УДК: 005:37

Білощицький Андрій Олександрович
Доктор технічних наук, завідувач кафедри інформаційних технологій
Діхтяренко Олександр Васильович
Аспірант кафедри основ інформатики
Лященко Тамара Олексіївна
Асистент кафедри основ інформатики

Київський національний університет будівництва і архітектури, Київ

ПЕРЕТВОРЕННЯ ФАЙЛІВ РІЗНИХ ТИПІВ ДО ЄДИНОГО ФОРМАТУ

Запропоновано спосіб модифікації текстових даних документа зі збереженням відповідності модифікованого документа оригінальному, а також можливість прямої обробки форматів MS Word, PDF та їх конвертація. Розглянуто три найпопулярніші формати збереження документів: DOC, DOCX і PDF, та можливі способи їх конвертації у базовий формат для подальшої роботи з вмістом.

Ключові слова: розбір форматів файлу, перетворення файлів, конвертація, антиплагіат

Предложен способ модификации текстовых данных документа с сохранением соответствия модифицированного документа оригинальному, а также возможность прямой обработки форматов MS Word, PDF и их конвертация. Рассмотрены три самые популярные формата хранения документов: DOC, DOCX и PDF, и возможные способы их конвертации в базовый формат для дальнейшей работы с содержимым.

Ключевые слова: разбор форматов файла, преобразования файлов, конвертация, антиплагиат

In this paper the method of modifying text data documents with matching modified the original document. As different formats are fundamentally different structure, the rational solution is to bring all formats to one standard for us form or format. It will not write tools to work with all types of files, and creates the opportunity to write tools to work with just one format converters and other formats to be chosen. This approach will allow for easy expansion in the future the number of formats supported, because adding support for new formats only need to write a converter. Also, the possibility of direct processing formats MS Word, PDF and convert. In the article the 3 most popular formats to save documents: DOC, DOCX and PDF, and their possible conversion into a basic format for further work with the content. A basic format of HTML, as supports saving and formatting all we need entities and has a simple structure that facilitates its handling. In order to process text data suggested build vocabulary index document that represents a table, the first column of which the position of words in the text, and the second - the actual words. This structure allows to process document Literal and apply to individual words any modification while retaining the possibility to compare the modified sample from the original.

Keywords: parsing file formats, file conversion, conversion, antiplagiat

Вступ

В рамках розробки системи пошуку збігів в наукових текстах виникає необхідність роботи з різними форматами файлів. Оскільки різні формати мають принципово різну будову, то раціональним рішенням буде звести всі формати до одного, стандартного для нас вигляду або формату. Це дозволить не писати засоби для роботи з усіма типами файлів, а створює можливість написати засоби для роботи з лише одним форматом і конвертери з інших форматів у обраний. Такий підхід дозволить у майбутньому легко розширювати кількість форматів, що підтримуються, адже для додавання підтримки нового формату необхідно лише написати конвертер. Слід уважно опрацьовувати єдиний обраний формат,

оскільки він, по-перше, має бути певною мірою універсальним, підтримувати збереження і відображення структур, що ми обробляються. По-друге, формат повинен надавати можливість легко обробляти дані, які він містить.

Мета статті

Мета статті – розглянути спосіб модифікації текстових даних документа зі збереженням відповідності модифікованого документа оригінальному. Тобто – це можливість знайти оригінал тексту в початковому документі незалежно від того, яких змін зазнав цей текст, чи його фрагмент, в модифікованому документі. Також розглядається можливість прямої обробки форматів MS Word, PDF та їх конвертація.

Виклад основного матеріалу

Поширені формати документів

DOC – формат, розроблений корпорацією Microsoft у 1990-х рр. Довгий час він був закритий і його структура не розголошувалася, через це працювати з цим форматом міг лише офісний пакет MS Office. Лише у 2008 р. Microsoft відкрила специфікації формату, але обмежила можливість його використання для комерційного програмного забезпечення (ПЗ). Крім того, специфікація описує не всі можливості формату, на даний момент вона ще доповнюється. Хоча сьогодні формат підтримується в тому числі і безплатними офісними пакетами, такими як OpenOffice, LibreOffice, повноцінну обробку і коректне відображення гарантує лише офісний пакет від Microsoft. Зокрема у відкритих офісних пакетах не відображаються деякі елементи вставлені з інших документів (наприклад схеми Visio). Формат може містити всередині себе будь-які інші файли, в тому числі й інші файли формату DOC.

DOCX (Office Open XML) – формат, розроблений корпорацією Microsoft в 2005 р. і позиціонується як відкритий, але захищений патентами Microsoft. Він являє собою zip-архів, що містить текстові дані у вигляді XML-файлів, графіка та інші елементи документа. Формат стандартизований такими організаціями, як ECMA та ISO, та має відкриту специфікацію на 7000 сторінок. Підтримується абсолютною більшістю сучасних офісних пакетів на всіх платформах. Так само, як і DOC, у DOCX файл можна вставити будь-який формат даних або інший файл. Крім того він має власну, відмінну від MathMl, реалізацію побудови математичних формул, хоча MathMl також підтримується.

PDF – формат, розроблений в 1993 р. корпорацією Adobe Systems і позиціонується як міжплатформний (Portable Document Format). Спочатку це був закритий комерційний формат, але в 2008 р. Adobe зробили його відкритим і опублікували специфікації. Формат набув широкого розповсюдження на всіх платформах, оскільки при правильній побудові (використанні вбудованих шрифтів) можна отримати абсолютно однаковий вигляд документу у будь-якому перегляді на будь-якій платформі. Він може містити у собі текстову інформацію, векторну та растрову графіку, форми і сценарії мовою JavaScript, а також прикріплені будь-які інші файли. Формат розвивається, почавши з версії 1.0 у 1993 р., зараз має версію 1.7, розробка якої завершена у 2011 р.

Вибір базового формату

Вищеописані формати документів є чи не найбільш розповсюдженими форматами збереження документів в країнах СНД. Тому при розробці програмного забезпечення, яке працює з текстовими документами, необхідно орієнтуватися саме на них. Як було зазначено, для того, щоб не розробляти засоби роботи з усіма типами форматів, можна розробити лише засоби роботи з одним, базовим, форматом та конвертери у базовий формат. Всі формати можуть містити у собі текстові дані, растрову чи векторну графіку, таблиці, формули, схеми, графіки, діаграми та безліч видів вкладень інших файлів. Спростити задачу до рівня, на якому її можна реалізувати, можна лише, якщо робити перетворення з втратами, наприклад графіки та діаграми перетворювати в графіку. Необхідно визначитися з тим, які дані документа необхідні, а якими можна знехтувати або перетворити у більш примітивний формат. У цій статті вирішується завдання перетворення документів зі збереженням тексту, частково форматування, табличних даних, графічних зображень. Векторна графіка та інші типи даних за можливості перетворюються в растрову графіку. Тому до базового формату, в який перетворюється документ, висуваються певні вимоги. Він повинен підтримувати форматування та відображення таких типів даних:

- текстових;
- табличних;
- графічних зображень.

При цьому краще, щоб формат мав просту структуру і з ним було зручно працювати. В цій роботі базовим форматом вибрано HTML.

HTML – стандартна мова розмітки для web-документів (рис. 1). Її розробка почалася у 1989 р., перший публічний опис датується 1991 р. На сьогодні ведеться розробка документації HTML5, що має закінчитися у 2014 р.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
    <head>
        <meta http-equiv="content-type" content="text/html; charset=windows-1251">
        <title>Заголовок</title>
    </head>
    <body>
        <p>Вміст документа.</p>
    </body>
</html>
```

Рис.1. Формат HTML документа

HTML файл – це звичайний текстовий документ зі спеціальною розміткою подібною до XML. Текстові дані розміщені всередині тегів, зображення зберігаються як окремі файли, а в сам документ вставляється спеціальний тег з посиланням на файл зображення. Також є можливість зберігати зображення в тексті, в закодованому (за допомогою BASE64) вигляді, але така можливість використовується не часто. Також існують спеціальні теги для форматування таблиць та інших сутностей. Для оформлення можуть використовуватися стилі CSS, які можуть бути як вписані в документ, так і винесені в окремий файл, та підключені за допомогою спеціального тега з посиланням на файл стилів. Аналогічно розміщуються скрипти мовою JavaScript, які використовуються для створення інтерактивних web-сторінок. Перевагою HTML над іншими вищеописаними форматами є те, що з програмного забезпечення для відкриття HTML сторінки необхідно мати лише web-браузер, а він є у комплекті поставки усіх сучасних користувацьких операційних систем (не серверних). Однак недоліком формату є його різне відображення в різних браузерах, навіть в межах однієї операційної системи. Хоча в рамках нашої задачі це не важливо, оскільки форматування не впливає на зміст документа. Крім того, оскільки HTML – це звичайний текстовий документ, то для його редагування вистачить найпримітивнішого текстового редактора.

Конвертація документів

Конвертація DOC. Оскільки цей формат став поширеним ще до того, як з'явилися його відкриті специфікації, були численні спроби зробити реінжиніринг для того, щоб хоча б прочитати дані документу. Зокрема є і наукові праці з теми конвертування формату DOC в TXT (http://jrnl.nau.edu.ua/index.php/SBT/article/viewFile/5291/5931). Але постановка задачі така, що необхідно дістати не лише текст, а і графічні та інші дані.

Аналіз наявного програмного забезпечення виявив, що найпростішим способом дістати всю необхідну інформацію з файлів DOC та DOCX файлів є використання бібліотек Microsoft Office Interop або LibreOffice Writer (запуск з параметрами у командній стрічці). Розробляти власне рішення надто трудомістко, а готові реалізації бібліотек або комерційні, і коштують від $500, або не задовольняють умови (не витягують з документу зображення, не конвертують графіки і діаграми тощо). Тому раціональним рішенням є придбання пакету MS Office (найнижча ціна 880 грн, http://office.microsoft.com/uk-ua/buy/) або використання безкоштовного LibreOffice Writer, де слід знехтувати деякими даними, які він не може перетворити у графічні зображення (діаграми Visio). Щодо формату PDF – ситуація дещо інша. Відкритих безплатних бібліотек немає, але наявні безплатні утиліти (http://sourceforge.net/projects/pdftohtml/), які можуть конвертувати файли та витягувати зображення, але з втратою форматування тексту та таблиць (перетворюються в текст).

Таким чином, використовуючи безплатне програмне забезпечення, можна реалізувати конвертацію трьох вищеописаних форматів у HTML не витрачаючи багато часу на написання коду.

Виділення з документа тексту та таблиць

Працювати з HTML документами, знаючи їх структуру, досить просто. Документ зазвичай починається з тегу, що визначає його версію, наприклад: <!DOCTYPE html> для HTML5. Вся інша інформація знаходиться всередині тега <html>, який має ще дві секції head і body. Всередині тега head може міститися різна метаінформація, така як кодування сторінки, заголовок, підключення зовнішніх CSS та JavaScript файлів. Всередині body знаходиться вже безпосередньо сама текстова інформація, відформатована за допомогою інших HTML тегів.

Для вставки в документ зображення використовується тег img з параметром src, що вказує на місцезнаходження файлу зображення:

<img src=”/path/to/image.jpg”>. Таблиці форматуються за допомогою тегів table, thead, tbody, th, tr, td. Слід зазначити, що теги thead і tbody необов’язкові. Табл. 1 у HTML документі буде мати вигляд, як показано на рис. 2.

Таблиця 1

Приклад таблиці

Заголовок колонки 1	Заголовок колонки 2
Рядок 1, колонка 1	Рядок 1, колонка 2
Рядок 2, колонка 1	Рядок 2, колонка 2

```
<table>
    <thead>
        <th>Заголовок колонки 1</th>
        <th>Заголовок колонки 2</th>
    </thead>
    <tbody>
        <tr>
            <td>Рядок 1, колонка 1</td>
            <td>Рядок 1, колонка 2</td>
        </tr>
        <tr>
            <td>Рядок 2, колонка 1</td>
            <td>Рядок 2, колонка 2</td>
        </tr>
    </tbody>
</table>
```

Рис. 2. Форматування таблиці всередині HTML документа

Звичайний текст може знаходитися всередині тегів div, p, span, li та ін. Таким чином досить просто виділити з документу рисунки, графіки та звичайний текст – вони знаходяться в різних тегах.

Редагування документу

Наступна задача даної статті – модифікація копії документу без втрати прив’язки до оригінального тексту. Тобто, якщо змінюємо текст, але необхідно мати можливість показати оригінал будь-якої ділянки документу. Це можна використовувати для перекладу документу на іншу мову, автокорекції помилок та ін. Далі буде розглянуто кілька варіантів реалізації такого функціоналу.

По перше, оскільки ми працюємо з HTML документом, можна використовувати атрибути тегів для збереження додаткової інформації, наприклад: кухонний. Такий підхід можна буде використати, але це не зовсім зручно, оскільки доведеться обгорнути тегами кожне слово, яке зазнає змін. Для того, щоб не робити помилок в структурі документу, необхідно брати до уваги вже існуючі теги, які теоретично можуть конфліктувати з новими. Тому була обрана інша реалізація.

Для початку зазначимо, що основний матеріал, з яким ми працюємо, це – слова. В системі пошуку збігів ряд модулів обробляють матеріал послівно (зміна форми слова, заміна синонімів тощо), тому раціональним рішенням буде використовувати структуру, яка надає можливість такого перебору. А для того, щоб не втратити зв’язку з оригінальним текстом, всі слова будуть проіндексовані. Індексом служитиме позиція першого символу слова, відлік починається з нульової позиції від початку документу. Для того, щоб можна було зібрати з набору слів речення – до слова також будуть додаватися розділові знаки, які після нього йдуть (якщо такі є). Таким чином проходячи в циклі по всіх словах (від більшого індексу до меншого) – можна за певними ознаками виокремити речення (наприклад, в кінці слова крапка, наступне починається з великої букви та ін.). Таким чином, початковий словарний індекс документа матиме вигляд табл. 2.

Таблиця 2

Словарний індекс документа

Позиція	Слово
0	Таким
6	чином
12	початковий
23	словарний

Зберігання тексту документу у вигляду такої структури дозволяє змінювати слова будь-яким чином, не втрачаючи оригінал, оскільки – відома координата початку нового слова і можна зіставити оригінал з видозміненим словом.

Індексація слів в HTML документі

HTML документ, крім тексту, містить багато інформації, яку необхідно відкинути перед індексацією. Документи, які конвертовані за допомогою MS Word, можуть містити службової інформації (тегів та їх атрибутів) більше, ніж самого тексту. Документи конвертовані за допомогою LibbreOffice Writer містять в собі також зображення, закодовані за допомогою BASE64 (MS Word зберігає зображення у окремі файли). Тому необхідно зробити очищення документу від тегів, але при цьому зберегти позиції слів на своїх місцях. Це питання вирішується досить просто, шляхом заміни всіх тегів на еквівалентну кількість пробілів. Таким чином, всі слова залишаються на своїх місцях (позиція початку слова відносно початку файлу документа). Вилучити слова з документа, в якому звичайний текст оточений лише пробілами – досить просто і очевидно.

Висновок

В статті було розглянуто три найпопулярніші формати збереження документів: DOC, DOCX і PDF та можливі способи їх конвертації у базовий формат для подальшої роботи з вмістом. Базовим форматом було вибрано HTML, оскільки він підтримує збереження та форматування усіх необхідних сутностей і має просту структуру, що спрощує його оброблення.

Для оброблення текстових даних запропоновано створювати словарний індекс документу, що являє собою таблицю, в першій колонці якої позиції слів у тексті, а в другій – власне слова. Така структура дозволяє опрацьовувати документ послівно та застосовувати до окремих слів будь-які модифікації, зберігаючи при цьому можливість зіставляти модифікований зразок з оригіналом.

Список літератури

1. Колесніков О.Є. Основні аспекти впровадження дистанційної освіти / О.Є. Колесніков, В.Д. Гогунський // Інформаційні технології в освіті, науці та виробництві. – Херсон-Одесса: Вип.1 (1) – 2012 – С.34-41.

2. Білощицький А.О. Ефективність методів пошуку збігів у текстах / А.О.Білощицький, О.В. Діхтяренко // Управління розвитком складних систем. – К.: КНУБА Вип. 14. – С. 144-147.

3. Высоцкий, В.Ю. Поисковые алгоритмы для автоматизированного обучения [Текст] / В.Ю. Высоцкий, В.Д. Гогунский // Інформаційні технології в освіті, науці та виробництві. – № 3(4), 2013. – С. 105-113.

4. Зеленков Ю.Г. Сравнительный анализ методов определения нечетких дубликатов для Web-документов [Електронний ресурс] / Ю.Г. Зеленков, И.В. Сегалович. – Режим доступу: http://download.yandex.ru/company/download /paper_65_v1.pdf.

5. Толчеев В.О. Анализ проблемы и разработка процедуры выявления нечетких дубликатов научных статей по библиографическим описаниям [Текст] / В.О. Толчеев. – изд. "Новые технологии", "Информационные технологии", 2011. № 2 (174). – С.17-21.

6. Буй Д.Б. Scopus та інші наукометричні бази: прості питання та нечіткі відповіді / Д.Б. Буй, А.О. Білощицький, В.Д. Гогунский // Вища школа. Наук.-практ. видання – Вип. 4 (118) / 2014 – С. 27-40.

7. Бурков В.Н. Параметры цитируемости научных публикаций в наукометрических базах данных / В.Н. Бурков, А.А. Белощицький, В.Д. Гогунский// Зб. наук. праць: Управління розвитком складних систем. – К.: КНУБА, 2013. – Вип. 15. – С. 134-139.

References

1. Kolesnikov, O. Ye., Gogunsky, V. D. (2012). Basic Aspects of Distance Education. – Kherson-Odessa, 1 (1), 34-41.

2. Biloshchytskyi, A., Dikhtyarenko, O. (2013). Effectiveness of methods to search for matches in the texts. Management of complex systems. Kyiv, Ukraine: KNUCA, 14, 144-147.

3. Vysotsky, V. Y. Gogunsky, V. D. (2013). Search algorithms for computer-aided instruction. Information technology in education, science and industry, 3 (4), 105-113.

4. Zelenkov, J. G., Segalovich I. V. Comparative analysis of duplicate detection methods for Web-documents [E resource]. – Mode of access: http://download.yandex.ru/company/download /paper_65_v1.pdf.

5. Tolcheev, V. O. (2011). Analysis PROBLEMS pazpabotka and near-duplicate detection window procedure of scientific articles on bibliogpaficheskim descriptions. "New technologies", "Infopmatsionnye technology", 2 (174), 17-21.

6. Buy, D. B., Biloshchytskyi, A. O., Gogunsky, V. D. (2014). Scopus and other scientometric database: simple questions and vague answers. Vyshcha shkola. Naukovo-praktychne vydannya, 4 (118), 27-40.

7. Burkov, V. N., Beloshchytskyi, A. O., Gogunsky, V. D. (2013). Options citation of scientific publications in scientometric databases. Management of complex systems. Kyiv, Ukraine: KNUCA, 15, 134-139.

Стаття надійшла до редколегії 12.05.2014

Рецензент: д-р техн. наук, проф. С.Д. Бушуєв, Київський національний університет будівництва і архітектури, Київ.