Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 22.Б08-мм

Расширение базы правил и улучшение точности их проверки в валидаторе текстов ВКР

Басиев Георгий Давидович

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ассистент кафедры ИАС Чернышев Г.А.

Санкт-Петербург
2024

Оглавление

Введение

Во время обучения в высших учебных заведениях студенты сталкиваются с необходимостью написания курсовых и выпускных квалификационных работ. В силу отсутствия опыта они часто допускают различные ошибки в оформлении работы и синтаксисе, на рутинное выявление которых уходит большое количество времени. Для автоматизации этого процесса с помощью языка Kotlin было создано веб-приложение Mundane Assignment Police[1]. Пользователь может загрузить в него PDF-файл своей работы и, в случае выявления ошибок, они будут выделены в тексте красной линией. Приложение находится в стадии разработки, в связи с чем в нем отстутствует часть необходимых правил, а существующие не всегда корректно проверяются. В рамках данной работы предлагается минимизация ложных срабатываний и расширение набора правил.

[1] https://github.com/darderion/map

1. Постановка задачи

Целью работы является расширение и улучшение функционала веб-приложения для проверки текстов курсовых и выпускных квалификационных работ.

Для её выполнения были поставлены следующие задачи:

1. повысить точность работы алгоритма, проверяющего, что все целые числа от одного до девяти написаны словами;
2. повысить точность работы алгоритма, проверяющего ссылки на единообразие;
3. повысить точность работы алгоритма, проверяющего аббревиатуры на единообразие;
4. реализовать алгоритм, проверяющий отсутствие в тексте использования двух пробелов подряд;
5. реализовать алгоритм, проверяющий количество скобок, используемых в одном предложении;
6. реализовать алгоритм, проверяющий отношение количества слов, заключенных в скобки, к общему количеству слов в предложении;
7. провести апробацию реализованных и исправленных алгоритмов на базе текстов имеющихся работ;

2. Обзор

В данном разделе рассматриваются существующие решения для автоматической проверки текстов.

2.1. Автоматизированная проверка текстов в других предметных областях

Автоматизированная проверка текста находит применение в разных предметных областях. Основными потребителями сервисов, предоставляющих подобную функциональность, являются люди, постоянно работающие с текстом: журналисты, редакторы, копирайтеры. Для задач, возникающих в этих областях, существует большое количество различных решений [2]. Они в основном нацелены на улучшение читаемости текста и повышение его художественной ценности.

Также актуальным вопросом является разработка автоматической системы проверки медицинских отчетов [1]. Важной задачей в этой области является обнаружение некорректных сокращений и противоречий в тексте.

2.2. Автоматизированная проверка текстов научных работ

В целях автоматической проверки научных текстов были разработаны две системы: ЛИНАР и КОНУТ [3]. ЛИНАР, в основном, концентрировалась на выявлении орфографических и стилистических ошибок. КОНУТ же обладал расширенной, по сравнению с ЛИНАР, функциональностью, и позволял выявлять ошибки в оформлении работы.

2.3. Различия

В связи с отсутствием открытого доступа к перечисленным раннее проектам, сравнить их эффективность по сравнению с MAP не представляется возможным. Очевидной является недостаточность их функ-

циональности для решения поставленной задачи, так как они не позволяют расширить набор правил для проверки конкретных требований по оформлению, выдвигаемых студентам Математико-механического факультета.

3. Общее описание построения правил

Для реализации правил в приложении были разработаны различные шаблоны, на основе которых можно создавать конкретные правила. Шаблоны различаются в зависимости от масштаба проверки. В данный момент в проекте используется восемь различных классов-шаблонов:

1. правила, проверяющие символы;
2. правила, проверяющие слова;
3. правила, проверяющие списки;
4. правила, проверяющие предложения;
5. правила, проверяющие ссылки;
6. правила, проверяющие строки;
7. правила, основанные на поиске с помощью регулярных выражений;
8. правила, проверяющие секции;

Область действия каждого правила можно ограничить определенной частью текста. Например, исключить титульный лист из проверки, или не рассматривать слова, являющиеся первыми в своей строке и так далее. Также часть правил допускает добавление набора исключительных случаев, при которых правило не считается нарушенным.

4. Исправленные правила

В данном разделе представляется работа, проделанная для повышения точности проверки уже существовавших правил.

4.1. Проверка написания целых чисел от одного до девяти словами

Для реализации правила, проверяющего написание целых чисел от одного до девяти словами, было написано несколько подправил. Каждое из них обрабатывает набор исключений, при которых написание числа в виде цифры является допустимым. В первую очередь, это такие случаи, как «рисунок/таблица 1», «7 Мб» и так далее. Однако, в ходе исследования было обнаружено, что большое количество случаев не было рассмотрено. Подправила также пропускали исключения, если часть слова, представляющего собой исключение, переносилась вместе с числом на другую строку. Для решения этой проблемы было реализовано дополнительное подправило. Помимо исправления перечисленных недочетов, была добавлена обработка арифметических символов и операторов сравнения, чтобы избежать определения математических формул в качестве нарушения правила.

4.2. Проверка одинаковых сокращений на единообразие

В ходе исследования было обнаружено, что правило, назначением которого является обработка случаев написания в тексте аббревиатур в разном виде (например, «ВУЗ» и «Вуз»), часто выделяет в качестве ошибки разные формы написания предлогов и союзов («по» — «По», «или» — «Или»), в связи с использованием в тексте «ИЛИ» в псевдокоде, либо аббревиатуры «ПО». Для решения этой проблемы в правило была добавлена возможность обработки исключительных случаев, представленных в отдельном словаре.

4.3. Проверка единообразия ссылок

Данное правило используется для обнаружения несоответствия ссылок единому стилю Подразумевается, что ссылки должны начинаться либо с «https»/«http», либо с «www». Предыдущая реализация допускала нарушение в обработке текста ссылок, пропуская одно из условий. По сути, ссылки, начинавшиеся с «http» являлись недопустимыми по умолчанию, несмотря на отсутствие поддержки протокола «HTTPS» для многих сайтов. Для исправления этой ошибки был изменен фильтр, обрабатывающий текст ссылки.

5. Реализованные правила

В данном разделе представляется работа, проделанная для расширения базы правил веб-приложения.

5.1. Проверка отсутствия двух символов пробела подряд

Для реализации правила, проверяющего отсутствие в тексте использования двух символов пробела подряд, был создан экземпляр класса символьного правила. Он предназначен для создания правил, осуществляющих проверку на основе символа. Все строки в тексте, в которых встречаются два пробела подряд, подсвечиваются красным.

5.2. Проверка количества скобок в предложении

Использование избыточного количества скобок в тексте курсовой или выпускной квалификационной работы является нерекомендуемым. Для того, чтобы отслеживать подобные ошибки, было реализовано правило на основе класса-шаблона, предназначенного для отслеживания неправильно построенных предложений. Внутри реализованного правила строки текста разбиваются на предложения, для каждого из которых подсчитывается количество скобок. В случае, если оно превышает лимит, правило считается нарушенным.

5.3. Проверка количества текста внутри скобок

Помимо общего числа скобок, необходимо также отслеживать, какое количество слов от общего числа в предложении содержится в скобках. Для этого было реализовано правило на основе того же класса-шаблона для предложений, распространяющееся на весь файл, кроме библиографии. Внутри правила строки текста также разбиваются на предложения, с помощью регулярных выражений отбираются все вхождения подстрок, заключенных в скобки. Затем количество слов в найденных

вхождениях соотносится с общим количеством слов в предложении. Если отношение превышает предельное значение, правило также считается нарушенным.

6. Эксперимент

Апробация проводилась на основе курсовых работ[2] студентов Математико-механического факультета СПбГУ.

6.1. Изменения в работе исправленных правил

Для оценки количества ложных срабатываний для каждого из правил было выбрано по пять работ. Для них в ручную было подсчитано количество ложных срабатываний в изначальной реализации и с учетом внесенных изменений. Результаты представлены в таблице 1.

Таблица 1: Ложные срабатывания до и после изменений

Правило	**Количество ложных срабатываний до изменений**	**Количество ложных срабатываний после изменений**	**Коэффициент уменьшения**
Написание целых чисел от одного до девяти	128	38	3,37
Единообразие сокращений	67	5	13,4
Единообразие ссылок	27	5	5,4

В случае с правилом, контролирующим единообразие ссылок, проблемой являлась не только генерация ложных срабатываний, но и пропуск некоторых нарушений. Благодаря внесенным изменениям, ее также удалось решить.

В связи с отсутствием функциональности, необходимой для правильной обработки таблиц в тексте курсовых работ, проверка написания целых чисел все еще генерирует большое количество ложных срабатываний.

[2]https://github.com/Darderion/map-dataset — репозиторий с базой курсовых работ на GitHub (дата обращения: 29.10.2023).

6.2. Тестирование новых правил

В использованной для остальных тестов базе курсовых работ не было найдено примеров использования двух символов пробела подряд и достаточного количества скобок. Для апробации соответствующих правил пришлось прибегнуть к тестовым файлам. Для остальных правил было также использовано несколько работ студентов.

Таблица 2: Апробация новых правил

Правило	**Количество найденных нарушений**	**Количество ложных срабатываний**	**Процент ложных срабатываний**
Два пробела подряд	7	0	0%
Количество скобок в предложении	23	9	39%
Объем текста внутри скобок	89	11	12%

Добавленные правила также генерируют ложные срабатывания в связи с отсутствием возможности определения в тексте курсовых и выпускных квалификационных работ таблиц, диаграмм и фрагментов программного кода.

Заключение

В ходе данной работы были выполнены следующие задачи:

1. повышена точность работы алгоритма, проверяющего, что все целые числа от одного до девяти написаны словами;
2. повышена точность работы алгоритма, проверяющего ссылки на единообразие;
3. повышена точность работы алгоритма, проверяющего аббревиатуры на единообразие;
4. реализован алгоритм, проверяющий отсутствие в тексте использования двух пробелов подряд;
5. реализован алгоритм, проверяющий количество скобок, используемых в одном предложении;
6. реализован алгоритм, проверяющий отношение количества слов, заключенных в скобки, к общему количеству слов в предложении;
7. проведена апробация реализованных и исправленных алгоритмов на базе текстов имеющихся работ;

В дальнейшем планируется расширение базы правил и разработка функциональности, позволяющей выявлять в тексте таблицы и диаграммы. Исходный код доступен на GitHub[3].

[3]Исходный код (изменения под именем пользователя GeorgeBasiev): https://github.com/Darderion/map (дата обращения: 23.12.2023)

Список литературы

[1] Pogrebnoi Dmitrii, Funkner Anastasia, Kovalchuk Sergey. RuMedSpellchecker: Correcting Spelling Errors for Natural Russian Language in Electronic Health Records Using Machine Learning Techniques // Computational Science – ICCS 2023: 23rd International Conference, Prague, Czech Republic, July 3–5, 2023, Proceedings, Part III. — Berlin, Heidelberg : Springer-Verlag, 2023. — P. 213–227. — URL: https://doi.org/10.1007/978-3-031-36024-4_16.

[2] Sorokin A., Shavrina T. Automated spelling correction for russian social media texts // Computational Linguistics and Intellectual Technologies (Dialogue 2016). — 2016. — URL: https://www.dialog-21.ru/media/3429/sorokinaashavrinato.pdf.

[3] Мальковский М. Г., Большакова Е. И. Интеллектуальная система контроля качества научно-технического текста // Интеллектуальные системы. — 1997. — URL: https://elibrary.ru/item.asp?id=42896186.