Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б08-мм

Тинарский Алексей Сергеевич

Расширение системы правил валидатора текстов выпускных квалификационных работ

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ассистент кафедры ИАС Чернышев Г.А.

Санкт-Петербург
2022

Оглавление

Введение

Выпусная квалификационная работа является одним из важных итогов обучения студента в вузе. Сложности при её написании могут возникнуть не только с содержательной частью, но и со следованием формальным правилам оформления. Проверка их соблюдения — весьма трудозатратный процесс как для студентов, так и для научных руководителей. Эту рутинную деятельность можно частично автоматизировать, при этом повысив качество контроля.

Для решения задачи автоматизации обработки ВКР и создано вебприложение Mundane Assignment Police. Оно отображает пользователю отчёт об ошибках, обнаруженных в оформлении ВКР, загруженной в формате PDF. В данный момент продолжается разработка приложения.

В приложении присутствует набор правил для проверки работ. Развитие проекта, в том числе, происходит за счет увеличения количества распознаваемых ошибок в оформлении ВКР. В рамках этой работы предлагается расширение системы правил Mundane Assignment Police, что позволит улучшить детализацию и качество проверки текстов выпускных квалификационных работ.

1 Постановка задачи

Целью работы является расширение системы правил веб-приложения для проверки PDF-файлов с текстами выпускных квалификационных работ.

Для её выполнения были поставлены следующие задачи:

1. разработать и реализовать алгоритм, проверяющий, что все целые числа от одного до девяти написаны словами;
2. разработать и реализовать алгоритм, проверяющий корректность использования открывающей и закрывающей кавычки;
3. разработать и реализовать алгоритм, проверяющий порядок следования разделов;
4. разработать и реализовать алгоритм, проверяющий порядок следования ссылок на литературу, перечисленных подряд;
5. разработать и реализовать алгоритм, проверяющий одинаковые сокращения на совпадение регистра символов;
6. разработать и реализовать алгоритм, проверяющий ссылки на потенциальную низкокачественность материалов;
7. провести анализ результатов выполненных алгоритмов на базе имеющихся работ.

2 Обзор

В данном разделе рассматриваются работы, решающие проблему автоматизированного контроля и оценки текстов.

2.1 Автоматизированная оценка текстов

Задачи автоматизированной оценки текста, такие как автоматическая оценка читаемости и сложности, являются важными приложениями обработки естественного языка. В связи с этим появилось большое количество работ для анализа и проверки текстов.

Одной из систем, решающих данные задачи, является фреймворк с открытым исходным кодом EXPATS [2]. Данная платформа позволяет пользователям разрабатывать и экспериментировать с различными моделями автоматической оценки текста, предлагая набор расширяемых компонентов, систему конфигурации и интерфейс командной строки. Также фреймворк поддерживает визуализацию моделей и их предсказаний.

2.2 Автоматизация контроля учебно-научных текстов

Помимо общей оценки качества учебно-научного текста, важной его характеристикой является соответствие выдвинутым стандартам. Некоторые узкоспециализированные системы автоматизированного контроля текстов на русском языке описаны в работе [3].

Одной из таких систем, ориентированных на автоматический контроль научнотехнических документов является ЛИНАР [5]. Данная система анализирует текст пакетом программ, каждая из которых выполняет один из видов контроля, например, контроль орфографии, проверка структуризации текста или анализ его лексического состава. Однако в составе ЛИНАР нет средств контроля правильности оформления учебно-научных текстов.

Другим примером системы, предназначенной не только для проверки, но и редактирования учебно-научных текстов, является КОНУТ [1]. Это текстовый редактор со встроенной проверкой формальных требований, например, наличие в тексте обязательных разделов и правильный порядок их расположения, а также возможностью оценки стиля и композиции текста.

Ещё одна система, выполняющая схожие задачи — Гамма [4] — разрабатывалась как подсистема для текстового редактора MS Word. Гамма находит типичные ошибки в оформлении учебно-научных текстов, такие как использование аббревиатур без их расшифровки, ссылки на отсутствующие в документе таблицы и другие.

2.3 Приложение Mundane Assignment Police

Среди найденных работ со схожей тематикой не было обнаружено прямого аналога MAP — приложения, обрабатывающее ВКР в формате PDF.

Приложение MAP состоит из двух основных частей — RESTful API сервиса, занимающегося обработкой ВКР в виде PDF файлов, и веб-приложения, обращающегося к нему.

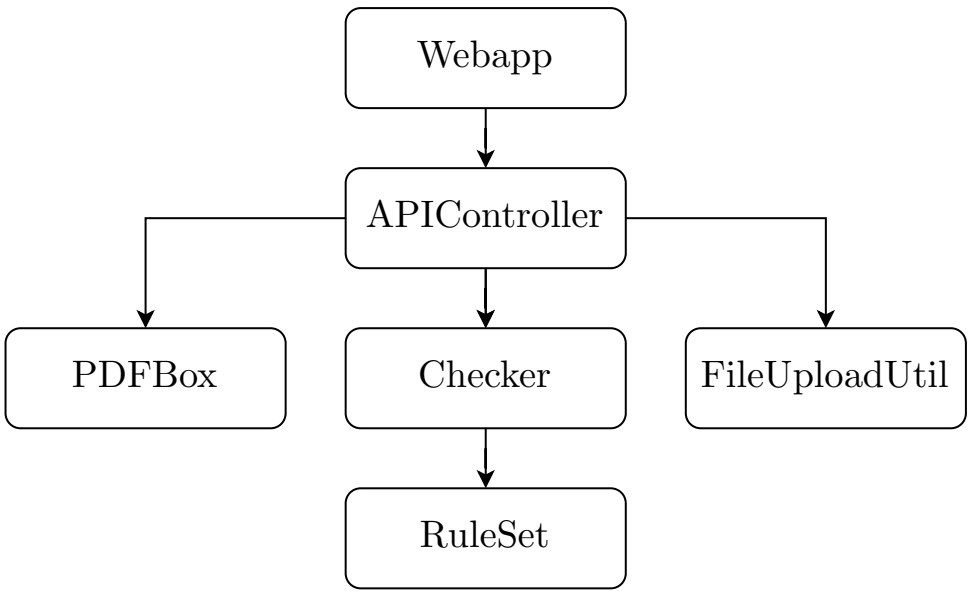

Рис. 1: Архитектура приложения MAP

На рисунке 1 изображены следующие элементы архитектуры приложения МАР:

- Webapp — веб-приложение, предоставляющее пользователю интерфейс для взаимодействия с системой по проверке ВКР;
- APIController — контроллер, обрабатывающий запросы веб-приложения;
- PDFBox — библиотека с открытым исходным кодом для работы с PDF-файлами;
- Checker — класс, отвечающий за проверку текста, извлеченного из PDF-документа, в соответствии с правилами из RuleSet;
- FileUploadUtil — утилитный класс, предоставляющий методы для сохранения, удаления и получения файлов;
- RuleSet — класс, содержащий список правил, в которых и описаны алгоритмы, валидирующие текст ВКР; именно в этот список были добавлены новые правила в рамках данной работы.

3 Метод

В этом разделе описывается процесс реализации алгоритмов правил.

3.1 Проверка написания целых чисел от одного до девяти словами

В первую очередь, были рассмотрены ситуации, в которых числа от одного до девяти пишутся цифрами. Это случаи, когда число:

- находится слева или справа от разрешенного слова (например, «рисунок», «таблица», «Гб»);
- является частью другого числа или десятичной дроби;
- является номером пункта списка;
- является номером ссылки на литературу.

Во всех остальных ситуациях было принято решение предупреждать пользователя о возможной ошибке. Для обработки вышеперечисленных пунктов было разработано несколько подправил, скомбинированных далее в итоговое правило.

Алгоритм не проверяет номера страниц, оглавление и список литературы. Программа оповещает пользователя об ошибке, если найдено целое число от одного до девяти, записанное цифрой, для которого не выполняется ни одно из условий:

- рядом с числом слева или справа находится слово из списка разрешенных (пробелы и точка игнорируются);
- рядом с числом слева или справа находится число, запятая или точка;
- данное число является первым символом в строке;

- слева от числа находится символ «[», при этом пробелы, запятые и слова (непрерывные последовательности из букв и цифр) игнорируются.

3.2 Проверка корректности использования открывающей и закрывающей кавычки

Цитата должна начинаться с открывающей кавычки «“», а заканчиваться закрывающей «”». Для проверки выполнения этого условия был разработан алгоритм правила, действующего во всех разделах ВКР.

При проверке все символы, кроме открывающей и закрывающей кавычки, игнорируются. Пользователь получает предупреждение о возможной ошибке, если:

- слева от закрывающей кавычки находится закрывающая кавычка;
- справа от открывающей кавычки находится открывающая кавычка;
- слева (справа) от закрывающей (открывающей) кавычки нет ни одной кавычки на ближайших 20 строках.

3.3 Проверка порядка следования разделов

Для секций был определен набор правил, следование которым обязательно для соблюдения корректности порядка разделов:

- введению не может предшествовать ни одна секция;
- постановке задачи может предшествовать только введение;
- обзору может предшествовать только постановка задачи;
- контенту (любой не названной в этом списке секции) может предшествовать обзор или контент;
- заключению может предшествовать только контент;

- списку литературы может предшествовать только заключение.

Для проверки соблюдения вышеперечисленных условий был разработан алгоритм правила, который действует в рамках оглавления. Для каждой секции рассматривается предшествующая ей. Если для данной пары разделов (текущий и предшествующий ему) нарушается одно из правил, перечисленных выше, то секция классифицируется как не следующая корректному порядку, и пользователь получает оповещение об ошибке.

3.4 Проверка порядка следования перечисленных подряд ссылок на литературу

При перечислении нескольких номеров ссылок на литературу в одних квадратных скобках нужно упорядочить их по возрастанию. Для проверки соблюдения этого правила был разработан алгоритм, действующий во всех разделах ВКР.

В первую очередь, для разработки алгоритма понадобилось создать новый тип правил, работающих с регулярными выражениями. Данные правила способны находить в тексте работы последовательности символов, соответствующие указанному регулярному выражению, а затем проверять, удовлетворяют ли они определенному предикату.

Алгоритм, основанный на вышеописанном типе правил, находит в тексте работы все последовательности чисел, пробелов и запятых в квадратных скобках при помощи соответствующего регулярного выражения. Для каждой такой последовательности создается массив целых чисел, в котором сохраняется исходный порядок. Если этот массив не упорядочен по возрастанию, то данная последовательность классифицируется как ошибочная и пользователь получает предупреждение об этом.

3.5 Проверка одинаковых сокращений на совпадение регистра символов

Одно из правил оформления ВКР заключается в следующем: для одного сокращения во всей работе следует использовать одинаковые по регистру символов аббревиатуры. Для проверки этого критерия на основе типа правил, работающих с регулярными выражениями, был разработан алгоритм, проверяющий все разделы, кроме списка литературы.

Разработанное правило находит все непрерывные последовательности из букв (слова). Аббревиатурой назовём слово, в котором есть хотя бы одна буква в верхнем регистре (первая игнорируется). Считается, что данное слово не соответствует вышеописанному правилу, если выполняются следующие два условия:

- хотя бы в одном варианте написания, встречающемся в тексте, это слово является аббревиатурой;
- в тексте встречаются хотя бы два различных по регистру символов написания данного слова.

3.6 Проверка ссылок на потенциальную низкокачественность материалов

Прежде всего был создан утилитный класс для получения и парсинга веб-страницы со списком ссылок на низкокачественные конференции[1]. После загрузки страницы класс кэширует данные на 10 часов и предоставляет пользователю готовый список. По истечении этого периода обращение к веб-ресурсу осуществляется вновь и список актуализируется.

Далее был разработан алгоритм правила, который сравнивает ссылки из списка литературы со всеми предоставленными вышеописанным

[1]https://beallslist.net — страница со списком ссылок на потенциально низкокачественные конференции, используемая в работе (дата обращения: 24.09.2022).

утилитным классом. Если ссылка из текста работы содержит один из адресов веб-ресурсов с низкокачественными конференциями, пользователь получает предупреждение об этом.

4 Эксперимент

Апробация проводилась на базе данных курсовых, магистерских и бакалаврских работ[2] Математико-механического факультета СПбГУ. Из них приложение обработало 21 работу со средним количеством ошибок (распознанных алгоритмами из данной работы) в 10.95 и медианным значением в одну ошибку (таблица 1).

Таблица 1: Апробация

Тип нарушения	Среднее	Медиана	Суммарное
Целые числа от одного до девяти написаны цифрами	36.48	28	766
Некорректное использование открывающей и закрывающей кавычки	3.43	0	72
Некорректный порядок следования разделов	1.33	0	28
Нарушение порядка следования ссылок на литературу, перечисленных подряд	0.52	0	11
Несовпадение одинаковых сокращений по регистру символов	23.95	13	503
Использование ссылок на низкокачественные конференции	0	0	0
Все нарушения, распознанные алгоритмами из данной работы	10.95	1	1380

Для оценки количества ложных срабатываний были выбраны три случайные работы. В них вручную было подсчитано количество ложных срабатываний для описанных в данной работе правил (таблица 2).

Среди полученных результатов апробации выделяется статистика по первому и последнему типу нарушений. Алгоритм правила, работающий с целыми числами от одного до девяти, имеет ложные срабатывания в таблицах, так как функциональность по их обнаружению в

[2]https://github.com/Darderion/map-dataset — репозиторий с базой выпускных квалификационных работ на Github (дата обращения: 30.09.2022).

Таблица 2: Количество ложных срабатываний

Тип нарушения	Количество найденных нарушений	Количество ложных срабатываний	Процент ложных срабатываний
Целые числа от одного до девяти написаны цифрами	89	34	38.2%
Некорректное использование открывающей и закрывающей кавычки	30	0	0%
Некорректный порядок следования разделов	3	1	33.3%
Нарушение порядка следования ссылок на литературу, перечисленных подряд	2	0	0%
Несовпадение одинаковых сокращений по регистру символов	37	5	13.5%
Использование ссылок на низкокачественные конференции	0	0	0%
Все нарушения, распознанные алгоритмами из данной работы	161	40	24.8%

приложении ещё не реализована. Для дополнительной проверки корректности алгоритма, выявляющего использование в работе ссылок на низкокачественные конференции, был проведен ручной разбор списка литературы пяти случайных ВКР. В них не было обнаружено низкокачественных источников.

Заключение

В ходе работы были выполнены следующие задачи:

1. разработан и реализован алгоритм, проверяющий, что все целые числа от одного до девяти написаны словами;
2. разработан и реализован алгоритм, проверяющий корректность использования открывающей и закрывающей кавычки;
3. разработан и реализован алгоритм, проверяющий порядок следования разделов;
4. разработан и реализован алгоритм, проверяющий порядок следования ссылок на литературу, перечисленных подряд;
5. разработан и реализован алгоритм, проверяющий одинаковые сокращения на совпадение регистра символов;
6. разработан и реализован алгоритм, проверяющий ссылки на потенциальную низкокачественность материалов;
7. проведён анализ результатов выполненных алгоритмов на базе имеющихся работ.

Открытый код проекта доступен по ссылке на репозиторий[3] Github.

Планируется дальнейшее расширение функционала веб-приложения за счёт увеличения количества проверяемых правил.

[3]https://github.com/tinarsky/map — репозиторий проекта на Github (дата обращения: 24.09.2022).

Список литературы

[1] Bolshakova E. Computer Assistance in Writing Technical and Scientific Texts // Proceedings of 2nd International Symposium “Las Humanidades en la Educación Técnica ante el Siglo XXI”, México. — 2000. — P. 59–63.

[2] Manabe Hitoshi, Hagiwara Masato. EXPATS: A Toolkit for Explainable Automated Text Scoring. — 2021. — URL: https://arxiv.org/abs/2104.03364 (online; accessed: 2022-10-15).

[3] Баева Н. В., Большакова Е. И. Проблемы автоматизации контроля учебно-научных текстов // Сборник научных трудов SWorld. — 2012. — Vol. 6. — P. 59–63. — URL: https://sworld.education/konfer27/381.pdf (online; accessed: 2022-10-15).

[4] Кирсанова А. В. Программные средства контроля учебно-научных текстов // Сборник тезисов лучших дипломных работ. — 2005. — P. 87–88.

[5] Мальковский М. Г., Большакова Е. И. Интеллектуальная система контроля качества научно-технического текста // Интеллектуальные системы. — 1997. — Vol. 2, no. 1-4. — P. 149–155. — URL: https://elibrary.ru/item.asp?id=42896186 (online; accessed: 2022-10-15).