Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Нафикова Лиана Ирековна

Расширение функциональности валидатора текстов выпускных квалификационных работ

Учебная практика

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2023

Оглавление

Введение

Во время обучения студенты сталкиваются с написанием различных научных работ. У студентов часто недостаточно опыта, поэтому они не могут с первого раза написать грамотный текст. Допускается достаточное количество стилистических и грамматических ошибок. Поэтому необходимо создание такой системы, которая могла бы автоматизировать проверку ВКР и облегчить работу студентам и преподавателям. С помощью веб-приложения Mundane Assignment Police, реализованного на языке Kotlin, осуществляется проверка выпускных квалификационных работ на наличие наиболее распространённых ошибок. В него загружается текст курсовой или дипломной работы в формате PDF, он обрабатывается. Затем студент или преподаватель может ознакомиться с найденными ошибками, которые будут подчёркнуты. В приложении уже сейчас реализовано множество проверяемых правил. Mundane Assignment Police находится в стадии разработки, происходит расширение функционала. В данной работе предлагается добавление новых правил проверки текста.

1. Постановка задачи

Целью работы является расширение функционала веб-приложения, проверяющего выпускные квалификационные работы студентов Математико-механического факультета СПбГУ на наличие ошибок оформления.

Для её выполнения были поставлены следующие задачи:

1. реализовать правило, запрещающее использование пробелов перед точкой или запятой;
2. реализовать правило, срабатывающее в случае, если после запятой нет пробела;
3. выдавать пользователю предупреждение, если в написанном предложении больше 30 слов;
4. реализовать правило, запрещающее нумерацию секций с нуля;
5. реализовать правило, срабатывающее в случае использования слов: «theorem», «definition», «lemma»;
6. реализовать правило, срабатывающее в случае неправильного использования аббревиатуры «вуз»;
7. реализовать правило, срабатывающее в работах, где есть эксперименты, в случае если нет описания используемых конфигураций.

2. Обзор

В данном разделе рассматриваются возможные аналоги и организация проверки правил.

Проблема автоматизации контроля учебных текстов была рассмотрена в работе [2]. Существуют три системы анализа и контроля текста: ЛИНАР, КОНУТ и Гамма. ЛИНАР [3] была настроена на орфографическую, грамматическую и стилистическую проверки текста, каждая из которых была организована отдельно, некоторые даже предлагали варианты исправления найденных ошибок. КОНУТ [1] и Гамма же предназначаются именно для обработки научных текстов. В них была добавлена формальная проверка. Например, проверка правильности разбиения на секции, анализ ссылок в литературе.

3. Высокоуроневое описание системы работы с PDF-файлами

Работа студента обладает вполне понятной структурой [4]. У неё есть: титульная страница, оглавление, введение, цели и задачи, обзор, решение, заключение и список литературы. В каждом из разделов, студенты могут допускать ошибки, поэтому правила разрабатываются так, чтобы их можно было проверить в любом из выбранных разделов. Правила в приложении реализуются в зависимости от уровня проверки текста: например, можно проверять отдельные символы или слова. Так, в приложении уже реализованы следующие шаблоны:

1. правила, проверяющие списки;
2. правила, проверяющие слова;
3. правила, проверяющие содержание;
4. правила, проверяющие символы;
5. правила, проверяющие ссылки.

Каждый шаблон создаётся с помощью билдера. Затем через созданное правило прогоняются соответствующие блоки текста, которые проверяются на соответствие шаблону. В случае, если строка, блок строк не удовлетворяют шаблону, то они подчёркиваются с пометкой соответствующей ошибки.

4. Реализация

В данном разделе представляется описание реализации добавленных правил.

4.1. Символьные правила

Реализация правил, обрабатывающих отдельные символы, схожа. Нарушение обоих правил («Используется пробел перед пунктуационным знаком», «Отсутствует пробел после запятой») проверяется лишь внутри секций. Проверяется соседний символ от запятой или точки слева или справа. Для первого правила необходимо, чтобы соседний слева от пунктуационного знака символ не был пробелом или символом перевода строки. Для второго — наоборот, соседний справа символ должен быть пробелом или символом перехода на новую строку.

4.2. Словарные правила

Словарные правила работают по аналогии с символьными: для каждого слова проверяется соседнее. Если соседнее слово «theorem», «definition» или «lemma», то подчёркивается вся строка. Для правила, реализуещего проверку, является ли соседнее слово неправильной формой слова «вуз», было составлено регулярное выражение, которое позволяет осуществлять проверку написания всех падежных форм слова через дефис и без него.

4.3. Строковые Правила

Нумерация секций с нуля проверяется с помощью шаблона правила, работающего с содержанием. В каждой строке списка содержания, проверяется, нет ли подстроки вида «.0.». Для первых двух символов также смотрится, не равны ли они соответственно «0.»

В проекте до этого не было классов, обрабатывающих отдельные предложения. Поэтому понадобилось реализовать Builder и класс правил предложения, а также алгоритм, разбивающий строки на отдельные блоки. Внут-

ри каждого такого блока строк содержится одно-два целых предложения. Конец предложения определяется по наличию пунктуационных знаков в строке. Для реализации конкретного правила, проверяющего длину предложения, также осуществляется проверка каждого отдельного слова в блоке. Если в предложении присутствует цитирование, то слова, находящиеся внутри цитаты, не входят в подсчёт. Чтобы подчёркнутые строки корректно отображались, предлагается блоки строк одного предложения с разных страниц отображать как два отдельных нарушения.

Для реализации правила, проверяющего конфигурацию в экспериментах, понадобилось также собрать отдельный словарь с словами, содержащими описанием конфигураций, и реализовать отдельные конструирующие классы. Добавлена возможность выставления фильтра для строк, обработка как одной строки, так и нескольких. В частности организована фильтрация строк по принадлежности к секции экспериментов. Часто в главу с экспериментами студенты включают несколько секций. Поэтому, чтобы включить всю главу в подсчёт, осуществлялся отдельно поиск страниц экспериментов и заключения. Далее общее количество страниц складывалось как все страницы между найденной первой страницей экспериментов и соответственно первой страницей заключения. Правило считается нарушеным, если в главе, посвящённой экспериментам, менее пяти слов, описывающих конфигурацию используемого устройства.

5. Эксперименты

Тестирование проводится на базе 19 работ[1] Математико-механического факультета СПбГУ.

5.1. Символьные правила

Отсутствие пробела чаще всего встречается внутри скобок между перечисляемыми объектами. Например, (i,j) или [10,20]. Лишний же пробел встречается как ложное срабатывание перед расширениями файлов: .txt. Также довольно часто встречающийся случай — многоточие внутри интервала. Например, (1, ..., n).

5.2. Словарные правила и нумерация секций с нуля

К сожалению, в данной базе не было работ, в которых встречаются нумерация секций с нуля, неправильное написание аббревиатуры «вуз» или слова «theorem», «definition», «lemma». Поэтому данные правила были протестированы на тестовых экземлярах. Проверялись отдельные случаи написания слов в начале, конце и середине строки в разных регистрах. Для правила написания «вуз» было проверено написание в разных падежах, с дефисом и без. Для нумерации секций с нуля были проверены случаи списка, начинающегося с нуля, списка, содержащего подсписок, начинающийся с нуля.

5.3. Длинные предложения

Ложные срабатывания в правиле поиска длинного предложения связаны с отсутствием модуля, обрабатывающего таблицы и схемы, а также с неправильным оформлением списков. После списка студенты не всегда ставят точку, из-за чего два предложения могут «склеиться» в одно. Результаты проверки данного правила отражены в таблице 1. Каждая строка посвящена отдельной работе из базы. В первом столбце указан тип работы,

[1] https://github.com/Darderion/map-dataset — репозиторий с базой выпускных квалификационных работ на Github (дата обращения: 19.10.2022).

Тип работы	Ошибки	Ложные срабатывания
Курсовые	4	0
	3	0
	9	3
	3	0
	8	2
	4	0
	9	0
	30	1
Бакалаврские	7	0
	6	0
	15	2
	19	3
	5	0
	12	0
Магистерские	14	0
	7	0
	47	0
	23	2
	22	5

Таблица 1: Длинные предложения

во втором отражено количество найденных ошибок, в третьем — ложные срабатывания в конкретной работе.

5.4. Конфигурация устройства в экспериментах

Проверка данного правила описана в таблице 2. Иногда студенты пишут данные для экспериментов вне главы с экспериментами, что может повлиять на поиск нужных страниц секции. В частности так произошло с одной из бакалаврских работ: данные экспериментов были описаны в обзоре.

Тип работы	Секция экспериментов	Наличие конфигурации	Ошибки
Бакалаврские	Нет		
	Нет		
	Есть	Есть	Нет нарушения
	Есть	Есть	Нет нарушения
	Есть	Нет	Есть нарушение
	Есть	Нет	Нет нарушения
Курсовые	Нет		
	Нет		
	Есть	Нет	Есть нарушение
	Нет		
	Есть	Нет	Есть нарушение
	Нет		
	Нет		
	Нет		
Магистерские	Есть	Есть	Нет нарушения
	Нет		
	Есть	Есть	Нет нарушения
	Есть	Нет	Есть нарушение
	Нет		

Таблица 2: Конфигурация в экспериментах

Заключение

Были выполнены следующие задачи:

1. реализовано правило, запрещающее использование пробелов перед точкой или запятой;
2. реализовано правило, срабатывающее в случае, если после запятой нет пробела;
3. реализовано правило, при срабатывании которого пользователю выдаётся предупреждение, если в написанном предложении больше 30 слов;
4. реализовано правило, запрещающее нумерацию секций с нуля;
5. реализовано правило, срабатывающее в случае использования слов: theorem, definition, lemma;
6. реализовано правило, срабатывающее в случае неправильного использования аббревиатуры «вуз»;
7. реализовано правило, срабатывающее в работах, где есть эксперименты, в случае если нет описания используемых конфигураций.

Открытый код проекта доступен по ссылке на репозиторий[2] Github.

[2]https://github.com/Liana2707/map — репозиторий проекта на Github (дата обращения: 15.10.2022).

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] Баева НВ and Большакова ЕИ. Проблемы автоматизации контроля учебно-научных текстов // Сборник научных трудов SWorld. Материалы международной научнопрактической конференции «Перспективные инновации в науке, образовании, производстве и транспорте‘2012». — 2012. — P. 59–62.

[3] Мальковский МГ and Большакова ЕИ. Интеллектуальная система контроля качества научно-технического текста // Интеллектуальные системы. — 1997. — Vol. 2, no. 1-4. — P. 149–155.

[4] Хмельницкая ЕВ. Методические указания по оформлению выпускной квалификационной работы. — 2018.