УДК 004.912

Ярмолюк Р.С.

Хмельницький національний університет, Україна

ПІДСИСТЕМА АВТОМАТИЧНОЇ КОРЕКЦІЇ ОРФОГРАФІЧНИХ ПОМИЛОК У ЗАПИСАХ ЕЛЕКТРОННОГО КАТАЛОГУ НА ОСНОВІ ТЕХНОЛОГІЇ HUNSPELL

Основною метою даного дослідження є розробка підходів до проектування підсистеми автоматичної корекції орфографічних помилок у записах електронного каталогу бібліотеки на основі словникової технології Hunspell.

The main purpose this research is to develop approaches to designing subsystems for automatic correction of spelling errors in your e-library catalog-based dictionary technology Hunspell.

Постановка проблеми. Електронний каталог бібліотеки являє собою складну метаінформаційну систему. Під час роботи з електронним каталогом виникають нестандартні ситуації, що призводять до появи у записах різного роду помилок та спотворень [1,2]. Система верифікації даних електронного каталогу (СВДЕК) є важливим елементом, що забезпечує пошук, оцінку, виправлення та уточнення помилкових даних в записах електронного каталогу [3]. На рисунку 1 представлено структурно-функціональну схему СВДЕК.

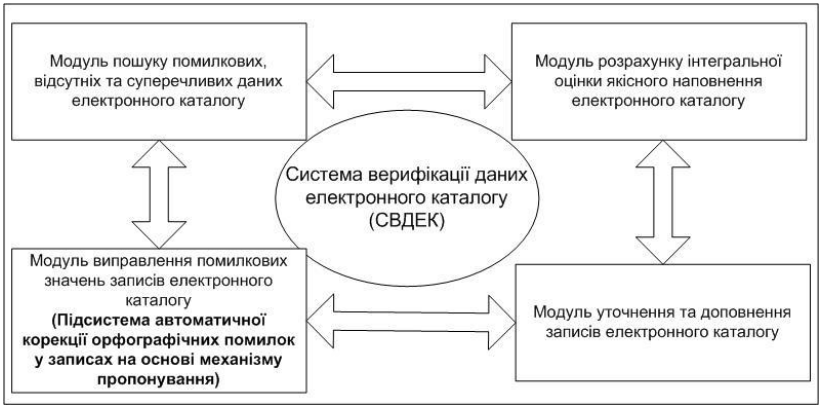

Рис. 1. **Структурно-функціональна схема СВДЕК**

Аналіз останніх досліджень і публікацій. В даний час найбільший інтерес представляє можливість автоматичної корекції помилкових записів електронного каталогу на природній мові. Хоча автоматично можлива лише корекція орфографічних помилок за допомогою методу пропонування можливих правильних форм для запису, а відповідно кінцевий вибір відбувається звичайно за участі людини. Але навіть при такому обмеженому автоматизмі точність і продуктивність вивірки записів електронного каталогу зростає [1,3].

Питання автоматичної корекції орфографічних помилок представлені у працях таких вчених, як Бабко-Малая О.Б., Гельбух А.Ф., Damerau F.J., Левенштейн В.И., Nielsen R., Ballard T., Вершинин М.И. та інші [1,4].

Формулювання цілей статті та актуальність досліджень. Підсистема автоматичної корекції орфографічних помилок є складовою частиною СВДЕК. Відповідно до структурно-функціональної схеми СВДЕК (Рисунок 1) підсистема автоматичної корекції орфографічних помилок є частиною модуля виправлення помилкових значень записів електронного каталогу.

Функціонально підсистема повинна реалізовувати можливості автоматичного пропонування варіантів для корекції записів, що не пройшли перевірку у модулі пошуку відсутніх, помилкових та суперечливих даних електронного каталогу [3].

Аналіз можливостей сучасних автоматизованих бібліотечних інформаційних систем (АБІС) показав відсутність інструментарію для корекції орфографічних помилок в записах баз даних електронного каталогу. Відсутність відповідної підсистеми для автоматичної корекції орфографії в таких поширених АБІС, як УФД/БІБЛІОТЕКА, МАРК-SQL, UNILIB, LIBER, ALEPH, Koha, ISIS, CDS Invenio, OpenBiblio, Evergreen, викликає багато труднощів для роботи коректора електронного каталогу. Зокрема, при роботі з електронним каталогом, виникає потреба ручної корекції. Тому, розробка підходів щодо проектування підсистеми автоматичної корекції на основі методу пропонування є важливою актуальною технічною та науковою проблемою при розробці програмних систем верифікації даних електронного каталогу бібліотеки.

Основною метою даного дослідження є розробка підходів до проектування підсистеми автоматичної корекції орфографічних помилок методом пропонування на основі словникової технології Hunspell. Аналіз технології створення словників пропонування за допомогою розмітки Hunspell. Створення рекомендацій для розробників відповідних модулів СВДЕК для АБІС.

Виклад основних матеріалів дослідження. Відповідно до структурних та логічних особливостей даних, що зберігаються в електронному каталозі бібліотеки, до підсистеми автоматичної корекції орфографічних помилок на основі словникового методу висувається ряд вимог:

•можливість роботи з декількома різномовними словниками;

•можливість створення користувацьких словників, що відображають специфіку певної предметної області або словників допустимих значень певних атрибутів;

•підтримка стандарту кодування символів UNICODE, зокрема систем кодування UTF-8.

•доступність готових загальнолексичних словників для різних мов;

•наявність ефективного механізму пропозицій для помилкових слів (пошук слів у словнику, які найбільше підходять за певним критерієм для даного неправильного слова);

•можливість інтеграції підсистеми, як в АБІС, так і в СВДЕК;

•можливість створення правил підстановки для методу пропонування;

•безкоштовність.

Проаналізувавши доступні на даний час системи перевірки орфографії, було відібрано словникову технологію Hunspell , яка задовольняє раніше викладені вимоги. Hunspell є повна програмна підтримка UNICODE та можливість створення власних правил пропонування у структурі словника. Тому для розробки підсистеми автоматичної корекції орфографічних помилок на основі словникового методу було вибрано ядро системи Hunspell.

Hunspell – вільна система перевірки орфографії, що використовується у таких вільних програмних продуктах,як: OpenOffice.org, LibreOffice, Thunderbird/Firefox, Opera, Google Chrome, The Bat! та інших. Для даної системи реалізовано інтерфейси та програмні порти для платформ: Delphi, Java, Perl, .NET, Python, Ruby.

Для перевірки орфографії за допомогою системи Hunspell необхідно два файли. Перший файл – словник, який містить у собі слова, другий - файл афіксів, який визначає значення спеціальних міток (атрибутів) в словнику [5].

Файл словника, як правило, має розширення *.dic та містить список слів по одному в рядку. В першому рядку словника вказується приблизна кількість слів в словнику. Дане значення використовується для оптимального розподілу пам’яті. Після кожного слова може слідувати знак «/» і одна або більше міток, які відповідають афіксам і

атрибутам. По замовчуванню, мітка являє собою один (зазвичай алфавітний) символ. В файлі словника Hunspell також може існувати поле для морфологічного опису, що відділяється табуляцією. Формат морфологічного опису визначається користувачем [5].

Файл афіксів, як правило, має розширення *.aff та може містити у собі необов'язкові атрибути. Наприклад, **SET** для визначення типу кодування символів в файлах афіксів і словників. **TRY** визначає символи для замін. **REP** визначає таблицю замін для виправлення декількох символів. **PFX** і **SFX** визначають класи префіксів і суфіксів, які позначені мітками афіксів.

Розглянемо більш докладніше мітки та атрибути, що призначені для організації роботи з правилами пропонування слів [5]:

- **TRY<***символи***>**. Hunspell можливо організовувати пропонування правильних слів , що відрізняються від помилкового на один символ в мітці **TRY**.
- **NOSUGGEST**<*мітка*>. Слова, що позначені даною міткою, не пропонуються. Мітку можливо використовувати для додавання виключень.
- **MAXNGRAMSUGS**<*число*>. Встановлює число пропозицій, що визначаються по ймовірності послідовної появи символів. Значення 0 відключає такі пропозиції.
- **NOSPLITSUGS**. Відключає пропозиції для слів з дефісом.
- **SUGSWITHDOTS**. Додає до пропозиції крапку, якщо невірне слово закінчується крапкою.
- **REP**<*кількість* > **REP**<*слово1*><*слово2*>. В файлах афіксів за допомогою таблиці заміни визначається фонетична інформація для конкретної мови . Спочатку іде мітка **REP** з заголовком таблиці, а потім два і більше рядків **REP** з даними. За допомогою даних таблиць Hunspell пропонує правильні варіанти написання для слів, якщо правильне написання слова відрізняється від введеного більш ніж на 1 символ.
- **MAP**<*кількість* > **MAP**<*рядок зв'язних символів*> За допомогою таблиці зв'язку символів для кожної конкретної мови в файлі афіксів можливо визначити інформацію про символи, що зв'язані один з одним більше ніж символи поза таблицею. За її допомогою в Hunspell пропонується правильне написання для слів, при введенні яких неправильна буква з відповідного набору була введена більше одного разу.

Розглянемо докладніше програмну реалізацію можливостей технології Hunspell. Далі у дослідженні використовується безкоштовний набір бібліотек NHunspell для платформи .NET

(http://nhunspell.sourceforge.net/), та об’єктно-орієнтовна мова програмування C#.

У просторі імен Nhunspell реалізовано 12 класів для доступу до Hunspell API функцій. Схема класів простору імен Nhunspell представлена на рисунку 2.

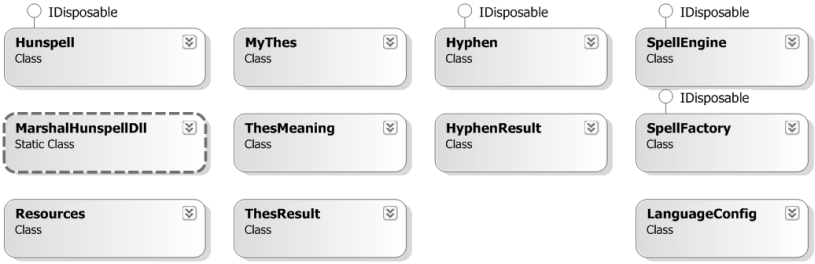

Рис. 2. **Схема класів Nhunspell**

Відповідно до схеми, класи можна згрупувати за функціональним призначеннями. Зокрема, публічний клас Hunspell реалізовує засоби перевірки орфографії, створення користувацьких словників слів та афіксів, морфологічний аналіз, механізм пропозицій для помилкових слів. Оскільки у даному класі реалізовані всі необхідні методи для забезпечення процесу автоматичної корекції орфографічних помилок, зупинимось на ньому більш докладніше. На рисунку 3 представлено перелік та сигнатуру усіх членів даного класу.

Клас інкапсулює у собі 4 перезавантаження конструктора класу **Hunspell**(). Як правило, використовують сигнатуру типу **Hunspell**(string affFile, string dictFile), де вхідні параметри affFile і dictFile є повним шляхом до словника афіксів і словника основ відповідно.

Метод **Load**(string affFile, string dictFile) має аналогічні перезавантаження, що й конструктор класу, і реалізовує можливість завантаження словників афіксів та основ у пам'ять.

Методи **Add**(string word) і **AddWithAffix**(string word, string example) призначені для додавання слів-виключень (word) та афіксів слів-виключень за певним правилом (example), тобто слів, яких немає у основних словниках, але які вважаються правильними. Слід зауважити, що слова не заносяться до файлів основних словників, а лише зберігаються у оперативній пам’яті, тому для організації їх використання необхідно створювати окремі файли для зберігання даних слів і використовувати їх завантаження при кожному запуску

системи. Метод повертає значення true, якщо додавання слова або афікса пройшло успішно якщо ні, то false.

Для безпосередньої перевірки орфографії використовується метод **Spell**(string word), word – слово, що перевіряється. Метод повертає значення true, якщо дане слово є коректним (тобто наявне у словнику), інакше false.

Методи **Stem**(string word), **Analyze**(string word) та **Generate**(string word, string sample) використовуються для морфологічного аналізу, зокрема перші два, для отримання основи слова та основи слова за морфологією відповідно. Останній метод використовується для генерації всіх варіацій слова word, за шаблоном sample. Методи повертають значення типу список List<string>.

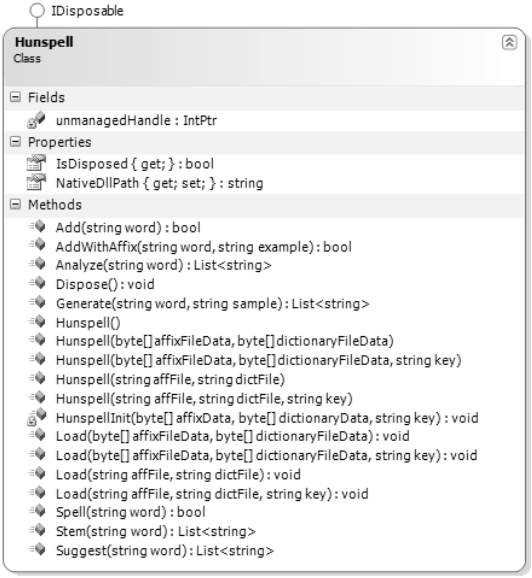

Рис. 3. **Перелік та сигнатура членів класу Hunspell**.

Реалізація механізму пропонування для помилкових слів здійснюється завдяки методу **Suggest**(string word), що повертає список (List<string>) можливих правильних слів для даного слова word. Список пропозицій генерується відповідно правил, що реалізовані у словнику афіксів та сортуються за визначеним пріоритетом. У випадку відсутності допустимих правильних слів список залишається порожнім. Реалізація даного методу на мові програмування C# з використанням API-процедур Hunspell представлена на рисунку4.

Отже маючи список слів, що є кандидатами для заміни слова із помилкою, коректор електронного каталогу у ручному режимі визначає який варіант є відповідно правильним [6].

```
public List<string> Suggest(string word)
{
    if (this.unmanagedHandle == IntPtr.Zero)
    {
        throw new InvalidOperationException("Dictionary is not loaded");
    }

    var result = new List<string>();

    IntPtr strings = MarshalHunspellDll.HunspellSuggest(this.unmanagedHandle, word);

    int stringCount = 0;
    IntPtr currentString = Marshal.ReadIntPtr(strings, stringCount * IntPtr.Size);

    while (currentString != IntPtr.Zero)
    {
        ++stringCount;
        result.Add(Marshal.PtrToStringUni(currentString));
        currentString = Marshal.ReadIntPtr(strings, stringCount * IntPtr.Size);
    }

    return result;
}
```

Рис. 4. **Реалізація методу пропонування у бібліотеці NHunspell.**

Підсистема перевірки орфографії, що реалізована на платформі .NET для операційної системи Windows складається з таких основних частин:

•Набір словників слів та афіксів – це файли з розширенням *.dic та *.aff відповідної структури Hunspell словників. Для корекції орфографії у полях бібліографічного запису, таких як «назва» або «анотація» достатньо і загальних словників та правил пропонування. Для корекції орфографії інших полів бібліографічного запису (автор, видання, тощо), необхідно створювати спеціалізовані користувацькі словники.

•Набір функцій та методів для роботи зі словниками Hunspell, що інкапсульовані відповідно до розрядності операційної системи у готові набори Hunspell API бібліотек (Hunspellx64.dll і Hunspellx86.dll). Дані бібліотеки написані на мові об’єктно-орієнтованого програмування C++ і доступні на сайті проекту (http://hunspell.sourceforge.net/).

•Набір методів для доступу до Hunspell API функцій, які інкапсульовані у класи відповідно до технології. У нашому випадку було обрано набір бібліотек для платформи .NET – Nhunspell (Nhunspell.dll). Дана бібліотека написана на мові об’єктно-

орієнтованого програмування C# і доступна на сайті проекту (http://nhunspell.sourceforge.net/).

•Підсистема автоматичної корекції орфографічних помилок і СКБД електронного каталогу бібліотеки – клієнт-серверний програмний комплекс, задача якого забезпечувати якість даних електронного каталогу. В залежності від платформи, на якій реалізована клієнтська частина, тобто підсистема перевірки орфографії, та від типу СКБД, використовується відповідна технологія доступу до даних. Однак, запропонована схема структурної організації підсистеми перевірки орфографії є крос-платформною і не залежить від обраної СКБД або технології реалізації.

Висновки. У результаті проведеного дослідження було запропоновано підходи для проектування підсистеми автоматичної корекції орфографічних помилок у записах електронного каталогу бібліотеки на основі словникової технології. Проаналізувавши функціональні вимоги, було обрано відкриту технологію Hunspell для побудови на її основі даної підсистеми. Запропонована схема структурної організації підсистеми на основі платформи .NET. Аналіз об'єктно-орієнтовної моделі платформи Nhunspell показав наявність у складі даної технології усього необхідного інструментарію для забезпечення автоматичної корекції орфографічних помилок.

Література

1. Вершинин М.И. Электронный каталог проблемы и решения / М. И. Вершинин. – СПб. : ПРОФЕССИЯ, 2007. – 233с.

2. Ярмолюк Р.С. Основні типи та джерела помилок у записах електронного каталогу / Р.С. Ярмолюк // Вісник Національного Університету «Львівська політехніка». Інформаційні системи та мережі, - 2010. - № 689. – С. 348-357.

3. Ярмолюк Р.С. Структурно-функціональна модель верифікації даних електронного каталогу / Р.С. Ярмолюк // Суч. пробл. діяльн. бібл..в ум. інф. сусп. м. 3-ої наук.-прак. конф., 29 вересня 2011р., Львів/ Національний університет "Львівська політехніка"; - Львів : Видавництво НУ "Львівська політехніка", 2011. – С. 217-224.

4. Ярмолюк Р.С. Задача аналізу текстових атрибутів в електронному каталозі / Р.С. Ярмолюк //Вісник Хмельницького Національного Університету, серія Технічні науки, - 2011. - № 3 – С.225-228.

5.Surhone L.M. Hunspell / L.M. Surhone, M.T. Tennoe, S.F. Henssonow – Betascript Publishing, 2010 – 116p.

6.Ярмолюк Р.С. Підходи до розрахунку інтегральної оцінки якісного наповнення електронного каталогу бібліотеки / Р. Ярмолюк // Іннов. комп. техн.. у в. шк.. : м. 3-ї наук.-практ. конф., 18–20 жовтня 2011 р., Львів / Нац. універ. "Львівська політехніка"; - Львів : Видавництво НУ " Львівська політехніка ", 2011. – С. 132-136.