Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 20Б.08-мм

Салтыков Павел Константинович

Реализация проверок правил оформления в валидаторе текстов ВКР

Отчёт по учебной практике

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2022

Оглавление

Введение

Выпускная квалификационная работа является завершающим этапом обучения в вузе. При её написании студенту необходимо соблюдать ряд основных требований к оформлению ВКР, а научному руководителю нужно контролировать выполнение этих требований. Обычно руководителю приходится несколько раз проверять работы своих студентов, поэтому такой процесс может занимать немало времени и сил. Однако автоматизация проверки соответствия требованиям оформления позволит немного освободить студентов и научных руководителей от этой рутины.

С целью снижения объёма работы по проверке текстов было реализовано веб-приложение[1] под названием “Mundane Assignment Police”, которое позволяет проверять работы студентов на наличие распространённых ошибок оформления. На данный момент ведётся командная работа над этим проектом.

В рамках этой работы предлагается усовершенствование валидатора текстов ВКР, посредством реализации проверок дополнительных правил оформления.

[1]Исходный код проекта: https://github.com/Darderion/map (дата обращения: 06.06.2022)

1. Постановка задачи

Целью данной работы является расширение функциональности валидатора текстов выпускных квалификационных работ, путём добавления новых правил оформления.

Для выполнения этой цели были поставлены следующие задачи:

1. сделать обзор валидатора ВКР;
2. реализовать проверку корректности оформления цитирования;
3. реализовать проверку отсутствия недопустимых символов в названиях секций;
4. реализовать проверку размеров секций;
5. реализовать проверку наличия пробелов вокруг скобок;
6. реализовать детектор сокращённых ссылок;
7. провести анализ работы добавленных проверок.

2. Обзор

Mundane Assignment Police — валидатор текстов ВКР, веб-приложение, которое обрабатывает загруженный PDF-файл с текстом выпускной квалификационной работы и отображает отчёт об ошибках, обнаруженных в оформлении работы.

Используемые технологии:

- Spring[2] используется в роли каркаса приложения.
- Kotlin[3] используется в качестве языка программирования для серверной части.
- Vue.js[4] используется в роли каркаса веб-приложения
- TypeScript[5] используется в качестве языка программирования для клиентской части.
- PDFBox[6] — библиотека, используемая для чтения PDF-файлов и выделения ошибок в тексте работы.

Ранее в приложении уже были реализованы проверки для следующих ошибок:

- неправильное использование дефиса;
- неправильное использование короткого тире;
- неправильное использование длинного тире;
- некорректные ссылки на литературу;
- один подраздел в разделе;
- нумерация введения, заключения и списка литературы.

[2] https://spring.io/ (дата обращения 07.06.2022)
[3] https://kotlinlang.org/ (дата обращения 07.06.2022)
[4] https://vuejs.org/ (дата обращения 07.06.2022)
[5] https://www.typescriptlang.org/ (дата обращения 07.06.2022)
[6] https://pdfbox.apache.org/ (дата обращения 07.06.2022)

Также стоит отметить, что приложение определяет структуру документа, что позволяет указывать область действия для правил. Определяются следующие области документа:

- титульный лист;
- оглавление;
- содержание;
- сноска;
- список литературы.

3. Описание реализации проверок

3.1. Оформление цитирования

Зачастую в Википедии ссылки на литературу ставят после предложения, а не у ключевых слов. Согласно требованиям к оформлению выпускной квалификационной работы, ссылки на литературу следует указывать непосредственно у главного слова, а не после точки.

Библиографические ссылки в тексте научной работы указываются в квадратных скобках. Поэтому для данного правила была реализована проверка, основанная на анализе соседнего слева символа от открывающей квадратной скобки, при этом пробелы игнорируются. Таким образом, если слева находится точка, то это считается ошибкой.

3.2. Недопустимые символы в названиях секций

В соответствии с требованиями недопустимо использовать символы “:”, “.”, “,” в названиях секций.

Для этого правила рассматриваются строки оглавления. Поскольку одним из запрещённых символов является точка, то для корректной проверки данного правила сначала необходимо из каждой строки оглавления удалить точки, которые не относятся к названиям разделов. Такими являются точки в нумерации разделов и отточие — точки, разделённые пробелом и находящиеся между названием раздела и номером страницы. И только потом проверяется отсутствие недопустимых символов в строках. При обнаружении таких символов выделяется строка в оглавлении с названием секции.

3.3. Размеры секций

Проведя анализ размеров секций нескольких бакалаврских и магистерских работ студентов математико-механического факультета СПбГУ, были выдвинуты следующие требования к размерам разделов:

- Введение и заключение должны быть строго меньше четырёх листов. Если размер составляет 3 листа, необходимо выдавать предупреждение.
- Список литературы не должен превышать половины работы.
- Любой раздел не должен занимать больше половины работы.

Как уже было упомянуто ранее, разработка проекта ведётся в команде. Задачей одного из участников команды была реализация сбора статистики, которая включает в себя подсчёт различных слов в тексте и определение размеров секций.

Автором данной работой была использована информация о размерах секций и реализована проверка секций на соответствие требованиям к размерам.

Правило является настраиваемым, поэтому в дальнейшем будет возможно использование требований, которые указывались бы пользователем веб-приложения перед проверкой выпускной квалификационной работы.

3.4. Пробелы вокруг скобок

Также нарушением требований к оформлению ВКР является отсутствие пробелов с внешней стороны скобок разных видов.

Для этого правила была реализована проверка, которая анализирует соседний со скобкой символ. Если скобка открывающая, проверяется символ слева от неё, игнорируя другие открывающие скобки. Если скобка является закрывающей, то рассматривается символ, находящийся справа от неё, при чём игнорируются не только другие закрывающие скобки, но и знаки препинания. Кроме того, была добавлена обработка особого случая, когда круглые скобки пустые (например, после названий функций). В этой ситуации отсутствие пробелов вокруг них не считается за ошибку.

3.5. Детектор сокращённых ссылок

Ещё одним нарушением является использование ссылок, полученных с помощью инструментов для сокращения ссылок. Это влечёт за собой ряд проблем. Например, по таким ссылкам нельзя понять, на какой сайт они ведут. Кроме того, сокращённые ссылки через некоторое время могут стать недействительными.

Прежде всего было реализовано обнаружение ссылок в тексте, основанное на том, что ссылки не содержат пробелов, а также начинаются с протокола (“http://” либо “https://”) или поддомена “www.”.

Был создан список доменных имён популярных на данный момент URL-сокращателей. Если доменное имя проверяемой ссылки находилось в этом списке, то данная ссылка считалась сокращённой. У такого подхода есть недостаток: невозможно собрать полный список, так как постоянно появляются новые домены, используемые URL-сокращателями. Поэтому была реализована дополнительная проверка, которая основана на трёх следующих признаках сокращённых ссылок [4]:

- длина доменного имени зачастую не больше пяти символов, не учитывая точку;
- путь к ресурсу в URL-адресе состоит из одной части, которая содержит только цифры и буквы, и длина которой часто менее десяти символов;
- используется перенаправление на веб-страницу с другим URL-адресом.

Для проверки последнего с помощью класса HttpURLConnection [2] из стандартной библиотеки Java был реализован алгоритм расширения ссылки [1]. Далее необходимо сравнить доменные имена полученной ссылки и исходной.

Таким образом, сокращённой ссылкой будет считаться такая ссылка, домен которой содержится в списке URL-сокращателей, или та, которая имеет три свойства сокращённой ссылки. В последнем случае выдаётся предупреждение.

4. Тестирование

Тестирование проводилось на наборе[7] курсовых, бакалаврских, и магистерских работ студентов Математико-механического факультета СПбГУ. Приложением было обработано 19 имеющихся работ.

4.1. Метрики

Для оценки работы алгоритмов были подсчитаны такие метрики как точность (Precision) и полнота (Recall) [3]. Для описания метрик введены следующие определения:

- Истинно положительный результат (*tp*, true positive) — верно определённая ошибка.
- Ложно положительный результат (*fp*, false positive) — ложно определённая ошибка.
- Ложно отрицательный результат (*fn*, false negative) — необнаруженная ошибка.

Таким образом, точность и полнота определяются формулами:

$$Precision = \frac{tp}{tp + fp},$$

$$Recall = \frac{tp}{tp + fn}.$$

4.2. Результаты

Результаты работы алгоритмов представлены в таблице 1.

4.3. Выводы

Все нарушения по размерам секций были верно обнаружены. В каждой из четырёх работ выдавалось предупреждение о том, что введение

[7]Коллекция тестовых работ: https://github.com/Darderion/map-dataset (дата обращения: 06.06.2022)

Тип проверяемой ошибки	*tp*	*fp*	*fn*	Precision	Recall
Оформление цитирования	2	0	0	1.00	1.00
Недопустимые символы в названиях секций	15	0	0	1.00	1.00
Размер секции не соответствует требованиям	4	0	0	1.00	1.00
Отсутствие пробелов вокруг скобок	89	266	0	0.25	1.00
Сокращённая ссылка	0	0	0	—	—

Таблица 1: Результаты работы алгоритмов.

занимает три страницы.

Поскольку отсутствие пробелов вокруг скобок в формулах и фрагментах кода не является ошибкой, такие нарушения считались ложными. В дальнейшем, чтобы увеличить точность выявления таких ошибок, необходимо убрать проверку этих областей документа, которые могут быть определены с помощью машинного обучения.

Так как имеющиеся работы не содержали сокращённых ссылок, алгоритм был протестирован на дополнительной работе[8]. В результате, была верно определена сокращённая ссылка в сноске на 15 странице.

[8] https://se.math.spbu.ru/thesis/texts/Vlasov_Il%27ja_Maksimovich_Bachelor_Report_2021_text.pdf (дата обращения: 07.06.2022)

Заключение

В ходе данной работы были выполнены следующие задачи:

1. сделан обзор валидатора ВКР;
2. реализована проверка корректности оформления цитирования;
3. реализована проверка наличия пробелов вокруг скобок;
4. реализована проверка размеров секций;
5. реализована проверка отсутствия недопустимых символов в названиях секций;
6. реализован детектор сокращенных ссылок;
7. проведён анализ работы добавленных проверок.

Исходный код доступен на GitHub[9].

[9]Исходный код (изменения под именем пользователя PavelSaltykov): https://github.com/Darderion/map (дата обращения: 06.06.2022)

Список литературы

[1] API Documentation — Unshorten.It! — URL: https://unshorten.it/api/documentation (online; accessed: 2022-06-07).

[2] HttpURLConnection (Java Platform SE 8) — Oracle Help Center. — URL: https://docs.oracle.com/javase/8/docs/api/java/net/HttpURLConnection.html (online; accessed: 2022-06-07).

[3] Wikipedia contributors. Precision and recall — Wikipedia, The Free Encyclopedia. — 2022. — URL: https://en.wikipedia.org/wiki/Precision_and_recall (online; accessed: 2022-06-07).

[4] Wikipedia contributors. URL shortening — Wikipedia, The Free Encyclopedia. — 2022. — URL: https://en.wikipedia.org/wiki/URL_shortening (online; accessed: 2022-06-07).