N.: ATR4S: toolkit with state-of-the-art automatic terms recognition methods in scala. arXiv preprint arXiv:1611.07804 (2016)
- (Astrakhantsev 2018) Astrakhantsev, N.: ATR4S: toolkit with State-of-the-Art automatic terms recognition methods in Scala. *Language Resources & Evaluation* 52, 853–872 (2018). doi: 10.1007/s10579-017-9409-4

- (Badenes-Olmedo et al. 2017) Badenes-Olmedo, C., Redondo-García, J.L., Corcho, O.: Efficient clustering from distributions over topics. Proceedings of the knowledge capture conference K-CAP 2017. Austin, TX, USA: ACM 17:1–17:8 (2017). doi: 10.1145/3148011.3148019
- (Baeza-Yates and Ribeiro-Neto 1999) Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier: Modern Information Retrieval. New York, NY: ACM Press, Addison-Wesley, Seiten 75 ff. ISBN 0-201-39829-X (1999)
- (Bordea et al. 2013) Bordea, G., Buitelaar, P., Polajnar, T.: Domain-independent term extraction through domain modelling. In: 10th Int Conf on Terminology and Artificial Intelligence, TIA 2013 (2013)
- (Brinkmann et al. 2011) Philippe H, Brinkmann H, Lavrov DV, Littlewood DT, Manuel M, Wörheide G, Baurain D: "Resolving difficult phylogenetic questions: why more sequences are not enough". PLoS Biology. 9 (3): e1000602 (March 2011). doi:10.1371/journal.pbio.1000602
- (Buitelaar et al. 2005) Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: Methods, evaluation, and applications. Frontiers in artificial intelligence and applications, vol. 123, IOS Press (2005)
- (Cearley et al. 2019) Cearley, D., Jones, N., Smith, D., Burke, B., Chandrasekaran, A., CK Lu: Top 10 Strategic Technology Trends for 2020. Gartner Inc. ID: G00432920, 21 Oct. 2019 <https://www.gartner.com/en/doc/432920-top-10-strategic-technology-trends-for-2020>
- (Chernyak and Berenstein 2006) Chernyak, L., Berenstein, A.: Method and apparatus for informational processing based on creation of term-proximity graphs and their embeddings into informational units. US patent application publication, No US 2006/0031219 A1, Feb. 9 (2006)
- (Chowdhury and Farrell 2019) Chowdhury, F. M., Farrell, R: An efficient approach for super and nested term indexing and retrieval. arXiv preprint arXiv:1905.09761v1 [cs.DS] (2019)

- (Chugunenko et al. 2018) Chugunenko, A., Kosa, V., Popov, R., Chaves-Fraga, D., Ermolayev, V.: Refining terminological saturation using string similarity measures. CEUR-WS vol. 2105 3–18 (2018) ISSN: 1613-0073
- (Church and Hanks 1990) Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. *Computational linguistics* 16(1), 22–29 (1990)
- (Church and Gale 1995) Church, K. W., Gale, W. A.: Inverse document frequency (idf): a measure of deviations from Poisson. In: *ACL 3rd Workshop on Very Large Corpora*, 121–130. Association for Computational Linguistics, Stroudsburg, PA, USA (1995). doi: 10.1007/978-94-017-2390-9_18
- (Corcho et al. 2015) Corcho, O., Gonzalez, R., Badenes-Olmedo, C., Dong, F.: Repository of indexed ROs. Deliverable No. 5.4. Dr Inventor project. (2015)
- (Corpas Pastor and Seghiri Domínguez 2010) Copras Pastor, G., Seghiri Domínguez, M.: Size matters: A quantitative approach to corpus representativeness. In R. Rabadán, M. Fernández López, T. Guzmán González (Eds.), *Lengua, traducción, recepción en honor de Julio César Santoyo*. León, Spain: Universidad de León, Secretariado Área de Publicaciones y Medios Audiovisuales 111–145 (2010)
- (Daille 1996) Daille, B.: Study and implementation of combined techniques for automatic extraction of terminology. In: Klavans, J., Resnik, P. (eds.) *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, 49–66. The MIT Press. Cambridge, Massachusetts (1996)
- (De La Briandais 1959) De La Briandais, R.: File searching using variable length keys. In: *Western Joint Computer Conference, IRE-AIEE-ACM'59 (Western)*, 295–298. ACM (1959)
- (Dice 1945) Dice, L. R.: Measures of the amount of ecologic association between species. *Ecology* 26(3), 297–302 (1945)
- (Dobrovolskyi and Keberle 2018) Dobrovolskyi, H., Keberle, N.: Collecting seminal scientific abstracts with topic modelling, snowball sampling and citation analysis. *Proceedings of the 14th international conference on ICT in education, research, and*

- industrial applications ICTERI 2018. Volume I: Main conference. Kyiv, Ukraine: CEUR-WS vol. 2105, 179–192 (2018)
- (Dobrovolskyi and Keberle 2020) Dobrovolskyi, H., Keberle, N.: Obtaining the minimal terminologically saturated document set with controlled snowball sampling. CEUR Workshop Proceedings, 2020, 2740, pp. 87-101 (2020) <http://ceur-ws.org/Vol-2740/20200087.pdf>
- (Doerre et al, 2002) Doerre, J., Gerstl, P., Goeser, S., Mueller, A., Seiffert, R.: Taxonomy generation for document collections. US patent, No US 6 446 061 B1, Sep. 3 (2002)
- (Dunning 1993) Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational linguistics 19(1), 61–74 (1993)
- (Ermolayev et al. 2013) Ermolayev, V., Akerkar, R., Terziyan, V., Cochez, M.: Toward Evolving Knowledge Ecosystems for Big Data Understanding. In: Akerkar, R. (ed.) Big Data Computing, pp. 3--56, Taylor & Francis 2013 (2013)
- (Ermolayev et al. 2014) Ermolayev, V., Batsakis, S., Keberle, N., Tatarintseva, O., Antoniou, G.: Ontologies of time: Review and trends. International Journal of Computer Science and Applications 11(3) 57–115 (2014)
- (Ermolayev 2018) Ermolayev, V.: OntoElecting requirements for domain ontologies. The case of time domain. EMISA Int J of Conceptual Modeling 13(Sp. Issue), 86–109 (2018). doi: 10.18417/emisa.si.hcm.9
- (Ermolayev et al. 2020) Ermolayev, V., Dobrovolskyi, H., Kosa, V., Yuschenko, E.: Terminological Analysis of the Gaps between Academic and Industrial R&D in Generative Adversarial Networks. Technical Report BWT-SAGOIT-IT-TR-2020-1, Dec. 2020, GroupBWT, Ukraine, 39 p. (2020)
- (Evans and Lefferts 1995) Evans, D. A., Lefferts, R. G.: Clarit-trec experiments. Information processing & management 31(3), 385–395 (1995). doi: 10.1016/0306-4573(94)00054-7
- (Fahmi et al. 2007) Fahmi, I., Bouma, G., van der Plas, L.: Improving statistical method using known terms for automatic term extraction. In: Computational Linguistics in the Netherlands, CLIN 17 (2007)

- (Ferrari et al. 2014) Ferrari, A., dell'Orletta, F., Spagnolo, G.O., Gnesi, S.: Measuring and improving the completeness of natural language requirements. In C. Salinesi, I. van de Weerd (Eds.) Requirements engineering: Foundation for software quality REFSQ 2014. Cham, Germany: Springer-Verlag LNCS vol. 8396, 23–38 (2014). doi: 10.1007/978-3-319-05843-6_3
- (Frantzi and Ananiadou 1999) Frantzi, K.T., Ananiadou, S.: The C-value/Nc-value domain independent method for multi-word term extraction. *Journal of Natural Language Processing* 6(3), 145–179 (1999). doi: 10.5715/jnlp.6.3_145
- (Gartner 2019) 5 Trends Appear on the Gartner Hype Cycle for Emerging Technologies, 2019. Gartner Inc. <https://www.gartner.com/smarterwithgartner/5-trends-appear-on-the-gartner-hype-cycle-for-emerging-technologies-2019/>
- (Girshick et al. 2014) Girshick, R., Donahue, J., Darrell. T., Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: Proc. 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, pp. 580–587 (2014). doi: 10.1109/CVPR.2014.81
- (Glänzel and Schubert 1988) Glänzel, W., Schubert, A.: Characteristic scores and scales in assessing citation impact. *Journal of Information Science* 14(2), 123–127 (1988). doi: 10.1177/016555158801400208
- (Glaser and Strauss 1967) Glaser, B.G., Strauss, A.: The discovery of grounded theory: Strategies for qualitative research. Chicago, IL, USA: Aldine (1967)
- (Gomaa and Fahmy 2013) Gomaa, W. H., Fahmy. A. A.: A Survey of Text Similarity Approaches. *Int J Comp Appl* 68(13), 13–18 (2013)
- (Gómez-Pérez et al. 2004) Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological engineering. London, UK: Springer-Verlag (2004). doi: 10.1007/b97353
- (Goodfellow et al. 2014) Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D, Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks. In: Proc. International Conference on Neural Information Processing Systems (NIPS 2014), 2672–2680 (2014)

- (Guarino et al. 2009) Guarino, N., Oberle, D., Staab, S.: What is an ontology? In S. Staab, R. Studer (Eds.) Handbook on ontologies. Berlin, Heidelberg, Germany: International handbooks on information systems, Springer-Verlag, 1–17 (2009). doi: 10.1007/978-3-540-92673-3_0
- (Gui et al. 2020) Gui, J. Sun, Z., Wen, Y., Tao, D., Ye, J.: A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. arXiv preprint, arXiv:2001.06937v1 [cs.LG] (2020)
- (Hamming 1959) Hamming, R. W.: Error detecting and error correcting codes. Bell System Technical Journal 29(2), 147–160 (1950)
- (Han et al. 2014) Han, H., Xu, S., Zhu, L.: Mining technical topic networks from Chinese patents. Proceedings of the 1st international workshop on patent mining and its applications IPAMIN 2014. Hildesheim, Germany: CEUR-WS vol. 1292 (2014)
- (Huang 2008) Huang, A.: Similarity Measures for Text Document Clustering. In: Proc. 6th New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand, 49–56 (2008)
- (Ioffe and Szegedy 2015) Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, arXiv e-prints (2015)
- (Jaccard 1912) Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist 11, 37–50 (1912)
- (Jaro 1989) Jaro, M. A.: Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. J Amer Stat Assoc 84(406), 414–420 (1989)
- (Karp and Rabin 1987) Karp, R. M., Rabin, M. O.: Efficient randomized pattern-matching algorithms. IBM Journal of Research and Development 31(2), 249–260 (1987)
- (Kim et al. 2003) Kim, J.-D., Ohta, T., Teteisi, Y., Tsujii, J.: GENIA corpus - a semantically annotated corpus for bio-textmining. Bioinformatics. 19(suppl. 1), i180-i182 (2003). doi: 10.1093/bioinformatics/btg1023

- (Knuth 1998) Knuth, D. E.: The Art of Computer Programming, Volume 3: (2nd Ed.) Sorting and Searching. AddisonWesley Longman Publishing Co., Inc., Redwood City, CA, USA (1998)
- (Korkontzelos et al. 2008) Korkontzelos, I., Klapaftis, I. P., Manandhar, S.: Reviewing and evaluating automatic term recognition techniques. In: Nordström B., Ranta A. (eds.) Advances in Natural Language Processing. GoTAL 2008. LNCS, vol. 5221, 248–259. Springer, Berlin, Heidelberg (2008). doi: 10.1007/978-3-540-85287-2_24
- (Kosa et al. 2017a) Kosa, V., Chugunenko, A., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., Birukou, A.: Semantic saturation in retrospective text document collections. CEUR-WS, vol. 1851, 1–8 (2017) ISSN: 1613-0073
- (Kosa et al. 2017b) Kosa, V., Chaves Fraga, D., Naumenko, D., Yuschenko, E., Badenes, C., Ermolayev, V., and Birukou, A.: Cross-evaluation of automated term extraction tools. Technical report TS-RTDC-TR-2017-1, 30.09.2017, Dept of Computer Science, Zaporizhzhia National University, Ukraine, 60 p. (2017) online: <http://ermolayev.com/TS-RTDS-TR-2017-1.pdf>. doi: 10.13140/RG.2.2.31187.07207
- (Kosa et al. 2018a) Kosa, V., Chaves-Fraga, D., Naumenko, D., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., Birukou, A.: Cross-evaluation of automated term extraction tools by measuring terminological saturation. Revised selected papers of ICTERI 2017. Cham, Germany: Springer-Verlag, CCIS vol. 826, 135–163 (2018) doi: 10.1007/978-3-319-76168-8_7, ISSN: 1865-0929
- (Kosa et al. 2018b) Kosa, V., Chaves-Fraga, D., Naumenko, D., Yuschenko, E., Moiseenko, S., Dobrovolskyi, H., Vasileyko, A., Badenes-Olmedo, C., Ermolayev, V., Corcho, O., Birukou, A.: The influence of the order of adding documents to datasets on terminological saturation. Technical report TS-RTDC-TR-2018-2-v2, 21.11.2018, Dept. of Computer Science, Zaporizhzhia National University, Ukraine (2018). doi: 10.13140/RG.2.2.28382.54086.
- (Kosa et al. 2019a) Kosa, V., Chaves-Fraga, D., Keberle, N., Birukou, A.: Similar terms grouping yields faster terminological saturation. Revised selected papers of ICTERI

2018. Cham, Germany: Springer-Verlag, CCIS vol. 1007, 43–70. (2019) doi: 10.1007/978-3-030-13929-2_3, ISSN: 1865-0929
- (Kosa et al. 2019b) Chaves-Fraga, D., Dobrovolskyi, H., Fedorenko, E., Ermolayev, V.: Optimizing automated term extraction for terminological saturation measurement. CEUR-WS, vol. 2387, 1–16 (2019) ISSN: 1613-0073
- (Kosa et al. 2020) Kosa, V., Chaves-Fraga, D., Dobrovolskiy, H., Ermolayev, V.: Optimized term extraction method based on computing merged partial C-values. Revised selected papers of ICTERI 2019. Cham, Germany: Springer-Verlag, CCIS vol. 1175, 24–49. (2020) doi: 10.1007/978-3-030-39459-2_2, ISSN: 1865-0929
- (Kosa and Ermolayev 2020) Kosa, V., Ermolayev, V.: Toward a theoretical framework of terminological saturation for ontology learning from texts. CEUR-WS vol. 2566, 40–51 (2020) ISSN: 1613-0073
- (Kosa et al. 2021) Kosa, V., Chaves-Fraga, D., Dobrovolskyi, H., Badenes-Olmedo, C., Corcho, O., Birukou, A. (2021) The choice of document ordering for extracting compact and representative terminological cores from domain-bounded paper collections. SN Computer Science J. Topical Issue on Methods and Models in ICT for Research and Applications – to appear
- (Kozakov et al. 2004) Kozakov, L., Park, Y., Fin, T., Drissi, Y., Doganata, Y., Cofino, T.: Glossary extraction and utilization in the information search and delivery system for IBM Technical Support. IBM System Journal 43(3), 546–563 (2004). doi: 10.1147/sj.433.0546
- (Krizhevsky et al. 2017) Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Comm. ACM 60(6), 84–90 (2017) doi: 10.1145/3065386
- (Lecy and Beatty 2012) Lecy, J.D., Beatty, K.E.: Representative literature reviews using constrained snowball sampling and citation network analysis. SSRN(2012). doi: 10.2139/ssrn.1992601
- (Lee et al. 2009) Lee, H., Ng, R. T., Shim, K.: Power-law based estimation of set similarity join size. Proc. of the VLDB Endowment 2(1), 658–669 (2009)

- (Levenshtein 1966) Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10 (8), 707–710 (1966)
- (Long et al. 2015) Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, 3431–3440 (2015) doi: 10.1109/CVPR.2015.7298965
- (Lossio-Ventura et al. 2013) Lossio-Ventura, J.A., Jonquet, C., Roche, M., Teisseire, M.: Combining C-value and keyword extraction methods for biomedical terms extraction. In: *Int Symposium on Languages in Biology and Medicine*, 45–49 (2013)
- (Lu and Browne, 2012) Lu, C. J., Browne, A. C.: Development of sub-term mapping tools (stmt). In: *AMIA 2012 Annual Symposium* (2012)
- (Lu et al. 2013) Lu, J., Lin, C., Wang, W., Li, C., Wang, H.: String similarity measures and joins with synonyms. In: *Proc. 2013 ACM SIGMOD Int Conf on the Management of Data*, 373–384 (2013)
- (Maedche and Staab 2001) Maedche, A., Staab, S.: Ontology learning for the Semantic Web. *IEEE Intelligent Systems* 16(2), 72–79 (2001). doi: 10.1109/5254.920602.
- (Manning and Schutze 1999) Manning, C. D., Schutze, H.: *Foundations of Statistical Natural Language Processing*. Bradford Book & MIT Press, Cambridge, MA, London, U.K. (1999)
- (Maynard et al. 2017) Maynard, D., Bontcheva, K., Augenstein, I.: *Natural Language Processing for the Semantic Web*. Morgan & Claypool (2017). doi: 10.2200/S00741ED1V01Y201611WBE015.
- (Medelyan and Witten 2006) Medelyan, O., Witten, I. H.: Thesaurus based automatic keyphrase indexing. In: Marchionini, G., Nelson, M. L., Marshall, C. C. (eds.) *ACM/IEEE Joint Conf on Digital Libraries*, 296–297 (2006). doi: 10.1145/1141753.1141819
- (Medlar et al. 2016) Medlar, A., Ilves, K., Wang, P., Buntine, W., Głowacka, D.: PULP: A system for exploratory search of scientific literature. *Proceedings of the 39th international ACM SIGIR conference on research and development in information*

- retrieval SIGIR'16. New York, NY, USA: ACM 1133–1136 (2016). doi: 10.1145/2911451.2911455.
- (Miller et al. 1990) Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D., Miller, K.: WordNet: An online lexical database. *Int. J. Lexicograph.* 3(4), 235–244 (1990)
- (Monger and Elkan 1996) Monger, A., Elkan, C.: The field-matching problem: algorithm and applications. In: *Proc. 2nd Int Conf on Knowledge Discovery and Data Mining*, 267–270, AAAI Press (1996)
- (Moore and Cloud 2007) Ramon E. Moore, Michael J. Cloud, in *Computational Functional Analysis (Second Edition)*, Woodhead Publishing Limited, 212 p. ISBN 978-1-904275-24-4 (2007)
- (Morse 2004) Morse, J. M.: Theoretical saturation. In M. S. Lewis-Beck, A. Bryman, T. F. Liao (eds.), *The Sage encyclopedia of social science research methods* (p. 1123). Thousand Oaks, CA: Sage. (2004). Retrieved from <http://sk.sagepub.com/reference/download/socialscience/n1011.pdf>
- (Nokel and Loukachevitch 2013) Nokel, M., Loukachevitch, N.: An experimental study of term extraction for real information-retrieval thesauri. In: *10th Int Conf on Terminology and Artificial Intelligence*, 69–76 (2013)
- (Oliver and V`azquez) Oliver, A., V`azquez, M.: TBXTools: a Free, Fast and Flexible Tool for Automatic Terminology Extraction. In: Angelova, G/, Bontcheva, K., Mitkov, R. (eds.): *Proc. Recent Advances in Natural Language Processing*, pp. 473-479, Hissar, Bulgaria, Sep. 7-9 (2015)
- (Osborne and Motta 2015) Osborne, F., Motta, E.: Klink-2: Integrating multiple Web sources to generate semantic topic networks. In M. Arenas et al. (eds.) *The Semantic Web ISWC 2015*. Springer-Verlag, Cham, Germany: LNCS vol. 9366, 408–424 (2015). doi: 10.1007/978-3-319-25007-6_24.
- (Osenton 2004) Osenton, T.: *The Death of Demand: Finding Growth in a Saturated Global Economy*. Financial Times Prentice Hall books. Upper Saddle River, New Jersey: FT Press. ISBN 9780131423312 (2004)

- (Pan et al. 2019) Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan and Y. Zheng: Recent Progress on Generative Adversarial Networks (GANs): A Survey," in *IEEE Access*, vol. 7, 36322–36333 (2019) doi: 10.1109/ACCESS.2019.2905015
- (Park et al. 2002) Park, Y., Byrd, R. J., Boguraev, B.: Automatic glossary extraction: beyond terminology identification. In: *19th Int Conf on Computational linguistics*, 1–7 (2002). doi: 10.3115/1072228.1072370
- (Peñas et al. 2001) Peñas, A., Verdejo, F., Gonzalo, J.: Corpus-based terminology extraction applied to information access. In: *Corpus Linguistics*, 458–465 (2001)
- (Pinto et al. 2004) Pinto, H.S., Tempich, C., Staab, S., Sure, Y.: DILIGENT: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In R.L. de Mántaras, L. Saitta (eds.) *Proceedings of the 16th European conference on artificial intelligence ECAI'04*. Valencia, Spain: IOS Press 393–397 (2004). doi: 10.5555/3000001.3000084.
- (Pohl and Mottelson 2019) Pohl, H., Mottelson, A.: How we guide, write, and cite at Chi. In *Extended abstracts of the 2019 CHI conference on human factors in computing systems CHI EA'19*, Glasgow, Scotland, UK. New York, NY, USA: ACM Paper No.: alt01 1–11 (2019). doi: 10.1145/3290607.3310429.
- (Qin et al. 2011) Qin, J., Wang, W., Lu, Y., Xiao, C., Lin, X.: Efficient exact edit similarity query processing with the asymmetric signature scheme. In: *Proc. of the 2011 ACM SIGMOD Int Conf on Management of data*, 1033–1044. ACM New York, USA (2011)
- (Riazanov and Voronkov 2001) Riazanov, A., Voronkov, A.: Adaptive saturation-based reasoning. In D. Bjørner, M. Broy, A.V. Zamulin (eds.) *Perspectives of System Informatics PSI 2001*. Springer-Verlag, Berlin, Heidelberg, Germany: LNCS vol. 2244, 55–61 (2001). doi: 10.1007/3-540-45575-2_11
- (Russakovsky et al. 2015) Russakovsky, O., Deng, J., Su, H. et al.: ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis* 115, 211–252 (2015). doi: 10.1007/s11263-015-0816-y

- (Salvaris et al. 2018) Salvaris M., Dean D., Tok W.H.: Generative Adversarial Networks. In: Deep Learning with Azure. Apress, Berkeley, CA. (2018). doi: 10.1007/978-1-4842-3679-6_8
- (Sarasua et al. 2015) Sarasua, C., Simperl, E., Noy, N., Bernstein, A., Leimeister, J. M.: Crowdsourcing and the semantic web: A research manifesto. Human Computation 2(1), 3–17 (2015). doi: 10.15346/hc.v2i1.2.
- (Salatino et al. 2019) Salatino A.A., Thanapalasingam T., Mannocci A., Osborne F., Motta E.: The Computer Science Ontology: A large-scale taxonomy of research areas. In D. Vrandečić et al. (eds.) The Semantic Web – ISWC 2018. Springer-Verlag, Cham, Germany: LNCS vol. 11137, 187–205. (2018). doi: 10.1007/978-3-030-00668-6_12
- (Savov et al. 2020) Savov, P., Jatowt, A., Nielek, R.: Identifying breakthrough scientific papers. Information Processing and Management 57 (2), 102–168 (2020). doi: 10.1016/j.ipm.2019.102168
- (Schmidhuber 2015) Schmidhuber, Jü.: Deep Learning. Scholarpedia. 10 (11), 32832 (2015). doi:10.4249/scholarpedia.32832
- (Schneider and Costas 2017) Schneider, J. W., Costas, R.: Identifying potential ‘breakthrough’ publications using refined citation analyses: Three related explorative approaches. Journal of the Association for Information Science and Technology 68(3), 709–723 (2017). doi: 10.1002/asi.23695
- (Schreiber et al. 1999) Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N.R., Van de Velde, W., Wielinga, B.J.: Knowledge engineering and management: The CommonKADS methodology. Cambridge, MA, USA: The MIT Press (1999)
- (Sclano and Velardi 2007) Sclano, F., Velardi, P.: TermExtractor: a Web application to learn the common terminology of interest groups and research communities. In: 9th Conf on Terminology and Artificial Intelligence, TIA 2007 (2007)

- (Settles 2009) Burr Settles: Active Learning Literature Survey. Computer Sciences Technical Report 1648 (2009). Available at: <https://minds.wisconsin.edu/bitstream/handle/1793/60660/TR1648.pdf>
- (Simonyan and Zisserman 2014) Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Proc. 2014 IEEE Conf. on Computer Vision and Pattern Recognition (2014)
- (Singhal 2001) Singhal, A.: Modern Information Retrieval: A Brief Overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4), 35–43 (2001)
- (Sørensen 1948) Sørensen, T.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Kongelige Danske Videnskabernes Selskab 5 (4), 1–34 (1948)
- (Studer et al. 1998) Studer, R., Benjamins, R., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25(1–2), 161–198 (1998). doi: 10.1016/S0169-023X(97)00056-6.
- (Suárez-Figueroa et al. 2012) Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A.: Ontology engineering in a networked world. Heidelberg, Germany: Springer-Verlag (2012). doi: 10.1007/978-3-642-24794-1
- (Sure et al. 2003) Sure, Y., Staab, S., Studer, R.: On-To-Knowledge methodology. In S. Staab, R. Studer (Eds.) Handbook on ontologies. Berlin, Heidelberg, Germany: International handbooks on information systems, Springer-Verlag 117–132 (2003). doi: 10.1007/978-3-540-24750-0_6.
- (Suthaharan 2016) Suthaharan, S.: Support Vector Machine. In: Machine Learning Models and Algorithms for Big Data Classification. Integrated Series in Information Systems, vol 36. Springer, Boston, MA. (2016) doi: 10.1007/978-1-4899-7641-3_9
- (Szegedy et al. 2015) Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In Proc:

- 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 1–9 (2015) doi: 10.1109/CVPR.2015.7298594
- (Tatarintseva et al. 2013) Tatarintseva, O., Ermolayev, V., Keller, B., Matzke, W.-E.: Quantifying ontology fitness in OntoElect using saturation- and vote-based metrics. In: Ermolayev, V., et al. (eds.) Revised Selected Papers of ICTERI 2013. CCIS, vol. 412, 136–162. Springer, Heidelberg (2013). doi: 10.1007/978-3-319-03998-5_8
- (Tsuruoka et al. 2007) Tsuruoka, Y., McNaught, J., Tsujii, J., Ananiadou, S.: Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics* 23(20), 2768–2774 (2007)
- (Tuarob et al. 2012) Suppawong Tuarob, Prasenjit Mitra and C. Lee Giles Taxonomy-based Query-dependent Schemes for Profile Similarity Measurement. *JIWES, SIGIR 2012 Workshop*, Portland, Oregon, USA (2012)
- (Varela et al. 2018) Varela, A.R., Pratt, M., Harris, J., Lecy, J., Salvo, D., Brownson, R.C., Hallal, P.C.: Mapping the historical development of physical activity and health research: A structured literature review and citation network analysis. *Preventive medicine* 111, 466–472 (2018). doi: 10.1016/j.ypmed.2017.10.020
- (Waddington et al. 2012) Waddington, H., White, H., Snilstveit, B., Hombrados, J.G., Vojtkova, M., Davies, P., et al.: How to do a good systematic review of effects in international development: A tool kit. *Journal of development effectiveness* 4(3), 359–387 (2012). doi: 10.1080/19439342.2012.711765
- (Wermter and Hahn 2005) Wermter, J., Hahn, U.: Finding new terminology in very large corpora. In: Clark, P., Schreiber, G. (eds.) *3rd Int Conf on Knowledge Capture*, 137–144, ACM (2005). doi: 10.1145/1088622.1088648
- (Winkler 1990) Winkler, W. E.: String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In: *Proc. Section on Survey Research Methods*. ASA, 354--359 (1990)
- (Wong et al. 2012) Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: A look back and into the future. *ACM Computing Surveys* 44(4) 20:1–20:36 (2012). doi: 10.1145/2333112.2333115

- (Yu et al. 2016) Yu, M., Li, G., Deng, D., Feng, J.: String similarity search and join: a survey. *Front. Comput. Sci.* 10(3), 399–417 (2016)
- (Zeiler and Fergus 2014) Zeiler, M.D., Fergus R.: Visualizing and Understanding Convolutional Networks. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds.) *Computer Vision – ECCV 2014*. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham. (2014). doi: 10.1007/978-3-319-10590-1_53
- (Zhang et al 2008) Zhang, Z., Iria, J., Brewster, C., Ciravegna, F.: A comparative evaluation of term recognition algorithms. In: *6th Int Conf on Language Resources and Evaluation*, 2108–2113 (2008)
- (Zhang et al 2016) Zhang, Z., Gao, J., Ciravegna, F.: Jate 2.0: Java automatic term extraction with Apache Solr. *Proceedings of the 10th international conference on language resources and evaluation LREC 2016*. Portorož, Slovenia: European Language Resources Association 2262–2269 (2016)

ДОДАТКИ

Додаток А. Сертифікати та дипломи, що підтверджують апробацію результатів





Ministry of Education and Science of Ukraine
 Alson-Adria-Universitäts Klagenfurt, Austria
 Aristotle University of Thessaloniki, Greece
 Université Côte d'Azur, France
 University of Vienna, Austria
 Institute of Information Technologies and Learning Tools, Ukraine
 Y.N. Karazin Kharkiv National University, Ukraine
 Kherson State University, Ukraine
 Lviv Polytechnic National University, Ukraine
 National Aerospace University "Kharkiv Aviation Institute", Ukraine
 Taras Shevchenko National University of Kyiv, Ukraine
 Zaporizhzhya National University, Ukraine
 Ukrainian Catholic University, Ukraine
 DataArt Solutions Inc.



Certificate of participation

This certifies that **Victoria Kosa** took part in
 the Information and Communication Technologies in Education,
 Research, and Industrial Applications

14th International Conference, ICTERI 2018
 Taras Shevchenko National University
 Kyiv, Ukraine, May 14-17, 2018



Local organization chair
 Prof. Anatoly Anisimov





15th International Conference ICTERI 2019
ICT in Education, Research, and Industrial Applications
Kherson State University, Kherson, Ukraine
12-15 June 2019

CERTIFICATE of APPRECIATION

This certificate is issued by the Organizing Committee of ICTERI 2019 in recognition of the efforts and performance of the participant

Victoria Kosa

Zaporizhzhia National University, Ukraine



DATAART

bwt



General Chair of ICTERI 2019

A handwritten signature in blue ink, belonging to Prof. Aleksander Spivakovsky.

Prof. Aleksander Spivakovsky



FACULTY OF
APPLIED SCIENCES

1 st International Masters Symposium on Advances in
Data Mining, Machine Learning, and Computer Vision
(MS-AMLV 2019), Ukrainian Catholic University,
Lviv, Ukraine, November 15-16 2019.

CERTIFICATE OF APPRECIATION

This certificate is issued by the Organizing Committee of MS-AMLV-2019 in
recognition of the effort and performance, as a presenter, of the talk
included in the program of the symposium

Victoria Kosa

Talk title: Toward a theoretical framework of terminological
saturation for ontology learning from texts

General Chair

Yaroslav Prytula
Dean of the Faculty of Applied Sciences



Запорізька обласна державна адміністрація



ДИПЛОМ

переможця обласного конкурсу
для обдарованої молоді у галузі науки

**Коса
Вікторія Вікторівна**

номінація: фізико-математичні науки

категорія: молоді науковці

Голова

Віталій ТУРИНОК

Запоріжжя 2019

Додаток Б. Приклад згенерованого каталогу колекції документів

Цей приклад представляє фрагмент автоматично згенерованого каталогу, у форматі .XLSX, документів колекції GAN (розділ 5.1), що було автоматично зібрано за допомогою розробленого обчислювального конверсу.

D	E	F	G	H	I	J	K	L	M	N	O	P
Venue	Publisher	Volume No	Issue No	Pages	DOI	DOI Link	MSF ID	Paper Title	Authors	Affiliations	Complete Citation	Abstract
2	Curran		1097-	1097-			2163605009	ImageNet Classification	alex krizhevsky,ilya	university	@inproceedings	We trained a large, deep
3	IEEE		None-	None-			1686810756	Very Deep Convolutional	karen	university	@article(simony	In this work we investiga
4	IEEE		580-587	580-587	10.1109/	https://dx.	2102605133	Rich Feature Hierarchies	ross girshick,jeff	university	@inproceedings	Object detection perfor
5	Springer US	115	3	211-252	10.1007/	http://dx.	2117539524	ImageNet Large Scale	olga russakovsky,li	stanford	@article(russak	The ImageNet Large Scal
6	IEEE		3431-	3431-	10.1109/	https://dx.	1903029394	Fully convolutional	jonathan long,evan	university	@inproceedings	Convolutional networks
7	arXiv:		None-	None-			2952186574	Visualizing and	matthew d zeiler,rob	new york	@inproceedings	Large Convolutional Netw
8	Springer		818-833	818-833	10.1007/	https://d	1849277567	Visualizing and	matthew d zeiler,rob	new york	@inproceedings	Large Convolutional Netw
9	Springer,		346-361	346-361	10.1007/	https://d	2179352600	Spatial Pyramid Pooling in	kaiming he,xiangyu	microsoft	@article(he2015	Existing deep convolutio
10	arXiv:		None-	None-			2552465644	Image-to-Image	philip isola,junyan	university	@inproceedings	We investigate condition
11	arXiv: Learning		None-	None-			2173520492	Unsupervised	alec radford,luke	None;None	@article(radford	When a large feedforwar
12	arXiv: Neural		None-	None-			1904365287	Improving neural networks	geoffrey e	university	@inproceedings	When a large feedforwar
13	International	JMLR.org	647-655	647-655			2155541015	DeCAF: A Deep	jeff	university	@inproceedings	We evaluate whether tea
14	arXiv:		None-	None-			2605287558	Unpaired Image-to-Image	junyan zhu,taesung	university	@inproceedings	Image-to-image translati
15	IEEE	35	8	1798-	10.1109/	https://dx.	2163922914	Representation Learning: A	yoshua bengio,aaron	None;None	@article(bengio	The success of machine l
16	arXiv: Learning		None-	None-			2432004435	Improved Techniques for	tim salimans,jan	openai;goo	@inproceedings	We present a variety of n
17	European	Springer,	694-711	694-711	10.1007/	https://dx.	2331128040	Perceptual Losses for Real-	justin	stanford	@inproceedings	We consider image
18	arXiv:		None-	None-			1994002998	Return of the Devil in the	ken chatfield,karen	university	@article(chatfel	The latest generation of
19	arXiv: Learning		None-	None-			2605135824	Improved Training of	ishaan Gulrajani,aruk	google;Non	@inproceedings	Generative Adversarial
20	Computer	IEEE	3128-	3128-	10.1109/	https://d	1905882502	Deep visual-semantic	andrej karpathy,ji	stanford	@inproceedings	We present a model that
21	Computer	IEEE	2625-	2625-	10.1109/	https://d	1947481528	Long-term recurrent	jeff donahue,lisa anne	university	@inproceedings	Models comprised of dee
22	IEEE		30-42	30-42	10.1109/	https://dx.	2147768505	Context-Dependent Pre-	george e dahl,dong	university	@article(dahl20	We propose a novel cont
23	Neural	MIT Press	2017-	2017-			603908379	Spatial transformer	max jaderberg,karen	google;goo	@inproceedings	Convolutional Neural Net
24	Computer	IEEE	512-519	512-519	10.1109/	https://dx.	2062118960	CNN Features Off-the-	ali sharif	None;None	@inproceedings	Recent results indicate th
25	IEEE		14-22	14-22	10.1109/	http://doi	1993882792	Acoustic Modeling Using	abdelrahman	university	@article(moham	Gaussian mixture models
26	Computer	IEEE	1717-	1717-	10.1109/	https://dx.	2161381512	Learning and Transferring	maxime	ecole	@inproceedings	Convolutional neural net
27	Neural	Neural	487-495	487-495			2134670479	Learning Deep Features for	bolei zhou,agata	massachus	@inproceedings	Scene recognition is one
28	International	IEEE	1529-	1529-	10.1109/	https://d	2124592697	Conditional Random Fields	shuai zheng,sadeep	university	@inproceedings	Pixel-level labelling tasks,
29	arXiv:		None-	None-			2950891598	Holistically-Nested Edge	saining xie,zhuowen	None;None	@inproceedings	We develop a new edge
30	International	PMLR	214-223	214-223			2739748921	Wasserstein Generative	martin	courant	@inproceedings	

Додаток В. Алгоритми, що імплементують обчислювальний метод

Алг. 3.1. Алгоритм **GSAT** генерації каталогу колекції документів

Алгоритм GSAT. Сформувати каталог колекції документів.

Вхідні дані:

DR-URL - URL репозиторію документів

Q - запит (строка) для вибору документів з репозиторію документів

Вихідні дані:

CAT - файл XSLX каталогу в табличній формі, що містить записи бібліографічної інформації про документи, що входять до колекції.

1. **initialize**(*CAT*)

// Запит до DR, отримати результат як список URL-адрес

// описів документа (метадані)

2. *List* := **DR-query**(*DR-URL*, *Q*)

3. **initialize**(*CAT* [1..|*List*|])

4. **for** *i* := 1 to |*List*|

// Отримати атрибути метаданих

5. **parse**(*List*[*i*], *CAT*[*i*])

// Отримати кількість цитувань у Google Scholar

6. *CAT*[*i*].*Citation-Counts-(GS)* := **GS-query**(*CAT*[*i*].*Title*,
CAT[*i*].*Authors*,
CAT[*i*].*Publication-Year*)

// Обчислити частоту цитування

7. *CAT*[*i*].*Citations-per-Year-GS* := *CAT*[*i*].*Citation-Counts-GS* /
(**year**() - *CAT*[*i*].*Publication-Year* + 1)

// Створити ім'я файлу

8. *CAT*[*i*].*Document-File-Name* := *CAT*[*i*].*Publication-Year* + "-" +
CAT[*i*].*Volume-No* + "(" +
CAT[*i*].*Issue-No* + ")-(" +
CAT[*i*].*Pages* + ")-" +
substr(*CAT*[*i*].*DOI*, "/")

9. **end for**

10. **return**(*CAT*)

Алг. 3.2. Алгоритм DDL для завантаження повнотекстових документів з репозиторію документів

Алгоритм DDL. Завантажити PDF документи з URL-адрес, наданих у *CAT.Full-text-download-URL*.

Вхідні дані:

CAT - каталог колекції - файл XSLX каталогу, що містить записи бібліографічної інформації про документи, що входять до колекції.

Вихідні дані:

PD - назва локального каталогу для збереження завантажених PDF-файлів.

```
1.  if (check-dir(PD) = false)
    then create-dir(PD)
    else prune(PD)
2.  for i:= 1 to |CAT|
3.    download(CAT[i].Full-text-download-URL, // URL для завантаження
               CAT[i].Document-File-Name+".PDF", // ім'я файлу для
                                                       PDF-DIR) // назва каталогу
                                                       // зберегти файл
4.  end for
5.  return true
```

Алг. 3.3. Алгоритм PDF2TXT для здобуття тексту з набору повнотекстових файлів PDF у звичайні текстові файли.

Алгоритм PDF2TXT. Конвертувати PDF документи з каталогу вхідних PDF файлів у плоскі тексти.

Вхідні дані:

- PD* - ім'я локального вхідного каталогу, що містить PDF файли які потрібно конвертувати у звичайні тексти.
- TD* - ім'я локального вихідного каталогу для збереження конвертованих TXT файлів.
- ENC* - вихідне кодування: "ASCII" або "UNICODE".

Вихідні дані:

TXT файли у каталозі *TD*, що містять звичайні тексти відповідних PDF файлів у каталозі *PD* і мають ті самі імена, що і PDF файли.

```
// Перевірити вхідні параметри та ініціалізувати
1. if (check-dir(PD) = false) then return false
2. if (check-dir(TD) = false)
   then create-dir(TD)
   else prune(TD)
3. if (ENC <> "ASCII" or ENC <> "UNICODE") then ENC := "UNICODE"
// Обробка файлів з PD
4. while ((FN := nextfile(PD)) <> "")
5.   PDFF := open-file(PD + FN + ".PDF", "read")
6.   TXTF := new-file(TD + FN + ".TXT", "write")
7.   RawTxt := extract-txt(PDFF) // містить виклик функції API
                                   // для здобуття текстової частини
                                   // PDF файлу в UNICODE
   // Обробити необроблений текст
   // перемістити 1ше речення з RawTxt в NextSentence
8.   while ((NextSentence := extract-sentence(RawTxt)) <> "")
   // видалити розриви строк та дефіси
9.   NextSentence := nl-hph-remove(NextSentence)
   // замінити лігатури на відповідні послідовності літер
   // таблиця лігатур зчитується з окремого файлу
10.  NextSentence := lig-replace(NextSentence)
   // кодувати як ENC
11.  if (ENC <> "ASCII") then unicode2ascii(NextSentence)
   // зберігти речення до вихідних даних txt і додати розрив строки
12.  write(TXTF, NextSentence + NL)
13.  end while
14.  close-file(TXTF)
15. end while
16. return true
```

Алг. 3.4. Алгоритм **GDS** для генерації набору даних для поточної ітерації потоку обчислень.

Алгоритм GDS. Створіть набір даних для поточної ітерації потоку обчислень.

Вхідні дані:

- CAT* - XSLX файл каталогу колекції.
- IN* - кількість ітерацій.
- INC* - розмір інкременту (параметр конфігурації).
- TD* - шлях до локального каталогу, що містить TXT файли.
- DD* - шлях до локального каталогу для зберігання сформованого набору даних.
- DO* - порядок документів обраних для інкременту (параметр конфігурації).
- DP* - розділ колекції на частини (параметр конфігурації).

Вихідні дані:

```
DI - файл набору даних, збережений у DD.
// Ініціалізувати та перевірити параметри
1. if (IN = 1) then cat-sort(CAT, DO)
2. if (check-dir(DD) = false) then create-dir(DD)
3. if ((IN-1)*INC) >= |CAT|
   then return "No documents for iteration" + str(IN)
// Створити набір даних
4. DI := "D" + str(IN, 4); DIF := open-file(DD + DI + ".TXT", "rewrite")
5. if ((not DP) and IN > 1)
   then // Incremental: DI contains DI-1 plus the increment
6.   DI-1F := "D" + str(IN - 1, 4)
7.   DI-1F := open-file(DD + DI-1F + ".TXT", "read")
8.   copyfile(DI-1F, DIF)
// Додати інкремент
9. for i := ((IN - 1) * INC + 1) to min(IN * INC, |CAT|)
10.  D := readfile(TD + CAT[i].Document-File-Name+".TXT")
12.  write(TXTF, D)
13. end for
14. return DD + DI-1F + ".TXT - generated"
```

Алг. 3.5. Алгоритм AC-CV для оптимізованого обчислення C-values за допомогою Aho-Corascik Corascik (Aho and Corasick 1975) для швидкого співставлення декількох строк.

Алгоритм AC-CV. Обчислити значення C-value за допомогою Aho-Corascik для пошуку вкладених термінів

Вхідні дані:

S - набір строк-кандидатів у терміни, здобутий лінгвістичною частиною конвеєру. У наборі строка-кандидат повторюється стільки разів, скільки відображається у текстовому наборі даних.

Структури:

B - набір здобутих кандидатів у терміни. Кожен *B[i]* є структурою, що містить (*term:string, cv:float*), де: *термін* є строкою-кандидатом у терміни, яка з'являється лише один раз у *B*; *cv* є її C-value що потрібно обчислити.

ST - таблиця строк кандидатів у терміни та вкладених строк кандидатів у терміни побудована за допомогою Aho-Corascik FSM. Стовпці: *ST.term; ST.nstterm, ST.nstcv*

Вихідні дані:

```

B - набір здобутих термінів
// Ініціалізувати
1. A := ahocorascik.automaton() // використаний алгоритм Aho-Corascik
                                // Методи: A.add_word();
                                // A.make_automaton(); A.iter()

// (1) Обчислити власні частоти, сформувати B
2. sort(S)
3. k := 0; s := ""; f := 0
4. for i := 1 to |S|
5.   if (s <> S[i])
      // Новий кандидат у терміни (B[k].term) у наборі термінів
      // Поки що припускається, що s не вкладено в решту елементів B
      // Частота терміна f на цьому етапі зберігається у B[k].cv
      then k += 1; append(B); f := 1; s := S[i];
          B[k].term := s; B[k].cv := f
      // Той самий кандидат у терміни
      else f += 1
6. end for
// (2) Згенерувати префіксне дерево Aho-Corascick та FSM для пошуку
    вкладених термінів
7. for i := 1 to |B|
8.   A.add_word(B[i].term, (i, B[i].term, B[i].cv))
9. end for
10. A.make_automaton()
// (3) Побудувати та проіндексувати таблицю вкладень ST
11. for i := 1 to |B|
12.   cur-term := B[i].term
13.   cur-cv := B[i].cv
14.   for (end-index, (insert-order, original-value)) in A.iter(cur-term)
15.     if original_value <> cur-term
           then append(ST, (original-value, cur-term, cur-cv))
16.   end for
17. end for
18. IST := index(ST, ST.term+ST.nstterm)
// (4) Скорегувати C-values у B[.].cv використовуючи ST
19. for i := 1 to |B|
20.   k := 0; cval := 0
21.   for ind in loc(IST, ST.term, B[i].term)
22.     cval -= ST[ind].nstcv; k += 1
23.   end for
24.   if (k > 0) then cval /= k
25.   B[i].cv := log(len(B[i].term) * (B[i].cv + cval))
26. end for
27. return B

```

Алг. 3.6. Алгоритм MPCV для злиття часткових C-value з двох наборів термінів

Алгоритм MPCV. Об'єднати два набори термінів з частковими C-values (*pcv*)

Вхідні дані:

B_i, B_{i+1} - набори здобутих термінів.

Кожен термін $B_i.term$ супроводжується своїм $B_i.pcv$.

B_i, B_{i+1} сортуються у порядку зменшення $B_i.pcv$.

Вихідні дані: Набір термінів B_{i+1} злитий з B_i у B_{i+1} .

```
1. resort := false
2. for k := 1 to  $|B_i|$ 
3.   match := false
4.   for m := 1 to  $|B_{i+1}|$ 
5.     if ( $B_i.term[k] = B_{i+1}.term[m]$ )
6.       then  $B_{i+1}.pcv[m] += B_i.pcv[k]$ ; match := true
7.   if (match <> true)
8.     then append( $B_i[k], B_{i+1}$ ); resort := true
9.   end for
10. end for
11. if (resort) then sort( $B_{i+1}, B_{i+1}.pcv, desc$ )
12. return  $B_{i+1}$ 
```

Алг. 3.7. Базовий алгоритм THD, адаптований з (Tatarintseva et al. 2013)

Алгоритм THD. Обчислити термінологічну різницю між двома наборами термінів. Рівність символічних строк використовується для співставлення строк термінів.

Вхідні дані:

T_i , - набір збережених значущих термінів попередньої ітерації.

B_{i+1} - набір здобутих термінів поточної ітерації.

У цих наборах термінів кожен термін $T_i.term$ або $B_{i+1.term}$ супроводжується його C-value: $T_i.cv$ або $B_{i+1.cv}$. T_i та B_{i+1} сортуються у порядку спадання C-value.

Вихідні дані: T_{i+1} , eps , $thd(T_{i+1}, T_i)$, $thdr(T_{i+1}, T_i)$

```
// Обчислити eps для  $B_{i+1}$ , та  $T_{i+1}$ 
1.  sum := 0
2.  for k := 1 to | $B_{i+1}$ |
3.    sum +=  $B_{i+1}[k].cv$ 
4.  end for
5.  eps := 0.0; count := 0
6.  for i := 1 to | $B_{i+1}$ |
7.    eps +=  $B_{i+1}[k].cv$ 
8.    if eps > sum / 2.0
9.      then
10.         eps :=  $B_{i+1}[k].cv$ 
11.         while eps =  $B_{i+1}[count+1].cv$ 
12.           count += 1; append( $B_{i+1}[count]$ ,  $T_{i+1}$ )
13.         end while
14.         break
15.       else
16.         count += 1; append( $B_{i+1}[count]$ ,  $T_{i+1}$ )
17.     end for

// Обчислити thd та thdr
18. sum := 0
19. thd := 0
20. for k := 1, | $T_{i+1}$ |
21.   sum +=  $T_{i+1}[k].cv$ 
22.   found := false
23.   for m := 1, | $T_i$ |
24.     if ( $T_{i+1}[k].term$  =  $T_i[m].term$ )
25.       then
26.         thd += | $T_{i+1}[k].cv$  -  $T_i[m].cv$ |
27.         found := true
28.       end for
29.     if (NOT found) then thd +=  $T_{i+1}[k].cv$ 
30.   end for
31. thdr := thd / sum

32. return eps,  $T_{i+1}$ , thd, thdr
```

Алг. 3.8. Алгоритм групування подібних термінів (STG)

Алгоритм STG. Групування подібних термінів у наборах термінів

Вхідні дані:

T - набір термінів. Кожен термін $T.term$ супроводжується його C-value $T.cv$. T сортується за спаданням $T.cv$.

M - ім'я функції міри строкової подібності для порівняння термінів

th - значення порога подібності терміна з діапазона $[0,1]$

Вихідні дані: T із згрупованими подібними термінами

```
1.  $sum := 0$ 
2. for  $k = 1, |T|$ 
3.    $term := T[k].term$ 
4.    $cv := T[k].cv$ 
5.    $count := 1$ 
6.   for  $m = k+1, |T|$ 
7.     if  $M(term, T[m].term, th)$ 
8.       then  $cv += T[m].cv$ ;  $count += 1$ ; remove( $T[m]$ )
9.   end for
10.   $T[k].cv := cv / count$ 
11. end for
12. return  $T$ 
```

Алг. 3.9. Алгоритм (М-ЖР) для обчислення міри строкової подібності Жаро

Алгоритм М-ЖР. Обчислити подібність Жаро для символічних строк

Вхідні дані:

S_1, S_2 - символічні строки для порівняння

th - значення порогу подібності з діапазона $[0,1]$

Структури:

BS_1, BS_s - Логічні масиви з розмірами $[1..|l|]$ та $[1..|s|]$, де l найдовший та s найкоротший з S_1, S_2 , що вказують позиції співпадінь в S_1 та S_2 значенням «true»

Вихідні дані: *true*, якщо обчислено $JaroSim \geq th$, *false* в іншому випадку

```
// Ініціалізувати та перевірити вхідні дані
1. if ( $|S_1|=0$  or  $|S_2|=0$ ) then return false
2. if ( $th < 0.0$  or  $th > 1.0$ ) then return false
3.  $md := \lfloor |l|/2 \rfloor - 1$  // відстань співпадіння
4.  $cm := 0$  // лічильник для співпадінь
5.  $ct := 0$  // лічильник для транспозицій
6. if ( $|S_1| \geq |S_2|$ ) then ( $l := S_1; s := S_2$ ) else ( $l := S_2; s := S_1$ )
7.  $BS_1[1..|l|] := BS_s[1..|s|] := false$ 
// Підрахувати співпадіння
8. for  $i := 1$  to  $|s|$ 
9.   for  $k := \max(1, i-md)$  to  $\min(i+md, |l|)$ 
10.    if  $BS_1[k]$  then iterate
11.    if  $s[i] \neq l[k]$  then iterate
12.     $BS_1[k] := BS_s[i] := true; cm += 1; break$ 
13.  end for
14. end for
15. if ( $cm = 0$ ) then return false
// Підрахувати транспозиції
16.  $k := 1;$ 
17. for  $i := 1$  to  $|s|$ 
18.  if ( $BS_s[i] = false$ ) then iterate
19.  while ( $BS_1[k] = false$ )  $k += 1$ 
20.  if ( $s[i] \neq l[k]$ ) then  $ct += 1$ 
21. end for
22.  $ct := ct / 2$ 
// Обчислити подібність Жаро
23.  $JaroSim := (cm / |s| + cm / |l| + (cm - ct) / cm) / 3$ 
24. if ( $JaroSim \geq th$ ) then return true else return false
```

Алг. 3.10. Алгоритм (М-ЖВ) для обчислення міри строкової подібності Жаро-Вінклера

Алгоритм М-ЖВ. Обчислити подібність Жаро-Вінклера для символічних строк
Вхідні дані:

S_1, S_2 - символічні строки для порівняння
 th - значення порога подібності з діапазона $[0,1]$
 lcp - довжина $[1 \dots 4]$ спільного префікса, що потрібно перевірити в S_1, S_2
 sf - значення коефіцієнта масштабування (p) в межах $[0.1 \dots 0.25]$

Структури:

BS_1, BS_s - Логічні масиви з розмірами $[1..|l|]$ та $[1..|s|]$, де l найдовша та s найкоротша S_1, S_2 , вказуючи позиції співпадінь у S_1 та S_2 значенням «true»

Вихідні дані: *true*, кщо обчислено $JWSim \geq th$, *false* в іншому випадку

```
// Ініціалізувати та перевірити вхідні дані
1. if (|S1| = 0 or |S2| = 0) then return false
2. if (th < 0.0 or th > 1.0) then return false
3. md := [|l|/2] - 1 // відстань співпадіння
4. cm := 0 // лічильник для співпадінь
5. ct := 0 // лічильник для транспозицій
6. if (|S1| ≥ |S2|) then (l := S1; s := S2) else (l := S2; s := S1)
7. BS1[1..|l|] := BSs[1..|s|] := false
8. if (lcp ≤ 0 or lcp > 4) then lcp := 4
9. if (sf < 0.1 or sf > 0.25) then sf := 0.1
// Підрахувати співпадіння
10. for i := 1 to |s|
11. for k := max(1, i-md) to min(i+md, |l|)
12. if BS1[k] then iterate
13. if s[i] <> l[k] then iterate
14. BS1[k] := BSs[i] := true; cm += 1; break
15. end for
16. end for
17. if (cm = 0) then return false
// Підрахувати транспозиції
18. k := 1;
19. for i := 1 to |s|
20. if (BSs[i] = false) then iterate
21. while (BS1[k] = false) k += 1
22. if (s[i] <> l[k]) then ct += 1
23. end for
24. ct := ct / 2
// Обчислити подібність Жаро
25. JaroSim := (cm / |s| + cm / |l| + (cm - ct) / cm) / 3
// Обчислити налаштування Вінклера
// Обчислити довжину спільного префікса
26. clcp := 0
27. for i := 1 to min(lcp, |s|)
28. if s[i] <> l[i] then break
29. end for
// Обчислити JWSim
30. JWSim := JaroSim + clcp * cp * (1 - JaroSim)
31. if (JWSim ≥ th) then return true else return false
```

Алг. 3.11. Алгоритм (М-ЖА) для обчислення міри строкової подібності Жакара

Алгоритм М-ЖА. Обчислити подібність Жакара для символьних строк

Вхідні дані:

S_1, S_2 - символьні строки для порівняння

th - значення порогу подібності з діапазона $[0,1]$

Вихідні дані: *true*, якщо обчислено *JaccardSim* $\geq th$, *false* в іншому випадку

```
// Ініціалізувати та перевірити вхідні дані
1. if ( $|S_1|=0$  or  $|S_2|=0$ ) then return false
2. if ( $th < 0.0$  or  $th > 1.0$ ) then return false
3. if ( $|S_1| \geq |S_2|$ ) then ( $l := S_1; s := S_2$ ) else ( $l := S_2; s := S_1$ )
// Обчислити подібність Жакара
// Розмір перетину
4.  $is := 0$ 
5. for  $i := 1$  to  $|l|$ 
6.    $k := 1$ 
7.   while  $k \leq |s|$ 
8.     if ( $l[i] = s[k]$ ) then  $is += 1$ ; remove( $s[k], s$ ); break
       else  $k += 1$ ; iterate
9.   end while
10. end for
// Подібність
11.  $JaccardSim := is / (|S_1| + |S_2|)$ 
12. if ( $JaccardSim \geq th$ ) then return true else return false
```

Алг. 3.12. Алгоритм (M-SD) для обчислення міри строкової подібності Соренсена-Дайса

Алгоритм M-SD. Обчислити подібність Соренсена-Дайса для символічних строк
Вхідні дані:

S_1, S_2 - символічні строки для порівняння
 th - значення порогу подібності з діапазона $[0,1]$

Структури:

bl, bs - масиви 2-символьних підстрок (бі-грамів) з розмірами $[1..|l|]$
та $[1..|s|]$, де l найдовший та s найкоротший
з S_1, S_2

Вихідні дані: $true$, якщо обчислено $SDSim \geq th$, $false$ в іншому випадку

```
// Ініціалізувати та перевірити вхідні дані
1. if ( $|S_1|=0$  or  $|S_2|=0$ ) then return false
2. if ( $th < 0.0$  or  $th > 1.0$ ) then return false
3. if ( $|S_1| \geq |S_2|$ ) then ( $l := S_1; s := S_2$ ) else ( $l := S_2; s := S_1$ )
// Створити масиви бі-грамів
4. for i := 1 to  $|l| - 1$  // bi-grams of l
5.    $bl[i] := l[i] + l[i+1]$ 
6. end for
7. for i := 1 to  $|s| - 1$  // bi-grams of s
8.    $bs[i] := s[i] + s[i+1]$ 
9. end for
// Обчислити подібність Соренсена-Дайса
// Розмір перетину
10.  $is := 0$ 
11. for i := 1 to  $|bl|$ 
12.    $k := 1$ 
13.   while  $k \leq |bs|$ 
14.     if ( $bl[i] = bs[k]$ ) then  $is += 1$ ; remove( $bs[k], bs$ ); break
15.     else  $k += 1$ ; iterate
16.   end while
17. end for
// Подібність
17.  $SDSim := is / (|bl| + |bs|)$ 
18. if ( $SDSim \geq th$ ) then return true else return false
```

Алг. 3.13. Вдосконалення (R-THD) базового алгоритму THD (розділ 4.5).

Алгоритм R-THD. Обчислити Термінологічну Різницю між двома наборами термінів з використанням мір строкової подібності M для співставлення термінів

Вхідні дані:

T_i , - набір збережених значущих термінів попередньої ітерації.

B_{i+1} - набір здобутих термінів поточної ітерації.

У цих наборах термінів, кожен термін $T_i.term$ або $B_{i+1.term}$ супроводжується його C-value: $T_i.cv$ або $B_{i+1.cv}$. T_i та B_{i+1} сортуються у порядку спадання C-values.

Вихідні дані: T_{i+1} , eps , $thd(T_{i+1}, T_i)$, $thdr(T_{i+1}, T_i)$

```
// Обчислити eps для  $B_{i+1}$ , та  $T_{i+1}$ 
1. sum := 0
2. for k := 1 to | $B_{i+1}$ |
3.   sum +=  $B_{i+1}[k].cv$ 
4. end for
5. eps := 0.0; count := 0
6. for i := 1 to | $B_{i+1}$ |
7.   eps +=  $B_{i+1}[k].cv$ 
8.   if eps > sum / 2.0
9.     then
10.      eps :=  $B_{i+1}[k].cv$ 
11.      while eps =  $B_{i+1}[count+1].cv$ 
12.        count += 1; append( $B_{i+1}[count]$ ,  $T_{i+1}$ )
13.      end while
14.      break
15.     else
16.      count += 1; append( $B_{i+1}[count]$ ,  $T_{i+1}$ )
17.   end for

// Обчислити thd та thdr
18. sum := 0
19. thd := 0
20. for k := 1, | $T_{i+1}$ |
21.   sum +=  $T_{i+1}[k].cv$ 
22.   found := false
23.   for m := 1, | $T_i$ |
24.     if (M( $T_{i+1}[k].term$ ,  $T_i[m].term$ , th))
25.       then
26.         thd += | $T_{i+1}[k].cv$  -  $T_i[m].cv$ |
27.         found := true
28.       end for
29.     if (NOT found) then thd +=  $T_{i+1}[k].cv$ 
30.   end for
31. thdr := thd / sum
32. return eps,  $T_{i+1}$ , thd, thdrInput:
```

Алг. 3.14. Алгоритм **ARNR** для видалення строк ARN з наборів термінів

Алгоритм ARNR. Видалити ARN з набору термінів T

Вхідні дані:

ARN - набір символічних строк накопиченого регулярного шуму, по одній на елемент

T - набір термінів. Кожен термін $T[i].term$ супроводжується його $T[i].cv$.

Вихідні дані: T без ARN

```
1. for  $i := 1$  to  $|ARN|$ 
2.   for  $j := 1$  to  $|T|$ 
3.     if  $ARN[i] = T[j].term$ 
4.       then remove( $T[j]$ ); break
5.   end for
6. end for
7. return  $T$ 
```

Додаток Г. Характеристики модулів розробленого програмного забезпечення

Функціональність	Мова	Доступ	Обмеження
Етап ⁶⁵ : Підготовчий; Функція ⁶⁶ : Створення Каталогу; Алгоритм : GSAT (розділ 3.2.1)			
Приймає URL-адресу сторінки журналу Springer. Аналізує сторінки випусків журналу та здобуває необхідні метадані в каталог. Бере кількість цитувань з Google Scholar ⁶⁷ .	PHP	https://github.com/bwtgroup/SSRTDC-Springer-article-parser	Тільки для репозиторію журналів Springer Link ⁶⁸
Вибирає метадані з PDF файлів оброблених статей. Бере кількість цитувань з Google Scholar.	Java	https://github.com/OntoElect/Code/tree/master/CatGen-PDF	Колекцію PDF документів потрібно попередньо завантажити в локальну папку
Приймає ідентифікатори статей здобуті конвеєром вибірки літератури снігова куля. Аналізує сторінки статей Microsoft Research (MSR) ⁶⁹ для пошуку необхідних метаданих. Бере кількість цитувань з Google Scholar.	Python 3	https://github.com/gen_dobr/snowball/blob/master/scripts/009_extend_items.py	Тільки для репозиторію статей MSR
Етап : Підготовчий; Функція : Завантаження Документів; Алгоритм : DDL (розділ 3.2.2)			
Здійснює пакетне завантаження статей, перелічених у каталозі. Перевіряє, чи надається повнотекстова URL-адреса MSR. Якщо ні, перевіряє Google Scholar для пошуку URL. Якщо ні, повідомляє про неможливість завантаження.	Python 3	https://github.com/gen_dobr/snowball/blob/master/scripts/011_download_pdfs.py	Частота запитів на завантаження може обмежуватися певними повнотекстовими репозиторіями.
Етап : Попередня обробка; Функція : Конвертація PDF у звичайний текст; Алгоритм : PDF2TXT (розділ 3.3.1)			
Приймає каталог з PDF файлами. Виводить каталог TXT файлів (UNICODE)	Python 3	https://github.com/gen_dobr/snowball/blob/master/scripts/013_ate_pdf2txt.py	
Приймає каталог з TXT файлами. Застосовує видалення переносів та розділення строк, заміну лігатур. Зберігає тексти у необхідному кодуванні.	Python 3	https://github.com/gen_dobr/snowball/blob/master/scripts/014_ate_clear_txt.py	
Етап : Передобробка; Функція : Генерація Наборів Даних; Алгоритм : GDS (розділ 3.3.2)			
Приймає: (i) групу значень параметрів конфігурації наборів даних (Таблиця 4.2); (ii) каталог з TXT файлами; та (iii)	Python 3	https://github.com/gen_dobr/snowball/blob/master	

⁶⁵ Етап в таблиці відповідає підпроцесу на діаграмі робочого процесу, розділ 3.1, Рис. 3.1.

⁶⁶ Функція відповідає модулю на діаграмі потоку обчислень, розділ 3.1, Рис. 3.3.

⁶⁷ <https://scholar.google.com/>

⁶⁸ <https://link.springer.com/journals/>

⁶⁹ <https://academic.microsoft.com/>

каталог колекції. Генерує набори даних із TXT файлів відповідно до значень параметрів конфігурації.		ster/scripts/015_ate_ge_nerate_datasets.py	
Етап: Здобуття Термінів; Функція: Здобуття Термінів (базовий метод); Прийняте програмне забезпечення: UPM Term Extractor			
Приймає каталог, що містить набори даних. Повертає каталог, що містить набори здобутих термінів. Використовує базовий метод C-Value (розділ 1.11) для статистичної частини.	Java	https://github.com/onto-logylearning-oeg/epnoi-legacy	Обробляє лише тексти, написані англійською мовою. Розмір набору даних не повинен перевищувати 15 Мб
Етап: Здобуття Термінів; Function: Здобуття Термінів (оптимізований метод); Алгоритм: АС-SV (розділ 3.4.1) – використовується в статистичній частині програмного забезпечення для оптимізації обчислення C-values			
Приймає каталог, що містить набори даних. Повертає каталог, що містить набори здобутих термінів. Використовує оптимізацію за Aho-Corasick методу C-Value (розділ 1.11) для статистичної частини.	Python 3	https://github.com/gen-dobr/snowball/blob/master/scripts/lib/ate.py	Обробляє лише тексти, написані англійською мовою.
Етап: Постобробка; Функція: Об'єднати Набори Термінів (оптимізований); Алгоритм: MRCSV (розділ 3.4.2)			
Приймає два послідовних набори термінів B_i та B_{i+1} що містять часткові C-values. Виводить B_{i+1} об'єднані з B_i як B_{i+1} . Вихідні дані містять об'єднані часткові C-values.	Python 3	https://github.com/gen-dobr/snowball/blob/master/scripts/016_ate_merge_terms_partial.py	
Етап: Постобробка; Функція: Групування Термінів; Алгоритми: M-JR, M-JW, M-JA, M-SD, STG (розділ 3.6.3)			
Приймає набір термінів. Виводить набір термінів зі згрупованими подібними термінами. Подібність термінів обчислюється за допомогою одного з чотирьох обраних мір (розділ 3.6.1) та відповідного порогу подібності (розділ 3.6.2)	Python 3	https://github.com/OntoElect/Code/tree/master/STG/core	
Етап: Постобробка; Функція: Обчислити Термінологічні Різниці (базовий); Алгоритм: THD (розділ 3.5)			
Приймає набір збережених значущих термінів T_i та набір здобутих термінів наступної ітерації B_{i+1} . Повертає: eps_{i+1} , T_{i+1} , $thd(T_i, T_{i+1})$, та $thdr(T_i, T_{i+1})$. Також повертає кількість здобутих та збережених термінів.	Python 3	https://github.com/gen-dobr/snowball/blob/master/scripts/lib/thd.py	
Етап: Постобробка; Функція: Обчислити Термінологічні Різниці (вдосконалений); Алгоритми: M-JR, M-JW, M-JA, M-SD (розділ 3.6.3), R-THD (розділ 3.6.4)			
Приймає набір збережених значущих термінів T_i , набір здобутих термінів наступної ітерації B_{i+1} , назву методу M для обчислення подібності двох термінів і значення порогу подібності терміна th .	Python 3	https://github.com/OntoElect/Code/tree/master/STG	

Повертає: eps_{i+1} , T_{i+1} , $thd(T_i, T_{i+1})$, та $thdr(T_i, T_{i+1})$. Також повертає кількість здобутих та збережених термінів.

Етап: Постобробка; **Функція:** Очищення Наборів Термінів; **Алгоритм:** ARNR (розділ 3.7)

Приймає набір стоп-термінів (складений вручну шляхом перевірки набору термінів із накопиченим регулярним шумом), каталог із наборами термінів, які потрібно очистити від шуму.

Python 3 https://github.com/gen-dobr/snowball/blob/master/scripts/017_ate_clear_terms.py

Повертає очищені набори термінів у тому ж самому каталозі.

Додаток Д. Довідки про впровадження розробленого методу та програмного забезпечення

Довідка про впровадження у промислове використання в проекті компанії
ТОВ ГРУПБВТ

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «ГРУПБВТ»
Код ЄДРПОУ: 43447154;
69106, Україна, Запорізька обл., м. Запоріжжя, вул. Глазунова, буд. 6, кв. 54;
Тел. (097) 533 – 82 – 66; ел. пошта: megamanmail@gmail.com

Вих. № 02/0421
від 08.04.2021

м. Запоріжжя

ДОВІДКА

про впровадження у виробничий процес
результатів дисертаційного дослідження Коси Вікторії Вікторівни «Метод
експериментального дослідження термінологічного насичення в колекціях
документів для здобуття знань»,
що подана на здобуття наукового ступеня доктора філософії
за спеціальністю 122 – комп'ютерні науки

Розроблений в дисертаційному дослідженні Коси В. В. «Метод експериментального дослідження термінологічного насичення в колекціях документів для здобуття знань», за спеціальністю 122 – комп'ютерні науки, обчислювальний метод для виявлення та вимірювання термінологічного насичення в колекціях професійних текстів, реалізований як пакет програм, було впроваджено в промислове використання компанією ТОВ «ГРУПБВТ».

Компанією ТОВ «ГРУПБВТ» надане програмне забезпечення було використано для виконання власного дослідницького проекту SAGOIT-IT: Strategic Analysis of R&D Gaps and Opportunities for Industrial Uptake in Trending IT Fields. Метою проекту було перевірити прогноз Gartner Inc. щодо трендів промислового впровадження однієї з перспективних інформаційних технологій, генеративних Змагальних Мереж, за допомогою аналізу термінологічної насиченості у відповідно зібраній колекції наукових статей. Результати проекту будуть використані ТОВ «ГРУПБВТ» для подальшого стратегічного аналізу та визначення пріоритетних напрямів розвитку компанії.

Ющенко Євген Ігорович
Директор ТОВ «ГРУПБВТ»
м. Запоріжжя, 69106, вул. Глазунова 6, 54
Код ЄДРПОУ 43447154

М.П. Б.П. Підпис



*Примітка: Ведення документообігу ТОВ «ГРУПБВТ» не передбачає печатки для цього типу документів

Довідка про впровадження в освітній процес у Міжнародній магістерській програмі з Комп'ютерних наук та науки про дані Українського католицького університету.

Український
Католицький
Університет

вул. Іл. Свенціцького, 17,
м. Львів, 79011
тел.: (38/032) 240-99-40
факс: (38/032) 240-99-50



Ukrainian
Catholic
University

вул. Sventsitskogo, 17
Lviv, 79011, Ukraine
email: info@ucu.edu.ua
www.ucu.edu.ua

ДОВІДКА

про впровадження в освітній процес

результатів дисертаційного дослідження Коси Вікторії Вікторівни

«Метод експериментального дослідження термінологічного насичення в колекціях документів для здобуття знань»,

за спеціальністю 122 – комп'ютерні науки

Розроблені в дисертаційному дослідженні Коси В. В. «Метод експериментального дослідження термінологічного насичення в колекціях документів для здобуття знань», за спеціальністю 122 – комп'ютерні науки, пакет програм та інформаційна технологія для вимірювання тематичної репрезентативності колекції наукових публікацій за допомогою аналізу термінологічної насиченості (див. <https://github.com/gendobr/snowball.git>) впроваджені в освітній процес факультету прикладних наук Українського Католицького Університету за освітньо-професійною програмою «Науки про дані» спеціальності 122 - Комп'ютерні науки другого (магістерського) рівня вищої освіти при викладанні дисципліни Академічне письмо. У 2020 році ці програми та технологія були використані студентами магістерської програми при виконанні курсового завдання за вищевказаною дисципліною.

Керівник магістерської програми


О. Молчановський

Декан факультету


Я. Притула



СВІДЧИТИ. СЛУЖИТИ. СПІЛКУВАТИСЯ.

Додаток Е. Питання та метод перевірки прогнозу Гартнер

Щоб мати всебічне та досить детальне розуміння потенціалу аналізованої технології, впливати на ринки та пропонувати конкурентну перевагу для перших впроваджувачів технології, особливо актуально мати відповіді на наступні важливі питання:

- **П1.** Чи є зрілими наукові внески досліджень та відповідне професійне співтовариство навколо технології?

Відповідь на це питання є важливою, оскільки зрілість: (i) передбачає наявність чітко сформованого набору знань про SotA в даній галузі і сформоване дослідницьке співтовариство, що стоїть за цією областю знань; та (ii) сприяє розробці чіткої ідентифікації прогалін у R&D, що становлять потенційний інтерес.

- **П2.** Чи є прогалини у знаннях про технологію між академічними дослідниками та промисловими впроваджувачами?

Відповідь на це питання є важливою, оскільки, якщо виявити прогалини, завдання, спрямовані на їх звуження або усунення, можна було б легше визначити в плані R&D компанії.

- **П3.** Яким є відставання у часі між ідентифікованою SotA та поточною SotT?

Відповідь на це питання є важливою для з'ясування того, чи існує ще час і вікно можливостей на ринку для того, хто впроваджує технологію, щоб використовувати її у своєму бізнесі.

Щоб відповісти на питання **П1 – П3**, ми розглядаємо термінологічні відбитки документів (наукових статей) які описують предметну область (GAN) обрану для нашого аналізу. Ці термінологічні відбитки здобуваються для різних підгруп документів:

- Написані науковцями з академічної сфери
- Написані промисловцями
- Написані спільно науковцями з академічної сфери та промисловцями

Термінологічні відбитки цих підгруп документів порівнюються за допомогою функції термінологічної різниці (*thd*) (розділ 2.3). На основі цих порівнянь даються відповіді на питання.

Щоб забезпечити обґрунтованість порівнянь відбитків, усі зібрані документи повинні бути релевантними щодо обраного тематичного фокусу (домену), а колекція повинна бути репрезентативною. Релевантність у нашому методі забезпечується методом вибірки документів (Dobrovolskyi and Keberle 2020). Репрезентативність гарантується шляхом перевірки термінологічного насичення у колекції документів за допомогою методу та програмного забезпечення, розроблених в дисертаційній роботі.

Метод вибірки документів (Dobrovolskyi and Keberle 2020) заснований на поєднанні вибірки методом снігової кулі, ймовірного моделювання тем, аналізу мереж цитування для пошуку у заголовках, ключових словах та анотаціях наукових робіт, проіндексованих Microsoft Academic⁷⁰. Метод потребує надання кількох релевантних та значущих документів щоб розпочати ітераційний процес. У якості таких документів, автори рекомендують надавати часто цитовані оглядові статті, опубліковані не раніше ніж п'ять років тому.

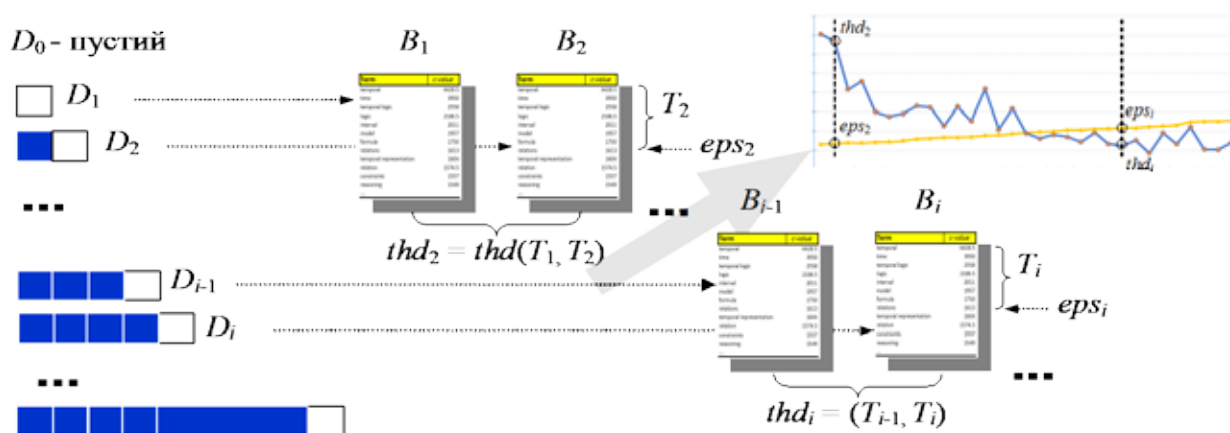
Метод вимірювання та аналізу термінологічної насиченості заснований на методі, представленому у розділі 2 та реалізованому в програмному забезпеченні конвеєру обробки (розділ 3). Отримана колекція документів була розділена на три частини на основі інформації про приналежність авторів окремих статей до промислових або академічних організацій. У цьому проекті розподіл було здійснено вручну.

Конфігурація пар наборів термінів, що використовуються в вимірюваннях термінологічних різниць відрізняється для відповідей на різні питання, як наведено нижче.

Щоб відповісти на питання **П1**, ми провели аналіз термінологічного насичення для кожної з трьох частин колекції окремо. Конфігурація цих вимірювань зображена на Рис. Г.1.

⁷⁰ <https://academic.microsoft.com/>

Для вимірювання насичення ми повинні використовувати наступну послідовність пар: (T_{i-1}, T_i) , $i = 1, \dots, \lceil N/INC \rceil$, де: N це кількість документів у частині колекції документів; $INC = 10$.



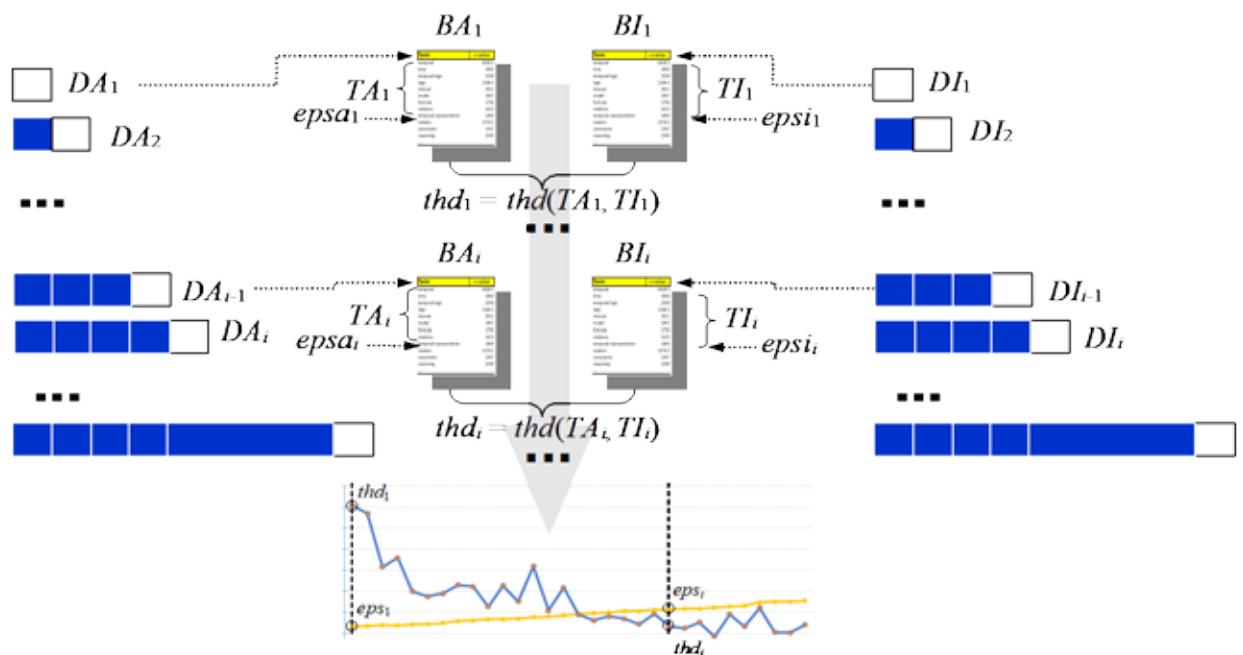
Легенда: D_i – набори даних у плоскому тексті;
 B_i – набори здобутих термінів;
 eps_i – індивідуальний поріг значущості термінів для B_i ;
 T_i – набори збережених значущих термінів;
 thd_i – термінологічна різниця для T_{i-1} та T_i ;
 i – номер ітерації

Рис. Г.1. Процес вимірювання термінологічної різниці для виявлення насичення в окремій частині колекції.

Щоб відповісти на **П2**, ми вимірювали термінологічні різниці між відповідними академічними та промисловими наборами даних на кожній ітерації послідовного процесу апроксимації. Подібним чином ми вимірювали термінологічні різниці між академічною та спільною, промисловою та спільною частинами колекції. Конфігурація цих вимірювань зображена на Рис. Г.2 для пари академічна та промислова частини.

Щоб відповісти на **П3**, ми заплановані провести ретроспективний аналіз термінологічних різниць. Ретроспектива базується на припущенні, що існує кількарічне відставання між академічним внеском (представленим в академічних публікаціях) та результативним промисловим впровадженням цього внеску (викладеним у промислових публікаціях). Звісно, впровадження технології відбувається пізніше. Отже, промислові статті можуть оперувати набором специфічних для певного домену термінів, який є дуже схожим на набір термінів в

наборі академічних статей, але за кілька років до цього. Це припущення опосередковано підтверджується Гартнер, яке для кожної розглянутої технології передбачає, скільки років знадобиться для досягнення плато продуктивності. Наприклад, відповідно прогнозам (Gartner 2019) GAN досягне плато через 5-10 років після 2019 року.



Легенда: DA_i, DI_i – набори даних різних частин (A – академічні, I - промислові) у плоскому тексті;
 BA_i, BI_i – набори термінів здобутих з різних частин наборів даних;
 eps_{A_i}, eps_{I_i} – індивідуальні пороги значущості термінів для BA_i, BI_i ;
 eps_i – $\min(eps_{A_i}, eps_{I_i})$
 TA_i, TI_i – набори збережених значущих термінів з різних частин наборів даних;
 thd_i – термінологічна різниця між TA_i та TI_i ;
 i – номер ітерації

Рис. Г.2. Обчислювальний процес для оцінки термінологічних різниць між наборами збережених значущих термінів різних частин колекції.

Одне з наших завдань у цьому проекті – з’ясувати, яка тривалість цього відставання у роках, що є показником широти вікна можливостей щодо впровадження технології. Для цього ми готуємо кілька зрізів академічної частини колекції. Ці зрізи містять статті, що опубліковані: у поточному році; до року тому; до двох років тому, ..., до десяти років тому. Потім ми досліджуємо термінологічні відмінності між кожним із підготовлених зрізів академічної частини та промисловою або спільною частиною колекції, як це було зроблено під час

відповіді на **П2**. Якщо є академічний зріз, який термінологічно найближчий до промислової або спільної частини, та термінологічна різниця менша за *eps*, то для пари підколекцій визначається відставання у часі.

Додаток Ж. Публікація, апробація та використання результатів роботи

Основні наукові результати цієї дисертаційної роботи було опубліковано у наступних 7 статтях (міжнародні періодичні видання з ISSN), що проіндексовані SCOPUS:

1. (Kosa et al. 2017a) Kosa, V., Chugunenko, A., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., Birukou, A.: Semantic saturation in retrospective text document collections. CEUR-WS, vol. 1851, 1–8 (2017) ISSN: 1613-0073
2. (Chugunenko et al. 2018) Chugunenko, A., Kosa, V., Popov, R., Chaves-Fraga, D., Ermolayev, V.: Refining terminological saturation using string similarity measures. CEUR-WS vol. 2105 3–18 (2018) ISSN: 1613-0073
3. (Kosa et al. 2018a) Kosa, V., Chaves-Fraga, D., Naumenko, D., Yuschenko, E., Badenes-Olmedo, C., Ermolayev, V., Birukou, A.: Cross-evaluation of automated term extraction tools by measuring terminological saturation. Revised selected papers of ICTERI 2017. Cham, Germany: Springer-Verlag, CCIS vol. 826, 135–163 (2018) doi: 10.1007/978-3-319-76168-8_7, ISSN: 1865-0929
4. (Kosa et al. 2019a) Kosa, V., Chaves-Fraga, D., Keberle, N., Birukou, A.: Similar terms grouping yields faster terminological saturation. Revised selected papers of ICTERI 2018. Cham, Germany: Springer-Verlag, CCIS vol. 1007, 43–70. (2019) doi: 10.1007/978-3-030-13929-2_3, ISSN: 1865-0929
5. (Kosa et al. 2019b) Chaves-Fraga, D., Dobrovolskyi, H., Fedorenko, E., Ermolayev, V.: Optimizing automated term extraction for terminological saturation measurement. CEUR-WS, vol. 2387, 1–16 (2019) ISSN: 1613-0073
6. (Kosa et al. 2020) Kosa, V., Chaves-Fraga, D., Dobrovolskiy, H., Ermolayev, V.: Optimized term extraction method based on computing merged partial C-values. Revised selected papers of ICTERI 2019. Cham, Germany: Springer-Verlag, CCIS vol. 1175, 24–49. (2020) doi: 10.1007/978-3-030-39459-2_2, ISSN: 1865-0929

7. (Kosa and Ermolayev 2020) Kosa,V., Ermolayev, V.: Toward a theoretical framework of terminological saturation for ontology learning from texts. CEUR-WS vol. 2566, 40–51 (2020) ISSN: 1613-0073

Аналіз повноти опублікування результатів дисертації наведено у Таблиці Д.1. У цій таблиці зіставляються частини дисертації, які містять результати, з опублікованою науковою роботою. Також наводиться кількість цитувань публікацій.

Таблиця Д.1. Повнота публікації результатів дослідження та цитування

Результат	Розділ(и) у дисертації	#	Посилання на Публікацію	Індексування	Кількість цитувань ⁷¹
Опубліковано					
Візія підходу, початкові формулювання питань дослідження та збір даних	ВСТУП, 1.12, 1.13, 4.1, 4.2	1.	(Kosa et al. 2017a)	SCOPUS: https://www.scopus.com/record/display.uri?eid=2-s2.0-85022043497&origin=resultlist	7
Вибір (базового) методу та програмного засобу для здобуття термінів. Збір та підготовка даних (синтетичні колекції). Експериментальна крос-оцінка.	1.8, 1.9, 1.12, 1.13, 3.1 – 3.3, 3.5, 3.7, 3.8, 4.1 - 4.4	2.	(Kosa et al. 2018a)	SCOPUS: https://www.scopus.com/record/display.uri?eid=2-s2.0-85044062624&origin=resultlist	8
Алгоритми мір подібності, пороги подібності, алгоритм групування термінів STG , вдосконалений алгоритм R-THD ; експериментальна оцінка впливу групування подібних термінів на термінологічне насичення.	1.10, 1.13, 3.6, 3.8, 4.1, 4.6	3.	(Chugunenko et al. 2018)	DBLP: https://dblp.uni-trier.de/pid/227/9912.html SCOPUS: https://www.scopus.com/record/display.uri?eid=2-s2.0-85048367546&origin=resultlist	6
		4.	(Kosa et al. 2019a)	DBLP: https://dblp.uni-trier.de/pid/227/9885.html [c1] SCOPUS: https://www.scopus.com/record/display.uri?eid=2-s2.0-	2

⁷¹ Кількість цитувань надається за даними Google Scholar (<https://scholar.google.com/citations?user=-yWzVVEAAAAJ>) від 28 січня 2021 року.

				85063447781&origin=result slist	
Оптимізований метод та алгоритми для обчислення злитих часткових C-value; доведення теореми про тотожність MPCV та C-value; експериментальна оцінка коректності методу	1.1 – 1.6, 1.11 – 1.13, 2.6, 3.3, 3.4, 3.8, 4.1, 4.7	5.	(Kosa et al. 2019b)	DBLP: https://dblp.uni-trier.de/pid/227/9885.html [c4] SCOPUS: <a href="https://www.scopus.com/record/display.uri?eid=2-s2.0-85068746071&origin=result
slist">https://www.scopus.com/record/display.uri?eid=2-s2.0-85068746071&origin=result slist	3
Експериментальна перевірка незалежності від домену та результативності методу для колекції великого промислового обсягу		6.	(Kosa et al. 2020)	DBLP: https://dblp.uni-trier.de/pid/227/9885.html [c3] SCOPUS: <a href="https://www.scopus.com/record/display.uri?eid=2-s2.0-85079095934&origin=result
slist">https://www.scopus.com/record/display.uri?eid=2-s2.0-85079095934&origin=result slist	2
Формулювання ключових тверджень формального фреймворку	2.1 – 2.5	7.	(Kosa and Ermolayev 2020)	SCOPUS: <a href="https://www.scopus.com/record/display.uri?eid=2-s2.0-85082439125&origin=result
slist">https://www.scopus.com/record/display.uri?eid=2-s2.0-85082439125&origin=result slist	
Подано до видавництва					
Вплив впорядкування документів на термінологічне насичення	1.7, 1.13, 3.3.2, 4.5, 5.3	8.	(Kosa et al. 2021)		

Окрім публікацій (Таблиця Д.1), деталі наукового доробку дисертації було оприлюднено у двох технічних звітах, із DOI, як зазначено у Таблиці Д.2.

Таблиця Д.2. Результати у загальнодоступних технічних звітах

#	Результат	Розділ(и) у дисертації	Звіт	URL
1.	Вибір (базового) методу та програмного засобу для здобуття термінів. Збір та підготовка даних (колекції документів). Експериментальна оцінка.	1.8, 1.9, 1.12, 1.13, 3.1 – 3.3, 3.5, 3.7, 3.8, 4.1 - 4.4	(Kosa et al. 2017b)	https://doi.org/10.13140/RG.2.2.31187.07207
2.	Вивчення впливу впорядкування документів на термінологічне насичення. Рейтинг можливих впорядкувань. Експериментальна оцінка.	1.7, 1.13, 3.3.2, 4.5, 5.3	(Kosa et al. 2018b)	https://doi.org/10.13140/RG.2.2.28382.54086

Апробація результатів дисертації

Результати досліджень доповідались (Додаток А) і були схвалені на:

- PhD симпозиумі 13ї міжнародної конференції ICT in Education, Research, and Industrial Applications (ICTERI 2017), Київ, 2017 р. – доповідь відзначена як найкраща
- 14їй міжнародній конференції ICT in Education, Research, and Industrial Applications (ICTERI 2018), Київ, 2018 р.
- 15їй міжнародній конференції «ICT in Education, Research, and Industrial Applications (ICTERI 2019), Херсон, 2019 р.
- Обласному конкурсі молодих науковців Запорізької обласної державної адміністрації «Молода наука», Запоріжжя, 2019 – визнана переможцем
- 1му симпозиумі Masters Symposium on Advances in Data Mining, Machine Learning, and Computer Vision (MS-AMLV 2019), Львів, 2019 р.

Практичне використання наукового доробку дисертації

Як представлено у розділах 5.1 та 5.2, методи та програмне забезпечення, розроблені в цій дисертаційній роботі, використовувались у промисловості, вищій школі (рівень магістратури), а також у наукових дослідженнях. Деякі кейси використання призвели до опублікування наукових результатів. Про факти використання доробку роботи свідчить інформація у Таблиці Д.3.

Таблиця Д.3. Результати, що використані організаціями та приватними особами

#	Використані результати	Контекст використання	Користувач / Публікація	URL
У промислових проектах				
1.	Перевірка прогнозу Гартнер щодо тенденцій впровадження технології GAN за допомогою аналізу термінологічного насичення	SAGOIT-IT - проект фінансований компанією ТОВ ГРУПБВТ, кінець 2020	ТОВ ГРУПБВТ	
У дослідницьких проектах				
2.	Базовий метод для вимірювання термінологічного насичення, експериментальна методологія.	проект OntoElect	(Ermolayev et al. 2014)	https://github.com/OntoElect

3.	Базовий метод та програмне забезпечення для вимірювання термінологічного насичення, експериментальна методологія.	проект OntoElect	(Ermolayev, V. 2018)	https://github.com/OntoElect
У вищій школі				
4.	Базовий метод та програмне забезпечення для вимірювання термінологічного насичення, експериментальна методологія, збір даних. Використання під час практичних занять.	Курс “Automated Term Extraction and Ontology Learning from Texts (DS.05.18)”, літо, 2018	Український католицький університет, факультет прикладних наук, магістерська програма з комп’ютерних наук та науки про дані	https://cms.u.edu.ua/course/view.php?id=1041#section-10
5.	Оптимізований метод та програмне забезпечення для вимірювання термінологічного насичення, експериментальна методологія, збір даних. Використання для виконання курсового завдання.	Курс “Academic Writing (1.20-21.ПКН19/М)”, кінець 2020	Український католицький університет, факультет прикладних наук, магістерська програма з комп’ютерних наук та науки про дані	https://cms.u.edu.ua/module/resource/view.php?id=134972
У магістерському проекті				
6.	Базовий метод та програмне забезпечення для вимірювання термінологічного насичення, експериментальна методологія, збір даних.	Для виконання магістерської кваліфікаційної роботи на кафедрі комп’ютерних наук ЗНУ (А. Чугуненко, 2017-2018)	(Chugunenko et al. 2018)	

Відомості у Таблицях Д.1 та Д.2 свідчать, що всі суттєві висновки та результати дисертаційної роботи є опублікованими, їх деталі представлені науково-технічними звітами. Інформація про використання результатів дисертації у Таблиці Д.3 свідчить про те, що доробок дослідження має достатній потенціал для подальшого впровадження, як інформаційної технології, в академічній сфері для досліджень і навчання студентів, а також в промисловості.