На правах рукописи

Чувилин Кирилл Владимирович

Автоматический синтез правил коррекции текстовых документов формата LaTeX

05.13.17 – Теоретические основы информатики

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Москва – 2013

Работа выполнена на кафедре «Интеллектуальные системы» факультета управления и прикладной математики Федерального государственного образовательного учреждения высшего профессионального образования «Московский физико–технический институт (государственный университет)».

Научный руководитель:

доктор физико–математических наук **Воронцов Константин Вячеславович**.

Официальные оппоненты:

Ульянов Михаил Васильевич, доктор технических наук, профессор,
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский государственный университет печати имени Ивана Федорова», профессор кафедры.

Гуров Сергей Исаевич, кандидат физико–математических наук, доцент, с. н. с.,
Обособленное подразделение факультет вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова, доцент.

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт системного анализа Российской академии наук.

Защита состоится «12» декабря 2013 г. в 14:00 часов на заседании диссертационного совета Д 002.017.02 при Федеральном государственном бюджетном учреждении «Вычислительный центр им. А. А. Дородницына Российской академии наук», расположенном по адресу: 119333, г. Москва, ул. Вавилова, 40.

С диссертацией можно ознакомиться в библиотеке ВЦ РАН.

Автореферат разослан «9» ноября 2013 г.

Ученый секретарь
диссертационного совета,
д. ф.–м. н. Рязанов В. В.

Общая характеристика работы

Актуальность темы. В связи с ростом числа электронных научных изданий постоянно увеличивается число издательств, редакционно-издательских отделов вузов и научных учреждений, индивидуальных авторов, использующих систему компьютерной верстки LaTeX. LaTeX является стандартом де-факто для научного общения и публикаций. Постоянно растет доля электронных изданий, к которым предъявляются повышенные требования оперативности публикаций.

При этом уровень подготовки пользователей в области компьютерной верстки, знания типографических правил и традиций остается невысоким. К таким правилам относятся оформление заголовков, списков, таблиц, библиографии, формул, чисел, и многое другое. Ошибки, связанные с несоблюдением этих правил, называются типографическими. При текущем уровне технологий исправление таких ошибок производится корректорами вручную, что требует значительных затрат времени. Большинство ошибок являются типовыми, что создает предпосылки для автоматизации процесса корректуры.

Автоматизация стадии корректуры при подготовке научных изданий позволила бы существенно сократить затраты и сроки и повысить качество верстки. В данной работе эта задача ставится как задача автоматической обработки текста и решается методами машинного обучения. Такой подход к проблеме автоматизации корректуры до сих пор не применялся.

Степень разработанности темы. Существуют инструменты для облегчения процесса ручной корректуры (André, Richy, 1999), но, тем не менее, обработка одной страницы занимает до двух часов. Вообще говоря, идея автоматизации коррекции текстов не нова (Большаков, 1988), и на данный момент существуют качественные инструменты для автоматического поиска и исправления орфографических ошибок[1], использующие словари и морфологический анализ словоформ текста. Кроме того, схожая проблема возникает для интеллектуальной коррекции ошибок в запросах поиска (Панина и др., 2013), с помощью лексических и статистических признаков. Но подобные подходы не применимы для исправления типографических ошибок, рассматриваемых в данной работе, которые связаны не только с текстовым содержанием документа, но и разметкой форматирования, и зачастую для описания ошибки не достаточно локальной информации в тексте, но также требуется знание контекста, дополнительной информации о позиции в структуре документа.

С другой стороны, существует область исследований, посвященная улучшению характе-

[1] http://extensions.services.openoffice.org/project/lightproof

ристик исходного кода программ (вероятности возникновения ошибок в отдельных модулях, степени связности модулей и др.). Известны методы (Williams, Hollingsworth, 2005; Князев, 2007), позволяющие оценивать характеристики, основываясь на анализе истории изменений репозиториев, и использовать их для поиска ошибок в коде. Они позволяют создавать рекомендательные системы (Madou et al., 2011) для улучшения качества кода программы при редактировании. Документы в формате LaTeX можно рассматривать как исходный код, который используется компилятором TeX, но в издательской практике не распространено использование репозиториев, пригодных для последующего анализа, нет единых стандартов, и, кроме того, текстовое содержимое документов не может быть подвержено подобной обработке.

Таким образом, возникает необходимость нового исследования, направленного непосредственно на автоматизацию процесса исправления типографических ошибок.

Цели и задачи исследования. Объектом исследования являются хорошо структурированные текстовые документы, которые могут быть описаны с помощью синтаксического дерева. Предмет исследования — алгоритмы автоматического синтеза правил коррекции структурированных тестовых документов по выборке пар «черновик–чистовик».

Целью диссертационного исследования является разработка методов, алгоритмов и технологий для создания автоматизированной системы, позволяющей многократно повысить эффективность труда корректоров при работе с текстовыми документами формата LaTeX.

Для достижения цели исследования в диссертации решаются следующие задачи.

1. Разработка эффективных алгоритмов для представления и сравнения файлов в формате LaTeX как древовидных структур данных.

2. Формализация описания правил коррекции типографических ошибок и разработка эффективных алгоритмов поиска мест ошибок в документах и синтеза правил для их исправления. Множество проблем вызваны тем, что при ручной обработке документов корректоры придерживаются недостаточно формализованных рекомендаций. И составление вручную достаточно полного описания набора правил для автоматического использования трудно реализуемо. Некоторые из используемых рекомендаций довольно сложны и сильно зависят от контекста, что требует сложных моделей для описания правил коррекции.

3. Задача автоматического синтеза правил коррекции текстовых документов формата LaTeX заключается в построении совокупности формальных инструкций, которые могут быть

использованы в алгоритмах локализации ошибок (определение фрагментов текста, содержащих ошибки) и исправления ошибки (построение ранжированного списка вариантов замены фрагмента текста, содержащего ошибку).

4. Разработка методики оценивания синтезированных правил коррекции для последующего ранжирования. Это необходимо при выборе наиболее подходящих вариантов найденной ошибки для предоставления их пользователю.

5. Экспериментальное исследование полноты и точности разработанных алгоритмов сравнения документов и построения правил коррекции с использованием корпуса реальных статей.

Научная новизна. В работе впервые предложен подход к синтезу правил коррекции текстовых документов по обучающей выборке, составленной из пар документов «черновик–чистовик». Задача автоматизации корректуры текстовых документов никогда ранее не ставилась как задача синтеза правил коррекции методами машинного обучения.

В работе предложен новый гибридный алгоритм для выявления различий между структурированными (обладающими синтаксическим деревом) текстовыми документами, который корректно учитывает логическую структуру текстов, но при этом, как минимум, в три раза быстрее алгоритма, основанного на сравнении только синтаксических деревьев.

Теоретическая и практическая значимость. Теоретическая ценность работы заключается в том, что предложены подход для синтеза правил автоматической коррекции по обучающей выборке, составленной из пар документов «черновик–чистовик», и методика оценки качества таких правил. Кроме того, разработан эффективный алгоритм сравнения синтаксических деревьев документов в формате LaTeX.

Практическая ценность результатов диссертации заключается в том, что разработанные методы, алгоритмы и технологии позволяют реализовать систему автоматизации корректуры, в несколько раз сокращающую трудозатраты при коррекции текстовых документов формата LaTeX. При этом автоматизируются процессы поиска различий между структурированными документами, поиска возможных типографических ошибок, синтеза правил коррекции, формирования наборов вариантов исправления.

Предлагаемый подход. В данной работе предлагается формально описывать правила автоматической коррекции. Для этого каждый документ в формате LaTeX отождествляется с синтаксическим деревом, для которого и формулируются правила [1].

Обучающая выборка составляется из пар документов: черновик (документ, не прошедший обработку профессиональным корректором) и чистовик (документ, содержащий корректорские правки). Для сравнения синтаксических деревьев используется гибридный алгоритм, который учитывает и текстовую природу документов LaTeX, и их древовидную структуру [2]. В результате работы алгоритма строится отображение вершин синтаксического дерева черновика в вершины дерева чистовика.

Построенное отображение используется для синтеза правил, из которых каждое характеризуется шаблоном (линейным или древовидным), применяющимся к вершинам синтаксического дерева. На основе предварительных оценок точности строятся групповые правила [6].

Для оптимизации построенного набора правил коррекции и последующего их ранжирования строятся оценки качества на основе статистики применимости правил к документам обучающей выборки [4].

Результаты, выносимые на защиту.

1. Алгоритм сравнения структурированных текстов, использующий их представление в виде синтаксических деревьев (на примере текстов формата LaTeX).

2. Алгоритмы построения линейных, древовидных и групповых правил коррекции документов по обучающей выборке пар документов «черновик–чистовик», позволившие достичь точности 76% и полноты 69% на коллекции из 85 пар документов.

3. Программа для построения набора правил коррекции документов и эмпирического оценивания полноты и точности построенного набора.

Достоверность результатов. Обоснованность и достоверность результатов и выводов подтверждена:

- сравнением реализованных алгоритмов и подходов с аналогами;
- опытом практического применения результатов исследования на реальных коллекциях текстовых документов;
- обсуждением результатов исследования на российских и международных научных конференциях;
- публикациями результатов исследования в рецензируемых научных изданиях, в том числе рекомендованных ВАК РФ.

Апробация результатов исследования. Основные результаты диссертационного исследования докладывались на следующих конференциях:

- 54-я научная конференция Московского физико-технического института (Долгопрудный, 2011 г.),
- Международная научная конференция студентов, аспирантов и молодых учёных «Ломоносов-2012» (Москва, 2012 г.),
- Вторая научная конференция молодых ученых «Теория и практика системного анализа» ТПСА-2012 (Рыбинск, 2012 г.),
- Девятая международная конференция «Интеллектуализация обработки информации» ИОИ-2012 (Черногория, Будва, 2012 г.),
- 55-я научная конференция Московского физико-технического института (Долгопрудный, 2012 г.),
- 16-я всероссийская конференция с международным участием «Математические методы распознавания образов — 2013» ММРО-16 (Казань, 2013 г.).

В рамках работы над диссертацией был реализован прототип системы полуавтоматической коррекции типографических ошибок. Проект «Самообучающаяся система для автоматизации коррекции документов в формате LaTeX» прошел отборочные этапы программы «Участник молодежного научно-инновационного конкурса» («У.М.Н.И.К.») и вошел в число победителей конкурса в 2012 году[2].

Основные результаты работы опубликованы в [3–6], в том числе в изданиях [1, 2], входящих в список ВАК.

Структура и объем диссертации. Диссертация состоит из введения, 4 глав основного содержания, заключения, библиографии и 4 приложений. Работа содержит 127 страниц основного текста, включая 24 иллюстрации. Перечень библиографических источников включает 70 наименований.

[2] http://miptic.ru/UMNIK/a_5lekjo.html

Содержание работы

Во Введении обоснована актуальность темы диссертационной работы, сформулирована цель и аргументирована научная новизна исследования, показана практическая значимость полученных результатов, представлены выносимые на защиту научные положения.

В первой главе приводится постановка задачи, обзор литературы по тематике задачи и структура предлагаемого в диссертации подхода.

В разделе 1.1 на примерах дается представление о типографических ошибках, которые встречаются в документах формата LaTeX.

Рассматривается постановка задачи автоматического синтеза правил коррекции текстовых документов формата LaTeX как задачи обучения по прецедентам.

Под правилом коррекции подразумевается формально описанная инструкция, которая может быть использована алгоритмом для:

- локализации ошибки в документе формата LaTeX(определение фрагмента исходного текста, содержащего ошибку),
- предложения варианта исправления (построения текста для замены фрагмента с ошибкой).

Пусть X — множество пар документов: черновик (документ, не прошедший обработку профессиональным корректором) и чистовик (документ, содержащий корректорские правки). R — множество правил коррекции документов. Дана обучающая выборка $X^m = \{x_1, \ldots, x_m\}$ из m пар документов. Требуется построить набор правил

$$R^n(X^m) = \{r_1, \ldots, r_n\} \subset R \times \ldots \times R$$

коррекции документов, который бы обладал наилучшими оценками полноты и точности.

В разделе 1.2 дается обзор методов и решений в областях исследований, смежных с рассматриваемой в диссертации задачей:

- автоматизация коррекции текстов,
- автоматический поиск и исправление орфографических ошибок,
- интеллектуальная коррекция ошибок в запросах поиска,
- улучшение характеристик исходного кода программ (вероятности возникновения ошибок в отдельных модулях, степени связности модулей и др.).

В разделе 1.3 приводится структура предлагаемого в диссертации подхода к решению задачи синтеза правил коррекции. Она состоит из четырех этапов.

На первом этапе строится синтаксическое дерево для каждого используемого документа формата LaTeX. В дальнейшем синтезируемые правила формулируются именно для деревьев.

На втором этапе выделяются различия между синтаксическими деревьями документов в каждой паре «черновик–чистовик» и строится отображение вершин первого дерева во второе.

На третьем этапе синтезируются правила коррекции, каждое из которых изменяет одну из вершин синтаксического дерева.

На четвертом этапе происходит построение групповых правил коррекции, которые образуются из построенных на предыдущем этапе правил и способны изменять несколько вершин синтаксического дерева.

Детально каждый из этапов описан в последующих главах диссертации.

Вторая глава посвящена формальному описанию структуры документов формата LaTeX.

В разделе 2.1 дается представление о TeX и LaTeX.

TeX представляет собой систему правил разметки текста и одновременно их обработчик — компилятор. Он был разработан американским математиком и программистом Дональдом Кнутом для верстки текстов с формулами. Он позволяет разделить физическое и логическое форматирование. LaTeXявляется наиболее распространенным раширением TeXа.

В разделе 2.2 описываются элементы разметки форматирования документа формата LaTeX.

Каждый документ LaTeX должен начинаться с команды, в которой указывается используемый класс — шаблон оформления:

```
\documentclass[<необязательные параметры>]{<имя класса>}
```

Далее могут идти команды для подключения дополнительных файлов со стилями, выбора настроек и т. п. Но весь текст документа с разметкой форматирования заключен в окружение `document`:

```
\begin{document}
    ...
\end{document}
```

В данной диссертации исследуется содержимое именно этого окружения и только его.

Каждая позиция в тексте документа может определяться набором состояний, в работе выделяются следующие: математическая формула (в противном случае — обычный текст), список, изображение, таблица, вертикальный режим (в противном случае — горизонтальный).

Весь исходный текст документа формата LaTeX состоит из элементов трех типов: символ, команда, окружение.

Символы являются минимальными элементами конструкции документа формата LaTeX. Каждый символ описывается шаблоном — фрагментом текста, который соответствует символу в коде документа. Некоторые символы могут обладать меткой конца, в этом случае считается, что символ ограничивает (или «включает в себя») другой код документа.

Команды представляют собой еще одни элементы конструкции документов LaTeX, которые могут использовать аргументы. Каждая команда определяется именем и шаблоном параметров. Имя команды состоит из знака \, непрерывной конечной последовательности латинских букв и может заканчиваться символом *. Шаблон параметров описывает сигнатуру размещения аргументов команды в исходном коде, которые всегда указываются сразу после имени.

Каждое окружение описывается именем, которое состоит из непрерывной конечной последовательности латинских букв и может заканчиваться символом *. В исходном коде окружения описываются с помощью вспомогательных команд `\begin` (начало окружения) и `\end` (конец окружения).

Каждый элемент исходного текста документа формата LaTeX обладает типом лексемы — логическим и функциональным значением. Один и тот же элемент может обладать разными типами лексем для различных состояний обработчика TeX. Также типом лексемы могут обладать параметры команд. Выделяются следующие типы: binaryOperator (бинарный математический оператор), brackets (скобки), cellbreak (конец ячейки), char (символ), command (команда), digit (цифра), equation (формула), floatingBox (плавающий бокс), hskip (горизонтальный отступ), image (изображение), index (верхний или нижний индекс), item (элемент списка), label (метка), length (линейное измерение), letter (буква слова), linebreak (обрыв строки), list (список), par (новый абзац), path (путь к файлу или папке), postOperator (математический постоператор), preOperator (математический преоператор), raw (необрабатываемые данные), space (пробел), table (таблица), tableParams (параметры таблицы), tag (тэг), vskip (вертикальный отступ), wrapper (обертка).

В разделе 2.3 описывается рассматриваемая в диссертации древовидная структура до-

кументов формата LaTeX. Файлы формата LaTeX, используемые при подготовке научных издательств (книг и сборников трудов), как правило, обладают естественной древовидной структурой (синтаксическим деревом), исследуя которую, можно получить всю необходимую информацию для описания корректорской правки. Узлы этой структуры будем называть токенами. Корнем является окружение `document`. Выделяются следующие типы токенов: тело окружения LaTeX, команда LaTeX, окружение LaTeX, метка, линейный размер, число, разделитель абзацев, путь к файлу, пробел, символ, параметры таблицы, слово, не распознаваемая последовательность символов (например, для окружения `verbatim`). Синтаксическое дерево взаимно однозначно определяет документ LaTeX.

Третья глава посвящена эффективному алгоритму сравнения текстовых документов, представимых в виде синтаксического дерева.

В работе [3] для построения различий между синтаксическими деревьями используется алгоритм, основанный на алгоритме Zhang–Shasha. Однако практический опыт позволил выявить следующие недостатки его применения. Во-первых, возникают проблемы, связанные с эффективностью: сложность алгоритма пропорциональна произведению числа ключевых корней для чернового и чистового деревьев. Это приводит к тому, что сравнение двух документов типичной длины занимает до трех минут, и становится невозможным использовать его для редактирования в режиме «онлайн». Во-вторых, существуют проблемы, связанные с потреблением памяти. Для работы алгоритма требуется хранить попарные расстояния между всеми поддеревьями чернового и чистового деревьев и соответствующими лесами. Это делает невозможным использование алгоритма для сравнения больших документов, соответствующих, например, главам книг.

С другой стороны, существуют алгоритмы сравнения текстовых файлов, избавленные от подобных недостатков. Но в этом случае возникают проблемы с качеством: полученное различие не учитывает структуру документов, и, в итоге, не соответствует логике корректора и не позволяет выявлять верные закономерности.

Поэтому предлагается гибридный алгоритм сравнения документов в формате LaTeX, использующий достоинства алгоритмов сравнения неформатированных текстов и синтаксических деревьев, и позволяющий сравнительно быстро выявлять различия, учитывающие логическую структуру, даже для больших документов.

В разделе 3.1 описываются алгоритмы построения редактирующего расстояния и отображений для линейных последовательностей элементов.

Мера различия (редактирующее расстояние) между линейными конечными последовательностями элементов, включая последовательности символов, которыми являются тексты, основано на расстоянии Левенштейна.

Определение 1 (Расстояние Левенштейна)**.** Пусть для изменения последовательности элементов разрешается применять операции трех типов: удаление элемента, вставка элемента, изменение элемента. Тогда расстоянием Левенштейна между двумя последовательностями называется минимальное количество таких операций.

Расстояние Левенштейна выражается следующими рекуррентными соотношениями:

$$\delta_L(a_1 \dots a_n a_{n+1}, b_1 \dots b_m b_{m+1}) = \min \begin{cases} \delta_L(a_1 \dots a_n, b_1 \dots b_m b_{m+1}) + \delta(a_{n+1}, \emptyset) \\ \delta_L(a_1 \dots a_n a_{n+1}, b_1 \dots b_m) + \delta(\emptyset, b_{m+1}) \\ \delta_L(a_1 \dots a_n, b_1 \dots b_m) + \delta(a_{n+1}, b_{m+1}) \end{cases},$$

где в классическом случае $\delta(a_{n+1}, \emptyset)$ (цена удаления элемента a_{n+1}), $\delta(\emptyset, b_{m+1})$ (цена вставки элемента b_{m+1}) и $\delta(a_{n+1}, b_{m+1})$ (цена изменения элемента a_{n+1} на b_{m+1} при $a_{n+1} \neq b_{m+1}$) приравниваются к 1. Но, вообще говоря, это могут быть другие неотрицательные числа, описывающие степень близости элементов.

Кроме того, в диссертации используется относительное расстояние Левенштейна — классическое расстояние Левенштейна, нормированное на длину наибольшей последовательности:

$$\delta_{RL}(a_1 \dots a_n, b_1 \dots b_m) = \frac{\delta_L(a_1 \dots a_n, b_1 \dots b_m)}{\max(n, m)},$$

которое, очевидно, может принимать значения от 0 до 1.

Алгоритмы, которые строят отображение, основанное на расстоянии Левенштейна, используют обратное отслеживание рекуррентных формул, описанных выше. Наиболее эффективным, с точки зрения количества потребляемой памяти, является алгоритм Хиршберга.

Документы в формате LaTeX обычно рассматриваются как текстовые файлы, поэтому возможно использование алгоритма Хиршберга, применимое к сравнению произвольных текстов. Естественно представлять текст как линейную последовательность символов и использовать алгоритм для сравнения таких последовательностей. Но часто оказывается (и это применимо к LaTeX-документам), что тексты содержат очень большое количество символов, и последовательности получаются чрезмерное длинными, что приводит к завышенному расходу памяти и низкой эффективности. Поэтому на практике сравниваемые тексты разбиваются

на неделимые фрагменты, обычно в местах переноса строк, и строится отображение последовательностей таких фрагментов.

В разделе 3.2 описывается алгоритм Zhang–Shasha построения редактирующего расстояния и отображений для деревьев.

Рассматриваются деревья, обладающие следующими свойствами: каждая вершина содержит ключ (элемент из заранее определенного набора), выбрана вершина, которая является корнем дерева, вершины, имеющие общего родителя, упорядочены. К дереву разрешается последовательно применять следующие операции: удаление вершины (все ее потомки переходят родителю), вставка новой вершины в произвольное место, изменение ключа вершины.

Определение 2 (Редактирующее расстояние)**.** Редактирующим расстоянием между двумя деревьями называется минимальное количество операций удаления вершины, вставки вершины и изменения ключа, позволяющих получить из первого дерева второе.

Алгоритм Zhang–Shasha алгоритм позволяет вычислять редактирующее расстояние между двумя деревьями и, кроме того, определять, какую операцию нужно применить к каждой вершине для реализации такого расстояния.

Определение 3 (Отображение деревьев)**.** Пусть заданы два дерева. Отображением первого дерева во второе называется правило, которое некоторым вершинам первого дерева взаимно однозначно сопоставляет некоторые вершины второго дерева так, чтобы порядок следования вершин сохранялся. Такие отображения принято записывать с помощью набора пар номеров вершин (прообраз, образ). Пусть отображение содержит пары (a, b) и (c, d). Тогда требуемые условия запишутся следующим образом:

$$a = c \Leftrightarrow b = d, \quad a < c \Leftrightarrow b < d.$$

Каждое такое отображение соответствует набору операций, используемых для построения редактирующего расстояния:

- если вершина первого дерева не имеет образа, то ее нужно удалить;
- если вершина второго дерева не имеет прообраза, то ее нужно вставить;
- если вершине первого дерева соответствует вершина второго с другим ключом, то нужно изменить ключ.

Таким образом, отображение, соответствующее минимальному количеству операций, реализует редактирующее расстояние.

Токену каждого типа синтаксического дерева LaTeX можно сопоставить ключ так, чтобы синтаксические деревья полностью удовлетворяли условиям применимости алгоритма Zhang–Shasha. Следующие особенности алгоритма Zhang–Shasha мешают эффективно применять его для таких деревьев:

- две матрицы расстояний для каждой пары вершин — не достаточно объема оперативной памяти персональных компьютеров для сравнения, например, глав книг;
- двойной цикл по ключевым корням — синтаксические деревья документов в формате LaTeX имеют тысячи ключевых корней, поэтому скорость алгоритма невысокая.

В разделе 3.3 для решения этих проблем предлагается гибридный алгоритм. Идея заключается в том, чтобы найти как можно больше совпадений и различий синтаксических деревьев, используя сравнение документов LaTeX, как текстов, а оставшиеся токены сравнить с помощью алгоритма Zhang–Shasha.

В первую очередь строятся последовательности фрагментов текста сравниваемых документов. Каждому токену синтаксического дерева соответствует набор последовательных символов в тексте документа. Поэтому можно говорить о границах токена: позициях начала (перед первым из этих символов) и конца (после последнего из символов). Эти позиции удобно использовать в качестве разделителей текста документа на фрагменты, поскольку они отражают логику структуры элементов LaTeX.

Затем находится отображение фрагментов текста с помощью алгоритма Хиршберга. Учитывается, что некоторые пары фрагментов могут иметь меньше различий, чем другие: в качестве цены изменения одного фрагмента текста на другой используется относительное расстояние Левенштейна для последовательностей символов, образующих эти фрагменты.

После построения отображения для каждого фрагмента текста сравниваемых документов возможны следующие случаи. Если фрагмент принадлежит первому документу и в качестве образа имеет пустое множество, то будем считать, что все его символы удаляются. Если фрагмент текста принадлежит второму документу и в качестве прообраза имеет пустое множество, то будем считать, что все его символы добавляются. Все остальные фрагменты разбиваются на пары: прообраз и образ. Если прообраз и образ совпадают, то будем считать, что каждый их символ не изменяется. Для не совпадающих образа и прообраза построим отображение символов с помощью алгоритма Хиршберга, рассматривая два этих фрагмента текста как две линейные последовательности символов.

Все символы текста разбиваются на классы (некоторые могут быть пусты), взаимно однозначно соответствующие токенам. Считается, что токен первого дерева удаляется, если удаляются все символы текста первого документа, которые ему соответствуют. Считается, что токен второго дерева добавляется, если добавляются все символы текста второго документа, которые ему соответствуют. Считается, что токен первого или второго дерева не изменяется, если не изменяются все символы текста документа, которые ему соответствуют. Если из синтаксических деревьев сравниваемых документов убрать все удаляемые, добавляемые и неизменяемые токены, останутся два дерева, состоящие из остальных токенов. Для постоения отображения этих токенов используется алгоритм Zhang–Shasha.

Четвертая глава посвящена синтезу, использованию и оценке качества правил коррекции документов формата LaTeX.

В разделе 4.1 Рассматриваются правила с линейным шаблоном. Каждое построенное правило характеризуется шаблоном (последовательностью соседних токенов с общим родителем), локализатором (токеном, к потомкам которого применяется шаблон) и действием (операцией, направленной на изменение синтаксического дерева).

Определение 4. Левая (правая) шаблонная цепочка радиуса r — это последовательность соседних токенов с общим родителем, длиной не больше r. Началом цепочки считается самый правый (левый) ее токен.

Пусть токен x чернового дерева удален или изменен на токен y. Тогда локализатор — родительский токен x, шаблон составляется из левой и правой шаблонных цепочек, наиболее близких к x и самого токена x. В таких случаях токен x будем называть целевым токеном правила. Действие правила заключается в удалении целевого токена или изменении его на токен y, в зависимости от типа правила.

Пусть в чистовое дерево добавлен токен y. Тогда локализатор — прообраз родительского токена y, если он существует; шаблон составляется из левой шаблонной цепочки, начинающейся в прообразе левого соседа y, если он существует, и аналогичной правой. Действие правила заключается в добавлении токена y между левой и правой шаблонными цепочками.

Считается, что токен l дерева соответствует локализатору правила, если выполняется совпадение типов токенов и типов их лексем.

Среди потомков l ищется непрерывная последовательность, совпадающая с шаблоном по следующим правилам:

- для всех токенов шаблонных цепочек должно выполняться совпадение типов и лексем

с соответствующими потомками l,

- для целевого токена должно выполняться полное совпадение с соответствующим потомком l.

Определение 5. Позиция правила в синтаксическом дереве документа LaTeX — это совокупность токена, соответствующего локализатору правила, и набора токенов, соответствующих шаблону. Порождающая позиция правила — позиция, которая соответствует элементу отображения синтаксических деревьев, из которого было синтезировано правило. Множество позиций или позиции правила на множестве документов — совокупность всех позиций в синтаксических деревьях этих документов, удовлетворяющих правилу.

Для предварительной оценки качества каждого правила вычисляются данные по обучающей выборке [5]. Обозначим: d_t — количество позиций правила на множестве черновиков, c_t — количество позиций правила на множестве чистовиков.

Определение 6 (Предварительная точность правила)**.** Предварительная (на обучающей выборке) точность правила — это отношение количества позиций, которые соответствуют только черновикам, к общему числу найденных позиций: $\frac{d_t - c_t}{d_t}$.

Это соответствует тому, что «идеальное» правило, обладающее точностью 1, применимо только к черновикам и не имеет позиций на чистовиках.

Набор токенов, образующих шаблон правила, можно задавать по-разному. Из результатов экспериментов [4] можно сделать вывод, что шаблоны максимальной длины не всегда дают лучший результат. В данной работе оптимальный шаблон выбирается по следующим критериям:

1. предварительная точность правила не должна быть меньше 0.9,
2. выбирается наименьший размер шаблона, позволяющий построить правило с допустимой точностью,
3. выбирается правило с наибольшей точностью из всех, обладающих шаблонами выбранного размера.

Но, как оказалось, правки, совершаемые корректорами, не всегда могут быть заданы тремя вышеописанными действиями. Например, при замене фрагмента `$(k-1)/(8L^2),$` на `$(k-1)/(8L^2)$,` должно произойти перемещение токена, соответствующего запятой, что вызовет нарушение порядка: токен, образованный запятой, является потомком токена

формулы (имеет меньший номер, чем номер токена формулы), а должен стать его правым соседом (иметь номер на 1 больше, чем токен формулы).

Для выделения подобных перемещений набор операций над деревьями был расширен операциями поднятия и опускания. В предположении, что найдено отображение некоторого дерева на другое, введены обозначения: D — множество удаленных вершин, I — множество добавленных вершин, $p(x)$ — родитель вершины x, $f(x)$ — образ вершины x (при этом $\forall x \in D\ f(x) = \emptyset$), $k(x)$ — ключ вершины x.

Определение 7. Поднятыми вершинами называются вершины $x_1, \ldots, x_k$ чернового дерева такие, что для $i = 1, \ldots, k$ выполняется:

- $x_i = x_1 + i - 1$ (последовательные),
- $p(x_i) = x_k + 1$ (являются последними потомками общего родителя).

При этом существуют вершины $y_1, \ldots, y_k$ чистового дерева такие, что для $i = 1, \ldots, k$ выполняется:

- $y_i = y_1 + i - 1$ (последовательные),
- $k(y_i) = k(x_i)$ (ключи соответствуют удаленным вершинам),
- $p(y_i) = p(y_1)$ (имеют общего родителя),
- $y_1 = f(p(x_1)) + 1$ (следуют за образом родителя $x_1, \ldots, x_k$).

Определение 8. Опущенными вершинами называются вершины $x_1, \ldots, x_k$ чернового дерева такие, что для $i = 1, \ldots, k$ выполняется:

- $x_i = x_1 + i - 1$ (последовательные),
- $p(x_i) = p(x_1)$ (имеют общего родителя).

При этом существуют вершины $y_1, \ldots, y_k$ чистового дерева такие, что для $i = 1, \ldots, k$ выполняется:

- $y_i = y_1 + i - 1$ (последовательные),
- $k(y_i) = k(x_i)$ (ключи соответствуют удаленным вершинам),
- $p(y_i) = f(x_1 - 1)$ (имеют общего родителя, являющегося образом вершины, предшествующей $x_1, \ldots, x_k$).

Для всех поднятых и опущенных вершин отображение деревьев дополняется парами (x_i, y_i), $i = 1, \ldots, k$.

В диссертации доказано следующее утверждение.

Теорема 1. Пусть есть два дерева T_1 и T_2, причем T_2 получено из T_1 с помощью операций вставки, удаления, изменения ключа, поднятия и опускания. Если $\varphi_1, \ldots, \varphi_n$ — последовательность операций с вершинами, реализующими какое-то отображение T_1 в T_2, то можно построить набор операций $\varphi'_1, \ldots, \varphi'_m$, реализующий выбранное отображение, такой, что $m \leqslant n$, и все операции поднятия и опускания выполняются после операций вставки, удаления и изменения ключа.

Смысл теоремы заключается в том, что для построения отображения деревьев, используя все пять операций, можно сначала воспользоваться алгоритмом для построения редактирующего расстояния, использующего операции вставки, удаления и изменения ключа, затем поднятые и опущенные вершины нужно искать среди удаленных (множество D), а их образы — среди добавленных (множество I).

Тем не менее, в дальнейшем были предложены более универсальные групповые правила, которые охватывают этот подход, а поиск их происходит эффективнее.

Для оценки качества набора правил был проведен эксперимент, в котором использовалось 85 пар черновых и чистовых статей конференции ИОИ-8. Моделировалось адаптивное обучение набора правил. Для этого обучающее множество пар документов, используемое для построения правил, постепенно увеличивалось: 2, 3, 4, 6, 9, 13, 19, 28, 42, 63. Обозначим через $S_1 \subset \ldots \subset S_{10}$ полученные десять обучающих множеств пар документов, S_{11} — множество всех пар документов. На каждом шаге контрольное множество формировалось из пар документов, которые добавлялись к обучающему множеству на следующем шаге: $S_{i+1} \setminus S_i$, $i = 1, \ldots, 10$.

Вычислялись оценки качества для синтезированных правил и наборов правил [5]. Последовательности множеств строились 50 раз, данные по всем построениям были усреднены.

Обозначим: d_t — количество позиций правила на множестве черновиков, c_t — количество позиций правила на множестве чистовиков.

Определение 9. Пусть $P(A_1), \ldots, P(A_k)$ — предварительные точности правил $A_1, \ldots, A_k$ соответственно, а их позиции правил таковы, что соответствуют изменению одного и того же токена. Весом правила A_i называется $W(A_i) = \frac{P(A_i)}{\sum_{j=1}^{k} P(A_j)}$.

Определение 10. Пусть $E(A_i)$ — число, равное 0, если правило A_i соответствует верной

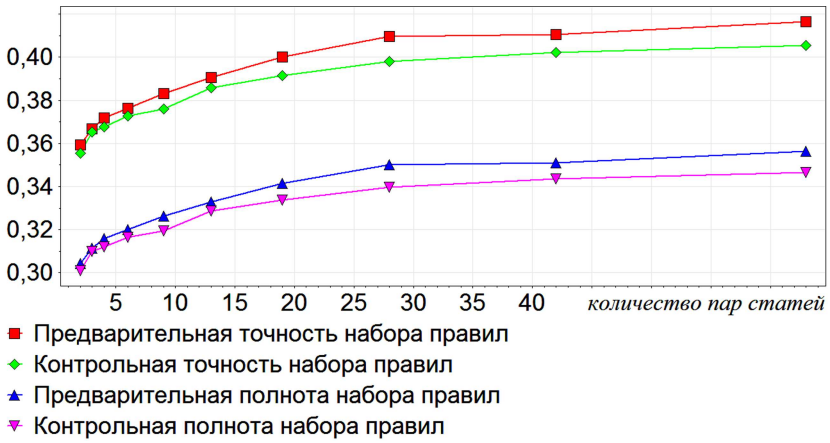

Рис. 1. Оценки точности и полноты набора правил с простой структурой.

правке, и 1 в противном случае. Тогда выражение

$$e_x = \sum_{i=1}^{k} W(A_i)E(A_i) = \frac{\sum_{i=1}^{k} P(A_i)E(A_i)}{\sum_{i=1}^{k} P(A_i)}$$

задает среднюю ошибку набора правил на выбранном токене.

Обозначим: E_t и E_c — суммы средних ошибок набора правил на всех токенах черновых деревьев обучающей и контрольной выборок соответственно, N_t и N_c — количества различных позиций всех правил набора на множествах черновиков обучающей и контрольной выборок соответственно, D_t и D_c — суммы редактирующих расстояний для всех пар черновых и чистовых деревьев обучающей и контрольной выборок соответственно.

Поскольку правила синтезируются только при добавлении, удалении или изменении токена, а сумма таких операций для двух деревьев равна редактирующему расстоянию, будут корректны следующие определения [5].

Определение 11. $\frac{N_t - E_t}{N_t}$ — предварительная (на обучающей выборке) точность набора правил. $\frac{N_c - E_c}{N_c}$ — контрольная (на контрольной выборке) точность набора правил.

Определение 12. $\frac{N_t - E_t}{D_t}$ — предварительная (на обучающей выборке) полнота набора правил. $\frac{N_c - E_c}{D_c}$ — контрольная (на контрольной выборке) полнота набора правил.

Результаты проведенных расчетов для наборов правил с простой структурой представлены на рисунке 1. Кривые, соответствующие предварительным и контрольным оценкам точности и полноты набора правил, расположены довольно близко друг другу. Это означает, что синтезированные предложенным способом правила обладают неплохой обобщающей способностью.

С другой стороны, и точность, и полнота наборов правил не превосходят 50%. Для точ-

ности это означает, что существуют различные правила со схожими шаблонами. Недостаток полноты можно объяснить тем, что рассмотренных типов правил недостаточно для описания действий корректора.

В разделе 4.2 вводится понятие групповых правил. На практике встречаются случаи, когда корректор изменяет, удаляет или добавляет более одного токена. Например, перенос одного токена на другую позицию представляет собой совокупность удаления и добавления токена. Для увеличения спектра обрабатываемых правок корректора мы будем использовать группировку правил.

Пусть для двух правил существуют позиции такие, что:

- токены, соответствующие локализаторам, совпадают;
- наборы токенов, соответствующих шаблонам, имеют общие элементы.

Тогда построим новое групповое правило, локализатор которого совпадает с локализатором рассматриваемых правил, а шаблон образуется объединением их шаблонов. Построенное правило добавляется в набор, если его предварительная точность выше, чем предварительная точность каждого из рассматриваемых правил.

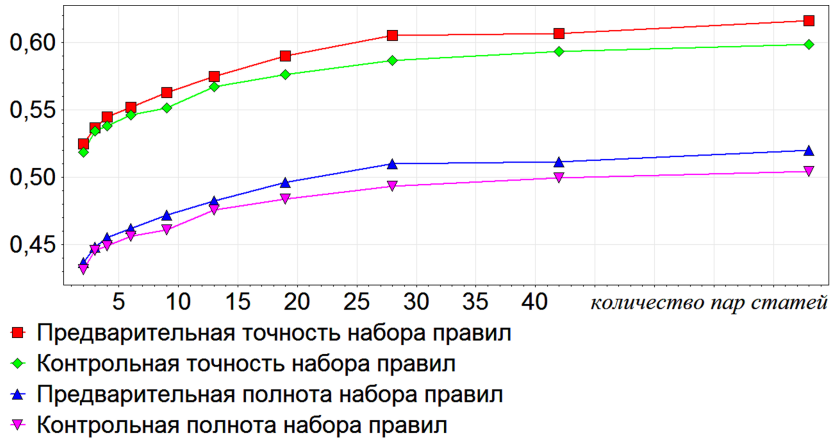

Рис. 2. Оценки точности и полноты набора правил с учетом группировки.

На рисунке 2 показаны оценки качества набора правил с учетом групповых правил, построенные в соответствии с экспериментом, описанным выше. Можно видеть, что такой подход позволил получить точность заметно больше половины, но полнота все еще находится на уровне 50%.

В разделе 4.3 описываются правила с древовидной структурой шаблона. Шаблон правила с простой структурой позволяет использовать только соседние токены для определения

позиции, что, вообще говоря, не означает использование всего текста, соответствующего этим токенам, поскольку не учитываются структура и содержимое поддеревьев, корни которых образуют шаблон.

Шаблон двевовидного правила будем строить из двух шаблонных деревьев: левого и правого. В этом случае длина шаблона — количество токенов в этих деревьях.

Шаблонные деревья и все их поддеревья проверяются на применимость с помощью проверки на каждом уровне, начиная с потомков токена l, условий:

- совпадение самых правых (для левых поддеревьев) или левых (для правых поддеревьев) токенов-потомков с токенами шаблона,
- применимость соответствующего поддерева для каждого токена-потомка.

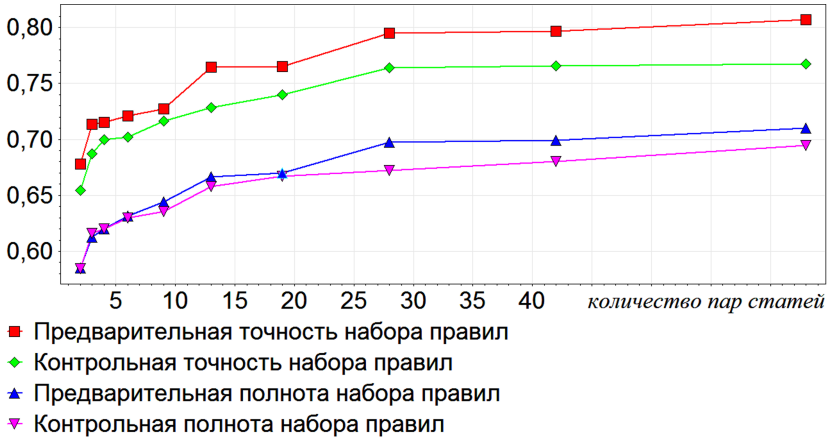

Рис. 3. Оценки точности и полноты набора правил с учетом группировки.

На рисунке 3 показаны оценки качества набора правил с учетом синтеза древовидных правил, построенные в соответствии с экспериментом, описанным выше. Можно видеть, что подобный подход позволил значительно увеличить точность синтезированного набора правил.

В Заключении перечислены основные результаты работы:

1. Впервые задача автоматической коррекции документов формата LaTeX формулируется как задача обучения по прецедентам, в которой обучающая выборка составляется из пар документов «черновик–чистовик».
2. Предложен алгоритм сравнения структурированных текстовых документов (на примере файлов формата LaTeX), использующий их представление в виде синтаксических деревьев. Алгоритм основан на выделении удаленных, добавленных и не измененных вершин деревьев с помощью сопоставления текстовых представлений документов.

3. Предложен алгоритм автоматического построения правил удаления, вставки или изменения отдельных вершин деревьев, обладающих линейными и древовидными шаблонами. Показано, что в некоторых случаях требуются правила, которые изменяют несколько вершин одновременно, и предложен алгоритм построения групповых правил.

4. Предложена методика построения оценок полноты и точности синтезированного набора правил и точности отдельных правил.

В Приложении А приведен список символов LaTeX, применяющихся для анализа используемых в данной работе документов.

В Приложении Б приведен список команд LaTeX, применяющихся для анализа используемых в данной работе документов.

В Приложении В приведен список окружений LaTeX, применяющихся для анализа используемых в данной работе документов.

В Приложении Г приведены примеры построенных правил коррекции документов.

Список публикаций

Статьи в изданиях, входящих в перечень ВАК:

1. Чувилин, К. В. Использование синтаксических деревьев для автоматизации коррекции документов в формате LaTeX / К. В. Чувилин // Компьютерные исследования и моделирование. — 2012. — Т. 4, № 4. — С. 871–883.

2. Чувилин, К. В. Гибридный алгоритм сравнения документов в формате LaTeX / К. В. Чувилин // Прикладная информатика. — 2013. — № 4 (46). — С. 56–64.

Публикации в других изданиях:

3. Чувилин, К. В. Синтез правил коррекции документов в формате LaTeX с помощью сопоставления синтаксических деревьев / К. В. Чувилин // Труды 15-й всероссийской конференции «Математические методы распознавания образов». — Москва: МАКС Пресс, 2011. — С. 597–600.

4. Чувилин, К. В. Автоматический синтез правил коррекции документов в формате LaTeX и их улучшение на основе статистической оценки качества / К. В. Чувилин // Труды II Все-

российской научной конференции молодых ученых с международным участием «Теория и практика системного анализа». — 2012. — С. 17–25.

5. Чувилин, К. В. Адаптивное обучение правил коррекции документов в формате LaTeX / К. В. Чувилин // Труды 9-й международной конференции «Интеллектуализация обработки информации». — Москва: МАКС Пресс, 2012. — С. 652–655.

6. Чувилин, К. В. Использование правил со сложной структурой для коррекции документов в формате LaTeX / К. В. Чувилин // Машинное обучение и анализ данных. — 2013. — Т. 1, № 5. — С. 632–640.

Чувилин К. В.