ISSN 1818-1015
eISSN 2313-5417

2022 Volume 29 No 4

MODELING AND ANALYSIS OF INFORMATION SYSTEMS

SCIENTIFIC JOURNAL

Start date of publication — 1999
Published quarterly

FOUNDER
P.G. Demidov Yaroslavl State University

EDITORIAL OFFICE
14 Sovetskaya str., Yaroslavl 150003, Russian Federation

Website: http://mais-journal.ru
E-mail: mais@uniyar.ac.ru
Phone: +7 (4852) 79-77-73

2022 Том 29 № 4

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

НАУЧНЫЙ ЖУРНАЛ

Издается с 1999 года
Выходит 4 раза в год

УЧРЕДИТЕЛЬ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Ярославский государственный университет им. П. Г. Демидова»

РЕДАКЦИЯ
ул. Советская, 14, Ярославль, 150003, Российская Федерация
Website: http://mais-journal.ru
E-mail: mais@uniyar.ac.ru
Телефон: +7 (4852) 79-77-73

Свидетельство о регистрации СМИ ПИ № ФС 77–66186 от 20.06.2016 выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Подписной индекс — 31907 в Объединенном каталоге «Урал-Пресс». Технический редактор, компьютерная вёрстка – М. С. Каряева. Корректор английского текста – М. С. Комар. Подписано в печать 16.12.2022. Дата выхода в свет 30.12.2022. Формат 200×265 мм. Объем 108 с. Тираж 40 экз. Свободная цена. Заказ 038/022. Адрес типографии: ул. Советская, 14, оф. 109, Ярославль, 150003, Россия. Адрес издателя: Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003, Россия.
Содержание предназначено для детей старше 12 лет.

Contents

Содержание

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 29, NO. 4, 2022
journal homepage: www.mais-journal.ru

THEORY OF DATA

Analysis of Students' Academic Performance using LMS Event Logs

N. D. Shaimov[1], I. A. Lomazova[1], A. A. Mitsyuk[1], I. Y. Samonenko[1]

DOI: 10.18255/1818-1015-2022-4-286-314

[1]HSE University, 20 Myasnitskaya str., Moscow 101000, Russia.

MSC2020: 68U35
Research article
Full text in Russian

Received June 5, 2022
After revision August 23, 2022
Accepted August 26, 2022

Modern educational process involves the use of electronic educational environments. These are special information systems that are both a means for storing educational materials and a tool for conducting tests, collecting homework, keeping a grade book, and working together. Such environments produce a large amount of data containing the recorded behavior of students and teachers within the educational process. This paper proposes an approach that allows one to analyze such data and discover typical student trajectories that lead to successful or unsuccessful learning outcomes. It is shown how process mining can be used to build models of the educational process based on the available data. We also show how you can evaluate the extent to which the synthesized model reflects the actual behavior of the system recorded in event logs. The paper contains not only a description of the proposed approach, but also a case study with its application to a real data set for an undergraduate educational program. It is clearly shown how, using our approach, it is possible to find out what factors lead to the formation of successful and unsuccessful student trajectories. The bottlenecks of the educational process were identified, as well as errors in the data, indicating the incorrect operation of the system. As a result of the analysis, points of special attention for administrators of the educational program were identified, as well as some signal events, the appearance of which in a student's individual trajectory can be an alarm. The application of the approach involves the use of free open source software, which further facilitates its deployment in a variety of educational organizations.

Keywords: process analysis; process mining; learning management systems; event logs

INFORMATION ABOUT THE AUTHORS

Nikita D. Shaimov	orcid.org/0000-0003-3843-5379. E-mail: nshaimov@hse.ru Postgraduate student.
Irina A. Lomazova correspondence author	orcid.org/0000-0002-9420-3751. E-mail: ilomazova@hse.ru Professor, Doctor of Sciences in Theoretical Foundations of Computer Science.
Alexey A. Mitsyuk	orcid.org/0000-0003-2352-3384. E-mail: amitsyuk@hse.ru Associate Professor, PhD in Computer Science.
Ilya Yu. Samonenko	orcid.org/0000-0002-3063-4640. E-mail: isamonenko@hse.ru Associate Professor, PhD in Sociology.

Funding: This work is supported by the Basic Research Program at the National Research University Higher School of Economics.

For citation: N. D. Shaimov, I. A. Lomazova, A. A. Mitsyuk, and I. Y. Samonenko, "Analysis of Students' Academic Performance using LMS Event Logs ", *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 286-314, 2022.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 29, № 4, 2022
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Анализ академической успеваемости студентов с использованием журналов событий электронной образовательной среды

Н. Д. Шаимов[1], И. А. Ломазова[1], А. А. Мицюк[1], И. Ю. Самоненко[1]

DOI: 10.18255/1818-1015-2022-4-286-314

[1]Национальный исследовательский университет «Высшая школа экономики», ул. Мясницкая, д. 20, г. Москва, 101000 Россия.

УДК 004.04
Научная статья
Полный текст на русском языке

Получена 5 июня 2022 г.
После доработки 23 августа 2022 г.
Принята к публикации 26 августа 2022 г.

Современный образовательный процесс предполагает использование электронных образовательных сред. Это специальные информационные системы, которые являются как средством для хранения учебных материалов, так и инструментом для проведения проверочных работ, сбора домашних заданий, ведения журнала оценок, совместной работы. Такие среды производят большое количество данных о поведении учащихся и преподавателей в рамках учебного процесса. В данной работе предлагается подход, позволяющий анализировать такие данные, извлекать из них типичные траектории учащихся, которые ведут к успешным или неудачным результатам обучения. Показано, как для построения моделей образовательного процесса на основе имеющихся данных могут быть использованы алгоритмы process mining. Также показано, как можно оценить, насколько синтезированная модель отражает реальное поведение системы, записанное в журналах событий. Работа содержит не только описание предлагаемого подхода, но и пример его применения к реальному набору данных для образовательной программы бакалавриата. Наглядно показано, как с использованием нашего подхода можно выяснить, какие факторы приводят к формированию успешных и неудачных траекторий студентов. Выявлены узкие места образовательного процесса, а также ошибки в данных, свидетельствующие о некорректной работе системы. В результате анализа выявлены точки особого внимания для администраторов образовательной программы, а также определены некоторые сигнальные события, появление которых в индивидуальной траектории студента может быть тревожным сигналом. Применение подхода предполагает использование только свободных программных инструментов с открытым исходным кодом, что дополнительно облегчает его внедрение в самых разных образовательных организациях.

Ключевые слова: моделирование процессов; извлечение и анализ моделей процессов; электронная образовательная среда; журналы событий

ИНФОРМАЦИЯ ОБ АВТОРАХ

Никита Денисович Шаимов	orcid.org/0000-0003-3843-5379. E-mail: nshaimov@hse.ru аспирант.
Ирина Александровна Ломазова автор для корреспонденции	orcid.org/0000-0002-9420-3751. E-mail: ilomazova@hse.ru профессор, доктор физ.-мат. наук.
Алексей Александрович Мицюк	orcid.org/0000-0003-2352-3384. E-mail: amitsyuk@hse.ru доцент, канд. комп. наук.
Илья Юрьевич Самоненко	orcid.org/0000-0002-3063-4640. E-mail: isamonenko@hse.ru доцент, канд. соц. наук.

Финансирование: Работа выполнена при поддержке Программы фундаментальных исследований Национального исследовательского университета «Высшая школа экономики».

Для цитирования: N. D. Shaimov, I. A. Lomazova, A. A. Mitsyuk, and I. Y. Samonenko, “Analysis of Students’ Academic Performance using LMS Event Logs ”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 286-314, 2022.

Введение

В настоящее время активно происходит цифровизация традиционных процессов во всех областях хозяйственной деятельности: от банковского дела, промышленности и продаж до юриспруденции и систем голосования. Современное образование всех ступеней также всё больше строится на базе специализированных информационных систем. Особенно ускорился процесс внедрения цифровых технологий в образовании после начала пандемии COVID-19, когда все школы и вузы существенную часть учебного процесса вынуждены были перенести в дистанционный формат.

Сегодня образовательный процесс в высших учебных заведениях, как правило, построен с использованием специализированных информационных систем, которые также называют *системами управления обучением* или *цифровыми образовательными средами* [1]. Первоначально такие системы разрабатывались специально для организации дистанционного образования, но сегодня они используются и для поддержки обучения в традиционном формате. Студенты не только получают учебные материалы в электронной системе, но выполняют в ней остальные виды образовательной деятельности: делают индивидуальные и командные задания, проходят тестирования, контрольные работы и экзамены. Современная зачётная книжка студента также перешла в электронный вид, как и экзаменационные ведомости.

Переход образовательного процесса в электронный вид несёт с собой как риски, так и возможности. Обсуждение рисков такого перехода находится за рамками данной работы. Вместо этого обратим внимание на то, что одна из неотъемлемых особенностей современного образовательного процесса — фиксация всех действий его участников. Так как все действия как студента, так и преподавателя осуществляются через информационную образовательную среду и могут записываться. Можно сохранить факт входа в систему, обращения к тому или иному материалу, прохождение теста, просмотр видеолекции и так далее. Событийные данные сочетаются с другими, представляющими журналы оценок, программы курсов, тексты переписки студентов с преподавателями и т.д. В результате формируются огромные объёмы данных, которые обычно складируются на защищённых серверах образовательных организаций. Такие массивы данных могут быть использованы с пользой для администраторов образовательного процесса, преподавателей и студентов, что и демонстрируется в данной работе.

На основе анализа данных образовательного процесса можно, например, выявлять студентов или преподавателей, которым требуется помощь. Анализ такого рода данных позволяет совершенствовать структуру образовательной программы в целом, выявляет слабые связки дисциплин, неэффективные организационные решения и так далее. Дополнительные возможности возникают в том случае, если образовательный процесс устроен вариативным образом, когда студенты могут самостоятельно строить свою траекторию обучения. Задача исследователей в области информационных систем — предложить инструменты, которые бы помогли участникам образовательного процесса, включая администраторов, действительно эффективно использовать имеющиеся массивы данных. При этом, так как речь идёт о социальном процессе, в который вовлечены многие действующие лица, важно не только найти что можно улучшить в процессе, но ещё и не навредить никому из его участников.

В данной работе рассматривается, как данные об академической успеваемости студентов университета, получаемые из системы управления обучением, могут использоваться для выявления проблемных мест и ошибок при построении образовательной программы, которые приводят к неуспеху и неудачам студентов. Для анализа данных используется подход интеллектуального анализа процессов или, как его называют на английском языке, process mining [2]. Методы, объединяемые этим названием, которые будут рассмотрены далее в разделе 1, позволяют не просто выявлять зависимости в наборах данных, но разработаны с целью выявления динамики и причинно-следственных связей между событиями, происходящими в исследуемой системе. Это особенно полезно для

анализа образовательного процесса, в рамках которого неудачное выстраивание траектории обучения или неудачное прохождение какой-то дисциплины могут привести к провалу студента не сразу, а по прошествии существенного временного промежутка.

1. Анализ процессов

Приведём теперь некоторые базовые сведения из области интеллектуального анализа процессов, которые познакомят с ней неподготовленного читателя. А затем введём основные понятия и определения, которые потребуются далее.

1.1. Общее описание

Интеллектуальный анализ процессов (process mining) активно развивается с начала XXI века и включает в себя составляющие элементы [2], которые показаны на Рис. 1.

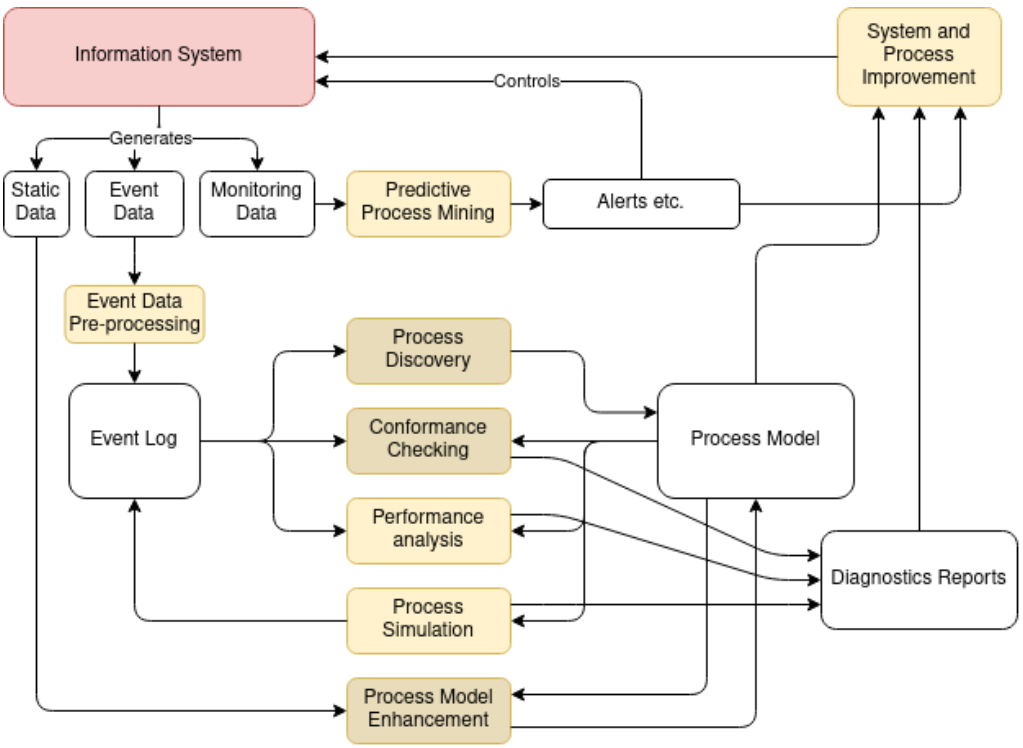


Fig. 1. Process mining

Рис. 1. Интеллектуальный анализ процессов

К ключевым задачам интеллектуального анализа процессов относят:

- автоматический синтез модели процесса по журналу событий (process discovery);
- оценку соответствия модели процесса и реального поведения процесса журнала событий информационной системы (conformance checking) [3];
- обогащение модели процесса дополнительными данными (process model enhancement).

Методы интеллектуального анализа и оптимизации процессов применяются в страховании [4], в программной инженерии [5], в электронной торговле [6], в промышленности [7—9], в медицине [10—12] и многих других областях человеческой деятельности [13].

Кроме вышеназванных основных задач, к ключевым следует отнести задачу предварительной обработки данных для получения журналов событий, с которыми могут работать алгоритмы анализа процессов. Большинство информационных систем всё ещё не настроены для генерации готовых журналов событий требуемого формата. В частности, именно так обстоит дело с данными,

получаемыми из электронных образовательных сред. В данной работе подготовке данных уделено особое внимание в разделе 4.2.

По мере внедрения методов анализа процессов в коммерческую деятельность многих компаний в дополнение к основным классическим составляющим дисциплины всё более активно развиваются новые направления, связанные с анализом производительности систем и предсказательными алгоритмами, которые позволяют анализировать данные о работе системы, что называется, «на лету» с выдачей предупреждений и предсказаний [14—16]. Не менее значимым является «А что, если ... ?» анализ, базирующийся на использовании методов симуляции моделей процессов [17—19].

Получаемые модели процессов, а также диагностические данные должны использоваться для настройки и усовершенствования работы информационной системы, которая поддерживает тот или иной процесс. Например, это может быть образовательный процесс и система управления образованием. Заметим, что усовершенствованию могут подвергаться как технические инструменты, так и сам поддерживаемый процесс, которые в действительности в процессно-ориентированных системах неразрывны [20].

Отметим, что большинство методов интеллектуального анализа процессов направлены не на анализ конкретных кейсов, а дают обобщённую картину того, как действуют и как взаимодействуют участники процесса. Это позволяет анализировать процесс в целом и соблюдать требования приватности данных.

1.2. Основные понятия

Анализ процесса начинается с журнала событий, который содержит записи о поведении информационной системы, поддерживающей этот процесс. Журналы событий разных систем могут выглядеть по-разному. В последние годы ассоциацией IEEE был разработан и стандартизирован общий формат журналов событий XES [21], который поддерживается всеми основными инструментами анализа процессов. В данной работе мы используем этот стандарт, впрочем, не все его возможности.

Процесс выполняется в соответствии с некоторой унифицированной процедурой. Выполнение этой процедуры для некоторого конкретного случая называется экземпляром процесса или кейсом. Каждый экземпляр имеет идентификатор. В случае образовательного процесса, экземпляр процесса обычно соответствует студенту.

Процесс определяется как множество событий. Каждое событие представляет собой запись (кортеж), включающий идентификатор экземпляра (case ID), отметку о времени события (timestamp), действие (activity) и набор дополнительных полей с необходимой информацией о событии. В случае образовательных процессов типом события может быть «выбор курса», «сдача экзамена», «переход на индивидуальный план» и прочее. Выделение действий и дополнительных полей обычно зависит от особенностей процесса и его аспектов, которые мы хотим анализировать. Например, дополнительные поля могут содержать информацию об исполнителе действия, используемых ресурсах, и пр.

Последовательность событий, относящихся к одному конкретному экземпляру процесса и упорядоченных в соответствии с временными отметками, называется трассой. При построении модели процесса важно разделение событий на трассы и их упорядочение в каждой трассе. Поэтому поля идентификатора экземпляра, временной отметки и поля с дополнительной информацией на этом этапе могут быть отброшены. Тогда трасса есть конечная последовательность действий. В свою очередь, упрощенный лог представляет собой мультимножество трасс, поскольку возможно, что трассы нескольких экземпляров процессов совпадают. После построения модели процесса опущенная информация может быть вновь привязана к элементам модели и использована для анализа процесса.

Таблица 1 содержит пример простого журнала событий L_{ex}. Трассы этого журнала определяются экземплярами 1, 2 и 3. Мультимножество трасс журнала L_{ex} есть $[\langle a, b, c, d\rangle^1, \langle a, c, b, d\rangle^1, \langle a, b, c, a, c, b, d\rangle^1]$, где верхний индекс 1 означает, что соответствующая трасса входит в мультимножество с кратностью 1.

Table 1. Simple event log **Таблица 1.** Простейший журнал событий

Case	Activity	Time	Actor
1	a	2022-06-29T10:00+03:00	Ivan
1	b	2022-06-29T10:02+03:00	Maria
2	a	2022-06-29T10:20+03:00	Ivan
1	c	2022-06-29T10:21+03:00	Elena
2	c	2022-06-29T10:25+03:00	Elena
1	d	2022-06-29T10:30+03:00	Fyodor
3	a	2022-06-29T10:31+03:00	Ivan
2	b	2022-06-29T10:36+03:00	Maria
2	d	2022-06-29T10:40+03:00	Fyodor
3	b	2022-06-29T10:42+03:00	Maria
3	c	2022-06-29T10:45+03:00	Elena
3	a	2022-06-29T10:46+03:00	Ivan
3	c	2022-06-29T10:48+03:00	Elena
3	b	2022-06-29T10:51+03:00	Fyodor
3	d	2022-06-29T10:55+03:00	Fyodor

В области process mining для моделирования процессов применяются различные формализмы и нотации: сети Петри, BPMN-модели, системы переходов и др. В этой работе мы используем графы частотного следования (dependency-frequency graphs).

Граф частотного следования — это ориентированный граф, в котором вершины представлены действиями и помечены их весом в журнале событий, а дуги обозначают причинно-частотную зависимость между соответствующими действиями. Дуги также помечены. Пометки на дугах в таком графе описывают силу причинно-частотной зависимости, которая вычисляется с помощью некоторого эвристического алгоритма.

На Рис. 2 показан пример графа частотного следования, построенного на основе журнала событий L_{ex} с помощью эвристического алгоритма [22] с настройками по умолчанию. Вершина a в этой модели помечена числом 4, так как действие a встречается в журнале событий 4 раза. Другие вершины помечены аналогичным образом.

Заметим, что пометки на дугах могут отображать силу причинно-следственной зависимости в разном виде. Это может быть либо частотность соответствующего отношения, либо мера относительной надёжности связи в диапазоне от 0 до 1, которая показывает степень уверенности относительно наличия причинно-следственной связи между соответствующими действиями (определение метрики см. в работе [22]) . Например, модель на Рис. 2 показывает, что действия a и b могут быть связаны причинно-следственным отношением (a является причиной b), но степень уверенности в этом не высока (0.667).

Чем больше примеров следования одного действия за другим имеется в журнале событий, тем больше будет степень уверенности в наличии отношения причинно-следственной зависимости между ними. Например, добавим в журнал событий дополнительные трассы и получим журнал L'_{ex}, содержащий такое поведение: $[\langle a, b, c, d\rangle^7, \langle a, c, b, d\rangle^1, \langle a, b, c, a, c, b, d\rangle^1]$. Тогда эвристический алгоритм со стандартными настройками выдаст модель, показанную на Рис. 3. В новой модели

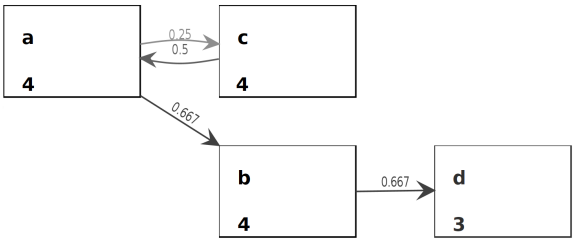


Fig. 2. Dependency-frequency graph for L_{ex} event log

Рис. 2. Граф частотного следования для журнала событий L_{ex}

степень уверенности в наличии причинно-следственной связи между действиями a и b повысилась до 0.889.

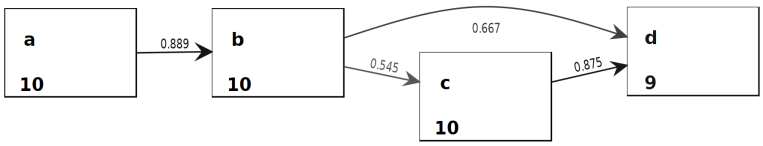


Fig. 3. Dependency-frequency graph for L'_{ex} event log

Рис. 3. Граф частотного следования для журнала событий L'_{ex}

Модель процесса не всегда точно отражает реально исполняемый процесс. Алгоритмы синтеза абстрагируются от некоторых деталей процесса с целью улучшения читаемости модели, выделения ключевых аспектов процесса и т.д. Важно уметь проверить, что модель соответствует журналу событий и в этом смысле верна.

Для оценки того, насколько хорошо данная модель процесса отражает реальное поведение системы, представленное в журнале событий, используются специальные метрики [3]. Наиболее важными являются метрики *соответствия* (fitness) между моделью и журналом событий и *точности* (precision) модели по отношению к журналу событий.

Метрика соответствия показывает, насколько хорошо модель позволяет воспроизвести трассы журнала событий. Степень соответствия измеряется числом в интервале от 0 до 1. В том случае когда модель позволяет исполнить в точности все трассы, записанные в журнале событий, т.е. степень соответствия равна 1, говорят об идеальном соответствии между моделью и журналом событий.

Как правило, модель может допускать поведение, не представленное в журнале событий, поскольку она нужна, в частности, для обобщения информации, содержащейся в журнале событий. Однако слишком общая модель может допускать много «лишнего» поведения. Для оценки такого «лишнего» поведения, допускаемого моделью, используется метрика точности. Точность модели также измеряется числом в интервале от 0 до 1. При точности 1 модель допускает только поведение, представленное в журнале событий. При точности 0 модель допускает любое поведение.

С практической точки зрения хорошими считаются модели, для которых их соответствие журналу событий близко или даже равно 1, но точность не слишком высока.

Отметим, что граф частотного следования отображает зависимости между событиями, но не всегда явно представляет допускаемые трассы. Например, графовая модель на Рис. 2 не имеет пути, соответствующего трассе $\langle a, b, c, d \rangle$ из журнала событий, на основе которого она построена. Чтобы было удобнее «накладывать» трассы на модель для вычисления метрик соответствия и точности, причинно-частотные модели переводят в эквивалентные сети Петри. Для сетей Петри разработаны различные алгоритмы вычисления этих метрик [23—26]. Далее в разделе 5.1 мы используем такой двух-этапный способ вычисления метрик соответствия и точности для графов частотного следования относительно журналов событий.

1.3. Программные инструменты

В области интеллектуального анализа процессов имеются как свободные, так и коммерческие программные инструменты, реализующие упомянутые выше алгоритмы анализа и синтеза моделей процессов.

Графические изображения моделей процессов, показанные на Рис. 2 и Рис. 3, получены с помощью реализации эвристического алгоритма синтеза в среде ProM Tools Framework [27]. Это свободный фреймворк для реализации алгоритмов анализа процессов, функционирующий на базе JVM, который появился раньше всех среди существующих на данный момент инструментов. Система ProM[1] разрабатывается открытым для участия исследовательским сообществом и доступна каждому. Именно ProM содержит реализации наибольшего количества (более 600) разных алгоритмов синтеза, анализа и усовершенствования моделей процессов.

Другой популярный свободный инструмент для анализа процессов — библиотека PM4Py[2]. Это относительно новая библиотека включает реализации базовых алгоритмов синтеза моделей процессов (среди прочих, например, и эвристического алгоритма) и их анализа на языке программирования Python [28]. Данная библиотека также разрабатывается исследовательским сообществом и довольно активно развивается. Что особенно важно, сообществом прилагаются существенные усилия для повышения эффективности работы реализаций алгоритмов, как с использованием чисто технологических подходов, например, путём исполнения задач на GPU [29], за счёт эффективного использования СУБД [30], так и на основе применения более эффективных алгоритмов [23, 31].

Существует также большое количество коммерческих инструментов, предназначенных для анализа и синтеза моделей процессов. В данной работе они не используются, а потому не рассматриваются.

2. Анализ процессов в образовании: обзор литературы

Обсудим некоторые работы, опубликованные в области анализа образовательных процессов.

Первым делом стоит упомянуть большой обзор, опубликованный в 2018 году [32]. В этом обзоре составлен каталог различных методов, которые могут применяться в задачах, характерных для образовательных процессов, а также рассматривается большое число применений. Всё это сделано на базе изучения существенного количества публикаций. Из доклада, в частности, следует, что одними из самых популярных алгоритмов синтеза в данной области являются нечёткий (Fuzzy miner) и эвристический (Heuristics miner) алгоритмы, а самая популярная методика проверки соответствия модельного и наблюдаемого поведения — проигрывание журнала событий на модели (token-based replay).

Исходные данные, по которым синтезируются модели, обычно получаются из разнообразных электронных образовательных систем [33—35].

Существенное внимание в области уделяется анализу образовательных траекторий с целью выявления и последующего устранения неудачных вариантов организации образовательного процесса, которые ведут к нежелательным результатам [33, 35—37]. Симптомами такой нежелательной организации могут быть, например, общая низкая академическая успеваемость или большое число отчисляющихся по разным причинам студентов. Так как область исследования образовательных процессов достаточно молода, многие работы последних лет посвящены сравнению эффективности различных алгоритмов автоматического синтеза моделей в применении к задачам анализа образовательных процессов [34, 35]. Чаще всего рассматриваются нечёткий или эвристический алгоритмы синтеза, но иногда исследователи включают в рассмотрение и индуктивный алгоритм. Некоторое исследователи [38] предлагают новые способы интерпретации классических моделей для

[1] http://promtools.org
[2] https://pm4py.fit.fraunhofer.de

применения в задачах анализа и моделирования образования или предлагают специализированные модели процессов [39].

В [38] предложена довольно любопытная модель. Набор академических задолженностей студента сравнивается с «рюкзаком», наполненным камнями, который, очевидно, затрудняет движение. По мере прохождения студентом тех или других контрольных мероприятий по одной из дисциплин рюкзак может наполняться новыми задолженностями или опустошаться. События изменения его наполнения фиксируются в журнале. Таким образом, становится возможным синтезировать разные траектории (представляются графами непосредственного следования), демонстрирующие типичные жизненные циклы рюкзаков студентов в ходе их обучения. Данная работа оказала существенное влияние на наше исследование, которое развивает идею построения траекторий на основе информации об академической успеваемости.

В некоторых работах рассматривается не организация процесса в целом, но поведение студентов в рамках конкретной составляющей образовательного процесса. Например, это может быть анализ выполнения студентами тестов в образовательной системе [40], или анализ поведения студентов в совместной работе над командным заданием [41], или выявление типичных шаблонов предоставления обратной связи в процессе обучения [42]. Авторы показывают, как на основе моделей, синтезированных с использованием нечёткого алгоритма синтеза, могут быть выявлены типичные варианты прохождения тестов студентами, а также отклоняющиеся от обычного порядка варианты поведения, которые потенциально могут приводить и приводят к провалу теста.

Характерные фрагменты поведения студентов в образовательной системе могут рассматриваться как шаблоны [43]. Поиск нежелательных (или, наоборот, желательных) шаблонов в типичных траекториях студентов позволяет получить более полное понимание того, как устроен образовательный процесс, а также выявить недостатки организации образовательной программы [44].

В некоторых случаях делаются даже попытки формирования индивидуальных рекомендаций по выбору наиболее актуальных для студента образовательных ресурсов (лекций, задач для самопроверки и т.д.) в рамках дисциплин, построенных по принципам самоуправляемого обучения [45]. При этом синтезированные модели процессов используются вместе с результатами анализа статистических данных о пользователях и позволяют выявлять характерные стратегии студентов и строить рекомендации на их базе. Авторы на практических данных демонстрируют, что построение рекомендаций с учётом в том числе и моделей процессов, делает предложения богаче и потенциально уменьшает риск неудачи с тестом и отчисления студентов. Когда речь идёт об обучении сложным навыкам, например, на курсах по моделированию, автоматически синтезированные модели могут повысить и качество преподавательских советов студенту, обогатить обратную связь, которая даётся по результатам выполнения заданий [46].

В работе [34] сделан вывод, что индуктивный алгоритм синтеза хорошо подходит для синтеза моделей образовательных процессов и дает более содержательные результаты по сравнению с Альфа-алгоритмом, эвристическим алгоритмом и алгоритмом синтеза эволюционных деревьев. Важным для нас выводом авторов этой работы является заключение о том, что при анализе образовательных процессов совершенно невозможно обойтись без предварительной обработки сырых данных. Впрочем, эта мысль является общей для process mining [2].

3. Постановка задачи

Основная цель нашей работы — сформулировать и описать подход для выявления ключевых характеристик образовательной программы, влияющих на успешное завершение студентом учебного года, на основании журналов событий, содержащих данные с результатами отдельных экзаменов.

Как конкретный пример мы анализируем данные одной из образовательных программ бакалавриата НИУ «Высшая школа экономики» за 2020/2021 год. Учебный год разделен на 4 модуля. Каждая дисциплина завершается экзаменом в одном из модулей. Оценка за экзамен ставится по

десятибалльной шкале, при этом оценка ниже 4 баллов считается неудовлетворительной. В случае получения неудовлетворительной оценки у студента возникает академическая задолженность по данной дисциплине и он имеет право на две пересдачи. Если у студента возникли три академические задолженности или если он не устранил академическую задолженность по итогу двух пересдач, то студент отчисляется с образовательной программы или ему предлагается индивидуальный учебный план для повторного изучения дисциплины. Существуют также и другие причины, по которым студент может не завершить учебный год успешно. Среди таких причин, например, уход в академический отпуск, перевод на другую образовательную программу и другие.

В нашей работе для каждого студента экзамен (или переэкзаменовка) может быть представлен двумя различными типами событий. Или студент «сдал экзамен» (тип события pass), т.е. получил оценку 4 балла или выше, или студент «не сдал экзамен» (тип события fail), т.е. получил оценку ниже 4 баллов. В случае неявки студента определяется была ли эта неявка по уважительной причине или нет.

Журнал событий, сформированный на основе данных из информационной системы университета, содержит следующую информацию:

1. Идентификатор (ID) студента;
2. Тип события;
3. Дата события.

Событиями являются, например, успешная или нет сдача конкретного экзамена (или переэкзаменовка), запись о пропуске экзамена, запись о неуспешном завершении учебного года по той или иной причине (больше деталей приводится в разделе 4).

Построенная на основании полученного журнала событий модель может быть использована для ответа на следующие вопросы:

1. Как выглядят образовательные траектории успешных студентов, и как устроены отклонения от данных траекторий?
2. Какие последовательности событий в наибольшей степени могут привести к тому, что студент не сможет успешно завершить курс? В результате будут выявлены наиболее «проблемные» для студентов дисциплины.
3. Существуют ли некорректные последовательности событий? Их наличие означает возможные ошибки в информационной системе университета или организации учебного процесса, которые необходимо устранить.

Предлагаемая модель может быть инструментом для руководителей образовательных программ и университетской администрации для анализа образовательных программ в целях их дальнейшего совершенствования и развития. Отдельно отметим, что большинство известных авторам университетских информационных систем в той или иной форме содержат информацию, используемую в данном исследовании, что придаёт нашему подходу универсальность.

4. Анализ академической успеваемости методами process mining: подход к решению задачи

Рассмотрим теперь в подробностях подход, который мы предлагаем для анализа данных об академической успеваемости и поиска ответов на вопросы, сформулированные в разделе 3. На Рис. 4 представлена общая схема подхода, предлагаемого в данной работе.

Сырые данные из разных источников предварительно обрабатываются, что даёт исходный набор данных. Описание используемых данных и метода их предварительной обработки содержится в разделе 4.1. Затем с помощью специально разработанных программных инструментов из сырого набора данных конструируется журнал событий (детали приводятся в разделе 4.2). Модели процессов синтезируются на основе журнала событий и, при необходимости, обогащаются дополнительной информацией, содержащейся в исходном наборе данных, но не принятой во внимание в ходе

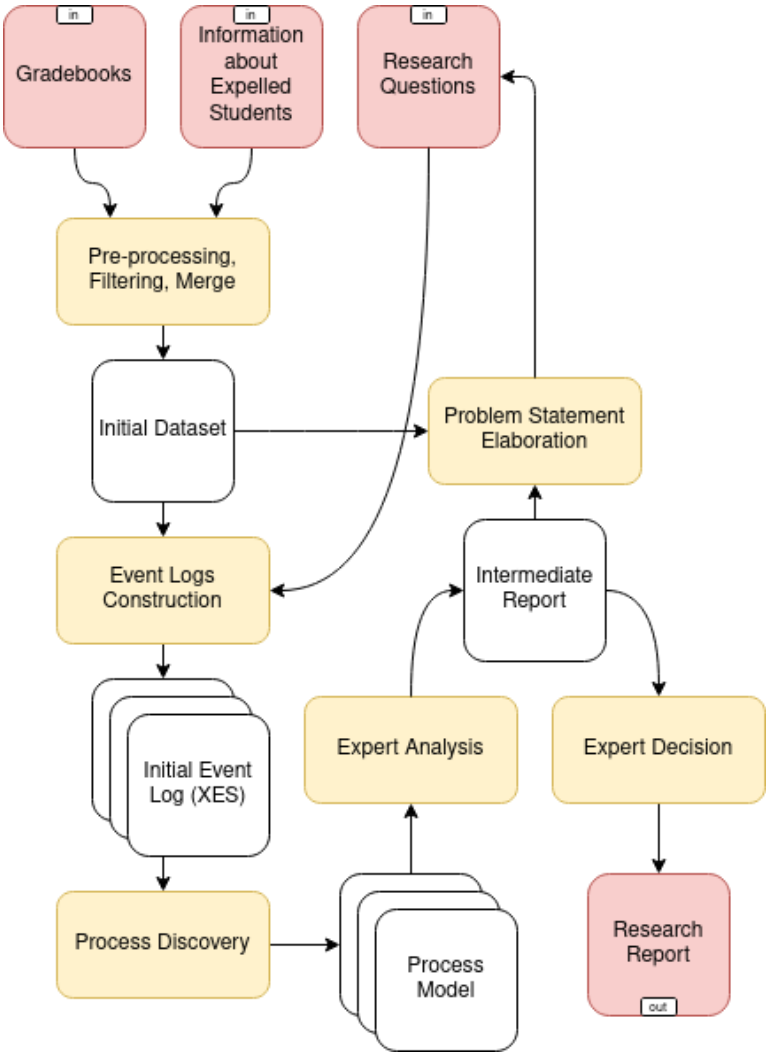


Fig. 4. General scheme for analysis of academic data

Рис. 4. Общая схема анализа данных об академической успеваемости студентов

конструирования первичного журнала событий (см. раздел 4.3). Наконец, получившиеся модели используются для формулировки ответов на поставленные в разделе 3 вопросы. Заметим тот важный факт, что сама формулировка задачи может изменяться, уточняться и детализироваться в ходе применения указанного подхода. Заметим также, что используемый подход в целом согласуется с описанным в литературе подходом PM^2 [47] для проведения проектов по анализу процессов.

4.1. Исходные данные

Как уже было отмечено, данные для данного исследования получены из внутренней электронной образовательной среды НИУ ВШЭ. Вся личная информация была анонимизирована, и студентам был присвоен уникальный идентификатор с помощью хэш-функции Python.

Для объединения и предобработки таблиц со студенческими оценками и экзаменационными ведомостями применена библиотека Pandas[3]. Вся несущественная для нашего исследования информация (например, информация о факультете или типе дисциплины) при этом была исключена из рассмотрения.

[3] https://pandas.pydata.org/

В результате получена таблица с таким набором *исходных данных*: [уникальный идентификатор студента]; [оценка по пятибалльной шкале]; [оценка по десятибалльной шкале]; [отметка присутствия на экзамене]; [дата экзамена]; [номер экзаменационной ведомости]; [тип записи (экзамен, переэкзаменовка, пересдача с комиссией)]; [учебный год]; [модуль (1–4)]; [название дисциплины]; [подразделение (кафедра), поддерживающее дисциплину]; [имя заполнившего ведомость профессора].

В том случае, если поле оценки по какой-то причине не заполнено, оно заполняется «0». В исходных данных даты заполнения ведомостей и проведения экзаменов были в разных форматах. Все они приведены в единый формат. Некоторые записи не содержали важной информации. Такие записи полностью удалены из набора данных. Обработанная таблица использована для конструирования журнала событий. Данный процесс описан далее в разделе 4.2.

4.2. Подготовка журнала событий

Рассмотрим теперь подробно процесс конструирования журнала событий из подготовленных данных образовательной информационной системы.

Для применения методов process mining нам необходимо преобразовать полученный массив данных из образовательной системы в более компактный журнал событий с унифицированным форматом данных. Для этого прежде всего определим формат и структуру журнала событий. Как уже было отмечено в разделе 1, каждая запись о событии представляет собой кортеж. Для определения события в составе записи обязательно должны быть представлены поля с идентификатором экземпляра процесса (case ID), отметкой о времени события (timestamp), а также указан тип действия (activity). При необходимости в записи также может содержаться набор дополнительных полей с другой информацией о событии. В качестве идентификатора экземпляра процесса мы пользуемся уникальным идентификатором (ID) студента. Отметкой о времени события послужит дата экзамена или пересдачи. Поле, фиксирующее тип действия, может быть заполнено разными способами, выбор каждого из которых определяется вопросами, для ответа на которые производится анализ процесса. Данное поле должно содержать всю информацию о действии, вызвавшем событие, которая может понадобится в ходе анализа процесса. В то же время необходимо избегать чрезмерного усложнения.

В данной работе поле тип действия для журнала событий формировалось экспериментальным путем. Отправной точкой стал вариант журнала событий, представленный в работе [38], где в качестве действий рассматриваются текущие задолженности студента. Однако при проведении первых экспериментов было выявлено множество недостатков такого формата журнала событий в контексте нашей задачи. В отличии от образовательной системы, рассматриваемой в [38], правила анализируемой в данной работе образовательной среды предполагают наличие ограничений на максимальное число задолженностей и сроки их устранения. Что ещё более важно, студенты в нашем случае имеют возможность пересдавать экзамены. Также в течение одного учебного года может быть несколько экзаменов по учебному курсу. Дальнейшие эксперименты привели к выбору другого варианта заполнения поля типа действия. Фокус сместился с задолженностей непосредственно на академическую траекторию студентов. Отслеживая экзамены и пересдачи в течение учебного года, мы получили модель, отражающую полную картину событий в ходе года. Полученная из полученного лога модель схожа с вариантами, используемыми в работах [40, 45], и позволяет выявлять шаблоны и взаимосвязи действий в траекториях студентов.

В конечном итоге мы определили общую структуру для поля типа действия так:

- номер учебного модуля (1-4);
- название учебного курса;
- тип события (экзамен, пересдача, пересдача с комиссией);
- результат действия (pass — успех, fail — неудача);

• (необязательный атрибут) причина отсутствия (уважительная, неуважительная).

Приведём несколько примеров заполнения поля тип действия у записей с использованием такой структуры:

• «4 module Algebra exam fail nonvalid miss»;
• «2 module Algebra retake pass»;
• «4 module Programing exam pass».

В ходе дальнейших экспериментов была выявлена необходимость отслеживать итоговое состояние студента, чтобы формулировать выводы об успешности траектории. Студенты, которые были отчислены по тем или иным причинам, взяли академический отпуск или перевелись на другую специальность, будут иметь соответствующее событие с указанным действием. В том случае, если студент успешно завершил учебный год и продолжит обучение, в конце трассы будет добавлено событие с действием «pass». Благодаря наличию таких событий можно также отследить студентов, которые были восстановлены в данном учебном году. Эти студенты будут иметь событие с отчислением в начале трассы.

Наконец, из собранных данных на основе приведенной выше структуры полей и столбцов был сконструирован журнал событий. Для этого нами был разработан алгоритм, использующий средства библиотеки Pandas. Алгоритм добавляет в журнал события для каждого студента в указанном формате преобразовывая данные из таблицы с данными. Если студент присутствует в таблице отчисленных, то будет добавлено событие с соответствующим типом действия и указанной временной меткой события. В противном случае в трассу, соответствующую студенту, добавляется событие с действием «pass» с временной отметкой конца учебного года.

Так как мы не собираемся строить модель для всех студентов сразу, наиболее рациональным решением будет разбить журнал на части. В нашем случае данные содержат информацию за один учебный год. Мы можем разделить студентов по курсам и получить 4 журнала, каждый из которых соответствует конкретному курсу. При наличии наборов данных за больший промежуток времени появляются возможности иного разделения студентов. Например, можно сгруппировать траектории студентов по году начала обучения, что позволяет проследить за группами студентов на протяжении нескольких курсов.

В журналах событий, которые мы получаем в данной работе, присутствует большое число различных событий: успешные или неудачные экзамены, переэкзаменовка, переэкзаменовка с комиссией, уход в академический отпуск по собственному желанию или по медицинским основаниям, отчисление в связи с академической задолженностью и т. п. Такая подробная детализация позволяет более точно моделировать траекторию студента в учебном процессе, но в то же время делает модель сложнее для восприятия. Поэтому одновременно мы также анализируем два упрощённых журнала событий.

В итоге мы получаем и анализируем далее три варианта построения журнала событий.

Базовый вариант конструирования предполагает учёт всех имеющихся событий без объединения различных событий. Такой подход особенно полезен для выявления различного рода аномалий в данных. Если в учетной системе образовательной организации произошла ошибка (например, была ошибочно указана дата экзамена так, что студент как будто бы сдавал экзамены уже после своего отчисления), модель процесса покажет соответствующий переход от отчисления к сдаче экзамена, что невозможно в реальности. Все подобные аномалии могут быть выявлены, проанализированы и, при необходимости, устранены.

Первый вариант модификации журнала событий предполагает объединение всех событий, связанных с конкретной учебной дисциплиной, в две группы: дисциплина успешно закрыта студентом (pass) и дисциплина не закрыта (fail). К первому виду событий относятся успешная сдача экзамена по дисциплине с первого раза, а также успешные пересдачи. Ко второму виду относятся все

события, связанные как с провалом экзамена по дисциплине, так и с провалом переэкзаменовок. В отличии от базового подхода в данном случае возможен повтор события с провалом дисциплины. Это означает появление циклов в модели. События, описывающие завершение обучения на данном курсе, также относятся к двум группам: успешное окончание (pass) и его отсутствие (non-pass). К первой группе относятся варианты успешного завершения учебного года: перевод на следующий курс или успешное завершение обучения. Ко второй группе относятся все прочие события: уход в академический отпуск, отчисление по собственному желанию, отчисление в связи с недобросовестным освоением образовательной программы и т. п. В рамках данного подхода группируются редкие события. Благодаря этому выделяются общие тенденции, а траектории студентов представляются в более общем виде без лишней детализации.

Второй вариант модификации журнала событий предполагает группировку всех событий, связанных с конкретной дисциплиной, такого вида: или дисциплина сдана с первого раза (pass), или с дисциплиной возникли проблемы проблемы (fail). Отличие данного подхода от предыдущего заключается в том, что любая пересдача (даже успешная) или пропуск экзамена (даже по уважительной причине) рассматриваются как проблемные ситуации. Заметим, что такие ситуации являются проблемными не только для студента, но и для учебной администрации, для которой усложняется организация образовательного процесса. Такой вариант журнала событий позволяет выделить наиболее проблемные области в образовательных траекториях.

Второй и третий варианты модификации исходного журнала событий представляют собой только некоторые возможные способы модификации. Для получения наиболее полной картины процесса рекомендуется использовать наиболее полный журнал событий, так как при модификации часть важных деталей будет утеряна. Модификация журнала уместна в тех случаях, когда необходимо рассмотреть определенный аспект образовательного процесса, а также когда детали не несут существенной ценности или когда алгоритм построения модели не способен качественно обработать зашумленные данные.

4.3. Построение моделей образовательного процесса

Рассмотрим теперь, какие методы используются в данной работе для построения моделей образовательного процесса.

Мы воспользуемся эвристическим алгоритмом синтеза (Heuristics miner) [48] в той его версии, что реализована в библиотеке PM4Py. Данный алгоритм позволяет получить модели в виде графа частотного следования, а также в виде сети Петри, что позволяет произвести расчеты соответствия (fitness) и точности (precision) модели по отношению к заданному журналу событий. Более того, это можно сделать разными методами: путём проигрывания журнала событий на модели (token-based replay) [23, 24] или с использованием выравниваний (aligment-based) [25, 26]. Мы используем в работе оба этих метода для повышения достоверности результатов.

Алгоритм синтеза имеет несколько входных параметров, которые влияют на вид получаемой модели. Среди прочего данные параметры позволяют задать пороговые значения для учёта событий и связей между ними в зависимости от их частоты. Это позволяет отсекать менее значимые события и связи в модели, что сделано для уменьшения влияния шума в данных на получаемую модель. Кроме того, восприятие и интерпретация модели, построенной без фильтрации редких вариантов поведения, крайне затруднительны. Пример подобной модели приведен на Рис. 5.

В нашей работе мы рассматриваем процесс без параллелизма. В связи с этим максимальное значение устанавливается для параметра эвристического алгоритма, отвечающего за уровень отсечения параллельных событий. Это обеспечивает полное исключение возможных параллельных событий в генерируемой модели, что является артефактом синтеза, а не реальным феноменом анализируемого процесса.

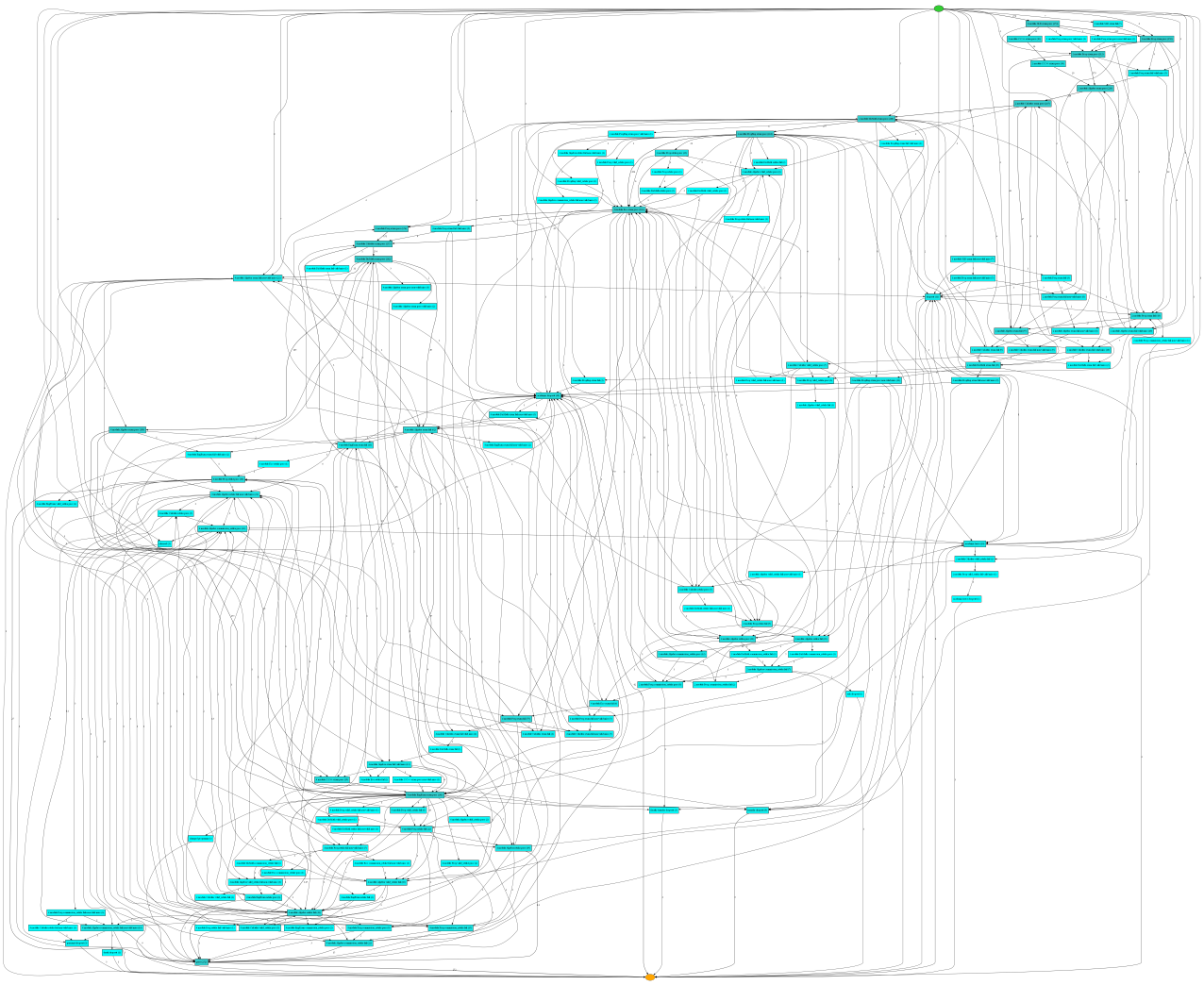


Fig. 5. Process model example without specifying threshold

Рис. 5. Пример модели процесса без использования пороговых значений

Возможность изменять входные параметры алгоритма позволяет выделить наиболее важные аспекты в модели. Для того чтобы изучить конкретные аспекты процесса более детально, можно прибегнуть также к фильтрации журнала событий. При помощи фильтров можно оставить в журнале событий только определенные трассы, соответствующие заданному критерию. Таким образом можно, например, получить модель, которая содержит только трассы студентов, которые провалили определенный экзамен.

Для поиска отклонений в учебных траекториях студентов и характерных закономерностей мы используем следующий подход к построению моделей.

На *первом этапе* анализируются модели, синтезированные из журнала событий. При этом применяются только различные параметры алгоритма синтеза. Обозначим подобные модели как *полные*. Путем задания пороговых значений для алгоритма можно получить достаточно компактную модель, выделяющую наиболее важные части всех траекторий.

Например, по журналу событий для первого курса на подобной модели можно выделить наиболее проблемные учебные курсы. На фрагменте модели, показанном на Рис. 6, можно выделить провалы экзаменов по программированию и алгебре во 2 модуле. Связь между данными событиями провалов демонстрирует, что в половине подобных случаев присутствуют оба провала. Далее

для всех найденных подобных аномалий можно задействовать фильтрацию журнала событий и построить модели для анализа только траекторий, содержащих конкретную аномалию. При необходимости можно вновь установить пороговые значения алгоритма для получения более читаемых моделей.

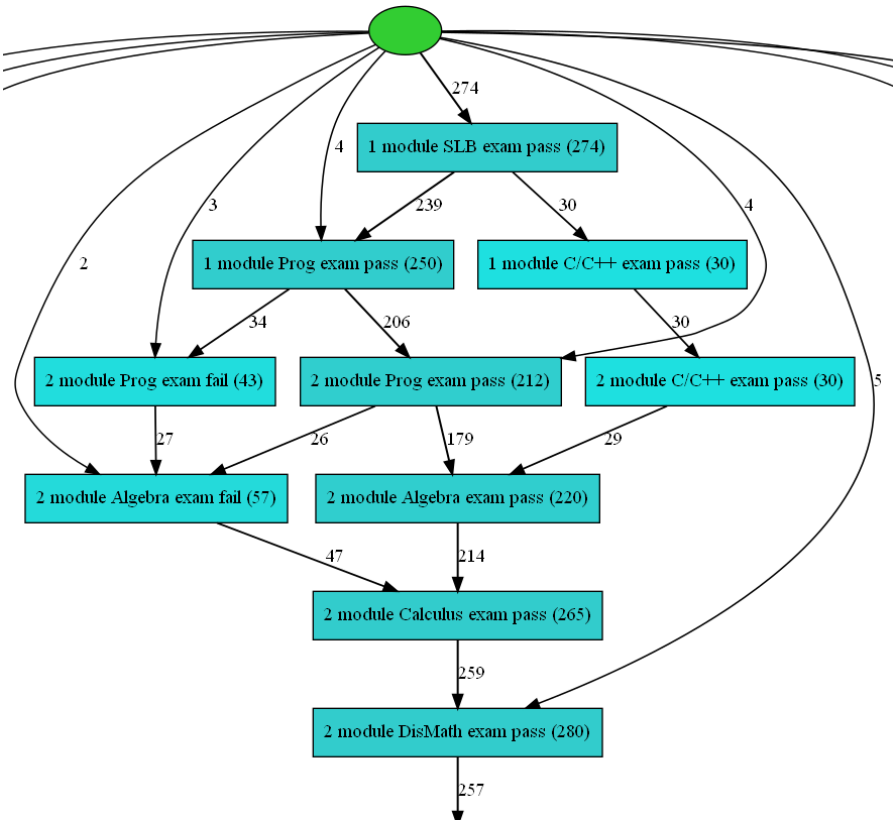


Fig. 6. Example fragment of process model for the 1-year students

Рис. 6. Пример фрагмента модели процесса для студентов 1 курса

Второй этап предполагает исключение (при помощи фильтров) из журнала событий «идеальных» траекторий студентов. Это траектории без отклонений, где все экзамены сданы с первого раза без пересдач. Обозначим данные модели как *модели отклонений*. Ввиду особенностей эвристического алгоритма, расчет относительной важности событий и переходов ведётся с учетом их частоты. Исключение траекторий части студентов из журнала событий позволяет выделить некоторые отклонения, которые в первом варианте не имели достаточного веса в модели, а потому были не видны.

Для уже упомянутого журнала событий для студентов первого курса полученная модель, содержащая только траектории с отклонениями, показала проблемы с некоторыми пересдачами экзаменов, которые отсутствовали в модели на первом этапе.

Как уже было сказано ранее, найденные аномалии можно дополнительно изучить после фильтрации журнала событий.

Третий этап развивает идею, использованную на втором этапе. В журнале событий, который содержит только траектории студентов с отклонениями, можно дополнительно исключить траектории с уникальными вариантами поведения. Таким образом, мы получим журнал событий, который содержит только повторяющиеся траектории отклонений. Такие модели обозначим как *модели повторных отклонений*. Модель, построенная по данному журналу событий, позволит рассмотреть

отклонения, которые встречались более одного раза и выделить области пересечений соответствующих траекторий.

На примере подобной модели для первого курса удалось обнаружить траекторию студентов, которые пропустили все экзамены во втором модуле и были отчислены за академическую неуспеваемость. Более того, в этой траектории один из экзаменов во втором модуле не был указан как пропущенный. При дополнительном рассмотрении выяснилось, что это ошибка в исходных данных, а действительные пропуски для данного курса не были отмечены в данных. Таким образом, различные подходы к анализу могут также помогать обнаруживать ошибки в данных. Как и на прошлых этапах, аномалии можно дополнительно изучить при помощи фильтрации журнала событий.

Используя данный подход из трех этапов можно получить полную картину траекторий студентов и изучить все возможные отклонения и аномалии. Результаты применения подхода к конкретным данным реальной образовательной программы рассмотрим в следующем разделе.

5. Анализ моделей образовательного процесса

Анализ будем проводить в соответствии с подходом, который описан в разделе 4.3. Для различения двух видов модификации журнала событий, описанных в разделе 4.2, обозначим первый вариант модификации журнала событий как вариант А, а второй вариант модификации — как вариант Б.

Данный раздел организован следующим образом. Первым делом в подразделе 5.1 приводится оценка синтезированных моделей с использованием классических для process mining метрик. Затем в подразделе 5.2 собраны различные наблюдения, сформулированные в ходе анализа синтезированных моделей образовательного процесса. Наконец, подраздел 5.3 содержит обобщённые результаты анализа и выводы.

5.1. Оценка синтезированных моделей процесса

Классические для process mining метрики соответствия и точности, упоминавшиеся ранее в разделе 1.2, позволяют выразить численно то, насколько модель процесса отражает наблюдаемую реальность, представленную журналом событий.

Результаты расчёта значений этих метрик для синтезированных моделей процессов приводятся в Таблице 2. При расчёте использованы два метода из библиотеки PM4Py — алгоритм, основанный на проигрывании журнала событий на модели (соответствующие результаты расчёта помечены в таблице с помощью латинской буквы «T»), а также алгоритм, использующий выравнивания (соответствующие результаты помечены буквой «A»).

В большинстве случаев оба алгоритма выдают схожий результат. Особенно это заметно для расчета точности модели. В наших экспериментах значение соответствия, посчитанное с использованием алгоритма, проигрывающего журнал событий на модели, в среднем выше, чем при использовании алгоритма, основанного на использовании выравниваний.

Для лога без модификаций характерно снижение значений метрик при переходе от полной модели к модели отклонений и увеличение при переходе от полной модели к модели повторных отклонений. Данное соотношение прослеживается на всех трех журналах событий. Пониженные значения метрик у модели отклонений можно объяснить большим числом уникальных трасс с отклонениями, которые тяжело выразить на одной общей модели без потерь точности и соответствия. В полной модели данные трассы составляют меньшую долю и оказывают меньше влияния при расчете метрик. В случае с моделями повторных отклонений причина повышенных значений может быть в малом числе кейсов у данных моделей.

Модифицированные логи показывают несколько иную картину. С точки зрения соответствия соотношения значений между полной моделью, моделью отклонений и моделью повторных

Table 2. Calculated fitness and precision values for discovered educational process models

Таблица 2. Рассчитанные значения соответствия и точности для синтезированных моделей образовательного процесса

Модель	Fitness T	Fitness A	Precision T	Precision A
Первый курс				
Полная модель	0.963	0.892	0.641	0.639
Модель отклонений	0.945	0.835	0.621	0.619
Модель повторных отклонений	0.979	0.959	0.746	0.746
Лог А. Полная модель	0.939	0.811	0.777	0.768
Лог А. Модель отклонений	0.888	0.635	0.786	0.780
Лог А. Модель повторных отклонений	0.966	0.902	0.620	0.611
Лог Б. Полная модель	0.941	0.851	0.794	0.794
Лог Б. Модель отклонений	0.897	0.715	0.767	0.767
Лог Б. Модель повторных отклонений	0.966	0.900	0.642	0.642
Второй курс				
Полная модель	0.989	0.967	0.617	0.616
Модель отклонений	0.982	0.940	0.579	0.577
Модель повторных отклонений	1.0	1.0	0.849	0.849
Лог А. Полная модель	0.946	0.855	0.872	0.867
Лог А. Модель отклонений	0.864	0.607	0.803	0.790
Лог А. Модель повторных отклонений	0.933	0.885	0.731	0.728
Лог Б. Полная модель	0.944	0.847	0.857	0.857
Лог Б. Модель отклонений	0.886	0.643	0.799	0.799
Лог Б. Модель повторных отклонений	0.977	0.935	0.692	0.692
Третий курс				
Полная модель	0.998	0.997	0.843	0.843
Модель отклонений	0.993	0.989	0.769	0.769
Модель повторных отклонений	1.0	1.0	0.930	0.930
Лог А. Полная модель	0.987	0.971	0.822	0.822
Лог А. Модель отклонений	0.935	0.846	0.646	0.646
Лог А. Модель повторных отклонений	1.0	1.0	0.902	0.902
Лог Б. Полная модель	0.992	0.974	0.896	0.896
Лог Б. Модель отклонений	0.961	0.879	0.704	0.704
Лог Б. Модель повторных отклонений	1.0	1.0	0.854	0.854

отклонений в большинстве случаев идентичны соотношению у полного журнала событий. Однако точность у модели повторных отклонений чаще всего оказывается ниже, чем у полной модели и модели отклонений, что связано с наличием циклов на модели в отличии от полного журнала событий, где циклы отсутствуют. Единственное исключение — это модель третьего курса. Для этой модели значения ведут себя также, как и у полного журнала событий. Данное исключение можно объяснить крайне низким числом экземпляров процесса (студентов) и отклонений для модели повторных отклонений у третьего курса.

При сравнении значений для одного и того же типа моделей в случае полного журнала событий и его модификаций можно заметить, что значения соответствия для модифицированных журналов ниже, чем у полной версии журнала событий. Снижение значения соответствия можно объяснить всё тем же наличием циклов в моделях, синтезированных на основе модификаций журнала событий. Циклы значительно усложняют как построение качественной модели, так и сам расчёт метрик. Однако значение точности в моделях, синтезированных на основе модифицированных журналов

событий, в целом выше, чем у моделей на основе полного журнала событий, что объясняется меньшим разнообразием событий.

Рассмотрим также значения для моделей, полученных после изменения параметров алгоритма построения моделей. Единственный параметр алгоритма, который будет изменяться в ходе исследования, — это порог значительности для отношения взаимной зависимости между событиями (dependency threshold). Данный параметр устанавливает минимально необходимый уровень значительности для включения в модель того или иного отношения между событиями [48]. Уровень значительности определяет используемую эвристику алгоритма. Данный параметр позволяет уменьшить объём модели без потери наиболее важных событий. Рассмотрим влияние разных значений данного параметра на значение соответствия и точности модели по отношению к журналу событий на основе данных, приводимых в Таблице 3.

Table 3. Dependence of fitness and precision from algorithm threshold

Таблица 3. Зависимость значений метрик соответствия и точности от порогового значения алгоритма

Модель	Fitness T	Fitness A	Precision T	Precision A
Полная модель второго курса	0.989	0.967	0.617	0.616
Пороговое значение 0.2	0.989	0.965	0.606	0.606
Пороговое значение 0.4	0.985	0.958	0.607	0.606
Пороговое значение 0.6	0.968	0.915	0.377	0.379
Пороговое значение 0.8	0.936	0.861	0.312	0.320
Полная модель третьего курса	0.998	0.997	0.843	0.843
Пороговое значение 0.2	0.996	0.990	0.845	0.845
Пороговое значение 0.4	0.996	0.990	0.845	0.845
Пороговое значение 0.6	0.978	0.946	0.624	0.632
Пороговое значение 0.8	0.944	0.916	0.527	0.547

Исходя из полученных данных можно заключить, что данный параметр оказывает заметное влияние на значение соответствия модели только при сравнении крайних значений порогового значения. При высоком пороговом значении заметно некоторое снижение значения соответствия относительно модели, для которой пороговое значение равно нулю. С точки зрения точности модели параметр значительно уменьшается при пороговом значении выше 0.5.

Полностью обойтись без использования данного параметра возможно лишь в редких случаях, тогда как чаще модели реального образовательного процесса получаются крайне объемными. Использование данного параметра алгоритма позволяет значительно упростить внешний вид модели и, соответственно, подчеркнуть ключевые особенности процесса в ходе анализа.

5.2. Наблюдения об образовательном процессе, сформулированные в ходе анализа моделей процессов

Приведём теперь некоторые наблюдения, которые были сделаны в ходе анализа большого количества моделей в виде графов частотного следования, синтезированных на базе журналов событий реального образовательного процесса.

Первый курс. Полная модель.

С использованием параметра эвристического алгоритма можно значительно сократить объем полной модели журнала событий и получить модель, отображающую наиболее важную часть анализируемого процесса. Анализ полной модели даёт возможность выделить значимые события. Например, в ходе анализа данной модели можно выделить проблемы с экзаменами по алгебре и программированию во 2 и 4 модулях. При этом для экзамена по программированию в 1 модуле

практически не наблюдается провалов. Также алгоритм выделяет на модели провалы по пересдаче экзамена по программированию в 4 модуле. После определения точек интереса на полной модели можно сделать определенные гипотезы для дальнейшего анализа и продолжить более подробное изучение выявленных точек при помощи фильтрации журнала событий. Далее при помощи фильтров получим урезанные журналы событий, оставляя только студентов с конкретным событием в траектории, и построим на их основе модели.

Рассмотрим модель с траекториями студентов, у которых был провал экзамена по программированию во 2 модуле. Это позволит нам обнаружить взаимосвязь данного события с другими событиями процесса. Половина таких студентов также имеет провал экзамена по алгебре во 2 модуле. А четверть студентов имеют провалы пересдач по алгебре или программированию за 2 модуль. Однако провал пересдач по двум этим предметам одновременно встречается редко. Лишь четверть студентов успешно сдаёт экзамен по алгебре в 4 модуле. Такое же количество проваливает программирование в 4 модуле и пересдачу алгебры с комиссией за 4 модуль. Всего в итоге только 25% студентов успешно завершает учебный год. Оставшиеся отчисляются, переводятся или уходят в академический отпуск.

Изучая траектории с провалом экзамена по алгебре во 2 модуле мы видим схожую картину. Половина таких студентов провалило экзамен по программированию во 2 модуле. Треть студентов имеют проблемы с пересдачей алгебры за 2 модуль и провал экзамена по программированию в 4 модуле. Только шестая часть студентов успешно сдают экзамен по алгебре в 4 модуле. В итоге к успешному завершению года приходит только треть из таких студентов.

По траекториям с провалом экзамена по программированию в 4 модуле можно заметить, что большинство студентов успешно сдали экзамен по программированию и по алгебре во 2 модуле, а провалы обычно наблюдались в паре: сразу и по программированию, и по алгебре. Однако экзамен по алгебре в 4 модуле успешно сдали только 40% студентов. При этом две трети студентов успешно завершили учебный год.

Можно обнаружить, что половина студентов, имеющих провал экзамена по алгебре в 4 модуле, также имеет провал экзамена по программированию в 4 модуле. Кроме того, лишь половина студентов успешно сдали экзамены по алгебре и программированию во 2 модуле с первой попытки. В итоге 60% из данных студентов успешно завершили учебный год.

Последняя выявленная точка интереса — это провалы пересдачи экзамена по программированию за 4 модуль. В траекториях, которые включают данное событие, у половины студентов наблюдаются провалы во 2 модуле сразу и экзамена по алгебре, и экзамена по программированию. Подавляющее большинство имели проблемы с экзаменом по алгебре в 4 модуле. Половина студентов также не справились с пересдачей экзамена по программированию за 4 модуль с комиссией, а среди траекторий часто наблюдаются провалы пересдач по иным учебным курсам. Только половина таких студентов смогли успешно завершить учебный год.

Важно также отметить, что для первого курса предусмотрены два варианта учебного курса по программированию. Часть студентов вместо обычного курса программирования проходит углублённый курс программирования на C/C++. Все траектории студентов, выбравших данный курс, приходят в точку с успешным завершением учебного года.

Первый курс. Модель отклонений.

После исключения из журнала событий траекторий студентов без каких-либо отклонений на новой модели в зависимости от используемого алгоритма могут появится новые детали, которые не проявлялись на фоне более часто происходящих успешных событий. Данная модель полезна для анализа журналов событий с траекториями большого числа студентов и отклонений в процессе обучения.

Сравнивая обнаруженные отклонения и аномалии на модели можно заметить, что многие из них совпадают с полной моделью траекторий. Однако есть некоторые отличия. Так, например, можно заметить повышенное число проблемных пересдач по алгебре за 2 и 4 модуль в добавок к уже обнаруженным ранее проблемам с пересдачами по программированию за 4 модуль. Кроме того, во втором модуле наблюдаются провалы с дискретной математикой, которые к тому же имеют связь с пропусками других экзаменов.

Рассматривать уже рассмотренные отклонения при помощи фильтрации журнала событий не имеет смысла, так как получаемые модели не будут отличаться друг от друга. Поэтому обратим внимание на студентов, проваливших пересдачу по алгебре.

Студенты, провалившие пересдачу по алгебре во 2 модуле, в половине случаев имели провал экзамена по программированию за 2 модуль, а четверть из них провалила вторую пересдачу с комиссией за 2 модуль алгебры. К тому же эти студенты в половине случаев имели провал экзаменов по алгебре и программированию в 4 модуле. Только половина таких студентов успешно завершили учебный год.

Студенты, провалившие пересдачи по алгебре в 4 модуле, показывают схожую картину. Четверть студентов имели провал экзаменов по алгебре и программированию за 2 модуль, а половина от этого количества имели провал по двум этим курсам одновременно. Половина студентов имели провал экзамена по программированию в 4 модуле. Интересно, что только половина студентов с провалом пересдачи по алгебре в 4 модуле провалили экзамен по данному предмету. Вторая половина от проваливших пересдачу пропустили первый экзамен по уважительной или неуважительной причине. Половина студентов провалили и вторую пересдачу с комиссией за 4 модуль алгебры. Однако в итоге 75% данных студентов смогли успешно завершить учебный год.

Траектории, следующие через провал экзамена по дискретной математике во 2 модуле, в большинстве своём содержат множество провалов или пропусков. Ни один из следующих по данным траекториям студентов не закончил год успешно. Практически все траектории обрываются отчислением во втором модуле.

Первый курс. Модель повторных отклонений.

Эта модель даёт возможность рассмотреть наиболее важные отклонения, так как здесь присутствуют только те отклонения от «идеальной» траектории, которые повторялись многократно. Воспроизводимость отклонения обычно является показателем распространенности проблемы или наличия у проблемы какой-то фундаментальной причины. Рассматривая траектории в виде единой модели можно наглядно увидеть точки пересечения повторных отклонений и выявить наиболее проблемные места образовательного процесса.

Для первого курса на подобной модели видна обособленная траектория с пропусками экзаменов, которая оканчивается отчислением за академическую неуспеваемость после 2 модуля. Более детальное изучение данной траектории выявило ошибку в исходных данных, где пропуск экзамена по дискретной математике во 2 модуле был указан как провал. В остальных случаях повторяющиеся отклонения чаще всего связаны с провалом экзамена по программированию в 4 модуле, реже — с провалом экзаменов по алгебре во 2 и 4 модулях. Во всех случаях, за исключением траектории с пропусками, все студенты в представленной модели успешно завершили учебный год.

Первый курс. Варианты журнала событий А и Б.

Как было указано ранее, любой вариант модификации журнала событий будет скрывать часть информации по сравнению с полным вариантом журнала событий. По результатам анализа моделей, синтезированных на основе данных журналов событий, не было выявлено ничего нового по сравнению с наблюдениями, сформулированными в ходе анализа моделей на основе полного журнала событий.

Однако данные журналы событий можно использовать несколько иначе. С использованием метрики *rework* из библиотеки PM4Py можно посчитать количество повторений событий в журнале. С учетом особенностей построения данных журналов событий, в таком случае мы получим список экзаменов, имеющих наибольшее число провалов. Обнаруженные события можно рассмотреть детальнее на моделях, синтезированных по полному журналу событий, с фильтром, который оставляет только траектории студентов с определенной проблемой. Таким образом можно ускорить процесс анализа образовательного процесса.

В результате применения метрики rework для данных журналов событий можно сделать вывод, что наибольшим числом повторений обладают провалы по алгебре и программированию во 2 и 4 модуле. Среди этих событий при рассмотрении по варианту журнала событий A особенно часто повторяются события, связанные с курсом алгебры в 4 модуле. Данная дисциплина в условном рейтинге повтора событий оказывается выше прочих минимум в 2 раза. Это является показателем необычно высокого числа неудачных пересдач.

Второй курс. Полная модель.

При анализе полной модели для второго курса, полученной при помощи эвристического алгоритма с установленным параметром отсечки, можно выделить несколько ключевых точек интереса. В 1 модуле наблюдаются многочисленные провалы экзамена курса «Архитектура вычислительных систем», затем во 2 модуле также наблюдаются провалы экзаменов по курсам «Конструирование программного обеспечения», «Алгоритмы и структуры данных» и «Теория вероятностей и математическая статистика». Наконец, в 4 модуле присутствуют провалы экзамена по курсу «Конструирование программного обеспечения». Важно отметить, что частота для всех указанных событий практически равная. При изучении модели также можно заметить несколько событий, связанных с провалами пересдач экзаменов по курсам «Алгоритмы и структуры данных» и «Теория вероятностей и математическая статистика» за 2 модуль. Более того, событие с провалом пересдачи по курсу «Теория вероятностей и математическая статистика» имеет повторы, что не предусмотрено образовательной системой и явно указывает на ошибку в исходных данных.

Воспользуемся фильтрацией журнала событий и рассмотрим более подробно провал экзамена по курсу «Архитектура вычислительных систем» в 1 модуле. Только половина таких студентов смогли успешно сдать с первого раза экзамен по данному курсу во 2 модуле. Дальнейшие траектории довольно разнообразны. Среди всех событий преобладают пропуски и провалы экзаменов. В итоге лишь 14% от общего числа таких студентов успешно завершили учебный год.

Рассмотрим также отдельно траектории студентов с провалом экзамена по курсу «Конструирование программного обеспечения» во 2 модуле. Среди подобных траекторий наблюдаются провалы экзаменов, упомянутых выше. Эти провалы затрагивают от четверти до половины студентов. Среди таких провалов больше всего выделяется провал экзамена по курсу «Алгоритмы и структуры данных» во 2 модуле. Также можно отметить провал курса «Теория вероятностей и математическая статистика» во 2 модуле. 30% таких студентов провалили экзамен и еще 30% пропустили экзамен в основном по уважительным причинам. 40% студентов с указанным провалом успешно завершили учебный год.

Траектории студентов с провалом экзамена по курсу «Алгоритмы и структуры данных» во 2 модуле также содержат провалы уже упомянутых выше курсов, затрагивающих от четверти до половины студентов. На данной модели выделяется большое число провалов пересдачи курса «Алгоритмы и структуры данных» за 2 модуль. Она была неудачна для около 60% от числа студентов, проваливших экзамен. А 40% провалили и вторую пересдачу с комиссией. Только половина студентов успешно завершили учебный год.

Модель траекторий с провалом экзамена по курсу «Теория вероятностей и математическая статистика» во 2 модуле показывает картину, идентичную двум, которые обсуждались ранее. Однако

для данного курса на полной модели можно также заметить большое число провалов данного экзамена из-за пропуска по уважительной причине. Среди траекторий тех, кто пропустил экзамен по уважительной причине, наблюдается малое число отклонений, за исключением того, что половина таких студентов провалили пересдачу после пропуска. Также около половины не закончили учебный год. Половина от этого числа перевелась в другое учебное учреждение, а другая половина была отчислена за академическую неуспеваемость.

Провалы экзамена по курсу «Конструирование программного обеспечения» в 4 модуле связаны с той же самой группой провалов всё тех же курсов, которые были явно видны на полной модели. Однако в отличии от прошлых случаев, частота отклонений ниже. Вторым отличием является присутствие в модели группы различных пересдач с комиссией дисциплин 2 курса. Их частота довольно низка, однако представлены все экзамены за 2 модуль. Также 60% из студентов имеющих указанный провал экзамена провалили и его пересдачу, а 28% от общего числа студентов также провалили и вторую пересдачу с комиссией. Несмотря на это 74% студентов смогли успешно завершить учебный год.

Рассмотрим также и обнаруженные проблемы с пересдачами. Среди студентов с провалом пересдачи экзамена по курсу «Алгоритмы и структуры данных» во 2 модуле заметно большое число провалов и второй пересдачи с комиссией данного экзамена, около 65% от числа таких студентов. Еще для 35% траекторий студентов можно выделить связь с провалами экзаменов по курсу «Конструирование программного обеспечения» во 2 или 4 модуле. Только половина указанных студентов успешно завершили учебный год.

По траекториям с провалом пересдачи экзамена по курсу «Теория вероятностей и математическая статистика» за 2 модуль видно, что большинство студентов с данным провалом имели пропуск по уважительной причине. Как было указано ранее, у данного события имеются повторы, а на основе анализа траекторий видна возможная причина ошибки в данных. Вполне вероятно, что данные повторы компенсируют отсутствующее событие с дополнительной пересдачей ввиду пропуска по уважительной причине. У четверти студентов имеется провал экзамена по курсу «Конструирование программного обеспечения» в 4 модуле. В итоге 40% студентов успешно завершили учебный год.

Второй курс. Модель отклонений.

Модель, построенная на основе журнала событий после исключения траекторий без отклонений, не показывает ничего нового относительно предыдущей модели, так как для второго курса наблюдается меньшее число студентов и меньшее число отклонений от «идеальной» траектории относительно первого курса.

Второй курс. Модель повторных отклонений.

Модель повторных отклонений в сравнении с первым курсом имеет меньшее число студентов, но, как и ранее, большинство успешно завершают учебный год.

На модели можно заметить обособленную траекторию, ведущую к отчислению за академическую неуспеваемость, содержащую провалы трех экзаменов во 2 модуле по курсам: «Алгоритмы и структуры данных», «Теория вероятностей и математическая статистика», «Конструирование программного обеспечения».

В основном все повторяющиеся траектории отклонений содержат пропуски экзаменов по уважительной или неуважительной причине. Среди пропусков можно отметить сдачу экзамена по курсу «Алгоритмы и структуры данных» во 2 модуле при пропуске по неуважительной причине. Такое возможно ввиду особенностей образовательного процесса. Студенты получают общую удовлетворительную оценку по дисциплине без успешной сдачи экзамена благодаря баллам, накопленным в ходе семестра.

Второй курс. Варианты журнала событий А и Б.

В сравнении с первым курсом повторений стало меньше, что говорит о большей частоте успешных пересдач. Среди повторений можно наблюдать группу экзаменов по курсам «Алгоритмы и структуры данных» и «Теория вероятностей и математическая статистика» за 2 модуль, а также «Конструирование программного обеспечения» за 2 и 4 модуль.

Третий курс. Полная модель.

В журнале событий, содержащем события образовательного процесса третьего года обучения, количество студентов в данных заметно меньше. Однако даже несмотря на это невооруженным глазом видно практически полное отсутствие отклонений на модели. Все присутствующие отклонения — это, в основном, единичные случаи. Единственный относительно часто встречающийся в траекториях вариант отклонения связан с курсом «Математические методы анализа данных» в 4 модуле. Также модель содержит сдачу экзамена по курсу «Проектирование архитектуры программных систем» в 4 модуле при наличии пропуска по уважительной или неуважительной причине.

Третий курс. Модель отклонений.

Модель отклонений в данном случае позволила взглянуть на все имеющиеся отклонения, так как все они имеют низкую частоту и теряются на фоне основной траектории. В то же время частота данных событий слишком мала для того, чтобы можно было сделать содержательные выводы.

Третий курс. Модель повторных отклонений.

Модель повторных отклонений еще раз подчеркнула то, что было выявлено при анализе полной модели и позволила сделать дополнительные выводы относительно упомянутых курсов. Так, провал экзамена по курсу «Математические методы анализа данных» в 4 модуле среди повторяющихся траекторий представлен только в виде пропуска по неуважительным причинам. При этом траектории студентов, содержащиеся в модели повторных отклонений, приводят к успешному, хоть и непростому, завершению учебного года.

Третий курс. Варианты журнала событий А и Б.

Повторения событий в основном представляют собой единичные случаи, что показывает практически полное отсутствие неуспешных пересдач.

Четвертый курс.

Количество траекторий студентов в журнале событий, как и число отклонений этих траекторий, в сравнении с третьим курсом уменьшилось настолько, что уже не позволяет провести какой-либо существенный анализ.

5.3. Результаты анализа образовательного процесса

Подведем итоги проведённого и сделаем обобщающие выводы из полученных наблюдений.

Анализ синтезированных моделей образовательного процесса с применением предложенного подхода позволяет в краткие сроки получить большое количество информации о зависимостях между конкретными дисциплинами, выявить проблемы и аномалии в учебных траекториях студентов. В комбинации с другими методами анализа данных можно легко обнаружить причины тех или иных нежелательных событий. Результаты анализа также позволяют выявить события в студенческих траекториях, которые сигнализируют о возможных дальнейших отклонениях, которые могут привести к негативным исходам.

Для первого курса удалось установить, что наибольшее число проблем связано с учебными курсами по алгебре и программированию. Чаще всего проблемы возникают с экзаменом по алгебре за 4 модуль. Дальнейший анализ показал, что между курсами по алгебре и программированию наблюдается корреляция, а их провал часто взаимосвязан. Провалы по алгебре и программированию во 2 модуле более опасны, чем в 4 модуле, так как чаще приводят к безуспешному завершению

учебного года. Важно отметить, что результат экзамена по программированию в 1 модуле не оказывает влияния на последующие провалы экзаменов по программированию во 2 и 4 модулях, что странно. Студенты, которые выбрали альтернативный курс программирования на C/C++, могут иметь отклонения в своих учебных траекториях. Однако даже не смотря на наличие отклонений в траекториях, все подобные студенты успешно завершают учебный год. Провал экзамена курса по дискретной математике в 2 модуле наблюдается только в траекториях с большим числом отклонений. То есть у среднего студента проблем с курсом не возникает. Среди студентов первого курса наблюдается группа неактивных студентов, пропускающих экзамены по неуважительной причине, что приводит к их отчислению за академическую неуспеваемость в конце 2 модуля. К неудачному завершению учебного года, в основном, приводят уникальные траектории отклонений. Траектории отклонений, для которых зарегистрировано несколько случаев, приводят рано или поздно к успешному завершению учебного года. Таким образом, можно дать рекомендацию администраторам академической программы с особой внимательностью относиться ко всем необычным проблемам, которые возникают у студентов. Обычные проблемы существенно реже приводят к отчислению, чем уникальные.

Для второго курса удалось установить, что наибольшее число проблем у студентов вызывают учебные курсы «Алгоритмы и структуры данных», «Теория вероятностей и математическая статистика» и «Конструирование программного обеспечения». Провалы данных курсов часто встречаются в студенческих траекториях вместе, что показывает взаимосвязь между ними. Также это говорит о том, что студенты, попавшие на отклоняющуюся траекторию, вероятнее будут иметь более одного отклонения на этой учебной траектории. Среди провалов экзаменов выделяется провал экзамена по курсу «Архитектура вычислительных систем» в 1 модуле. Студенты, провалившие данный экзамен, часто имеют множество других отклонений в своих учебных траекториях. Кроме того, эти студенты редко успешно заканчивают учебный год. Можно порекомендовать администраторам академической программы сразу же брать студентов, которые споткнулись на данном курсе, под индивидуальную «опеку», чтобы помочь справиться и не отчислиться. Наконец, удалось обнаружить ошибки в исходных данных, допущенных на этапе сбора информации.

Для третьего курса наблюдается низкое число отклонений. Студенты на данном этапе обучения «оступаются» гораздо реже, а все отклонения наблюдаются ближе к концу учебного года. К тому же провалы экзаменов редко получают продолжение в виде провалов пересдачи и большинство студентов с отклонениями в траекториях всё же пересдают экзамены и успешно завершают учебный год.

Заключение

В данной работе предложен общий подход, пригодный для анализа данных, получаемых из электронных образовательных сред, которые лежат в основе современного образования. Данный подход позволяет извлечь пользу из большого количества данных о движении студентов по их индивидуальным образовательным траекториям, которые записываются подобными системами. Данная работа не только описывает новый подход, основанный на использовании методов process mining, но и содержит пример применения этого подхода к реальному набору данных учебного процесса на одной из программ бакалавриата, который получен из информационной образовательной среды НИУ «Высшая школа экономики».

Синтезированные модели процессов наглядно представляют образовательные траектории успешных студентов, а также то, как устроены отклонения от данных траекторий. Показано, что в комбинации с другими методами анализа данных можно легко обнаружить причины тех или иных событий и траекторий. В результате анализа выявлены точки особого внимания для администраторов образовательной программы, а также определены некоторые сигнальные события, появление которых в индивидуальной траектории студента может быть тревожным сигналом. Наконец,

показано, что наш подход позволяет выявлять и некорректные последовательности событий, наличие которых в учебных траекториях свидетельствует об ошибках в данных или о некорректной работе образовательной среды.

Конечно, не на все открытые вопросы даны ответы в данной работе, а потому отметим также несколько направлений дальнейших исследований.

Увеличение разнообразия рассматриваемых типов действий в журнале событий позволяет строить большое количество разнообразных моделей процессов и отвечать на самые разные вопросы. Например, можно рассматривать журналы событий, отдельными действиями в которых будут получение отличной, хорошей, удовлетворительной оценок. Такие модели можно использовать для выявления типичных траекторий не просто успевающих студентов, но отличников.

Интерес представляет также анализ моделей прохождения отдельных дисциплин. Для этого необходимо использовать журналы событий, содержащие данные не только об итоговых экзаменах, но и о промежуточных контрольных и домашних работах. Интересным является также проведение сравнительного анализа журналов событий для различных образовательных программ.

Анализ данных позволяет выявлять цепочки событий в учебных траекториях, которые с большей вероятностью приведут к положительному исходу. Это означает, что есть возможность построения прогноза для образовательной траектории студента. Модуль такого прогнозирования может быть основой для рекомендательной системы, осуществляющей помощь в выборе индивидуальной учебной траектории из вариативных дисциплин учебного плана. Представляется, что разработка и внедрение подобных систем в образовательный процесс может помочь снизить число отчислений, а значит, сделать жизнь студентов, преподавателей, администраторов образования немного проще.

References

[1] R. Jaakonmäki, J. vom Brocke, S. Dietze, H. Drachsler, A. Fortenbacher, R. Helbig, M. D. Kickmeier-Rust, I. Marenzi, A. Suarez, and H. Yun, *Learning Analytics Cookbook - How to Support Learning Processes Through Data Analytics and Visualization*, ser. Springer Briefs in Business Process Management. Springer, 2020.

[2] W. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016, ISBN: 978-3-662-49850-7. DOI: 10.1007/978-3-662-49851-4. [Online]. Available: https://doi.org/10.1007/978-3--49851-4.

[3] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking - Relating Processes and Models*. Springer, 2018, ISBN: 978-3-319-99413-0. DOI: 10.1007/978-3-319-99414-7. [Online]. Available: https://doi.org/10.1007/978-3-319-99414-7.

[4] S. Suriadi, M. T. Wynn, C. Ouyang, A. H. M. ter Hofstede, and N. J. van Dijk, «Understanding Process Behaviours in a Large Insurance Company in Australia: A Case Study», in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 7908, Springer, 2013, pp. 449–464.

[5] M. Mittal and A. Sureka, «Process mining software repositories from student projects in an undergraduate software engineering course», in *ICSE Companion*, ACM, 2014, pp. 344–353.

[6] A. Mitsyuk, A. Kalenkova, S. Shershakov, and W. van der Aalst, «Using process mining for the analysis of an e-trade system: A case study», *Biznes-informatika*, no. 3 (29), pp. 15–27, 2014.

[7] S.-k. Lee, B. Kim, M. Huh, S. Cho, S. Park, and D. Lee, «Mining transportation logs for understanding the after-assembly block manufacturing process in the shipbuilding industry», *Expert Syst. Appl.*, vol. 40, no. 1, pp. 83–95, 2013.

[8] Á. Valencia-Parra, B. Ramos-Gutiérrez, A. J. Varela-Vaca, M. T. G. López, and A. G. Bernal, «Enabling process mining in aircraft manufactures: extracting event logs and discovering processes from complex data», in *BPM (Industry Forum)*, ser. CEUR Workshop Proceedings, vol. 2428, CEUR-WS.org, 2019, pp. 166–177.

[9] K. Smit and J. Mens, «Process Mining in The Rail Industry: A Qualitative Analysis of Success Factors and Remaining Challenges», in *Bled eConference*, University of Maribor Press / Association for Information Systems, 2019, p. 25.

[10] J. Munoz-Gama, N. Martin, C. Fernández-Llatas, O. A. Johnson, M. Sepúlveda, E. Helm, V. Galvez-Yanjari, E. Rojas, A. Martinez-Millana, D. Aloini, I. A. Amantea, R. Andrews, M. Arias, I. Beerepoot, E. Benevento, A. Burattin, D. Capurro, J. Carmona, M. Comuzzi, B. Dalmas, R. de la Fuente, C. D. Francescomarino, C. D. Ciccio, R. Gatta, C. Ghidini, F. Gonzalez-Lopez, G. Ibáñez-Sánchez, H. B. Klasky, A. P. Kurniati, X. Lu, F. Mannhardt, R. Mans, M. Marcos, R. M. de Carvalho, M. Pegoraro, S. K. Poon, L. Pufahl, H. A. Reijers, S. Remy, S. Rinderle-Ma, L. Sacchi, F. Seoane, M. Song, A. Stefanini, E. Sulis, A. H. M. ter Hofstede, P. J. Toussaint, V. Traver, Z. Valero-Ramon, I. van de Weerd, W. van der Aalst, R. J. B. Vanwersch, M. Weske, M. T. Wynn, and F. Zerbato, «Process mining for healthcare: Characteristics and challenges», *J. Biomed. Informatics*, vol. 127, p. 103 994, 2022.

[11] A. Guzzo, A. Rullo, and E. Vocaturo, «Process mining applications in the healthcare domain: A comprehensive review», *WIREs Data Mining Knowl. Discov.*, vol. 12, no. 2, 2022.

[12] M. R. Dallagassa, C. dos Santos Garcia, E. E. Scalabrin, S. O. Ioshii, and D. R. Carvalho, «Opportunities and challenges for applying process mining in healthcare: a systematic mapping study», *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 1, pp. 165–182, 2022.

[13] C. dos Santos Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, and E. E. Scalabrin, «Process mining techniques and applications - A systematic mapping study», *Expert Syst. Appl.*, vol. 133, pp. 260–295, 2019.

[14] M. Dumas and F. M. Maggi, «Enabling Process Innovation via Deviance Mining and Predictive Monitoring», in *BPM - Driving Innovation in a Digital World*, J. vom Brocke and T. Schmiedel, Eds., Springer, 2015, pp. 145–154.

[15] I. Teinemaa, M. Dumas, F. M. Maggi, and C. D. Francescomarino, «Predictive Business Process Monitoring with Structured and Unstructured Data», in *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, M. L. Rosa, P. Loos, and O. Pastor, Eds., ser. Lecture Notes in Computer Science, vol. 9850, Springer, 2016, pp. 401–417.

[16] I. Teinemaa, N. Tax, M. de Leoni, M. Dumas, and F. M. Maggi, «Alarm-Based Prescriptive Process Monitoring», in *Business Process Management Forum - BPM Forum 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings*, M. Weske, M. Montali, I. Weber, and J. vom Brocke, Eds., ser. Lecture Notes in Business Information Processing, vol. 329, Springer, 2018, pp. 91–107.

[17] W. van der Aalst, «Business Process Simulation Survival Guide», in *Handbook on Business Process Management 1, Introduction, Methods, and Information Systems, 2nd Ed*, ser. International Handbooks on Information Systems, J. vom Brocke and M. Rosemann, Eds., Springer, 2015, pp. 337–370.

[18] A. A. Mitsyuk, I. S. Shugurov, A. A. Kalenkova, and W. van der Aalst, «Generating event logs for high-level process models», *Simul. Model. Pract. Theory*, vol. 74, pp. 1–16, 2017. DOI: 10.1016/j.simpat.2017.01.003. [Online]. Available: https://doi.org/10.1016/j.simpat.2017.01.003.

[19] W. van der Aalst, «Process mining and simulation: a match made in heaven!», in *Proceedings of the 50th Computer Simulation Conference, SummerSim 2018, Bordeaux, France, July 09-12, 2018*, ACM, 2018, 4:1–4:12. [Online]. Available: http://dl.acm.org/citation.cfm?id=3275386.

[20] W. van der Aalst, «Process-Aware Information Systems: Design, Enactment, and Analysis», in *Wiley Encyclopedia of Computer Science and Engineering*, B. W. Wah, Ed., John Wiley & Sons, Inc., 2008.

[21] G. Acampora, A. Vitiello, B. N. D. Stefano, W. van der Aalst, C. W. Günther, and E. Verbeek, «IEEE 1849: The XES Standard: The Second IEEE Standard Sponsored by IEEE Computational Intelligence Society [Society Briefs]», *IEEE Comput. Intell. Mag.*, vol. 12, no. 2, pp. 4–8, 2017.

[22] A. J. M. M. Weijters and W. van der Aalst, «Rediscovering workflow models from event-based data using little thumb», *Integr. Comput. Aided Eng.*, vol. 10, no. 2, pp. 151–162, 2003. DOI: 10.3233/ica-2003-10205. [Online]. Available: https://doi.org/10.3233/ica-2003-10205.

[23] A. Berti and W. van der Aalst, «A Novel Token-Based Replay Technique to Speed Up Conformance Checking and Process Enhancement», *Trans. Petri Nets Other Model. Concurr.*, vol. 15, pp. 1–26, 2021.

[24] J. Munoz-Gama and J. Carmona, «A Fresh Look at Precision in Process Conformance», vol. 6336, Sep. 2010, pp. 211–226, ISBN: 978-3-642-15617-5. DOI: 10.1007/978-3-642-15618-2_16.

[25] J. Buijs, B. Dongen, and W. Aalst, «Quality Dimensions in Process Discovery: The Importance of Fitness, Precision, Generalization and Simplicity», *International Journal of Cooperative Information Systems*, vol. 23, p. 1 440 001, Mar. 2014. DOI: 10.1142/S0218843014400012.

[26] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. Dongen, and W. Aalst, «Measuring precision of modeled behavior», *Information Systems and e-Business Management*, vol. 13, Jan. 2014. DOI: 10.1007/s10257-014-0234-7.

[27] E. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. van der Aalst, «ProM 6: The Process Mining Toolkit», in *Proceedings of the Business Process Management 2010 Demonstration Track, Hoboken, NJ, USA, September 14-16, 2010*, M. L. Rosa, Ed., ser. CEUR Workshop Proceedings, vol. 615, CEUR-WS.org, 2010. [Online]. Available: http://ceur-ws.org/Vol-615/paper13.pdf.

[28] A. Berti, S. J. van Zelst, and W. van der Aalst, «Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science», *CoRR*, vol. abs/1905.06169, 2019. arXiv: 1905.06169. [Online]. Available: http://arxiv.org/abs/1905.06169.

[29] A. Berti, M. P. Nghia, and W. van der Aalst, «PM4Py-GPU: A High-Performance General-Purpose Library for Process Mining», in *Research Challenges in Information Science - 16th International Conference, RCIS 2022, Barcelona, Spain, May 17-20, 2022, Proceedings*, R. S. S. Guizzardi, J. Ralyté, and X. Franch, Eds., ser. Lecture Notes in Business Information Processing, vol. 446, Springer, 2022, pp. 727–734.

[30] S. A. Shershakov, «VTMine for Visio: A Graphical Tool for Modeling in Process Mining», *Autom. Control. Comput. Sci.*, vol. 55, no. 7, pp. 847–865, 2021. DOI: 10.3103/S0146411621070282. [Online]. Available: https://doi.org/10.3103/S0146411621070282.

[31] I. S. Shugurov and A. A. Mitsyuk, «Applying MapReduce to conformance checking», *Proceedings of ISP RAS*, vol. 28, no. 3, pp. 103–122, 2016. [Online]. Available: https://ispranproceedings.elpub.ru/jour/issue/download/9/17#page=104.

[32] A. Bogarín, R. Cerezo, and C. Romero, «A survey on educational process mining», *WIREs Data Mining Knowl. Discov.*, vol. 8, no. 1, 2018. DOI: 10.1002/widm.1230. [Online]. Available: https://doi.org/10.1002/widm.1230.

[33] J. C. Vidal, B. Vázquez-Barreiros, M. Lama, and M. Mucientes, «Recompiling learning processes from event logs», *Knowl. Based Syst.*, vol. 100, pp. 160–174, 2016. DOI: 10.1016/j.knosys.2016.03.003. [Online]. Available: https://doi.org/10.1016/j.knosys.2016.03.003.

[34] A. Bogarín, R. Cerezo, and C. Romero, «Discovering learning processes using Inductive Miner: A case study with Learning Management Systems (LMSs).», *Psicothema*, vol. 30 3, pp. 322–329, 2018.

[35] H. Al-Qaheri and M. Panda, «An Education Process Mining Framework: Unveiling Meaningful Information for Understanding Students' Learning Behavior and Improving Teaching Quality», *Inf.*, vol. 13, no. 1, p. 29, 2022. DOI: 10.3390/info13010029. [Online]. Available: https://doi.org/10.3390/info13010029.

[36] E. M. Real, E. P. Pimentel, L. V. de Oliveira, J. C. Braga, and I. Stiubiener, «Educational Process Mining for Verifying Student Learning Paths in an Introductory Programming Course», in *IEEE Frontiers in Education Conference, FIE 2020, Uppsala, Sweden, October 21-24, 2020*, IEEE, 2020, pp. 1–9. DOI: 10.1109/FIE44824.2020.9274125. [Online]. Available: https://doi.org/10.1109/FIE44824.2020.9274125.

[37] J. P. Salazar-Fernandez, M. Sepúlveda, J. Munoz-Gama, and M. Nussbaum, «Curricular Analytics to Characterize Educational Trajectories in High-Failure Rate Courses That Lead to Late Dropout», *Applied Sciences*, vol. 11, no. 4, 2021, ISSN: 2076-3417. DOI: 10.3390/app11041436. [Online]. Available: https://www.mdpi.com/2076-3417/11/4/1436.

[38] J. P. Salazar-Fernandez, J. Munoz-Gama, J. Maldonado-Mahauad, D. Bustamante, and M. Sepúlveda, «Backpack Process Model (BPPM): A Process Mining Approach for Curricular Analytics», *Applied Sciences*, vol. 11, no. 9, 2021, ISSN: 2076-3417.

[39] I. A. Lomazova, A. A. Mitsyuk, and A. M. Sharipova, *Modeling MOOC learnflow with Petri net extensions*, 2021. DOI: 10.48550/ARXIV.2111.04419. [Online]. Available: https://arxiv.org/abs/2111.04419.

[40] L. Juhanák, J. Zounek, and L. Rohlíková, «Using process mining to analyze students' quiz-taking behavior patterns in a learning management system», *Comput. Hum. Behav.*, vol. 92, pp. 496–506, 2019. DOI: 10.1016/j.chb.2017.12.015. [Online]. Available: https://doi.org/10.1016/j.chb.2017.12.015.

[41] V. Southavilay, K. Yacef, and R. A. Calvo, «Process Mining to Support Students' Collaborative Writing», in *EDM*, www.educationaldatamining.org, 2010, pp. 257–266.

[42] G. Deeva and J. D. Weerdt, «Understanding Automated Feedback in Learning Processes by Mining Local Patterns», in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing, vol. 342, Springer, 2018, pp. 56–68.

[43] D. Codish, E. Rabin, and G. Ravid, «User behavior pattern detection in unstructured processes - a learning management system case study», *Interact. Learn. Environ.*, vol. 27, no. 5-6, pp. 699–725, 2019.

[44] J. Maldonado-Mahauad, M. Pérez-Sanagustín, R. F. Kizilcec, N. Morales, and J. Munoz-Gama, «Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses», *Comput. Hum. Behav.*, vol. 80, pp. 179–196, 2018.

[45] W. Hachicha, L. Ghorbel, R. Champagnat, C. A. Zayani, and I. Amous, «Using Process Mining for Learning Resource Recommendation: A Moodle Case Study», in *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES-2021, Virtual Event / Szczecin, Poland, 8-10 September 2021*, J. Watróbski, W. Salabun, C. Toro, C. Zanni-Merk, R. J. Howlett, and L. C. Jain, Eds., ser. Procedia Computer Science, vol. 192, Elsevier, 2021, pp. 853–862. DOI: 10.1016/j.procs.2021.08.088. [Online]. Available: https://doi.org/10.1016/j.procs.2021.08.088.

[46] G. Sedrakyan, J. D. Weerdt, and M. Snoeck, «Process-mining enabled feedback: "Tell me what I did wrong" vs. "tell me how to do it right"», *Comput. Hum. Behav.*, vol. 57, pp. 352–376, 2016.

[47] M. L. van Eck, X. Lu, S. J. J. Leemans, and W. van der Aalst, «PM^2: Process Mining Project Methodology», in *CAiSE*, ser. Lecture Notes in Computer Science, vol. 9097, Springer, 2015, pp. 297–313.

[48] A. Weijters, W. Aalst, and A. Medeiros, *Process Mining with the Heuristics Miner-algorithm*. Jan. 2006, vol. 166, pp. 1–34.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 29, NO. 4, 2022
journal homepage: www.mais-journal.ru

THEORY OF DATA

Detecting Mentions of Green Practices in Social Media Based on Text Classification

A. V. Glazkova[1], O. V. Zakharova[1], A. V. Zakharov[1], N. N. Moskvina[1], T. R. Enikeev[2], A. N. Hodyrev[1], V. K. Borovinskiy[1], I. N. Pupysheva[1]

DOI: 10.18255/1818-1015-2022-4-316-332

[1]University of Tyumen, 6 Volodarskogo str., Tyumen 625003, Russia.
[2]Novosibirsk State University, 1 Pirogova str., Novosibirsk 630090, Russia.

MSC2020: 68T50
Research article
Full text in Russian

Received October 6, 2022
After revision November 11, 2022
Accepted November 16, 2022

The paper is devoted to the task of searching for mentions of green practices in social media texts. The relevance of this task is dictated by the need to expand existing knowledge about the use of green practices in society and the spread of existing green practices. This paper uses a text corpus consisting of the texts published on the environmental communities of the VKontakte social network. The corpus is equipped with an expert markup of the mention of nine types of green practices. As part of this work, a semi-automatic approach is proposed to the collection of additional texts to reduce the class imbalance in the corpus. The approach includes the following steps: detecting the most frequent words for each practice type; automatic collecting texts in social media that contain the detected frequent words; expert verification and filtering of collected texts. The four machine learning models are compared to find the mentions of green practices on the two variants of the corpus: original and augmented using the proposed approach. Among the listed models, the highest averaged F1-score (81.32%) was achieved by Conversational RuBERT fine-tuned on the augmented corpus. Conversational RuBERT model was chosen for the implementation of the application prototype. The main function of the prototype is to detect the presence of the mention of nine types of green practices in the text. The prototype is implemented in the form of the Telegram chatbot.

Keywords: text classification; social network analysis; machine learning; BERT; green practices; natural language processing

INFORMATION ABOUT THE AUTHORS

Anna Valerevna Glazkova correspondence author	orcid.org/0000-0001-8409-6457. E-mail: a.v.glazkova@utmn.ru PhD, Associate Professor of the Department of Software.
Olga Vladimirovna Zakharova	orcid.org/0000-0002-1404-4915. E-mail: o.v.zakharova@utmn.ru PhD, Head of Project Office at Green Solutions Lab, Associate Professor, Department of State and Municipal Administration.
Anton Viktorovich Zakharov	orcid.org/0000-0002-0093-049X. E-mail: a.v.zakharov@utmn.ru PhD, Chief Scientific Officer, Department of the Ishim Pedagogical Institute.
Natalya Nikolayevna Moskvina	orcid.org/0000-0001-5198-276X. E-mail: n.n.moskvina@utmn.ru PhD, Associate Professor, Department of Physical Geography and Ecology.
Timur Ruslanovich Enikeev	orcid.org/0000-0001-8195-1278. E-mail: t.enikeev@g.nsu.ru Student.
Arseniy Nikolaevich Hodyrev	orcid.org/0000-0001-7151-9852. E-mail: stud0000247809@study.utmn.ru Student.
Vsevolod Konstantinovich Borovinskiy	orcid.org/0000-0001-6193-6548. E-mail: stud0000224807@study.utmn.ru Student.
Irina Nikolayevna Pupysheva	orcid.org/0000-0003-2870-4870. E-mail: i.n.pupysheva@utmn.ru PhD, Associate Professor of the Department of Philosophy.

Funding: The work was carried out during the Big Mathematical Workshop of the Mathematical Center in Akademgorodok.

For citation: A. V. Glazkova, O. V. Zakharova, A. V. Zakharov, N. N. Moskvina, T. R. Enikeev, A. N. Hodyrev, V. K. Borovinskiy, and I. N. Pupysheva, "Detecting Mentions of Green Practices in Social Media Based on Text Classification", *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 316-332, 2022.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 29, № 4, 2022
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Поиск упоминаний экологических практик в социальных сетях с помощью методов классификации текстов

А. В. Глазкова[1], О. В. Захарова[1], А. В. Захаров[1], Н. Н. Москвина[1], Т. Р. Еникеев[2], А. Н. Ходырев[1], В. К. Боровинский[1], И. Н. Пупышева[1]

DOI: 10.18255/1818-1015-2022-4-316-332

[1]Тюменский государственный университет, ул. Володарского, д. 6, г. Тюмень, 625003 Россия.
[2]Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090 Россия.

УДК 004.912
Научная статья
Полный текст на русском языке

Получена 6 октября 2022 г.
После доработки 11 ноября 2022 г.
Принята к публикации 16 ноября 2022 г.

Работа посвящена решению задачи поиска упоминаний экологических практик в текстах социальных сетей. Авторами составлен корпус текстов экологических сообществ социальной сети ВКонтакте, снабженный экспертной разметкой упоминаний девяти видов экологических практик. Предложен полуавтоматический подход к сбору дополнительных текстов для уменьшения несбалансированности видов экологических практик, представленных в корпусе. Подход включает в себя следующие этапы: определение наиболее частотных слов, характеризующих упоминания практик; автоматический сбор текстов, включающих в себя найденные частотные слова; экспертная проверка и фильтрация собранных текстов. Проведено сравнение четырех моделей машинного обучения для поиска упоминаний практик на двух вариантах корпуса: исходном и дополненном. Лучший усредненный показатель F-меры (81.32%) достигнут моделью Conversational RuBERT, дообученной на текстах дополненного корпуса. Данная модель выбрана в качестве основы для реализации прототипа приложения для поиска упоминаний экологических практик, реализованного в форме чат-бота Telegram.

Ключевые слова: классификация текстов; анализ социальных сетей; машинное обучение; BERT; экологические практики; обработка естественного языка

ИНФОРМАЦИЯ ОБ АВТОРАХ

Анна Валерьевна Глазкова автор для корреспонденции	orcid.org/0000-0001-8409-6457. E-mail: a.v.glazkova@utmn.ru канд. тех. наук, оцент кафедры программного обеспечения.
Ольга Владимировна Захарова	orcid.org/0000-0002-1404-4915. E-mail: o.v.zakharova@utmn.ru канд. филос. наук, руководитель проектного офиса Green Solutions Lab, доцент кафедры государственного и муниципального управления.
Антон Викторович Захаров	orcid.org/0000-0002-0093-049X. E-mail: a.v.zakharov@utmn.ru канд. пед. наук, нач. научного отд. Ишимского пед. института им. П. П. Ершова.
Наталья Николаевна Москвина	orcid.org/0000-0001-5198-276X. E-mail: n.n.moskvina@utmn.ru канд. геогр. наук, доцент кафедры физической географии и экологии.
Тимур Русланович Еникеев	orcid.org/0000-0001-8195-1278. E-mail: t.enikeev@g.nsu.ru студент.
Арсений Николаевич Ходырев	orcid.org/0000-0001-7151-9852. E-mail: stud0000247809@study.utmn.ru студент.
Всеволод Константинович Боровинский	orcid.org/0000-0001-6193-6548. E-mail: stud0000224807@study.utmn.ru студент.
Ирина Николаевна Пупышева	orcid.org/0000-0003-2870-4870. E-mail: i.n.pupysheva@utmn.ru канд. филос. наук, доцент кафедры философии.

Финансирование: Исследование выполнено в рамках работы на Большой математической мастерской, организованной Математическим центром в Академгородке в 2022 году.

Для цитирования: A. V. Glazkova, O. V. Zakharova, A. V. Zakharov, N. N. Moskvina, T. R. Enikeev, A. N. Hodyrev, V. K. Borovinskiy, and I. N. Pupysheva, “Detecting Mentions of Green Practices in Social Media Based on Text Classification”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 316-332, 2022.

Введение

В условиях ухудшающейся экологической ситуации и дефицита ресурсов необходимо активно привлекать общество к социальным *экологическим (зеленым) практикам*, то есть к повседневным действиям, направленным на гармонизацию отношений человека и его окружающей среды [1]. Согласно данным Всероссийского центра изучения общественного мнения, некоторые из этих действий, способные повлиять на экологическую ситуацию, не только мало распространены, но и остаются практически незамеченными [2, 3]. Привлечение общества к экологическим практикам возможно, например, путем масштабирования уже имеющихся на сегодняшний день экологических практик, особенно тех, которые направлены на сокращение потребления, а значит, на сокращение используемых ресурсов и производимых загрязнений. В зависимости от того, как экологические практики влияют на потребление, их можно разделить на два типа: адаптационные и трансформационные. *Адаптационными* практиками будем называть практики, которые являются реакцией общества на ухудшающуюся экологическую обстановку, но не предполагают сокращения потребления, а *трансформационными* – практики, которые рассчитаны на сокращение производства товаров и услуг и потребления обществом вещества и энергии.

Чтобы эффективно внедрять экологические практики, необходимо обладать определённой информационной базой о том, какие экологические практики уже существуют в обществе, насколько распространены среди них те, которые ведут к сокращению потребления, кто является инициатором подобных практик, кто их поддерживает и так далее. Однако на сегодняшний день знаний о распространенности экологических практик очень мало, поскольку сбор требуемого объема информации традиционными социологическими методами (анкетирование, интервью) является очень трудоемким и занимает много времени [4]. Тем не менее, в социальных сетях в настоящее время сформирован значительный объем неструктурированной текстовой информации, связанной с экологической тематикой. Автоматический анализ текстов экологических сообществ в социальных сетях позволил бы собрать и структурировать большое количество текстовых данных в рассматриваемой предметной области, ускорить их обработку и сделать выводы о распространенности тех или иных видов практик. В связи с этим возникает необходимость разработки методов автоматического получения информации об экологических практиках в социальных сетях. В данной работе описывается подход к автоматическому поиску упоминаний экологических практик в текстах социальных сетей с помощью методов классификации текстов. Авторами представлен корпус текстов социальных сетей, снабженный экспертной разметкой девяти видов экологических практик. Предлагается подход к полуавтоматическому дополнению исходного корпуса текстов для уменьшения дисбаланса количества различных видов практик. Приводятся и обсуждаются результаты сравнения нескольких моделей машинного обучения для автоматического поиска упоминаний экологических практик. Одним из результатов работы является прототип приложения для поиска упоминаний экологических практик в текстах социальных сетей, реализованный в форме чат-бота Telegram.

Работа структурирована следующим образом. Раздел 1 содержит обзор текущего состояния области классификации текстов применительно к анализу текстов социальных сетей. В разделе 2 приводится постановка задачи. Раздел 3 содержит описание используемого корпуса текстов. В разделе 4 приводится перечень использованных моделей машинного обучения. Описание полученных результатов содержится в разделе 5.

1. Обзор смежных работ

Данная работа связана с анализом текстов, размещенных в социальных сетях, и, в частности, их классификацией с применением методов машинного обучения, обработки естественного языка и компьютерной лингвистики. Открытость и разнообразие текстовых данных, размещенных в социальных сетях, предоставляет широкие возможности для изучения общественного мнения

с помощью инструментов компьютерного анализа и позволяет анализировать пути распространения социально значимой информации в онлайн-источниках [5]. Таким образом, социальные сети служат своеобразным индикатором общественных взглядов и тенденций. Изучение контента социальных сетей является важной задачей для научного, политического и коммерческого сообществ [6], что делает автоматическую обработку и анализ постов актуальной тематикой исследований в области компьютерных наук.

Эффективность классификации текстов в области социальных сетей росла параллельно развитию методологии обработки естественного языка (natural language processing). Для решения данной задачи привлекались различные методы, начиная от классификаторов, основанных на применении правил, и заканчивая современными моделями, базирующимися на использовании глубоких нейронных сетей [7]. На сегодняшний день большинство задач классификации текстов решаются с использованием методов машинного обучения и, в частности, глубокого обучения.

Среди традиционных методов машинного обучения наивный байесовский классификатор и логистическая регрессия применяются для анализа тональности (sentiment analysis) постов в Twitter [8, 9] и «риторики вражды» (hate speech detection) [10, 11]. Также для классификации текстов социальных сетей широко используются метод ближайших соседей, метод опорных векторов и случайный лес (например, в работах [12—14]). Среди нейросетевых моделей распространено использование сетей долгой краткосрочной памяти (Long Short-Term Memory, LSTM) [15], которые применялись для анализа тональности [16—18], поиска оскорбительного контента [19, 20] и недостоверной («фейковой») информации [21, 22], а также сверточных нейронных сетей (Convolutional Neural Networks, CNN), использование которых для задачи классификации текстов было впервые предложено в статье [23]. В частности, сверточные нейронные сети применялись для классификации постов социальных сетей в работах [24, 25]. В работах [26, 27] были предложены подходы к гибридному использованию LSTM и CNN. На сегодняшний день, наиболее высокое качество во многих задачах классификации текстов показывают нейросетевые модели, основанные на архитектуре Transformer [28] и, в частности, на использовании лингвистических моделей Bidirectional Encoder Representations from Transformers (BERT) [29], RoBERTa [30] и их модификаций. Так, в ряде соревнований по машинному обучению, связанных с тематикой классификации текстов социальных сетей и проводимых в рамках крупнейших конференций, лучшие результаты были получены с помощью BERT и ее вариаций (например, [31—33]).

В течение последних лет было опубликовано большое количество исследований, связанных с классификацией текстов социальных сетей русскоязычного сегмента Интернета. Авторами данных исследований представлено множество разнообразных подходов. Так, в работах [34—36] рассматриваются подходы к автоматической классификации постов в Twitter с помощью словарей. В статьях [37—39] описываются результаты применения методов машинного обучения к задаче классификации текстов социальных сетей. В работах [40, 41] оценивается эффективность использования признаков различной природы для классификации постов. Подходы, основанные на применении глубоких нейронных сетей, представлены, в частности, в статьях [42—45].

Помимо классификации постов социальных сетей, данная работа также связана с анализом текстов экологической тематики. Ряд работ в данной области посвящен библиометрическому анализу научных текстов для выявления основных трендов экологической повестки [46]. В частности, в работе [47] представлены результаты обширного частотного анализа текстов экологических журналов для выявления динамики словоупотребления тех или иных терминов. В статье [48] проведена кластеризация аннотаций статей экологической тематики. В работах [49—51] представлены результаты частотного анализа публикаций в более узких предметных областях. Некоторые исследования посвящены применению методов машинного обучения в области анализа экологических текстов. Так, в работе [52] предложен подход к автоматическому подбору статей по тематике

биоразнообразия с помощью логистической регрессии и сверточных нейронных сетей. Авторы получили достаточно высокое качество бинарной классификации (ROC AUC ≥ 98%) на датасете, содержащем заголовки и аннотации статей. При этом эксперименты, проведенные в указанной работе, показали, что на датасетах сравнительно небольшой размерности логистическая регрессия и нейронная сеть продемонстрировали близкие показатели качества. В работе [53] предлагается модель для автоматического извлечения таксономических категорий на основе нейронных сетей с Transformer-архитектурой. Предложенная модель продемонстрировала лучшее показатели F-меры в сравнении с другими моделями для распознавания таксономических категорий на двух корпусах текстов (> 75% для COPIOUS [54] и > 88% для Bacteria Biotope [55]). Однако качество на тестовой выборке, содержащей тексты биомедицинской тематики, оказалось существенно ниже, что позволило сделать вывод о важности разработки предметно-ориентированных моделей для анализа биомедицинских и экологических текстов.

Обзор смежных работ показывает, что недавние достижения в области обработки естественного языка демонстрируют высокое качество автоматического анализа текстов для решения множества практических задач, в том числе задачи классификации текстов социальных сетей. Технологии искусственного интеллекта используются в различных областях научного знания. Тем не менее, автоматический анализ естественного языка в области экологических исследований, представлен в настоящее время преимущественно в форме частотного анализа и лишь точечно в виде моделей машинного обучения. Развитие методологии анализа текстов экологической тематики, основанной на применении машинного обучения, представляется перспективной задачей, решение которой служит преодолению существующего пробела в области автоматического анализа экологических текстов.

2. Постановка задачи

Целью данной работы является разработка подхода к автоматическому поиску упоминаний экологических практик в текстах социальных сетей. При наличии нескольких видов экологических практик и размеченного корпуса текстов задача автоматического поиска упоминаний экологических практик в тексте на естественном языке может быть решена как задача классификации. В таком случае задача может быть представлена в формализованном виде следующим образом. Дано множество текстов $T = \{t_1, t_2, ..., t_n\}$ и множество экологических практик $P = \{p_1, p_2, ..., p_m\}$. Требуется найти решающую функцию, приближающую неизвестную целевую зависимость $F : T \longrightarrow 2^P$, значения которой известны только на обучающей выборке. При этом каждому тексту $t_i \in T$, $1 \leq i \leq n$, соответствует некое подмножество экологических практик $P_i \subseteq P$, то есть один текст может содержать упоминания нескольких практик или не содержать упоминаний практик.

В данной работе использован подход, основанный на разбиении задачи классификации с несколькими метками (multi-label classification) на бинарные по схеме «один против остальных» (one-vs-rest). В таком случае задача с несколькими метками преобразуется в m задач бинарной классификации (m – количество рассматриваемых практик), целью каждой из которых является определение класса, к которому относится текст $t_i \in T$. В бинарной постановке каждый классификатор определяет наличие в тексте упоминания одной из экологических практик. Преимуществами подхода «один против остальных» являются его вычислительная эффективность за счет использования бинарных классификаторов и интерпретируемость, то есть возможность получить информацию о каждом классе в отдельности, использовав соответствующий классификатор.

3. Корпус текстов

Для решения задачи поиска упоминаний экологических практик в текстах социальных сетей был использован корпус постов экологических сообществ социальной сети ВКонтакте. Корпус включает в себя посты шести крупнейших экологических сообществ Тюменской области, собранные

в период с января по июнь 2021 года с помощью инструмента VK API[1]. Корпус не включает в себя посты, не содержащие текстовой информации, а также дублирующиеся посты. Общий объем используемого текстового корпуса – 1987 текстов. Корпус также снабжен информацией о количестве комментариев и лайков для каждого поста.

3.1. Разметка корпуса

Используемый в работе текстовый корпус снабжен экспертной разметкой упоминаний экологических практик, включающей в себя практики следующих видов:

- адаптационные практики:
  - сортировать отходы (P1);
  - изучать маркировку товаров (P2);
  - перерабатывать отходы (P3);
  - подписывать петиции (P4);
- трансформационные практики:
  - отказываться от покупок (P5);
  - обменивать (P6);
  - совместно использовать (P7);
  - продвигать ответственное потребление (P8);
  - ремонтировать (P9).

Разметка представляет собой выделение фрагмента, содержащего упоминание практики, открывающим и закрывающим тэгами в соответствии со следующей схемой:

<номер практики>фрагмент, содержащий упоминание практики</номер практики>.

Например, в тексте «На улице холодает, чтобы <3>ускорить свою сдачу сырья</3>, не забывайте заранее <1>максимально его сортировать и подготавливать к сдаче</1>» выделены фрагменты с упоминаниями практик с номерами 3 и 1 («перерабатывать отходы» и «сортировать отходы» соответственно).

Разметка корпуса проводилась двумя экспертами из Тюменского государственного университета, имеющими опыт в области изучения экологических практик. На первом этапе разметки от экспертов требовалось независимо друг от друга выделить в постах максимально короткие, но семантически полные упоминания экологических практик. На втором этапе проводилась проверка и корректировка разметки. Консенсус в случае расхождения между двумя выполненными разметками достигался в ходе дискуссии обоими экспертами.

3.2. Количественный анализ корпуса

На рисунке 1 представлено распределение количества постов, содержащих упоминания экологических практик, в исходном корпусе текстов. Наиболее упоминаемыми практиками в корпусе являются сортировка отходов (37% от общего количества текстов, содержащих упоминания экологических практик), переработка отходов (27.3%) и продвижение ответственного потребления (17.2%). При этом для ряда практик корпус содержит лишь единичные случаи упоминания. В частности, миноритарными являются практики совместного использования (3 текста, содержащих упоминания, или 0.2% от общего количества текстов, содержащих упоминания экологических практик) и ремонта (11 текстов, 0.6%). Поскольку количество текстов, упоминающих разные практики, значительно различается, разработка классификаторов требует предварительного принятия мер по балансировке размеров классов.

Таблица 1 содержит основные количественные характеристики корпуса текстов. Как видно из данных, представленных в таблице, около трети постов экологических сообществ, входящих в состав корпуса, не содержит упоминаний экологических практик (34.17%), и примерно такое же

[1] https://dev.vk.com/reference

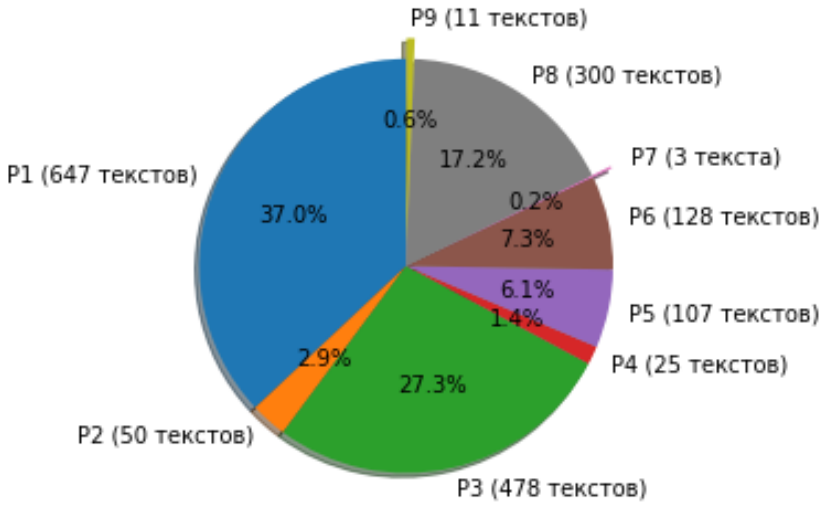
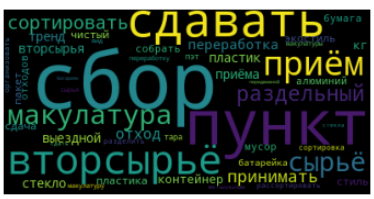
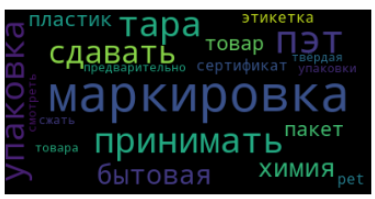


Fig. 1. The distribution of the number of texts containing green practice mentions in the corpus

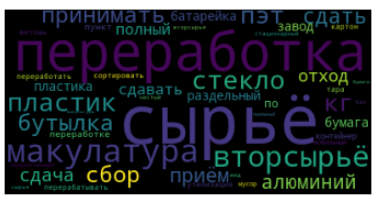
Рис. 1. Распределение количества текстов, содержащих упоминания экологических практик, в корпусе



P1)

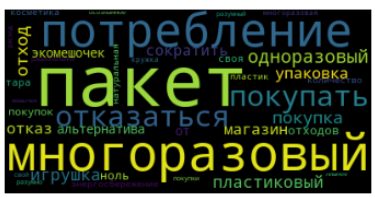


P2)

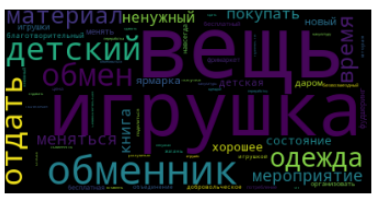


P3)

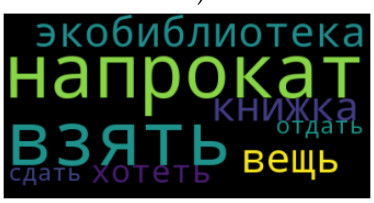
P4)



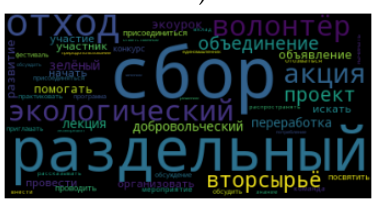
P5)



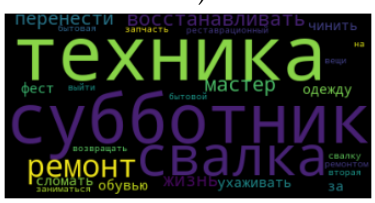
P6)



P7)



P8)



P9)

Fig. 2. The visual representation of the word frequency in the fragments of posts containing the mentions of green practices

Рис. 2. Визуальное представление частотности слов во фрагментах постов, содержащих упоминания экологических практик

количество постов (31%) содержит более одного упоминания. На рисунке 2 в виде «облаков слов» показано визуальное представление частотности слов, входящих в фрагменты постов, содержащих упоминания экологических практик. Эмпирический анализ данных визуальных представлений позволяет сделать выводы о том, что высокочастотные слова для некоторых практик в значительной мере совпадают (в частности, для практик сортировки отходов и переработки отходов).

Table 1. The quantitative characteristics of the corpus

Таблица 1. Количественные характеристики корпуса

Характеристика	Значение
Среднее количество символов в посте	761.67
Среднее количество слов в посте	113.74
Количество постов	1987
Количество постов, не содержащих упоминаний экологических практик	679
Количество постов, содержащих более одного упоминания экологических практик	616

3.3. Дополнение корпуса

Поскольку исходный корпус текстов являлся несбалансированным и упоминания некоторых практик встречались лишь в виде единичных примеров, было решено провести сбор дополнительных текстов для расширения исходного датасета и укрупнения миноритарных категорий постов. В качестве источника дополнительных текстов использовалась социальная сеть ВКонтакте. Так как посты, содержащие упоминания искомых миноритарных экологических практик, являются достаточно редкими, и частотные слова, описывающие различные практики, в некоторой мере пересекаются, не представляется возможным полностью автоматизировать процесс поиска дополнительных текстов. Исходя из этого, в данной работе был использован следующий подход к полуавтоматическому дополнению исходного корпуса текстов.

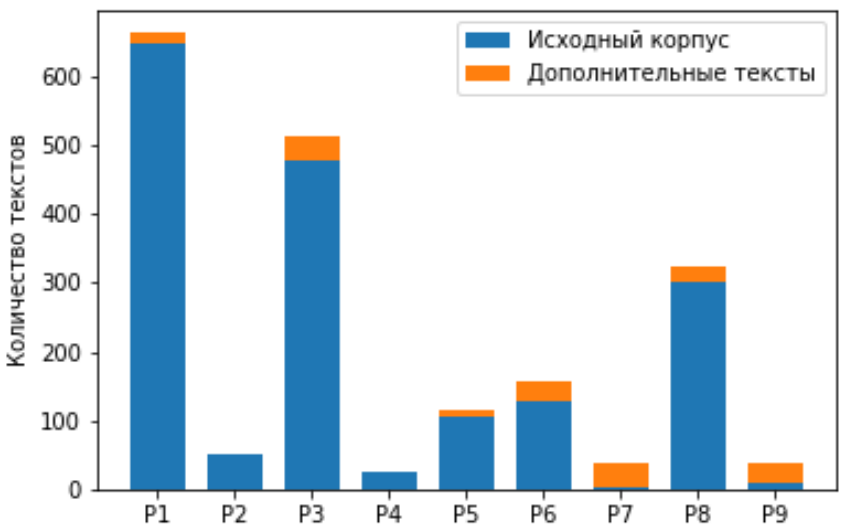


Fig. 3. The number of texts per green practice in the original and augmented corpora

Рис. 3. Количество текстов для каждой экологической практики в исходном и дополненном корпусах

1. На первом этапе для каждой экологической практики были составлены списки слов, наиболее частотных внутри фрагментов, содержащих упоминания этой практики.
2. Далее, на основании выявленных частотных слов с помощью сервиса автоматического поиска целевой аудитории в социальных сетях был получен набор постов, содержащих сочетания частотных слов, присутствующих в упоминаниях миноритарных экологических практик. В данной работе использовался сервис TargetHunter[2].
3. На третьем этапе было проведено экспертное оценивание автоматически собранных текстов с целью уточнения наличия в них упоминаний экологических практик. При этом эксперты проверяли наличие не только миноритарных, но и других видов практик.

На рисунке 3 представлено соотношение количества текстов, содержащих упоминания различных экологических практик, в исходном и дополненном датасетах. Поскольку в ходе экспертного оценивания большая часть автоматически собранных текстов была отмечена как не содержащая упоминаний миноритарных практик, дисбаланс классов в дополненном корпусе сохраняется. Несмотря на это, количество текстов, включающих в себя упоминания миноритарных практик было увеличено в несколько раз.

4. Модели

Данный раздел содержит описание моделей машинного обучения для поиска упоминаний экологических практик в текстах социальных сетей с помощью методов машинного обучения. Для экспериментов по проведению бинарной классификации текстов были использованы несколько моделей машинного обучения. Во-первых, были применены три широко используемых алгоритма машинного обучения: *логистическая регрессия*, *случайный лес* и *метод опорных векторов*. Данные классификаторы были реализованы с помощью библиотеки Scikit-learn [56] и языка программирования Python. При реализации случайного леса было использовано количество деревьев, равное 50. Для остальных гиперпараметров были выбраны настройки по умолчанию. Начальное число, используемое генератором случайных чисел (random_state), было зафиксировано в значении 0. В качестве модели для представления текстов была выбрана модель Bag-of-words («мешок слов») с максимальным размером словаря (max_features), равным 5000. Во-вторых, в ходе экспериментов была использована русскоязычная лингвистическая модель *Conversational RuBERT*[3] [57] (далее – RuBERT), которая представляет собой мультиязычную BERT [29], дополнительно обученную на текстах русскоязычной Википедии, новостных русскоязычных текстах, а также корпусах OpenSubtitles [58], Dirty, Pikabu и Taiga [59]. Для каждой экологической практики проводилось дообучение (fine-tuning) модели RuBERT в течение трех эпох с использованием следующих гиперпараметров: максимальная длина входной последовательности – 256 токенов, размер батча – 8, скорость обучения – 4e-5. Для реализации использовалась библиотека Simple Transformers[4] для языка программирования Python.

5. Результаты

Оценка качества моделей проводилась отдельно на исходном и дополненном датасетах. При этом была использована стратифицированная кросс-валидация с разбиением данных на три части. Для представления результатов использовалась F-мера с усреднением по обоим классам (macro averaging). Результаты моделей представлены в таблице 2, лучший результат для каждой практики выделен полужирным шрифтом. Рисунок 4 иллюстрирует различия в результатах, полученных до и после пополнения корпуса.

Результаты, полученные на исходном корпусе для видов практик, меньше всего представленных в датасете, являются ожидаемо низкими. При этом стоит отметить, что для ряда практик на

[2] https://targethunter.ru/
[3] https://huggingface.co/DeepPavlov/rubert-base-cased-conversational
[4] https://simpletransformers.ai/

исходных данных традиционные методы машинного обучения показали лучшие результаты в сравнении с RuBERT (в частности, практики отказа от покупок и обмена). В целом, достаточно высокое качество (более 75%) на исходном корпусе было достигнуто только для трех наиболее широко представленных экологических практик (сортировка отходов, переработка отходов и продвижение ответственного потребления) и для практики изучения маркировки товаров (вероятно, в связи с наличием специфических слов, входящих в фрагменты, упоминающие данную практику).

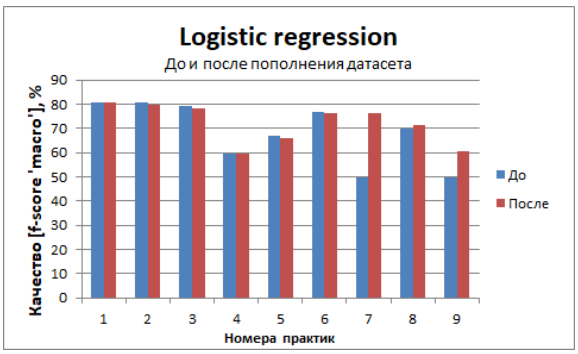
Несмотря на сохраняющуюся несбалансированность корпуса в дополненном датасете, результаты, полученные классификаторами в большинстве случаев значительно улучшились. При этом для большей части практик лучший показатель качества достигнут с помощью модели RuBERT. Так, наиболее высокое значение F-меры получено для практик переработки отходов (82.57% на дополненном датасете, прирост 4.82% по сравнению с исходными данными) и ремонта (82.29% на дополненном датасете, прирост 32.42%). Поскольку при обучении каждого бинарного классификатора используются тексты, относящиеся к разным видам практик, добавление дополнительных текстов в корпус позволило улучшить качество классификации не только при поиске тех практик, примеры упоминаний которых были добавлены при дополнении корпуса, но и при поиске практик, количество упоминаний которых не увеличилось или увеличилось незначительно при пополнении датасета (например, практики подписания петиций и отказа от покупок). В целом, добавление дополнительных текстов в корпус значительно повысило качество классификаторов. При этом по сравнению с результатами на исходном корпусе в случае модели RuBERT показатели качества выросли для восьми экологических практик из девяти рассмотренных, в случае метода опорных векторов – для семи, в случае логической регрессии – для пяти и в случае случайного леса – для двух. Средний прирост качества составил 18.73% для RuBERT, 5.28% для метода опорных векторов, 3.95% для логистической регрессии и 1.15% для случайного леса. Таким образом, эксперименты, проведенные на данном текстовом корпусе, показали, что RuBERT демонстрирует достаточно высокое качество классификации даже при наличии небольшого количества примеров в классах.

Table 2. The results for comparing models (F-score, %): LR – Logistic Regression, RF – Random Forest, SVM – Support Vector Machines, BERT – Conversational RuBERT

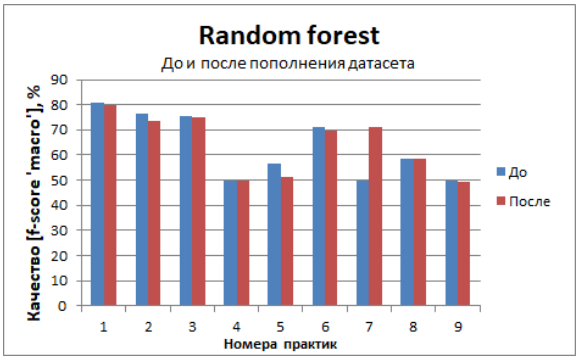
Таблица 2. Результаты сравнения моделей (F-мера, %): LR – логистическая регрессия, RF – случайный лес, SVM – метод опорных векторов, BERT – Conversational RuBERT

Практика	Исходный корпус					Дополненный корпус				
	Количество постов	LR	RF	SVM	BERT	Количество постов	LR	RF	SVM	BERT
P1	647	80.48	80.59	77.13	**83.2**	663	80.68	80.06	78.33	81.15
P2	50	80.56	76.41	**82.9**	78.18	51	79.77	73.53	81.86	80.63
P3	478	79.2	75.37	75.02	77.75	511	78.12	75.2	76	**82.57**
P4	25	59.66	49.62	74.75	52.22	25	59.68	49.64	80.57	**81.46**
P5	107	66.87	56.42	65.9	48.65	116	65.99	51.14	68.35	**79.57**
P6	128	76.68	70.91	79.15	48.35	158	76.22	69.83	78.98	**81.26**
P7	3	49.96	49.96	49.96	49.95	40	76.38	71.1	79.13	**81.92**
P8	300	69.89	58.78	70.69	75.14	324	71.31	58.35	70.77	**80.99**
P9	11	49.83	49.83	49.79	49.87	40	60.54	49.43	58.83	**82.29**

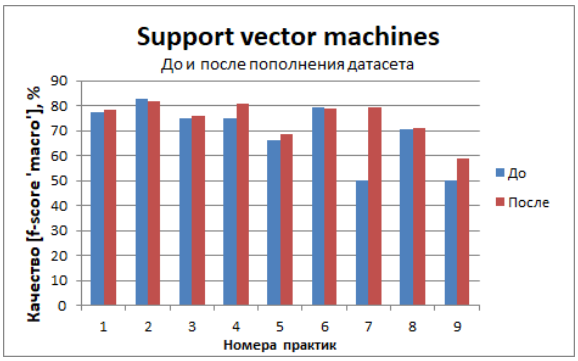
Модель RuBERT, дообученная (fine-tuned) на текстах дополненного корпуса, показала наилучшее среднее качество классификации текстов, содержащих упоминания практик, с помощью подхода «один против остальных» (81.32%). Исходя из этого, данная модель была выбрана для реализации прототипа приложения для поиска упоминаний экологических практик в текстах социальных



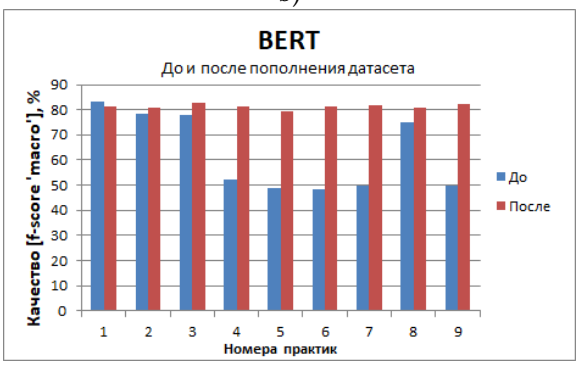
a)



b)



c)



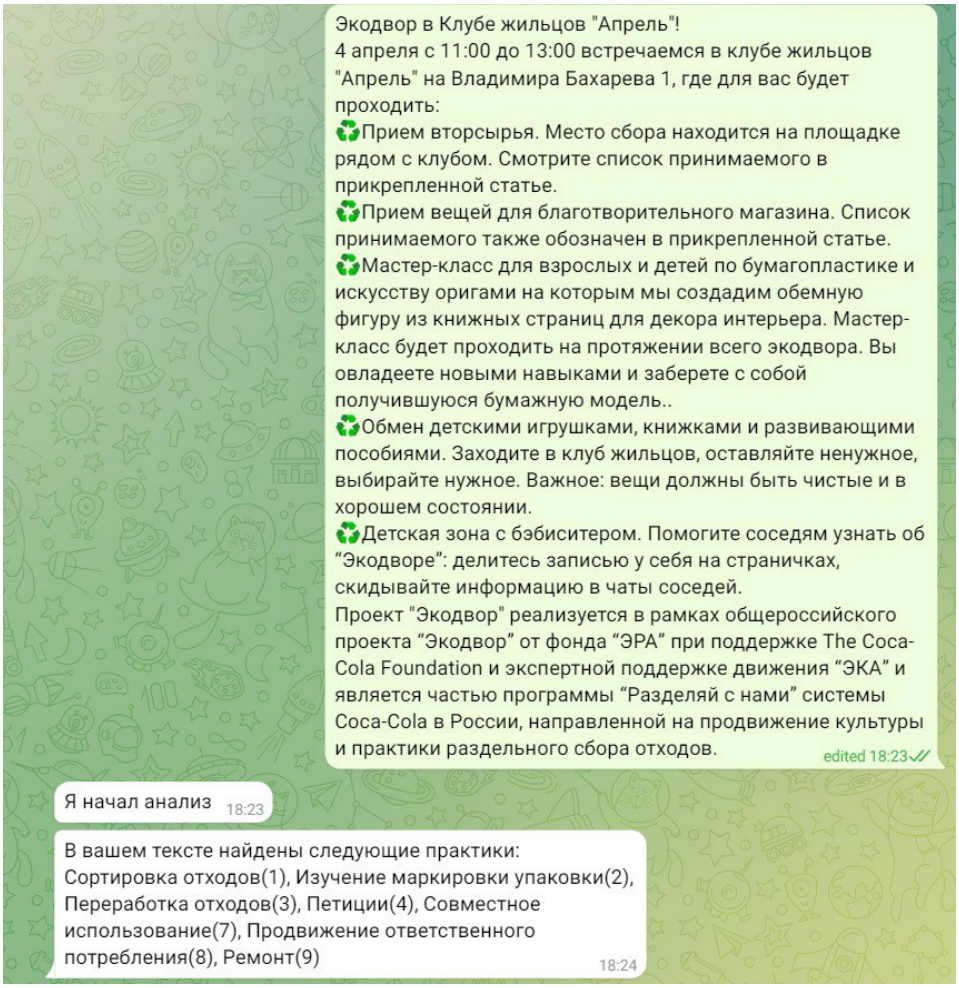
d)

Fig. 4. The difference of the results obtained for the original and augmented corpora: a) Logistic Regression, b) Random Forest, c) Support Vector Machines, d) Conversational RuBERT

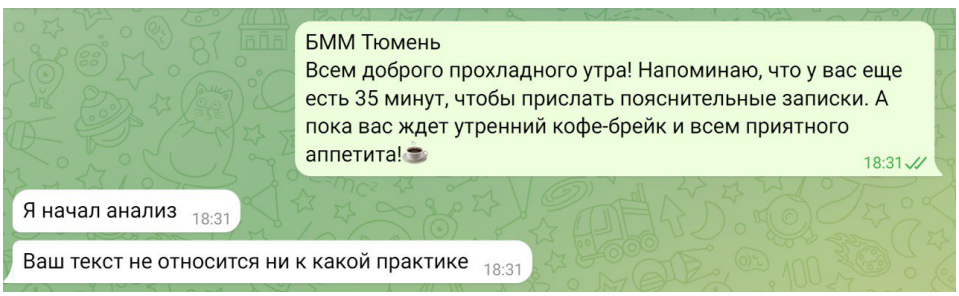
Рис. 4. Различие в результатах, полученных на исходном и дополненном корпусах: a) логистическая регрессия, b) случайный лес, c) метод опорных векторов, d) Conversational RuBERT

сетей. Прототип приложения[5], реализованный в форме чат-бота Telegram, определяет наличие в тексте упоминаний рассмотренных в данной работе девяти типов экологических практик. Когда текст отправляется боту, пользователь получает сообщение о том, что начался анализ текста, в свою очередь бот поочередно запускает девять обученных классификаторов, что занимает в сумме около 63 секунд. В дальнейшем планируется проведение мероприятий по оптимизации работы сервиса для уменьшения временных затрат на получение прогнозов классификаторов. По окончании анализа выводится информация о том, какие практики были найдены в тексте (рисунок 5). Помимо основного функционала, чат-бот содержит кнопки «О проекте», «Команда проекта». Из каждого состояния «беседы» с ботом можно вернуться в главное меню или на шаг назад. Для реализации чат-бота был выбран язык программирования Python и библиотека pyTelegramBotAPI[6].

[5] https://t.me/dlya_proekta_bot
[6] https://github.com/eternnoir/pyTelegramBotAPI



a)



b)

Fig. 5. The output of the prototype: a) the mentions of seven green practices have been found, b) no practice mentions have been found

Рис. 5. Результаты работы прототипа приложения: а) в тексте найдены упоминания семи экологических практик, b) в тексте не найдены упоминания практик

Заключение

Данная работа посвящена решению задачи поиска упоминаний экологических практик в текстах социальных сетей. Автоматизация поиска упоминаний экологических практик позволит, анализируя большие объемы контента социальных сетей, делать выводы о текущей распространенности различных видов экологических практик, эффективности существующих способов их внедрения

и, как следствие, возможных путях масштабирования уже имеющихся практик. Решение данной задачи является значимым с социальной точки зрения, поскольку экологические практики выполняют функцию важнейшего инструмента защиты окружающей среды, сокращения потребления и переработки отходов.

В рамках данного исследования впервые было проведено сравнение эффективности нескольких моделей машинного обучения для задачи поиска упоминаний экологических практик. Задача поиска экологических практик была сформулирована как задача бинарной классификации текстов с помощью подхода «один против остальных». Работа выполнена на корпусе текстов социальной сети ВКонтакте, снабженном экспертной разметкой упоминаний экологических практик различных типов. Для уменьшения влияния несбалансированности исходных данных был предложен подход к полуавтоматическому дополнению корпуса текстов, заключающийся в выделении наиболее частотных слов для каждой экологической практики, сборе дополнительных текстов, содержащих выделенные частотные слова, с помощью существующих сервисов для анализа социальных сетей и экспертной проверке собранных текстов. В результате сравнения моделей машинного обучения лучшее качество классификации было получено с помощью лингвистической модели Conversational RuBERT, дообученной с помощью дополненного корпуса текстов. Результат работы представлен в виде прототипа приложения – чат-бота Telegram для поиска упоминаний экологических практик в тексте на естественном языке.

В рамках дальнейшей работы будут протестированы подходы к генерации дополнительных текстов для миноритарных видов экологических практик (в частности, с помощью генерации текстов с использованием RuGPT-3[7] и перефразирования [60]) и извлечению точных фрагментов (span detection), содержащих упоминания практик, а также опробованы различные подходы к дообучению моделей (в частности, сопоставительное дообучение по аналогии с работой [61]). Кроме того, дальнейшие планы включают в себя завершение разработки приложения и дополнение его функционала.

References

[1] O. Zakharova, I. Pupysheva, T. Payusova, A. Zakharov, and S. L., “Green Values in Crowdfunding Projects”, *Glocalism*, no. 1, p. 6, 2021. DOI: 10.12893/gjcpi.2021.1.6.

[2] *VCIOM. Jekologicheskaja povestka: za desjat’ mesjacev do vyborov v Gosdumu (analiticheskij doklad). 2020-12-30*, http://www.wciom.ru, Accessed: 2021-03-18.

[3] Y. V. Ermolaeva and M. V. Rybakova, “Civil social practices of waste recycling in Russia (Moscow and Kazan)”, *IIOAB Journal*, vol. 10, no. S1, pp. 153–156, 2019.

[4] O. Zakharova, T. Payusova, I. Akhmedova, and L. Suvorova, “Green Practices: Ways to Investigation”, *Sotsiologicheskie issledovaniya*, no. 4, pp. 25–36, 2021. DOI: 10.31857/S013216250012084-5.

[5] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, “Detection and resolution of rumours in social media: A survey”, *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–36, 2018. DOI: 10.1145/3161603.

[6] D. Rogers, A. Preece, M. Innes, and I. Spasić, “Real-time text classification of user-generated content on social media: Systematic review”, *IEEE Transactions on Computational Social Systems*, 2021. DOI: 10.1109/TCSS.2021.3120138.

[7] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, “A Survey on Text Classification: From Traditional to Deep Learning”, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 2, pp. 1–41, 2022. DOI: 10.1145/3495162.

[7] https://github.com/ai-forever/ru-gpts

[8] F. C. Permana, Y. Rosmansyah, and A. S. Abdullah, "Naive Bayes as opinion classifier to evaluate students satisfaction based on student sentiment in Twitter Social Media", in *Journal of Physics: Conference Series*, IOP Publishing, vol. 893, 2017, p. 012 051. DOI: 10.1088/1742-6596/893/1/012051.

[9] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, "Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naive Bayes, decision tree, and random forest algorithm", *Procedia Computer Science*, vol. 161, pp. 765–772, 2019. DOI: 10.1016/j.procs.2019.11.181.

[10] N. R. Fatahillah, P. Suryati, and C. Haryawan, "Implementation of Naive Bayes classifier algorithm on social media (Twitter) to the teaching of Indonesian hate speech", in *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, IEEE, 2017, pp. 128–131. DOI: 10.1109/SIET.2017.8304122.

[11] K. K. Kiilu, G. Okeyo, R. Rimiru, and K. Ogada, "Using Naive Bayes algorithm in detection of hate tweets", *International Journal of Scientific and Research Publications*, vol. 8, no. 3, pp. 99–107, 2018. DOI: 10.29322/IJSRP.8.3.2018.p7517.

[12] Z. Peng, Q. Hu, and J. Dang, "Multi-kernel SVM based depression recognition using social media data", *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 43–57, 2019. DOI: 10.1007/s13042-017-0697-1.

[13] P. Karthika, R. Murugeswari, and R. Manoranjithem, "Sentiment analysis of social media network using random forest algorithm", in *2019 IEEE international conference on intelligent techniques in control, optimization and signal processing (INCOS)*, IEEE, 2019, pp. 1–5. DOI: 10.1109/INCOS45849.2019.8951367.

[14] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM", in *2015 International Conference on Data and Software Engineering (ICoDSE)*, IEEE, 2015, pp. 170–174. DOI: 10.1109/ICODSE.2015.7436992.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis", *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018. DOI: 10.1007/s12559-018-9549-x.

[17] M. Tripathi, "Sentiment analysis of Nepali COVID19 tweets using NB SVM and LSTM", *Journal of Artificial Intelligence*, vol. 3, no. 03, pp. 151–168, 2021. DOI: 0.36548/jaicn.2021.3.001.

[18] R. Monika, S. Deivalakshmi, and B. Janet, "Sentiment analysis of US airlines tweets using LSTM/RNN", in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, IEEE, 2019, pp. 92–95. DOI: 10.1109/IACC48062.2019.8971592.

[19] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets", in *Proceedings of the 26th international conference on World Wide Web companion*, 2017, pp. 759–760. DOI: 10.1145/3041021.3054223.

[20] A. Bisht, A. Singh, H. Bhadauria, J. Virmani, *et al.*, "Detection of hate speech and offensive language in Twitter data using LSTM model", in *Recent trends in image and signal processing in computer vision*, Springer, 2020, pp. 243–264. DOI: 10.1007/978-981-15-2740-1_17.

[21] V. Rupapara, F. Rustam, A. Amaar, P. B. Washington, E. Lee, and I. Ashraf, "Deepfake tweets classification using stacked Bi-LSTM and words embedding", *PeerJ Computer Science*, vol. 7, e745, 2021. DOI: 10.7717/peerj-cs.745.

[22] A. Wani, I. Joshi, S. Khandve, V. Wagh, and R. Joshi, "Evaluating deep learning approaches for COVID19 fake news detection", in *International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation*, Springer, 2021, pp. 153–163. DOI: 10.1007/978-3-030-73696-5_15.

[23] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification", in *Twenty-ninth AAAI conference on artificial intelligence*, 2015. DOI: 10.5555/2886521.2886636.

[24] S. Bansal, "A Mutli-Task Mutlimodal Framework for Tweet Classification Based on CNN (Grand Challenge)", in *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, IEEE, 2020, pp. 456–460. DOI: 10.1109/BigMM50055.2020.00075.

[25] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis", *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021. DOI: 10.1016/j.future.2020.08.005.

[26] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model", in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2016, pp. 225–230. DOI: 10.18653/v1/P16-2037.

[27] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis", in *International cross-domain conference for machine learning and knowledge extraction*, Springer, 2018, pp. 179–191. DOI: 10.1007/978-3-319-99740-7_12.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[30] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach", *arXiv preprint arXiv:1907.11692*, 2019. DOI: 10.48550/arXiv.1907.11692.

[31] A. El Mahdaouy, A. El Mekki, K. Essefar, A. Skiredj, and I. Berrada, "CS-UM6P at SemEval-2022 Task 6: Transformer-based Models for Intended Sarcasm Detection in English and Arabic", in *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 2022, pp. 844–850. DOI: 10.18653/v1/2022.semeval-1.117.

[32] M. Du, S. D. Gollapalli, and S.-K. Ng, "NUS-IDS at CheckThat! 2022: Identifying Check-worthiness of Tweets using CheckthaT5", *Working Notes of CLEF*, 2022.

[33] A. Glazkova, M. Glazkov, and T. Trifonov, "g2tmn at constraint@ aaai2021: exploiting CT-BERT and ensembling learning for COVID-19 fake news detection", in *International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation*, Springer, 2021, pp. 116–127. DOI: 10.1007/978-3-030-73696-5_12.

[34] Y. Rubtsova, "Constructing a corpus for sentiment classification training", *Software & Systems*, no. 1 (109), pp. 72–78, 2015. DOI: 10.15827/0236-235X.109.072-078.

[35] I. Bolshakova and K. Lagutina, "Avtomaticheskaja klassifikacija tekstov na russkom jazyke s pomoshh'ju tonal'nogo slovarja", no. 14, pp. 6–13, 2022.

[36] A. Kotelnikova, D. Paschenko, and E. Razova, "Lexicon-based methods and BERT model for sentiment analysis of Russian text corpora", in *CEUR Workshop Proceedings*, 2021, pp. 73–81.

[37] N. Loukachevitch and Y. Rubtsova, "SentiRuEval-2016: overcoming time gap and data sparsity in tweet sentiment analysis", in *Computational Linguistics and Intellectual Technologies*, 2016, pp. 416–426.

[38] A. Chernyaev, A. Spryiskov, A. Ivashko, and Y. Bidulya, "A rumor detection in Russian tweets", in *International Conference on Speech and Computer*, Springer, 2020, pp. 108–118. DOI: 10.1007/978-3-030-60276-5_11.

[39] E. Mikhalkova, Y. Karyakin, and I. Glukhikh, "Large Scale Retrieval of Social Network Pages by Interests of Their Followers", in *Computational Science – ICCS 2018*, Cham: Springer International Publishing, 2018, pp. 234–246. DOI: 10.1007/978-3-319-93698-7_18.

[40] E. Pronoza, P. Panicheva, O. Koltsova, and P. Rosso, "Detecting ethnicity-targeted hate speech in Russian social media texts", *Information Processing & Management*, vol. 58, no. 6, p. 102 674, 2021, ISSN: 0306-4573. DOI: 10.1016/j.ipm.2021.102674.

[41] K. V. Lagutina, N. S. Lagutina, and E. I. Boychuk, "Text classification by genre based on rhythm features", *Modeling and analysis of information systems*, pp. 280–291, 2021. DOI: 10.18255/1818-1015-2021-3-280-291.

[42] K. Svetlov and K. Platonov, "Sentiment analysis of posts and comments in the accounts of Russian politicians on the social network", in *2019 25th Conference of Open Innovations Association (FRUCT)*, IEEE, 2019, pp. 299–305. DOI: 10.23919/FRUCT48121.2019.8981501.

[43] I. Kozitsin, A. Chkhartishvili, A. Marchenko, D. Norkin, S. Osipov, I. Uteshev, V. Goiko, R. Palkin, and M. Myagkov, "Modeling political preferences of Russian users exemplified by the social network Vkontakte", *Mathematical Models and Computer Simulations*, vol. 12, no. 2, pp. 185–194, 2020. DOI: 10.1134/S2070048220020088.

[44] P. Basina, V. Goiko, E. Petrov, and V. Bakulin, "Classification community publications of the "VKontakte" for assessing the quality of life of the population", *Computational Linguistics and Intellectual Technologies*, p. 18, 2022. DOI: 10.28995/2075-7182-2022-21-1001-1016.

[45] A. Sboev, I. Moloshnikov, A. Naumov, A. Levochkina, and R. Rybka, "The Russian Language Corpus and a Neural Network to Analyse Internet Tweet Reports About COVID-19", *PoS*, vol. DLCP2021, p. 017, 2021. DOI: 10.22323/1.410.0017.

[46] M. J. Farrell, L. Brierley, A. Willoughby, A. Yates, and N. Mideo, "Past and future uses of text mining in ecology and evolution", *Proceedings of the Royal Society B*, vol. 289, no. 1975, p. 20 212 721, 2022. DOI: 10.1098/rspb.2021.2721.

[47] S. C. Anderson, P. R. Elsen, B. B. Hughes, R. K. Tonietto, M. C. Bletz, D. A. Gill, M. A. Holgerson, S. E. Kuebbing, C. McDonough MacKenzie, M. H. Meek, *et al.*, "Trends in ecology and conservation over eight decades", *Frontiers in Ecology and the Environment*, vol. 19, no. 5, pp. 274–282, 2021. DOI: 10.1002/fee.2320.

[48] J. Knott, E. LaRue, S. Ward, E. McCallen, K. Ordonez, F. Wagner, I. Jo, J. Elliott, and S. Fei, "A roadmap for exploring the thematic content of ecology journals", *Ecosphere*, vol. 10, no. 8, e02801, 2019. DOI: 10.1002/ecs2.2801.

[49] F. R. Dayeen, A. S. Sharma, and S. Derrible, "A text mining analysis of the climate change literature in industrial ecology", *Journal of Industrial Ecology*, vol. 24, no. 2, pp. 276–284, 2020. DOI: 10.1111/jiec.12998.

[50] F. Romero-Perdomo, J. D. Carvajalino-Umaña, J. L. Moreno-Gallego, N. Ardila, and M. Á. González-Curbelo, "Research Trends on Climate Change and Circular Economy from a Knowledge Mapping Perspective", *Sustainability*, vol. 14, no. 1, p. 521, 2022. DOI: 10.3390/su14010521.

[51] O. J. Luiz, J. D. Olden, M. J. Kennard, D. A. Crook, M. M. Douglas, T. M. Saunders, and A. J. King, "Trait-based ecology of fishes: A quantitative assessment of literature trends and knowledge gaps using topic modelling", *Fish and Fisheries*, vol. 20, no. 6, pp. 1100–1110, 2019. DOI: 10.1111/faf.12399.

[52] R. Cornford, S. Deinet, A. De Palma, S. L. Hill, L. McRae, B. Pettit, V. Marconi, A. Purvis, and R. Freeman, "Fast, scalable, and automated identification of articles for biodiversity and macroecological datasets", *Global Ecology and Biogeography*, vol. 30, no. 1, pp. 339–347, 2021. DOI: 10.1111/geb.13219.

[53] N. Le Guillarme and W. Thuiller, "TaxoNERD: deep neural models for the recognition of taxonomic entities in the ecological and evolutionary literature", *Methods in Ecology and Evolution*, vol. 13, no. 3, pp. 625–641, 2022. DOI: 10.1111/2041-210X.13778.

[54] N. T. Nguyen, R. S. Gabud, and S. Ananiadou, "COPIOUS: A gold standard corpus of named entities towards extracting species occurrence from biodiversity literature", *Biodiversity data journal*, no. 7, 2019. DOI: 10.3897/BDJ.7.e29626.

[55] R. Bossy, L. Deléger, E. Chaix, M. Ba, and C. Nédellec, "Bacteria biotope at BioNLP open shared tasks 2019", in *Proceedings of the 5th workshop on BioNLP open shared tasks*, 2019, pp. 121–131. DOI: 10.18653/v1/D19-5719.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in Python", *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[57] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for Russian language", in *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 2019, pp. 333–339.

[58] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles", 2016.

[59] T. Shavrina and O. Shapovalova, "To the methodology of corpus construction for machine learning: "Taiga" syntax tree corpus and parser", *Proceedings of the "Corpora"*, pp. 78–84, 2017.

[60] A. Fenogenova, "Russian paraphrasers: Paraphrase with transformers", in *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, 2021, pp. 11–19.

[61] I. Bondarenko, "Contrastive fine-tuning to improve generalization in deep NER", 2022. DOI: 10.28995/2075-7182-2022-21-70-80.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 29, NO. 4, 2022
journal homepage: www.mais-journal.ru

THEORY OF DATA

Classification of Russian Texts by Genres Based on Modern Embeddings and Rhythm

K. V. Lagutina[1]

DOI: 10.18255/1818-1015-2022-4-334-347

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

MSC2020: 68T50
Research article
Full text in Russian

Received August 17, 2022
After revision November 4, 2022
Accepted November 9, 2022

The article investigates modern vector text models for solving the problem of genre classification of Russian-language texts. Models include ELMo embeddings, BERT language model with pre-training and a complex of numerical rhythm features based on lexico-grammatical features. The experiments were carried out on a corpus of 10,000 texts in five genres: novels, scientific articles, reviews, posts from the social network Vkontakte, news from OpenCorpora.
Visualization and analysis of statistics for rhythm features made it possible to identify both the most diverse genres in terms of rhythm: novels and reviews, and the least ones: scientific articles. Subsequently, these genres were classified best with the help of rhythm features and the neural network-classifier LSTM. Clustering and classifying texts by genre using ELMo and BERT embeddings made it possible to separate one genre from another with a small number of errors. The multi-classification F-score reached 99%. The study confirms the efficiency of modern embeddings in the tasks of computational linguistics, and also allows to highlight the advantages and limitations of the complex of rhythm features on the material of genre classification.

Keywords: stylometry; natural language processing; rhythm features; genres; text classification; BERT; ELMo

INFORMATION ABOUT THE AUTHORS

Ksenia Vladimirovna Lagutina
correspondence author
orcid.org/0000-0002-1742-3240. E-mail: lagutinakv@mail.ru
PhD.

Funding: The work is supported by the President of Russian Federation Scholarship for young scientists and postgraduates No. SP-2109.2021.5.

For citation: K. V. Lagutina, “Classification of Russian Texts by Genres Based on Modern Embeddings and Rhythm”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 334-347, 2022.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 29, № 4, 2022
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Классификация русскоязычных текстов по жанрам на основе современных эмбеддингов и ритма

К. В. Лагутина[1]

DOI: 10.18255/1818-1015-2022-4-334-347

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, д. 14, г. Ярославль, 150003 Россия.

УДК 004.912
Научная статья
Полный текст на русском языке

Получена 17 августа 2022 г.
После доработки 4 ноября 2022 г.
Принята к публикации 9 ноября 2022 г.

В статье исследуются современные векторные модели текстов для решения задачи классификации русскоязычных текстов по жанрам. Модели включают эмбеддинги ELMo, языковую модель BERT с предобучением и комплекс числовых ритмических характеристик на основе лексико-грамматических средств. Эксперименты проводились на корпусе из 10 000 текстов пяти жанров: романы, научные статьи, отзывы, посты из социальной сети Вконтакте, новости из OpenCorpora.

Визуализация и анализ статистики для ритмических характеристик позволили выделить как наиболее разнообразные по ритму жанры: романы и отзывы, так и наименее – научные статьи. Именно эти жанры были впоследствии классифицированы лучше всего с помощью ритма и нейросети-классификатора LSTM. Кластеризация и классификация текстов по жанрам с помощью эмбеддингов ELMo и BERT позволила отделить один жанр от другого с небольшим количеством ошибок. F-мера мультиклассификации достигла 99%. Исследование подтверждает эффективность современных эмбеддингов в задачах компьютерной лингвистики, а также позволяет выделить достоинства и ограничения комплекса ритмических характеристик на материале классификации по жанрам.

Ключевые слова: стилометрия; обработка естественного языка; ритмические характеристики; жанры; классификация текстов; BERT; ELMo

ИНФОРМАЦИЯ ОБ АВТОРАХ

Ксения Владимировна Лагутина
автор для корреспонденции

orcid.org/0000-0002-1742-3240. E-mail: lagutinakv@mail.ru
кандидат технических наук.

Финансирование: Работа поддержана стипендией Президента Российской Федерации для молодых ученых и аспирантов, осуществляющих перспективные научные исследования и разработки по приоритетным направлениям модернизации российской экономики: № СП-2109.2021.5.

Для цитирования: K. V. Lagutina, “Classification of Russian Texts by Genres Based on Modern Embeddings and Rhythm”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 334-347, 2022.

Введение

Изучение стилевых особенностей русского языка — важная задача современной российской компьютерной лингвистики. Стилевые характеристики текста отражают в том числе и структурные особенности текстов, поэтому они являются маркерами различных жанров и часто используются для автоматического анализа в этой области [1].

Автоматическая классификация текстов по жанрам является одной из фундаментальных задач обработки текстов. Зная жанр текста, можно эффективнее решить и другие проблемы компьютерной лингвистики: определить часть речи или значение слова или словосочетания, найти подходящий по смыслу запроса документ и т. п. [2]. Как и многие задачи автоматического анализа текстов, для классификации англоязычных текстов по жанрам уже предложено несколько высококачественных решений [3, 4].

Русскоязычные тексты представляют собой более широкое поле для исследований. Учёные часто классифицируют тексты на основе стандартных характеристик или ограничиваются литературными жанрами [5]. Современные эмбеддинги, например, BERT, и вовсе остаются недоисследованными в области анализа жанров.

Автор статьи ставит перед собой задачу классифицировать русскоязычные тексты по жанрам на романы, научные статьи, отзывы, новости и посты в социальной сети Вконтакте с помощью современных числовых характеристик. Характеристики включают в себя эмбеддинги BERT и ELMo, а также разработанный автором ранее комплекс ритмических характеристик [6]. Также в цели исследования входит визуализация числовых характеристик для наглядного анализа жанров и интерпретации результатов классификации.

1. Обзор смежных работ

В научной литературе последних лет активно изучаются возможности предобученных языковых моделей и эмбеддингов, построенных на основе архитектуры Трансформер, а именно, подходы ELMo, GPT и BERT [7, 8]. Среди них ELMo и BERT хорошо подходят не только для генерации текстов, но и для классификационных задач.

Авторы исследования [4] применяют эмбеддинги GloVe, ELMo и BERT для классификации новостей по жанрам и структуре, используя нейронную сеть BiLSTM в качестве классификатора. ELMo и BERT достигают очень близких результатов по качеству: около 80 % точности для мультиклассификации. Эксперименты проводились на 853 англоязычных новостях. Следует отметить, что для достаточно небольшого корпуса текстов достигнуто отличное качество классификации.

Подобные высокие результаты часто достигаются для текстов на английском языке, поскольку для него разработано немало стандартных корпусов, размеченных для решения разнообразных задач компьютерной лигнвистики. Это позволяет проводить масштабные исследования с различными эмбеддингами: word2vec, GloVe, FastText, ELMo, BERT, — и демонстрировать высокую эффективность моделей на их основе сразу для нескольких задач классификации [9]. F-мера для классификации по темам стабильно достигает 80–90 %.

Для национальных языков современные эмбеддинги в области задач классификации также исследованы достаточно хорошо, но для исследования жанра или стиля практически не применяются.

Для русского языка модель RuBERT, являющаяся адаптацией BERT, предобучена и применена для классификации по тональности [10]. F-мера достигла 72 %.

Эмбеддинги BERT и ELMo применялись и участниками научного соревнования RuShiftEval по детекции изменения семантики слов в русскоязычном тексте [11]. Лучшие результаты (коэффициент корреляции Спирмена 80 %) показала модель XLM-RoBERTa, основанная на мультиязычной версии BERT, в комбинации с системой разрешения неоднозначности смысла слова.

Оригинальные версии ELMo и BERT для русского языка достигли только 50–55 % коэффициента корреляции Спирмена [12].

Глазкова [13] успешно классифицирует фрагменты биографических текстов на русском языке по десяти темам. Модель RuBERT достигает высокого значения F-меры в 93 % и превосходит мультиязычную версию BERT, word2vec и стандартный подход TF-IDF в сочетании с SVM.

В области классификации текстов по жанрам для русского языка применялись свёрточные нейронные сети и эмбеддинги word2vec [14]. Точность классификации на пять жанров: история, детективы, детская литература, поэзия, фантастика, — достигла около 78 %.

Автор статьи в своей предыдущей работе [5] вместе с коллегами исследовала комплекс ритмических характеристик для анализа шести жанров: художественные романы, научные статьи, политические статьи, рекламные статьи, отзывы, твиты. С помощью этих характеристик и классификаторов AdaBoost и LSTM были достигнуты достаточно высокие значения метрик качества: не менее 76 % F-меры для всех жанров, кроме рекламы.

Таким образом, в области классификации текстов по жанрам на русском языке актуальные языковые модели ELMo и BERT пока ещё почти не изучены. Но их высокие результаты для анализа жанров англоязычных корпусов, а также для других задач классификации русскоязычных текстов позволяют ожидать высокого качества решения и для проблемы, исследуемой в данной статье. Кроме того, сравнение нейросетевых и лингвистических моделей текстов, в частности, модели ритма, даст хороший материал для анализа достоинств и ограничений данных характеристик текста.

2. Корпус текстов пяти жанров

Для анализа русскоязычных текстов в различных стилях были выбраны пять жанров: художественные романы, научные статьи, отзывы, новости и посты в социальной сети Вконтакте. Каждому жанру соответствует 2 000 текстов, у текста может быть только один жанр.

Художественные тексты в данном исследовании — это фрагменты романов русскоязычных писателей XIX–XXI веков. Фрагменты содержат около 20 000 знаков и целое количество абзацев. Исходные тексты достаточно велики и существенно отличаются по объёму как друг от друга, так и от текстов других жанров, поэтому было принято решение выделить из них фрагменты заданного размера случайным образом, чтобы уравнять и уменьшить объёмы текстов. В результате тексты в среднем содержат 2 982 слова.

Научные статьи были собраны из журналов Грамота, Диалог и Кардиология. Они также были разделены на фрагменты объёмом около 2 000 знаков и целым количеством абзацев. В результате тексты в среднем содержат 191 слово.

Тексты отзывов включают в себя положительные, отрицательные и нейтральные русскоязычные отзывы на фильмы с интернет-ресурса Кинопоиск. Они в среднем содержат 406 слов.

Новостные тексты были взяты из корпуса текстов OpenCorpora [15]. Он содержит в том числе и новостные тексты из онлайн-медиа. Они в среднем содержат 433 слова.

Посты в социальной сети Вконтакте были собраны через API сайта из 50 различных групп, посвящённых разнообразным тематикам: наука, книги, кино и т. п. Выбирались не посты о новостях, которые часто являются дублями статей онлайн-медиа, а оригинальные авторские тексты, созданные именно для социальной сети. Они в среднем содержат 127 слов.

Кроме того, при сборе корпуса собирались только тексты, содержащие ритмические средства. Поэтому отсеивалось примерно 17 % текстов OpenCorpora, 2 % отзывов, 4 % научных статей, 30 % постов Вконтакте. Все художественные фрагменты содержали ритмические средства.

Таким образом, был сформирован корпус из 10 000 русскоязычных текстов пяти жанров, где каждый текст содержит хотя бы одно ритмическое средство.

3. Моделирование текстов

Для исследования жанров автор использует три современные векторные модели, где каждый текст представляется в виде кортежа из числовых характеристик. Это эмбеддинги, полученные с помощью языковой модели BERT, эмбеддинги ELMo и комплекс ритмических характеристик, которые ранее исследовались автором [6].

Модель BERT была взята в версии для русского языка — RuBERT cased, т. е. RuBERT, учитывающий регистр букв. У данной модели, как и у оригинального BERT, имеется ограничение: длина исходного текста не может превышать 512 токенов. Токенами в данном случае считаются слова, знаки препинания и специальные маркеры BERT [CLS] и [SEP], обозначающие начало текста и конец предложения. Тексты некоторых жанров, например, фрагменты художественных романов, могут быть большего размера, поэтому для подсчёта эмбеддингов их необходимо разделить на части.

Автор разбивала тексты на абзацы и считала эмбеддинги отдельно для каждого. Таким образом текст представляется как матрица из 768 столбцов, в которой каждая строка содержит эмбеддинг для своего абзаца. 768 — это длина одного BERT-эмбеддинга. Далее для каждого столбца считалось среднее арифметическое, так получался итоговый вектор для текста.

Модель ELMo также была взята в версии для русского языка из Python-библиотеки DeepPavlov. ELMo строит эмбеддинги для каждого слова и работает существенно медленнее BERT, поэтому для ускорения подсчётов автор из каждого текста выбирала фрагмент меньшего объёма, состоящий из целого количества предложений и обладающий размером около 1000 знаков. Как будет показано далее, данного объёма текста будет вполне достаточно для успешной классификации.

Таким образом, ELMo для каждого текста строит матрицу, где строки — это эмбеддинги отдельных слов, а количество столбцов — 2560 (число является длиной стандартного эмбеддинга ELMo). Как и для BERT, здесь также считаются средние значения по столбцам, чтобы получить итоговый вектор для текста.

Третья модель текстов — комплекс ритмических характеристик. Для их получения сначала в текстах ищутся ритмические средства: анафора, эпифора, симплока, анадиплозис, эпаналепсис, многосоюзие, диакопа, эпизевксис, хиазм, апозиопеза, повторяющиеся вопросительные и восклицательные предложения. Определения ритмических средств и алгоритмы их поиска приведены в предыдущих работах автора [6, 16]. Комплекс чистовых характеристик на основе представленных средств для данного исследования был расширен, чтобы изучить структуру ритма более подробно. Также из набора была исключена характеристика “доля уникальных слов”, так как по результатам экспериментов она оказалась наименее полезной. Итоговый комплекс характеристик выглядит следующим образом:

- количество появлений в тексте конкретного средства, делённое на количество предложений;
- доли существительных, прилагательных, глаголов, наречий, имён собственных, местоимений, соединительных союзов, подчинительных союзов, междометий и предлогов среди слов, составляющих средства;
- максимальное и среднее расстояния между первым и последним повторяющимся в средстве словом. Расстояние измеряется в количестве слов.

В обновлённой версии комплекса ритмических характеристик исследуются не только самостоятельные, но и служебные части речи, а также размеры ритмических средств.

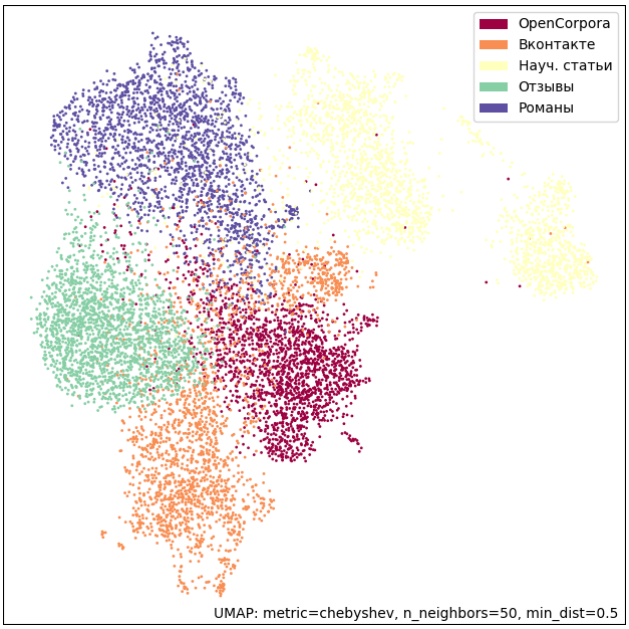
Таким образом, каждый текст представляется в виде эмбеддингов BERT (длина вектора — 768), эмбеддингов ELMo (длина вектора — 2560) или комплекса числовых характеристик, основанных на статистике ритмических средств (длина вектора — 24).

4. Визуализация числовых характеристик

Для того, чтобы визуализировать векторные модели и проанализировать особенности ритма в каждом жанре, для каждого набора характеристик были построены графики и диаграммы с разметкой по жанрам.

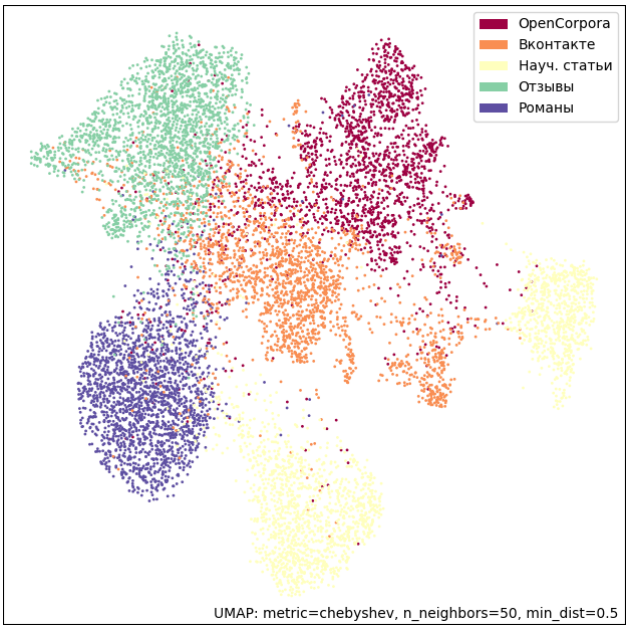
Так как эмбеддинги содержат числовые характеристики, которые невозможно интерпретировать отдельно друг от друга, было принято решение кластеризовать тексты на основе моделей BERT и ELMo и сопоставить полученные кластеры с исходными жанрами. Для визуализации был использован алгоритм UMap, который уменьшает размерность данных до двумерного пространства, кластеризуя объекты на основе их расстояния до k-ближайших соседей.

На рис. 1 представлены результаты кластеризации при k равном 50 и метрике Чебышёва для измерения расстояния. Жанры текстов отмечены различными цветами. И BERT, и ELMo позволяют отделить все жанры друг от друга: у каждого жанра имеется собственный кластер, мало пересекающийся с другими. Для научных статей появляется сразу два кластера, причём кластеры BERT ближе друг к другу, чем кластеры ELMo. Посты Вконтакте, наоборот, у BERT также делятся на два кластера (или один протяжённый, на который накладывается кластер OpenCorpora), у ELMo можно выделить один кластер. ELMo Новости из OpenCorpora и посты Вконтакте наиболее часто смешиваются друг с другом и с остальными жанрами. В целом, по результатам кластеризации можно ожидать высокого качества классификации с помощью эмбеддингов.



a)

Fig. 1. UMap clusters for embeddings
a) BERT, b) ELMo



b)

Рис. 1. Кластеры UMap для эмбеддингов
a) BERT, b) ELMo

Комплекс ритмических характеристик более разнообразен по содержанию, поэтому он визуализируется по нескольким группам характеристик на отдельных диаграммах.

Гистограмма на рис. 2 создана на основе количеств появлений в тексте конкретных средств и визуализирует для каждого жанра средний процент каждого средства среди всех ритмических средств. Справа в легенде перечислены все ритмические средства сверху вниз, на гистограмме их доли отмечены снизу вверх. Каждый столбец соответствует жанру текстов, числа — это проценты конкретных средств, число над столбцом — процент апозиопезиса среди всего ритма.

Гистограмма демонстрирует, что самыми частыми ритмическими средствами являются диакопа и многосоюзие, их проценты самые большие в каждом жанре: 48–84 % и 7–19 % соответственно. Значительные доли в некоторых жанрах имеют анафора, эпифора, эпаналепсис, эпизевксис и повторяющиеся вопросительные и восклицательные предложения: от 2 до 11 % в большинстве случаев. Наиболее разнообразен по римту жанр художественных романов, на второе место по этому параметру можно поставить жанр отзывов. Научные статьи и новости OpenCorpora, наоборот, содержат мало различных типов средств, а количественно в них преобладает диакопа.

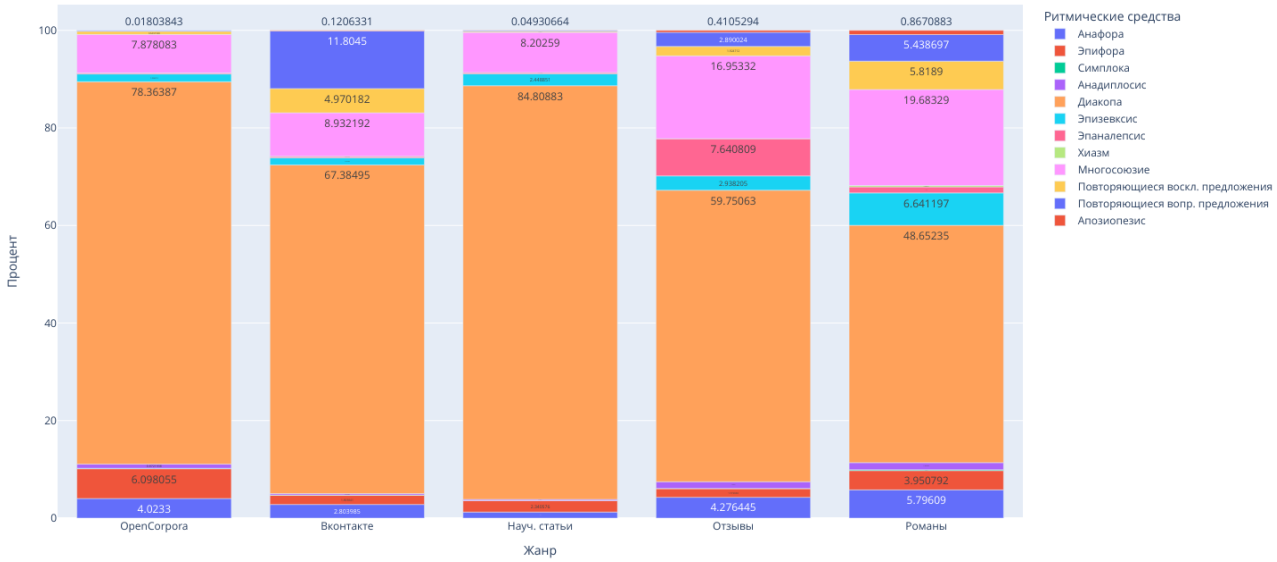


Fig. 2. The histogram with percents of rhythm features

Рис. 2. Гистограмма с процентами ритмических средств

Наиболее часто встречающиеся части речи отображены на тепловой карте на рис. 3. Это существительные (NOUN), прилагательные (ADJS), глаголы (VERB), наречия (ADVB), местоимения (PRON) и соединительные союзы (CCONJ). В столбцах тепловой карты указаны части речи, в строках — жанры, в ячейках — средний процент данной части речи в текстах данного жанра. Справа диапазон значений сопоставляется с диапазоном цветов: тёмные оттенки соответствуют меньшим долям, светлые — большим.

Тепловая карта показывает, что в романах и отзывах две части речи наиболее часто образуют ритмические средства: существительные (25 % и 22 %) и соединительные союзы (22 % и 19 %). В романах достаточно часто встречаются и глаголы: 16 %. В остальных жанрах самый большой процент среди всех частей речи занимают существительные, а соединительные союзы, хоть и имеют небольшой процент 9–10 %, но так же занимают второе место по количеству. Как следует из предыдущей гистограммы, существительные соответствуют диакопам, а многосоюзия — соединительным союзам.

Максимальное и среднее расстояния между первым и последним повторяющимся в средстве словом визуализированы на коробчатых диаграммах, см. рис. 4. Для каждого жанра отображено собственное распределение значений характеристики. Прямоугольник обозначает первый и третий квартили, вертикальная линия внутри него — медиану, белый круг — среднее значение, чёрные отрезки — минимальное и максимальное значение. Выбросы исключены из диаграммы.

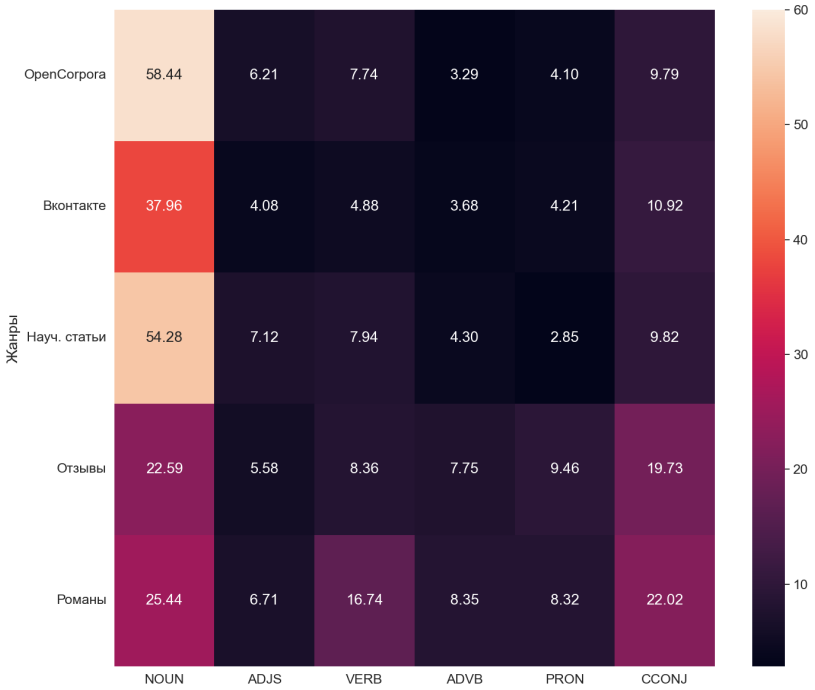
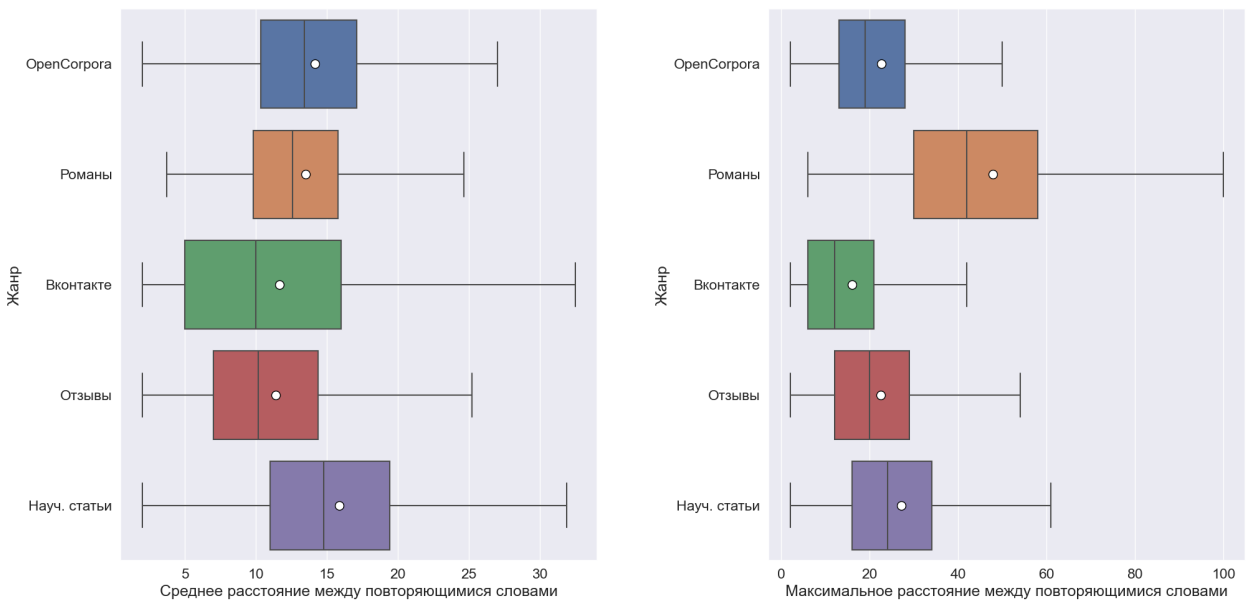


Fig. 3. The heat map with percents of the most frequent parts of speech

Рис. 3. Тепловая карта с процентами наиболее частых частей речи



a) b)

Fig. 4. The boxplot for a) average, b) maximal distance among words in rhythm features

Рис. 4. Коробчатая диаграмма для a) среднего, b) максимального расстояния между словами в ритмических средствах

Диаграммы показывают, что медианный размер ритмического средства составляет 10–15 слов от первого до последнего повторяющегося слова включительно. В целом жанры по статистике средних расстояний похожи друг на друга, только посты Вконтакте и научные статьи имеют чуть

больший диапазон значений, в том числе и между первым и третьим квартилем. По максимальному расстоянию ярко выделяются романы: все их статистические параметры больше, чем у других жанров.

Таким образом, визуализация демонстрирует, что жанры существенно отличаются друг от друга и по ритму, и по эмбеддингам на основе языковых моделей BERT и ELMo. Следовательно, векторные модели на основе данных характеристик могут быть хорошими маркерами жанров и обеспечивать качественную классификацию.

5. Классификация по жанрам

Постановка экспериментов

Тексты были классифицированы по жанрам на основе нескольких векторных моделей: эмбеддингов ELMo и BERT, ритмических характеристик, а также дополнительно комбинации BERT с ритмическими характеристиками.

Классификация проводилась двумя способами:

- мультиклассовая классификация на пять жанров;
- бинарная классификация для каждого жанра, когда тексты классифицировались на принадлежащие и не принадлежащие конкретному жанру.

Для обоих способов применялись одни и те же классификаторы, включающие в себя два стандартных классификатора машинного обучения и две нейронные сети:

- классификатор AdaBoost — мета-алгоритм машинного обучения, который объединяет результаты 50 классификаторов-деревьев решений, корректирующих неправильно классифицированные тексты;
- классификатор RandomForest — мета-алгоритм машинного обучения, который усредняет результаты 50 классификаторов-деревьев решений;
- двунаправленная LSTM — рекуррентная нейронная сеть со слоем двунаправленной долгой краткосрочной памяти (LSTM) с 64 блоками и полносвязным выходным слоем, использующим функцию активации Softmax для мультиклассовой классификации и Sigmoid для бинарной;
- GRU — рекуррентная нейронная сеть со слоем Gated Recurrent Unit (GRU) с 4 блоками и полносвязным выходным слоем, использующим функцию активации Softmax для мультиклассовой классификации и Sigmoid для бинарной.

Данные алгоритмы были выбраны как лучшие алгоритмы по классификации текстов как с помощью ритма [5], так и с помощью ELMo [7] и BERT [8].

Для классификации корпус был разделён случайным образом на обучающую и тестовую выборки в отношении 4:1, чтобы провести пятикратную кросс-валидацию. Оценка качества выполнялась с помощью трёх стандартных мер: точность, полнота и F-мера. Во всех случаях стандартное отклонение для всех мер не превысило 4 % для алгоритма RandomForest и 2 % для остальных классификаторов, что свидетельствует о высокой стабильности классификации по жанрам.

Мультиклассификация жанров

Результаты мультиклассовой классификации представлены в таблице 1. Обе модели на основе эмбеддингов превосходят ритмические характеристики на 20–30 % по точности, полноте и F-мере. Лучшей по качеству моделью можно назвать BERT, так как она в комбинации с LSTM практически безошибочно различает жанры (99 % F-меры). Комбинация эмбеддингов с ритмом не повышает результаты классификации, что проиллюстрировано строками BERT + Ритм.

Модель ELMo практически так же хороша, как и BERT: 95–96 % F-меры, что всего на 3–4 % меньше лучших результатов.

Table 1. Multi-class text classification by genres

Таблица 1. Мультиклассовая классификация текстов по жанрам

Классификатор	Модель	Точность	Полнота	F-мера
AdaBoost	Ритм	64.6	62.8	63.7
AdaBoost	ELMo	84.9	84.8	84.8
AdaBoost	BERT	91.9	91.1	91.5
AdaBoost	BERT + Ритм	90.5	88.7	89.6
RandomForest	Ритм	63.2	57.2	60.0
RandomForest	ELMo	89.3	89.4	89.4
RandomForest	BERT	**94.9**	**94.8**	**94.8**
RandomForest	BERT + Ритм	94.3	93.6	93.9
LSTM	Ритм	67.8	67.3	67.6
LSTM	ELMo	96.6	96.6	96.6
LSTM	BERT	**99.2**	**99.2**	**99.2**
LSTM	BERT + Ритм	**99.3**	**99.2**	**99.3**
GRU	Ритм	68.7	67.8	68.2
GRU	ELMo	95.3	95.3	95.3
GRU	BERT	**98.7**	**98.7**	**98.7**
GRU	BERT + Ритм	**98.6**	**98.6**	**98.6**

Что касается классификаторов, нейронные сети LSTM и GRU одинаково обеспечивают высокий уровень классификации до 98–99 %. Среди стандартных классификаторов RandomForest превосходит AdaBoost на 3–4 % и достигает почти 95 % F-меры, что также является очень хорошим результатом, лишь немногим ниже уровня нейросетей.

Ритмические характеристики не обеспечивают высокого качества для пяти жанров: лучшие результаты с классификатором GRU достигают всего лишь 68 % F-меры.

Ошибки мультиклассификации для BERT и ритмических характеристик представлены в виде тепловых карт на рис. 5. Тепловые карты 5a и 5b указывают, сколько текстов из жанра в строке были приняты за жанр в столбце, а в последнем столбце указано общее количество ошибок для каждого жанра. Тепловые карты 5c и 5d демонстрируют те же данные, но в процентном соотношении относительно общего числа ошибок. На главной диагонали условно указаны нули.

При классификации с помощью BERT все романы отнесены правильно к своему жанру, а девять новостей классифицированы как другие жанры. Пять новостей и постов Вконтакте приняты за отзывы. За научную статью ошибочно не принят ни один текст.

При классификации с помощью ритмических характеристик романы классифицируются лучше всего: 19 ошибок, 52 % которых отнесены к новостям из OpenCorpora. Новости классифицируются с наибольшим количеством ошибок, 152, причём тексты этого жанра путаются со всеми остальными жанрами. Среди научных статей тоже много ошибок, 142, и это преимущественно отнесение к жанру новостей (59 % ошибок). За новости из OpenCorpora принято наибольшее количество текстов из других жанров в процентном соотношении: 40–59 %.

Таким образом, наиболее качественно классифицируются романы, а наименее — новости из OpenCorpora. Тем не менее, эмбеддинги BERT классифицируют все жанры с высокой точностью, полнотой и F-мерой.

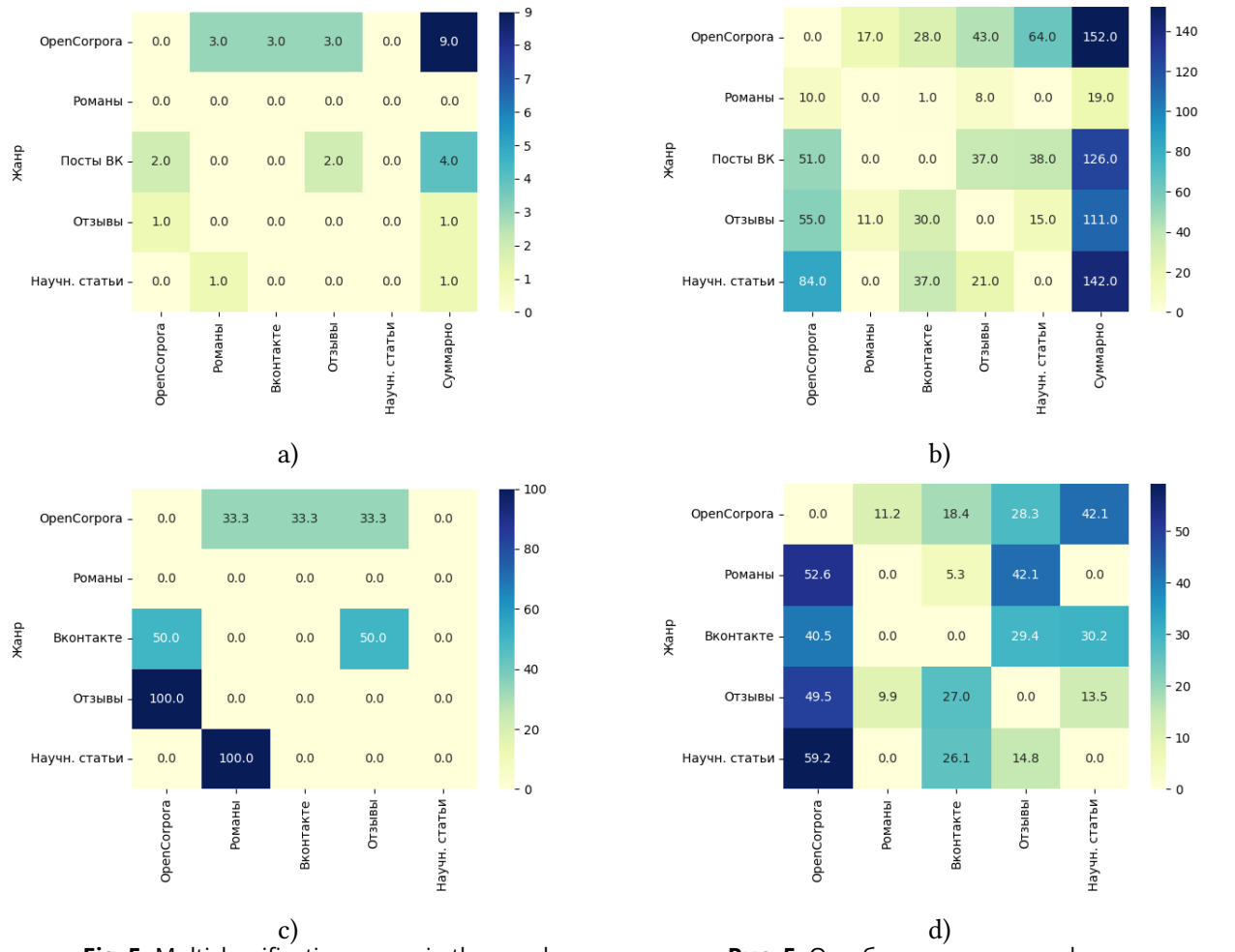


Fig. 5. Multiclassification errors in the number of texts for the a) BERT, b) rhythm, and in percent for the c) BERT, d) rhythm

Рис. 5. Ошибки мультиклассификации в количестве текстов для a) BERT, b) ритма, и в процентах для c) BERT, d) ритма

Верификация жанров

Для того, чтобы разобраться глубже в достоинствах, недостатках и ограничениях моделей текста, была проведена верификация жанров текстов для лучшей и худшей модели: эмбеддингов BERT и ритмических характеристик. Алгоритмом классификации была нейросеть LSTM. Её результаты представлены в таблице 2.

В целом, результаты верификации подтверждают результаты обработки ошибок: наиболее хорошо отделяются от остальных романы (96.0–99.8 % F-меры), наименее — новости из OpenCorpora (67.3–99.1 % F-меры). Это верно и для комплекса ритмических характеристик, и для эмбеддингов BERT. Кроме того, с высоким качеством верифицируются отзывы: 84.2–99.8 % F-меры.

BERT хорошо отделяет любой жанр от остальных. Комплекс ритмических характеристик лучше всего верифицирует романы и отзывы, а тексты OpenCorpora верифицируются слабее.

Чтобы определить набор жанров, для которых мультиклассификация с помощью ритма будет наиболее эффективной, из корпуса были исключены новости из OpenCorpora, а для остальных жанров была проведена мультиклассификация и верификация. Далее были исключены посты Вконтакте как худшие по верификации из четырёх жанров, и мультиклассификация с верификацией повторились. Любопытно, что среди трёх жанров: романы, отзывы, научные статьи, — наименьшее качество верификации оказалось у отзывов (85.3 % F-меры), а качество верификации научных статей

Table 2. Text verification by genres

Таблица 2. Верификация текстов по жанрам

Модель	Жанр	Точность	Полнота	F-мера
Ритм	OpenCorpora	73.9	61.8	67.3
Ритм	Романы	**95.0**	**97.0**	**96.0**
Ритм	Вконтакте	86.0	73.6	79.3
Ритм	Отзывы	**85.2**	**83.1**	**84.2**
Ритм	Научн. статьи	82.5	76.5	79.4
BERT	OpenCorpora	99.3	99.0	99.1
BERT	Романы	**99.6**	**99.9**	**99.8**
BERT	Вконтакте	99.6	99.4	99.5
BERT	Отзывы	**99.8**	**99.8**	**99.8**
BERT	Научн. статьи	99.7	99.7	99.7

оказалось вторым (92.3 % F-меры). И как завершение данного эксперимента, была проведена бинарная классификация для романов и научных статей.

Результаты каждой мультиклассификации и итоговой бинарной классификации приведены на рис. 6. На диаграмме по горизонтали указано количество классов, по вертикали — значения точности, полноты и F-меры. Уже исключение текстов OpenCorpora увеличивает качество классификации на 10 %, до 77 % для всех характеристик. А романы, научные статьи и отзывы вместе классифицируются отлично: 91.5 % F-меры. Романы и научные статьи классифицируются с ошибкой в одном тексте, когда фрагмент научной статьи принят за фрагмент романа. Стоит отметить, что в данном фрагменте было несколько цитат, написанных в литературном стиле.

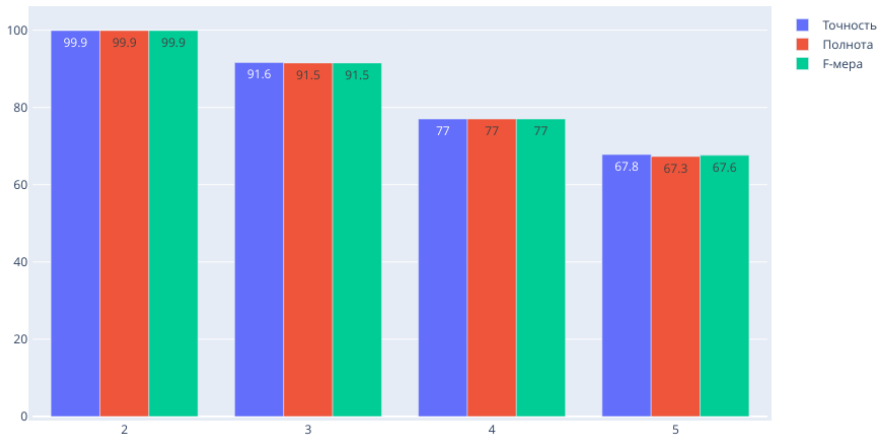


Fig. 6. Quality of multiclassification while reducing the number of classes

Рис. 6. Качество мультиклассификации при уменьшении количества классов

Таким образом, можно выделить три жанра: романы, научные статьи и отзывы, которые наиболее хорошо классифицируются с помощью комплекса ритмических характеристик. Наиболее вероятная причина таких результатов заключается в том, что в этих жанрах имеются общепринятые рекомендации к стилю письма, что может влиять и на особенности ритма. Новости и посты социальной сети Вконтакте более разнообразны по авторскому стилю.

Заключение

В данной статье автор классифицировала корпус из 10 000 русскоязычных текстов на пять жанров с помощью трёх векторных моделей: BERT, ELMo и комплекса ритмических характеристик. Лучшее качество было достигнуто с помощью комбинации эмбеддингов BERT с нейросетевым классификатором LSTM: 99 % F-меры как для мультиклассификации, так и для верификации отдельных жанров. Эмбеддинги ELMo показали близкий результат: 96 % F-меры.

Комплекс ритмических характеристик оказался полезен для романов, научных статей и отзывов на фильмы. Романы и отзывы наиболее разнообразны по ритмическим средствам среди всех жанров, как показали их визуализация и анализ. Кроме того, для данных жанров, как и для научных статей, существуют общепринятые рекомендации к их написанию, которые и могли повлиять на сходство стилей текстов в одном жанре. Новости OpenCorpora и посты Вконтакте гораздо более разнородны по стилю и тематике.

Результаты визуализации и классификации жанров для комплекса ритмических характеристик дают широкие возможности для их интерпретации с филологической точки зрения, что может быть направлением для будущих исследований. Также в дальнейшем можно рассмотреть и другие корпуса данных, включающие в себя большее количество Интернет-текстов различных жанров, чтобы протестировать современные эмбеддинги на более трудных задачах классификации длинных текстов.

References

[1] L. A. Kochetova and V. V. Popov, “Research of Axiological Dominants in Press Release Genre based on Automatic Extraction of Key Words from Corpus”, *Nauchnyi dialog*, no. 6, 2019, In Russian.

[2] B. Kessler, G. Numberg, and H. Schütze, “Automatic detection of text genre”, in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 1997, pp. 32–38.

[3] A. Onan, “An ensemble scheme based on language function analysis and feature engineering for text genre classification”, *Journal of Information Science*, vol. 44, no. 1, pp. 28–47, 2018.

[4] Z. Dai and R. Huang, “A Joint Model for Structure-based News Genre Classification with Application to Text Summarization”, in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 3332–3342.

[5] K. V. Lagutina, N. S. Lagutina, and E. I. Boychuk, “Text classification by genre based on rhythm features”, *Modeling and analysis of information systems*, vol. 28, no. 3, pp. 280–291, 2021.

[6] K. Lagutina, A. Poletaev, N. Lagutina, E. Boychuk, and I. Paramonov, “Automatic extraction of rhythm figures and analysis of their dynamics in prose of 19th-21st centuries”, *Proceedings of the 26th Conference of Open Innovations Association FRUCT*, pp. 247–255, 2020.

[7] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep Contextualized Word Representations”, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[9] C. Wang, P. Nulty, and D. Lillis, "A comparative study on word embeddings in deep learning for text classification", in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, 2020, pp. 37–46.

[10] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for Russian language", in *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 2019, pp. 333–339.

[11] A. Kutuzov, L. Pivovarova, *et al.*, "RuShiftEval: a shared task on semantic shift detection for Russian", in *Computational Linguistics and Intellectual Technologies Papers from the Annual International Conference "Dialogue" (2021)*, vol. 20, 2021, pp. 533–545.

[12] J. Rodina, Y. Trofimova, A. Kutuzov, and E. Artemova, "ELMo and BERT in semantic change detection for Russian", in *International Conference on Analysis of Images, Social Networks and Texts*, Springer, 2020, pp. 175–186.

[13] A. V. Glazkova, "Topical classification of text fragments accounting for their nearest context", *Automation and Remote Control*, vol. 81, no. 12, pp. 2262–2276, 2020.

[14] I. A. Batraeva, A. D. Nartsev, and A. S. Lezgyan, "Using the analysis of semantic proximity of words in solving the problem of determining the genre of texts within deep learning", *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika*, no. 50, pp. 14–22, 2020, In Russian.

[15] V. Bocharov, S. Alexeeva, D. Granovsky, E. Protopopova, M. Stepanova, and A. Surikov, "Crowdsourcing morphological annotation", in *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue". Volume 1*, 2013, pp. 109–114.

[16] K. Lagutina, N. Lagutina, E. Boychuk, V. Larionov, and I. Paramonov, "Authorship verification of literary texts with rhythm features", in *28th Conference of Open Innovations Association FRUCT*, IEEE, 2021, pp. 240–251.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 29, NO. 4, 2022
journal homepage: www.mais-journal.ru

THEORY OF DATA

A Model for Automated Business Writing Assessment

D. D. Zafievsky[1], N. S. Lagutina[1], O. A. Melnikova[1], A. Y. Poletaev[1]

DOI: 10.18255/1818-1015-2022-4-348-365

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

MSC2020: 68T50
Research article
Full text in Russian

Received September 27, 2022
After revision November 14, 2022
Accepted November 16, 2022

This study is aimed at building an automated model for business writing assessment, based on 14 rubrics that integrate EFL teacher assessment frameworks and identify expected performance against various criteria (including language, task fulfillment, content knowledge, register, format, and cohesion). We developed algorithms for determining the corresponding numerical features using methods and tools for automatic text analysis. The algorithms are based on a syntactic analysis with the use of dictionaries. The model performance was subsequently evaluated on a corpus of 20 teacher-assessed business letters. Heat maps and UMAP results represent comparison between teachers' and automated score reports. Results showed no significant discrepancies between teachers' and automated score reports, yet detected bias in teachers' reports. Findings suggest that the developed model has proved to be an efficient tool for natural language processing with high interpretability of the results, the roadmap for further improvement and a valid and unbiased alternative to teachers' assessment. The results may lay the groundwork for developing an automatic students' language profile. Although the model was specifically designed for business letter assessment, it can be easily adapted for assessing other writing tasks, e.g. by replacing dictionaries.

Keywords: natural language processing; text features; automated essay scoring; business letter

INFORMATION ABOUT THE AUTHORS

Daniil Dmitrievich Zafievsky	orcid.org/0000-0001-8266-2283. E-mail: zafievsky@mail.ru student.
Nadezhda Stanislavona Lagutina correspondence author	orcid.org/0000-0002-6137-8643. E-mail: lagutinans@gmail.com PhD, associate professor.
Oksana Andreyevna Melnikova	orcid.org/0000-0001-8814-7696. E-mail: oam8108@gmail.com associate professor.
Anatoliy Yurievich Poletaev	orcid.org/0000-0003-0116-4739. E-mail: anatoliy-poletaev@mail.ru post-graduate student.

Funding: The reported study was funded by YarSU Program according to the research project No. P2-GM5-2021.

For citation: D. D. Zafievsky, N. S. Lagutina, O. A. Melnikova, and A. Y. Poletaev, "A Model for Automated Business Writing Assessment", *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 348-365, 2022.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 29, № 4, 2022
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Модель текста для автоматической оценки делового письма на заданную тему

Д. Д. Зафиевский[1], Н. С. Лагутина[1], О. А. Мельникова[1], А. Ю. Полетаев[1]

DOI: 10.18255/1818-1015-2022-4-348-365

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, д. 14, г. Ярославль, 150003 Россия.

УДК 004.912
Научная статья
Полный текст на русском языке

Получена 27 сентября 2022 г.
После доработки 14 ноября 2022 г.
Принята к публикации 16 ноября 2022 г.

В статье описана модель текста, предназначенная для автоматической оценки связного текста в виде письма на заданную тему. Параметры оценки сформулированы и формализованы в виде 14 критериев при помощи экспертов в области обучения английскому языку. Критерии включают параметры, относящиеся к анализу лексики, включая особенности предметной области, тематики текста, стилю и формату письма, средствам логической связи предложений. Авторами разработаны алгоритмы определения соответствующих числовых характеристик с использованием методов и инструментов автоматического анализа текстов. Алгоритмы основаны на анализе состава и структуры предложений, для чего используются, в том числе данные специализированных словарей. Характеристики ориентированы на проверку электронного делового письма, но могут быть адаптированы к анализу других письменных текстов, например, с помощью замены словарей. На основе разработанных алгоритмов создана система автоматической оценки текстов. Проведён эксперимент по анализу результатов работы этой системы на корпусе из 20 текстов, предварительно размеченных преподавателями английского языка. Автоматическая оценка и оценка экспертов сравнивались с помощью тепловых карт и технологии двумерного представления векторов UMAP, применённой к характеристическим векторам текстов. В большинстве случаев не было выявлено значимых различий между этими оценками, кроме того, автоматическая оценка оказалась более объективной. Таким образом, разработанная модель успешно справилась с поставленной задачей и может применяться для оценки текстов, написанных человеком. Результаты будут использованы в проекте автоматического построения языкового профиля учащегося. Достоинствами модели являются хорошая интерпретируемость получаемых результатов, объективность, перспективы развития.

Ключевые слова: автоматическая обработка текста; параметры текста; автоматизированная оценка эссе; деловое письмо

ИНФОРМАЦИЯ ОБ АВТОРАХ

Даниил Дмитриевич Зафиевский — orcid.org/0000-0001-8266-2283. E-mail: zafievsky@mail.ru
студент.

Надежда Станиславовна Лагутина
автор для корреспонденции — orcid.org/0000-0002-6137-8643. E-mail: lagutinans@gmail.com
канд. физ.-мат. наук, доцент.

Оксана Андреевна Мельникова — orcid.org/0000-0001-8814-7696. E-mail: oam8108@gmail.com
доцент.

Анатолий Юрьевич Полетаев — orcid.org/0000-0003-0116-4739. E-mail: anatoliy-poletaev@mail.ru
аспирант.

Финансирование: Исследование выполнено в рамках Программы развития ЯрГУ, проект № П2-ГМ5-2021.

Для цитирования: D. D. Zafievsky, N. S. Lagutina, O. A. Melnikova, and A. Y. Poletaev, “A Model for Automated Business Writing Assessment”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 348-365, 2022.

Введение

Изменение процесса межличностного взаимодействия вследствие процессов глобализации и цифровизации повлекли за собой постановку ряда новых задач перед сообществом преподавателей, и в первую очередь перед сообществом преподавателей иностранного языка. Одной из таких задач сегодня является обучение эффективной онлайн-коммуникации как преобладающему способу взаимодействия в межкультурном пространстве. Актуальность данной задачи подтверждается включением компетенций и стратегий онлайн-коммуникации в обновленную версию стандарта для описания уровней владения иностранным языком Common European Framework of Reference (CEFR). Обучение такому виду коммуникации на практике представляет особую сложность вследствие наличия объективных различий в стратегиях для достижения эффективности при онлайн- и оффлайн-взаимодействии, которые часто не находят отражения в общепринятых методических подходах к обучению иностранному языку [1].

Важнейшее место в методике преподавания иностранного языка и, в частности, онлайн-коммуникации на иностранном языке, занимает обучение продуктивным видам речевой деятельности, ключевым аспектом которых является овладение навыком письма [2]. Поскольку письменная речь предполагает одновременное владение коммуникативными, лингвистическими и экстралингвистическими компетенциями на достаточном для эффективной коммуникации уровне [3], обучение навыкам письма и особенно их оценка нередко сопровождаются рядом проблем, которые усугубились в ходе пандемии COVID-19 [1, 4]. Оценка письменной речи является одним из способов измерения знаний, практических умений, логического и критического мышления и творческих способностей [5]. Чаще всего текст, написанный учащимся, квалифицируется человеком-экспертом на основе некоторых критериев. Однако ручная оценка требует много времени, особенно при большом количестве экзаменуемых, и достаточно необъективна, так как зависит от множества субъективных факторов: строгости оценщика, его последовательности, усталости [6]. Решением проблемы может стать система автоматизированной оценки письменной речи (Automated Essay Scoring, AES). AES — современная, актуальная задача в области обработки естественного языка [7]. Следует отметить, что термин «эссе (essay)» в области автоматической обработки текста имеет более широкую трактовку по сравнению с классической лингвистикой. В качестве эссе рассматривается короткий связный письменный текст на естественном языке.

Целью систем AES является имитация человека-оценщика. Для её достижения таких систем использует методы и инструменты обработки естественного языка и машинного обучения. При этом основой любого решения служит построение модели текста, т. е. выбор числовых параметров и формирование соответствующего вектора характеристик. Условно можно выделить две группы параметров: определяемые экспертами-людьми и получаемые автоматически с помощью методов глубокого обучения. Методы глубокого обучения показывают хорошие результаты в различных задачах, связанных с классификацией текстов. Однако следует отметить ряд проблем: хороший результат получается только при наличии больших объёмов данных для обучения, мало исследованы оценки тематических текстов в рамках предметной области, результаты работы не поддаются интерпретации, что затрудняет анализ ошибок [8]. Часть проблем даёт возможность решить группа характеристик, определяемых экспертами. Эти параметры включают меры, определяющие лексическую сложность и разнообразие текста, его синтаксические особенности, шаблоны, величины для кодирования аспектов, связанных с письменной речью. Этот подход может работать и с небольшими корпусами текстов, а также учитывать особенности предметной области, позволяет интерпретировать результаты и проводить анализ ошибок [9].

Поэтому, когда перед авторами работы встала задача автоматизации построения профиля языковых компетенций обучающихся английскому языку, в рамках которой потребовалась оценка делового письма на заданную тему, было принято решение построить и исследовать модель текста

письма на основе параметров, предложенных экспертами-лингвистами. Построение модели текста, описываемое в данной статье, заключается в разработке алгоритмов оценки уровня владения компетенциями языкового профиля и формировании соответствующего вектора числовых характеристик, а также анализе корпуса писем с применением разработанной модели.

1. Обзор связанных работ

Алгоритмы автоматической оценки писем разрабатываются с шестидесятых годов прошлого века. В настоящий момент наиболее популярны подходы на основе нейронных сетей.

Одна из первых моделей на основе рекуррентной нейронной сети (RNN) для решения задачи AES была предложена в работе [10]. Многоуровневая нейронная сеть обучалась на размеченном корпусе текстов и предсказывала оценку эссе из диапазона $[0, 1]$. Характерной работой в рамках обсуждаемого подхода является [11]. Её авторы предложили двухуровневую двунаправленную нейронную сеть LSTM для оценки эссе. Моделью текста послужили функции, извлеченные с помощью word2vec для обучения LSTM, точность решения в среднем составила 0,87.

Однако другие исследователи показали [12], что качество работы таких методов сильно зависит от квалификации и согласованности экспертов, оценивающих эссе, из которых формируется обучающий корпус. Чтобы решить эту проблему, авторы предложили двухэтапную архитектуру, которая объединила вероятностную психологическую модель оценки текстов и нейронную сеть DNN-AES. Эксперименты показали, что такой метод хорошо подходит для ситуаций, когда сложно подготовить качественные обучающие данные. Другим слабым местом моделей на основе RNN является оценка связности текста. Для решения этой задачи было предложено несколько моделей DNN-AES, учитывающих особенности когерентности отдельных частей эссе [13, 14]. Проблема решается дополнением в базовые нейронные сети элементов, моделирующих связность и семантику текста. Авторы работы [15] сформулировали и решили вспомогательную задачу извлечения семантических признаков на уровне отдельных предложений и на уровне документа в целом. Эти параметры были добавлены к другим характеристикам для получения целостной оценки письма и привели к росту качества решения задачи.

Одной из самых популярных современных моделей текста на основе глубоких нейронных сетей является BERT, выпущенная командой Google AI Language в 2018 году. BERT успешно применялся для AES и показал хорошие результаты [16—18].

К сожалению, методические и числовые особенности оценки эссе очень разнообразны и затрудняют использование каждой разработанной системы для других наборов данных или условий оцениваемой задачи. На эту проблему указывают авторы работы [19] и решают её с помощью структуры, объединяющей несколько моделей AES с учетом характеристик каждой модели с использованием теории оценки ответов. В этой структуре несколько моделей AES сначала обучаются независимо, а затем используются для получения интегрированной оценки целевых эссе. Для оценки эссе по заданному стандарту учёные [20] предложили мультимодальный подход к машинному обучению. Они подсчитывают грамматические и статистические параметры текста с помощью библиотеки spaCy и дополняют их эмбеддингами слов GloVe. Подготовленные таким образом векторы предложений передают в нейронную сеть LSTM для нахождения окончательного результата. Bi-LSTM проверяет каждое предложение в двух направлениях, чтобы найти семантику.

Описанные системы на основе нейронных сетей моделируют текст автоматически, с помощью эмбеддингов, однако сложные экспертные лингвистические характеристики тоже играют важную роль. В работе [21] предлагается использовать объединение нечёткой онтологии с методами латентно-семантического анализа (LSA). При этом извлекаются два типа функций: синтаксические и морфологические. На основе дерева синтаксического анализа, поиска подобия слов и предложений получаются параметры, характеризующие семантику текста. Синтаксические и морфологические

особенности объединяются для получения окончательной оценки. Похожая модель, агрегирующая простые и сложные параметры текста, используется в работе [22].

Авторы работы [23] замечают, что многословные выражения связаны с владением языком, но игнорируются в работах по автоматической оценке обучающихся. Поэтому в своей статье авторы оценивают их важность для автоматизированной оценки эссе и сравнивают с другими классическими функциями. Хотя эксперименты показали, что для автоматического оценивания классические функции более значимы, чем многословные выражения, тем не менее, предложенные на основе многословных выражений параметры позволяют анализировать текст более детально.

Подавляющее большинство работ над AES сосредоточено на комплексной оценке, которая обобщает качество эссе с помощью единой оценки. Корпуса эссе, аннотированные вручную, с такими оценками являются общедоступными, что облегчает разработку систем AES, особенно на основе методов машинного обучения. Эти системы дают возможность автоматизировать оценку миллионов эссе, написанных для стандартных тестов, что экономит много усилий по ручному оцениванию. Однако главным недостатком такого подхода является невозможность интерпретации результата и, соответственно, отсутствие обратной связи с обучающимся. Возвращение единственной оценки не говорит о том, какие аспекты письма способствовали этой оценке и тому, как ее можно улучшить [24].

Кроме этого в области автоматической оценки письма остаются вопросы: какие параметры в наибольшей степени способствуют точности оценки, можно ли обобщить эти характеристики на другие наборы данных с другой схемой оценивания и другими целевыми группами учащихся, какие конкретные лингвистические свойства письменной речи учащегося способствуют получению более высокой оценки владения языком. Исследуемые в данной статье характеристики, сформированные с помощью экспертов-лингвистов, дают возможность решать эти вопросы и строить образовательные системы с обратной связью.

2. Критерии оценки компетенций языкового профиля

Для полной, всесторонней и объективной оценки предложенного учащемуся письменного задания экспертами была создана матрица критериальных шкал для оценивания, включившая в себя как опыт зарубежных коллег, отражённый в системе уровней владения иностранным языков CEFR и критериях оценки международных экзаменов IELTS Cambridge Assessment English, так и опыт отечественных методистов, нашедший отражение в критериях оценки письменных заданий ЕГЭ и ОГЭ. Составленная матрица включает в себя четыре блока критериев: решение коммуникативной задачи (Communicative Achievement), организация речи (Organisation), лексика (Vocabulary), грамматика (Grammar).

Блок «Решение коммуникативной задачи» предназначен для контроля уровня сформированности коммуникативной компетенции, а именно оценки эффективности решения коммуникативной задачи: написание письма начальнику с просьбой о назначении на определенную должность. Для этого был разработан и выделен ряд критериев, нацеленных на определение способности автора письма решать коммуникативные микро-задачи, т. е. проверка наличия указанной в задании информации в письме. Данный блок также нацелен на контроль сформированности медиативной компетенции, в данном случае — умения вежливо и эффективно разрешить потенциально конфликтную ситуацию, для которого был выделен отдельный критерий оценки использованных в письме норм вежливости и умения эффективно и доброжелательно выразить свою точку зрения. Блок также включает в себя проверку сформированности лингвистической и экстралингвистической компетенций: стилистического оформления речи, необходимого для решения поставленной задачи, что предполагает наличие определенных фоновых знаний по затронутой проблеме.

Блок «Организация речи» включает в себя критерии проверки когезии — внутренней, лексико-грамматической связности текста, и когерентности — логичности изложения, связности дискурса,

а также правильности структурирования текста и его соответствия заявленному формату электронного письма.

Блок «Лексика» оценивает словарный запас и разнообразие используемой лексики, а также соответствие употребляемой лексики уровню по общеевропейской шкале CEFR. Данный блок также включает в себя критерии оценки точности выбора лексических единиц и словосочетаний, корректности их употребления и соответствия теме и ситуации общения.

Блок «Грамматика» предназначен для контроля уровня сформированности грамматических навыков: разнообразия употребляемых грамматических структур и явлений, точности и корректности их употребления и соответствия употребляемых структур и явлений теме и ситуации общения.

Авторами работы была предложена модель текста для проверки деловых писем. Основой модели служит набор формализованных числовых параметров (критериев), приведенный в таблице 1. Каждый из критериев оценки делового письма принимает значение один или ноль.

Table 1. Criteria used in evaluating texts

Таблица 1. Критерии, применяемые при оценке текстов

Критерий	Описание критерия	Метод оценки
K1	Письмо содержит информацию про презентацию и связанные с ней факты	Алгоритм определения схожести слов
K2	Письмо содержит информацию про коллегу и связанные с ним факты	Алгоритм поиска слова
K3	Письмо содержит информацию про приложение и связанные с ним факты	Алгоритм определения схожести слов
K4	Письмо содержит упоминание про навыки публичных выступлений автора	Алгоритм анализа темы предложения
K5	Письмо содержит упоминание про навыки публичных выступлений коллеги	Алгоритм анализа темы предложения
K6	Письмо содержит обращение к начальнику	Алгоритм поиска слова
K7	Письмо содержит просьбу о назначении автора	Работа со словарём
K8	Стиль письма соответствует заявленному формату задания	Работа со словарём
K9	Присутствует обращение	Регулярное выражение
K10	Присутствует разделение на абзацы	Регулярное выражение
K11	Присутствуют клишированные выражения	Работа со словарём
K12	Присутствует заключение (подпись)	Регулярное выражение
K13	Присутствуют связующие элементы и средства логической связи	Работа со словарём
K14	Наблюдается разнообразие корректно подобранных связующих элементов и средств логической связи	Работа со словарём

В первую очередь, письменное задание оценивается с точки зрения успешности выполнения коммуникативной задачи, то есть соответствия содержания письма заявленной теме и ситуации. Контроль решения коммуникативной задачи осуществлялся с помощью критериев K1–K7. В данном случае главной целью учащегося является обращение к начальнику с просьбой о назначении автора выступающим на презентации проекта. Для выполнения этой цели в письме требуется:

- упомянуть предстоящую презентацию (K1);
- упомянуть коллегу (K2) и навыки его публичных выступлений (K5);
- упомянуть приложение (K3);
- упомянуть навыки своих публичных выступлений (K4);
- обратиться к начальнику (K6) с просьбой о назначении автора выступающим на презентации проекта (K7).

Критерий считается выполненным, если учащимся было упомянуто слово или словосочетание, обозначающее представленные в описании критерия явления, людей и предметы, или использована грамматическая структура, на них указывающая.

Другим важным критерием успешности решения коммуникативной задачи является отсутствие в тексте письма фактических ошибок и/или информации, противоречащей заданию или заявленной ситуации общения.

Ещё одним критерием успешности решения коммуникативной задачи стало стилевое оформление (К8). Поскольку письмо, которое требуется написать, является примером деловой коммуникации, для решения поставленной коммуникативной задачи учащемуся необходимо продемонстрировать владение деловым стилем общения, в частности, умением вести деловую переписку, используя специальные стилистически маркированные языковые средства. Для проверки соответствия стилевого оформления нами был составлен список таких средств, т. е. наиболее часто употребляемой лексики, а также наиболее употребимых грамматических явлений, на базе ведущих и наиболее популярных учебно-методических комплексов по бизнес-английскому и деловой переписке, например, Oxford English for Emails. Критерий считался удовлетворённым, если учащийся не только употребил указанные в списке средства, но и не использовал средств, характерных для иных функциональных стилей речи.

Блок «Организация речи» содержит критерии, проверяющие соответствие заявленному формату задания, т. е. электронному письму начальнику (К9–К13). Такое письмо должно быть разделено на абзацы, в нём должны присутствовать обращение и заключение, характерные для электронной деловой переписки. Учащемуся также необходимо продемонстрировать владение соответствующими стилю клишированными выражениями.

С помощью критерия К14 осуществлялась оценка когезии текста, т. е. наличия связующих элементов и средств логической связи. Учащемуся важно не просто использовать разнообразные средства, но и употребить их корректно в соответствии с речевой ситуацией. Формализация этого критерия — наиболее сложная задача, в данной работе для её решения использовались специально составленные словари фраз и словосочетаний. Лексика и грамматика оцениваются опосредованно во всех критериях.

3. Алгоритмы определения параметров текста

Для оценки соответствия текста критериям К1–К6 были разработаны следующие алгоритмы: поиска слов в тексте, поиска сходных слов в тексте и анализа темы предложения.

Оценка соответствия текста критериям К9, К10 и К12 производится при помощи регулярных выражений, поскольку в соответствующих критериях можно явно выделить конструкции, которые должны присутствовать в тексте. Так для критерия К10 такой конструкцией является пустая строка между параграфами (два символа перехода на новую строку в строковом представлении текста). Для получения балла за критерий К9 необходимо иметь в первом абзаце письма формальное обращение к начальнику, такое как «Good morning, Dear Mrs. Addams» или «Good afternoon, Mrs. Addams» и подобные им. При оценке критерия К12 применяется регулярное выражение, отслеживающее наличие в подписи полного имени человека, например, Ivanov Ivan Ivanovich или Mrs. Addams. Следует отметить, что система обнаруживает и более сложные имена, такие как Ludwig Heinrich Edler von Mises, Muhammad ibn Musa al-Khwarizmi и множество других.

Для критериев К7, К8, К11, К13, К14 при проверке соответствия им текста используются специально составленные словари слов-связок, клишированных и формальных выражений.

3.1. Алгоритм поиска слов

Алгоритм выполняет последовательный поиск необходимого слова с начала текста. Например, при применении для определения соответствия текста критерию К2 происходит поиск упоминания

информации о коллеге отслеживанием использования слова «colleague» или любой из допустимых форм полного имени коллеги автора письма, а при проверке соответствия критерию К6 ищется обращение автора письма к начальнику. При оценке данных критериев достаточно самого использования слова в письме без учета контекста, а само их применение явно продиктовано постановкой задания.

3.2. Алгоритм поиска сходных слов

Алгоритм поиска сходных слов используется в случаях, когда необходимо определить, есть ли в тексте слово, максимально приближенное к заданному. На вход алгоритм получает слово для поиска, множество лемматизированных слов из письма без стоп-слов для английского языка (получаемых из библиотеки NLTK), а также множество слов-исключений.

Для решения этой задачи формируется множество, содержащее как само исходное слово, так и его гипонимы и гиперонимы. При необходимости есть возможность указания специальных слов-исключений, которые являются контекстуальными синонимами искомого слова. Они тоже будут включены в это множество. Наличие такой возможности обусловлено тем, что используемая библиотека не учитывает особенности предметной области. Таким образом получается обойти данную проблему. После формирования множества ищется его пересечение с множеством слов текста, очищенного от стоп-слов из библиотеки NLTK. Если пересечение не пусто — ставим за критерий один балл и прекращаем работу алгоритма. В противном случае необходимо выделить из искомого слова корень и начать искать однокоренное слово в тексте. При обнаружении — за критерий ставится один балл, а иначе ставится ноль баллов.

В алгоритме для оценки соответствия критерию К1 используется слово «presentation», а множество слов-исключений пустое. В алгоритме для оценки соответствия критерию К3 применяется слово «application», а множество слов-исключений содержит только слово «product».

3.3. Алгоритм анализа темы предложения

Алгоритм 1 анализа темы предложения применяется для поиска в тексте упоминаний о навыках публичного выступления автора текста и его коллеги. Подходящие предложения из всех предложений письма определяются с помощью ключевого глагола, существительного и прилагательного, а также их гипонимов и гиперонимов, кроме того, может учитываться присутствие в предложении отрицания. По наличию ключевых слов или сходных с ними слов в тексте принимается решение о соответствии предложения требуемой теме. Поскольку одну мысль возможно выразить множеством способов, данный алгоритм может быть применен для одного критерия несколько раз для охвата большего количества возможных комбинаций с различными наборами ключевых слов. При работе с прилагательными наличие сравнительных прилагательных или прилагательных превосходной степени (JJR и JJS соответственно) в тексте следует учитывать как альтернативу ключевому прилагательному или сходному к нему слову. Возможность учета отрицания в предложении необходима по причине того, что негативная характеристика навыков публичного предложения коллеги может быть описана не только при помощи негативного прилагательного, но и положительного с применением отрицания в тексте. Таким образом алгоритм учитывает большее количество возможных описаний негативной характеристики коллеги.

При помощи NLTK_mark_pos_tags происходит токенизация предложения и определение PoS-тега для каждого слова.

При помощи NLTK_build_lemmatized_sentence_word_set происходит составление лемматизированного множества слов текущего предложения.

При реализации обоих функций используется библиотека NLTK.

При проверке соответствия критерию К4 используются следующие данные: «speaker», «represent», «adequate» — ключевые слова, «I», «myself» — местоимения. Отрицание не ищется.

Algorithm 1. Sentence topic analysis algorithm

Algorithm 1. Алгоритм анализа темы предложения

```
Data: sentence_list, noun_verb_word_set, adj_word_set, pronoun_list, check_negation
for sentence in sentence_list do
    sentence_pos_tags ← NLTK_mark_pos_tags(sentence);
    words_in_sentence_set ←
     NLTK_build_lemmatized_sentence_word_set(sentence_pos_tags);
    intersection_size ← len(words_in_sentence_set.intersection(noun_verb_word_set));
    pronoun_contained ← adjective_contained ← negation_contained ← False;
    for current_pos in sentence_pos_tags do
        if current_pos[0].lower() in pronoun_list then
            pronoun_contained ← True;
        end
        if current_pos[1] in ['JJS', 'JJR'] or current_pos[0].lower() in adj_word_set then
            adjective_contained ← True;
        end
        if check_negation and current_pos[0].lower() in ['n't', 'not', 'no', 'never'] then
            negation_contained ← True;
        end
    end
    sentence_suitability ← adjective_contained or negation_contained;
    sentence_suitability ← pronoun_contained and sentence_suitability;
    if intersection_size ≠ 0 and sentence_suitability then
        return 1;
    end
end
return 0;
```

При проверке соответствия критерию K5 используется два набора данных, а оценка ставится как максимум из ответов. Первый набор данных: «speaker», «represent», «adequate» — ключевые слова, «he», «she», «his», «her», «himself», «herself», «john» — набор местоимений. Отрицание ищется. Второй набор данных: «speaker», «represent», «incompetent» — ключевые слова, «he», «she», «his», «her», «himself», «herself», «john» — набор местоимений. Отрицание не ищется.

4. Результаты экспериментов

Разработанные алгоритмы легли в основу автоматической системы оценивания связного текста. С использованием этой системы была проведена экспериментальная оценка двадцати писем на английском языке, размеченных экспертами-лингвистами по указанным критериям. Результаты оценки эксперта-лингвиста 1 и системы 2 представлены в виде тепловых карт, на которых насыщенность цвета прямо пропорциональна величине оценки по указанному критерию. Оценки эксперта более тонкие за счет возможности выставления половинных баллов; система же выставляет по каждому критерию только один или ноль баллов.

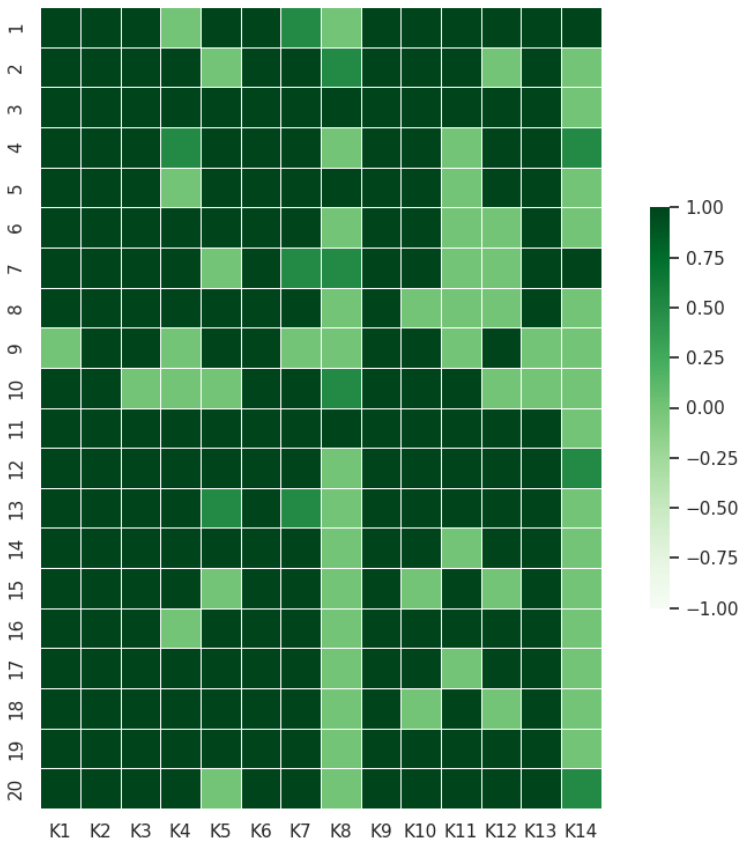


Fig. 1. Heat map of the lingist's estimates

Рис. 1. Тепловая карта оценок эксперта-лингвиста

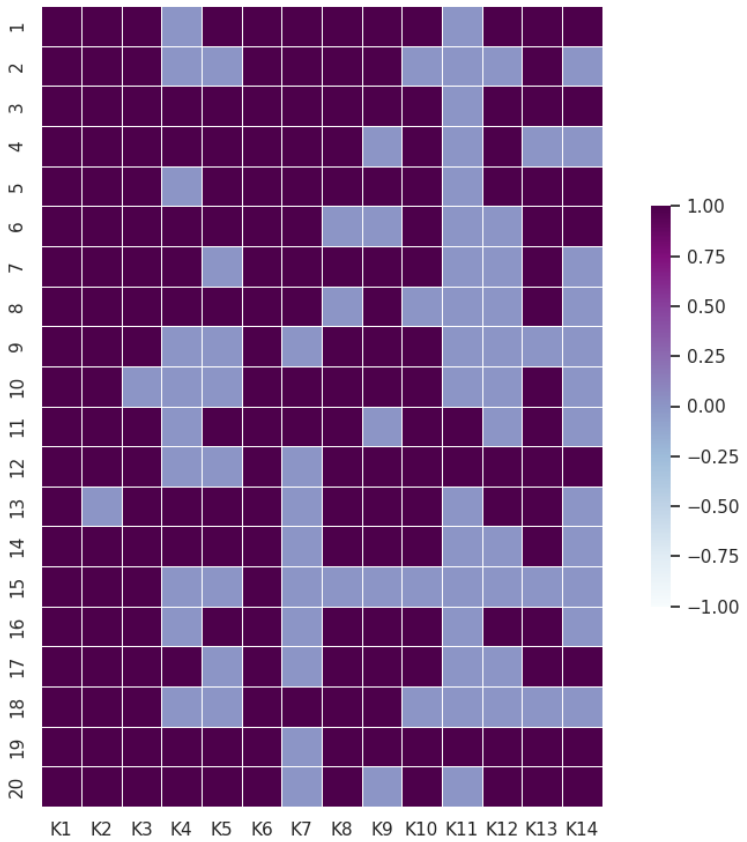


Fig. 2. Heat map of estimates of the developed system

Рис. 2. Тепловая карта оценок разработанной системы

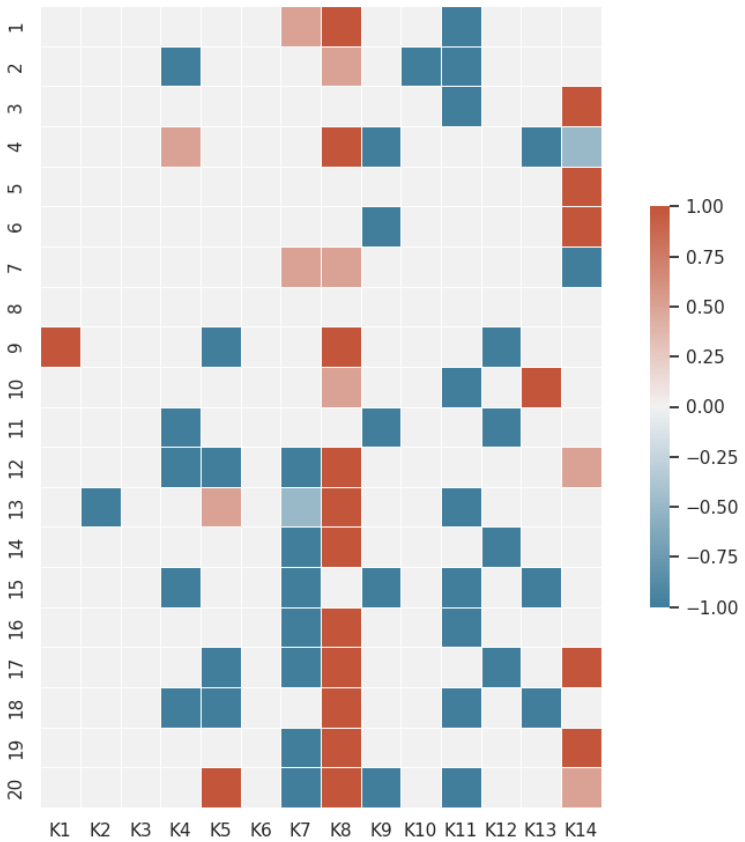


Fig. 3. Estimation difference heat map

Рис. 3. Тепловая карта разницы оценок

Различия между оценками системы и эксперта также представлены в виде соответствующей тепловой карты 3. На ней красными тонами показаны ложно-позитивные ошибки системы (система поставила балл за критерий, а эксперт — нет), а синими тонами показаны ложно-негативные (система не поставила балл, а эксперт — поставил). Белые ячейки обозначают совпадение оценок. По данной карте видно, что присутствуют как хорошо оцениваемые критерии: К1, К2, К3, К6 и К10, так и достаточно проблемные: К7, К8, К11, К14.

Был также проведён кластерный анализ текстов, на которых проводились эксперименты. Для представления четырнадцатимерных векторов оценок на двухмерном пространстве применялся алгоритм нелинейного снижения размерности UMAP. Поскольку алгоритм использует стартовые точки для выполнения работы, то построение графиков выполнялось неоднократно с различными значениями. Благодаря этому были получены представленные графики 4, 5, позволяющие отобразить расхождения оценки системы и экспертов на деловых письмах.

Как видно из обоих графиков, система автоматически разделила письма на два кластера. Первый кластер (письма 2, 8, 15, 18), несмотря на явные лингвистические различия в самих текстах, имеет ряд сходств, а именно одинаковые оценки по ряду критериев. Так, все четыре работы получили 1 экспертный балл за критерии К1–К4. Три работы из четырёх (2, 15, 18) получили 0 баллов за крите-

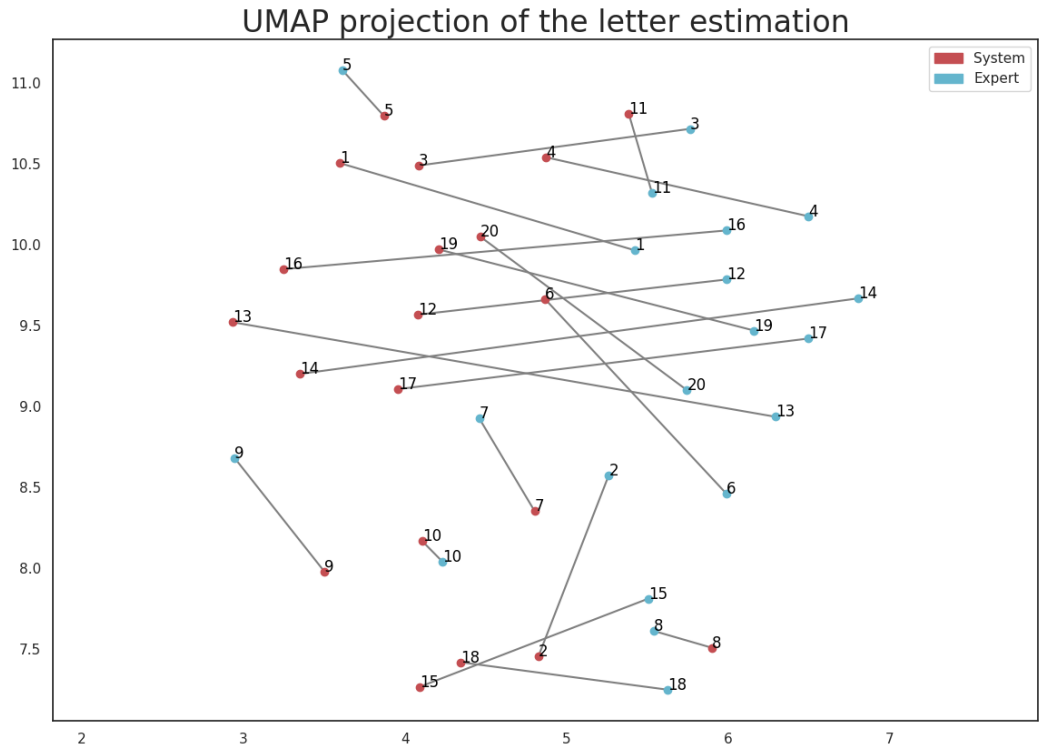


Fig. 4. Two-dimensional representation of score vectors using UMAP 1

Рис. 4. Двумерное представление векторов оценок при помощи UMAP 1

рии К4 и К14 при экспертной оценке в 1 балл. Все четыре письма не удовлетворяют критерию К8, но удовлетворяют критерию К9. В 8, 15 и 18 письмах отсутствует разделение на абзацы (критерий К0). В трех из четырех случаев (2, 15, 18) присутствует достаточное количество клишированных выражений (критерий К11). Во всех 4 письмах также отсутствует подпись (критерий К12), присутствуют логические средства связи (критерий К13), но они употреблены либо некорректно, либо в недостаточном количестве (критерий К14).

Во втором кластере (письма 1, 3, 4, 6, 11, 12, 13, 14, 16, 17, 19, 20) также прослеживается сходство. Во всех работах критерии К1–К3 и критерий К13 были оценены экспертом в 1 балл, в 10 случаях из 12 (кроме 1 и 16 писем) — так же был оценен критерий К4. В 10 случаях (кроме 3 и 11 писем) критерий К8 был обнулен экспертом, при этом в 9 из 10 таких писем по данному критерию был автоматически ошибочно выставлен 1 балл. Во всех письмах находилась информация, удовлетворяющая критериям К6 и К7, однако при этом в 7 из 12 писем по критерию К7 было автоматически выставлено 0 баллов. В 10 из 12 писем (кроме 6 и 20) присутствуют заключение и подпись (критерий К12), но отсутствует разнообразие корректно подобранных связующих элементов — в данном случае такое явление наблюдается во всех письмах кроме писем 4 и 12.

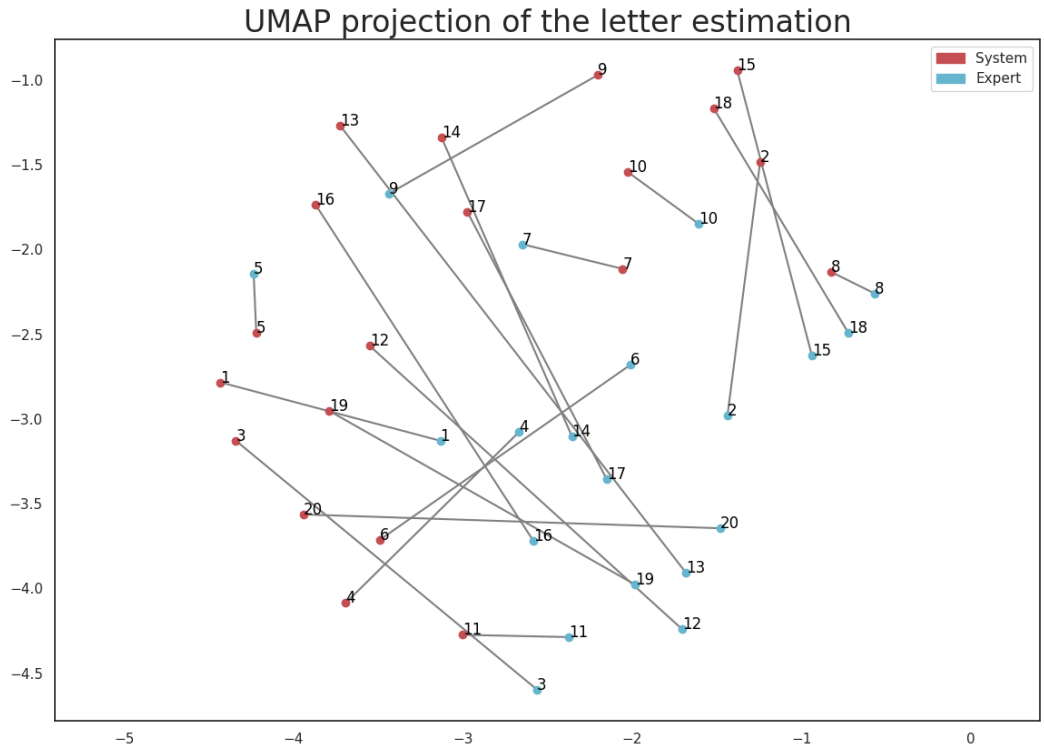


Fig. 5. Two-dimensional representation of score vectors using UMAP 2

Рис. 5. Двумерное представление векторов оценок при помощи UMAP 2

5. Обсуждение результатов

Как видно из полученных результатов, в части «Решение коммуникативной задачи» наибольшее количество проблем при автоматической оценке вызвали критерии К4, К5, К7 и К8. Это может быть связано с тем, что для решения данных задач учащиеся, как правило, использовали не ключевые слова и выражения, а языковые средства, которые не называли напрямую, но указывали на необходимые факты или явления, т. е. были логически связаны с искомыми фактами или явлениями. Так, в частности, при оценке соответствия критерию К4 (письмо содержит упоминание про навыки публичных выступлений автора) в письмах 2, 11, 12, 15 и 18 были использованы выражения и слова, которые отражают причину или следствие обладания навыками публичных выступлений (здесь и далее все ошибки сохранены): «The presentation will sound more dignified on my behalf» (письмо 11, следствие того, что у автора есть все необходимые для этого навыки), «I have excellent knowledge of these skills and can present our product» (письмо 18, причина и следствие + оценочное прилагательное excellent). В ряде случаев, как, например, в письме 11, решению задачи способствует исключительно контекст, когерентность и использование оценочной лексики: «I am well acquainted with the features of our application, because I was directly involved in its creation», которое позиционно находится сразу после упоминания навыков публичных выступлений коллеги и просьбы о своем назначении. Все вышеупомянутые случаи крайне трудно поддаются формализации, поскольку, во-первых, не содержат в себе каких-либо постоянных ключевых слов или выражений, которые можно было бы выделить в отдельную группу, а во-вторых, зависят от позиционирования и когерентности, которая является крайне субъективным и нечетким явлением по своей природе.

Похожая картина видна и при анализе результатов автоматический проверки критерия К5 (письмо содержит упоминание про навыки публичных выступлений коллеги). Успешности автоматической проверки в данном случае также препятствовало решение учащимися данной задачи с помощью логической связи, контекстуализации и когерентности: письмо 18, «he does not have the skills to speak in public» подразумевается, что он не сможет выступить должным образом; письмо 17: «if I make a presentation of the application, there are more chances to impress customers», следовательно, у коллеги шансов меньше.

Ошибки наблюдаются также и при автоматической оценке критерия К7 (письмо содержит просьбу о назначении автора). Как и при оценке критериев К4 и К5, возможной причиной этому явлению послужила невозможность оценить системой имплицитную логическую связь. Письмо 17: «Please, do not find me rude but if I make a presentation of the application, there are more chances to impress customers. I hope you will do a right choice», просьба о своем назначении имплицитно присутствует и выражается с помощью прилагательного many в сравнительной степени и выражения «make the right choice», где учащимся была допущена ошибка в употреблении коллокаций. В ряде случаев учащимися были использованы слова и выражения, которые могут быть отнесены к синонимам, антонимам, гипонимам и гиперонимам ключевых выражений или являться самими ключевыми выражениями, но не удовлетворять условиям иных критериев. Письмо 15: «I want to ask you assign me as the application presenter». Технически учащийся решает коммуникативную задачу, используя выражения, прямо называющие, а не указывающие на явление. Однако в данном случае при составлении этого предложения учащимся был допущен ряд других ошибок, что могло стать возможной причиной того, что система не смогла оценить соответствие данному критерию.

Наибольшее количество ошибок в блоке «Решение коммуникативной задачи» было выявлено при автоматической оценке критерия К8 (стиль письма соответствует заявленному формату задания): в 14 из 20 писем. Это может объясняться, во-первых, тем, что принадлежность текста к тому или иному функциональному стилю проявляется с помощью языковых единиц разных уровней языковой системы, из чего следует, что проверка такой принадлежности должна представлять собой сложную, разветвленную систему ряда параметров. Реализация такой системы на практике крайне затруднительна и, помимо этого, будет осложняться тем, что общепринятых утвержденных подробных списков и словарей таких единиц на настоящий момент не существует. Во-вторых, даже при условии использования учащимся стилистически корректных единиц, совершенные им/ей ошибки, как правило, нивелируют все корректно употребленные единицы. Так, автором письма 12 были корректно употреблено как и клишированное выражение «Dear Mrs Addams, I am writing to you about. . . », так и лексические средства, которые чаще употребляются именно в текстах официально-делового стиля, однако использования сокращений: «That's why», неуместной оценочной лексики: «bad public speaker, was a total failure» привело к обнулению экспертом баллов по данному критерию.

Наибольшие затруднения при автоматической проверке вызвал блок «Организация речи». Так, в данном блоке больше всего ошибок при автоматической проверке писем наблюдалось в критериях К9, К11, К12, К13 и К14. В частности, ошибки при проверке критериев К9 (присутствует обращение) и К12 (присутствует заключение или подпись), которые проверялись с помощью регулярных выражений, в первую очередь связаны с тем, что содержащиеся в письмах обращения и подписи могли либо не иметь отступа, либо содержать в себе стилистические и/или лексические ошибки. Так, например, в письме 6 формально присутствует обращение «Hello Mrs Addams», однако учащимся была допущена стилистическая и пунктуационная ошибка. В ряде случаев в качестве заключения учащиеся использовали гиперонимы вместо конкретного имени, что формально считается подписью, хоть и не способствует решению коммуникативной задачи: в письме 11 «Yours sincerely, An employee of your company».

Интересно, что анализ расхождений оценок экспертов и автоматической системы сработал и в обратную сторону: выявил недостаточную объективность человека. При проверке критерия К11 причиной «ошибочной оценки» стал исключительно человеческий фактор: во всех 9 письмах, в которых за данный критерий было автоматически выставлено 0 баллов, действительно наблюдается недостаток клишированных выражений, однако эксперты излишне лояльно отнеслись к работам.

Многочисленные ошибки при автоматической оценке критериев К13 и К14 (присутствуют связующие элементы и средства логической связи и наблюдается разнообразие корректно подобранных связующих элементов и средств логической связи) вполне логичны и могут объясняться самой природой механизмов когезии. Унификация требований к таким механизмам крайне сложна в силу объективных причин, поскольку, во-первых, связность текста обеспечивается за счет языковых средств разного уровня, а не только лексического, а во-вторых, сама корректность употребления таких средств будет всегда зависеть от контекста и от их позиции в тексте, а также от стилистической принадлежности таких средств к тому или иному функциональному стилю. В частности, в письме 19 автором было употреблено «And early I was a public speaker a few times so I understand that I must do». Само употребление «and» в данном случае не является ошибкой и вполне допустимо в официально-деловой переписке, однако в данном случае его позиция в начале предложения некорректна.

Однако в целом можно с уверенностью утверждать, что в подавляющем большинстве случаев система справилась с оценкой письма по выделенным критериям. Выявленные расхождения в оценке системой и в экспертной оценке объясняются комплексной, многоаспектной природой проверяемых явлений и сложностью их унификации, а также человеческим фактором, т. е. более лояльным отношением эксперта и более строгой и объективной автоматической оценкой.

Заключение

Модель текста, описанная в статье, включает набор числовых параметров, формализующих детали оценки письма на заданную тему. Основная часть этих параметров легко адаптируется к проверке связных текстов в других предметных областях. Проведенные эксперименты показали, что в большинстве случаев значимых различий между оценкой эксперта и оценкой системы выявлено не было. Это свидетельствует о том, что автоматическая проверка письменного задания является весьма эффективным инструментом оценивания, обладающим огромным и неоспоримым потенциалом в обучении иностранным языкам. Она также более объективна: система строго придерживается заранее заданных параметров, тем самым полностью исключая человеческий фактор. Те критерии, в которых наблюдались расхождения в оценках эксперта и системы сложны для алгоритмизации. Работа над формализацией таких критериев представляет собой перспективу для дальнейшего исследования как с точки зрения анализа естественного языка, так и с точки зрения методики преподавания иностранных языков и лингвистики.

Интересно, что данное исследование также поставило ряд актуальных вопросов перед преподавательским сообществом и выявило необходимость в дальнейшем пересмотре принятой системы оценивания письменных заданий экспертом, в частности, выделения более четких и строгих критериев оценки письма, а также объективности самого оценивания. В перспективе данного исследования также лежит анализ ошибок, допущенных экспертами, а также исследование возможных причин их возникновения. Это связано с особенностями методов обработки естественного языка, основанных на правилах. Результаты работы этих методов могут быть интерпретированы и проанализированы с разных точек зрения. Перспективой развития такого анализа является построение автоматических систем обучения с обратной связью не только для учеников, но и преподавателей.

References

[1] A. Al-Bargi, "Exploring Online Writing Assessment Amid Covid-19: Challenges and Opportunities from Teachers' Perspectives", *Arab World English Journal*, pp. 3–21, 2022.

[2] N. P. Soboleva and M. A. Nilova, "Obuchenie pis'mu studentov gumanitarnyh special'nostej s ispol'zovaniem sovremennyh obrazovatel'nyh tekhnologij", *Kazanskij vestnik molodyh uchyonyh*, vol. 2, no. 5 (8), pp. 57–59, 2018.

[3] M. Fareed, A. Ashraf, and M. Bilal, "ESL learners' writing skills: Problems, factors and suggestions", *Journal of education and social sciences*, vol. 4, no. 2, pp. 81–92, 2016.

[4] K. N. A. Al-Mwzaiji and A. A. F. Alzubi, "Online self-evaluation: the EFL writing skills in focus", *Asian-Pacific Journal of Second and Foreign Language Education*, vol. 7, no. 1, pp. 1–16, 2022.

[5] M. A. Hussein, H. Hassan, and M. Nassef, "Automated language essay scoring systems: A literature review", *PeerJ Computer Science*, vol. 5, e208, 2019.

[6] H. John Bernardin, S. Thomason, M. Ronald Buckley, and J. S. Kane, "Rater rating-level bias and accuracy in performance appraisals: The impact of rater personality, performance management competence, and rater accountability", *Human Resource Management*, vol. 55, no. 2, pp. 321–340, 2016.

[7] Z. Ke and V. Ng, "Automated Essay Scoring: A Survey of the State of the Art.", in *IJCAI*, vol. 19, 2019, pp. 6300–6308.

[8] M. Uto, "A review of deep-neural automated essay scoring models", *Behaviormetrika*, vol. 48, no. 2, pp. 459–484, 2021.

[9] S. Vajjala, "Automated assessment of non-native learner essays: Investigating the role of linguistic features", *International Journal of Artificial Intelligence in Education*, vol. 28, no. 1, pp. 79–105, 2018.

[10] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring", in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1882–1891.

[11] L. Xia, J. Liu, and Z. Zhang, "Automatic essay scoring model based on two-layer bi-directional long-short term memory network", in *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, 2019, pp. 133–137.

[12] M. Uto and M. Okano, "Robust neural automated essay scoring using item response theory", in *International Conference on Artificial Intelligence in Education*, Springer, 2020, pp. 549–561.

[13] Y. Tay, M. Phan, L. A. Tuan, and S. C. Hui, "Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018, pp. 5948–5955.

[14] Y. Farag, H. Yannakoudakis, and T. Briscoe, "Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics (ACL), 2018, pp. 263–271.

[15] Y. Yang and J. Zhong, "Automated essay scoring via example-based learning", in *International Conference on Web Engineering*, Springer, 2021, pp. 201–208.

[16] E. Mayfield and A. W. Black, "Should you fine-tune BERT for automated essay scoring?", in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 151–162.

[17] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking", in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1560–1569.

[18] M. Uto, Y. Xie, and M. Ueno, "Neural automated essay scoring incorporating handcrafted features", in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6077–6088.

[19] I. Aomi, E. Tsutsumi, M. Uto, and M. Ueno, "Integration of automated essay scoring models using item response theory", in *International Conference on Artificial Intelligence in Education*, Springer, 2021, pp. 54–59.

[20] W. Zhu and Y. Sun, "Automated Essay Scoring System using Multi-Model Machine Learning", in *CS & IT Conference Proceedings*, CS & IT Conference Proceedings, vol. 10, 2020, pp. 109–117.

[21] S. M. Darwish and S. K. Mohamed, "Automated essay evaluation based on fusion of fuzzy ontology and latent semantic analysis", in *International Conference on Advanced Machine Learning Technologies and Applications*, Springer, 2019, pp. 566–575.

[22] Y. Salim, V. Stevanus, E. Barlian, A. C. Sari, and D. Suhartono, "Automated English Digital Essay Grader Using Machine Learning", in *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*, IEEE, 2019, pp. 1–6.

[23] R. Wilkens, D. Seibert, X. Wang, and T. François, "MWE for Essay Scoring English as a Foreign Language", in *2nd Workshop on Tools and Resources for REAding DIfficulties (READI)*, 2022, pp. 62–69.

[24] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review", *Artificial Intelligence Review*, pp. 1–33, 2021.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 29, NO. 4, 2022
journal homepage: www.mais-journal.ru

ALGORITHMS

Remarks on the Reachability Graphs of Petri Nets

Y. A. Belov[1]

DOI: 10.18255/1818-1015-2022-4-366-371

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

MSC2020: 68Q85
Research article
Full text in Russian

Received October 7, 2022
After revision November 28, 2022
Accepted November 30, 2022

The question is considered – which graphs are isomorphic to the reachability graphs of Petri nets. Reachability graphs, or sets of achievable states, represent sets of all possible different network states resulting from a given initial state s_0 by a finite chain of permissible transitions. They have a natural structure of an oriented graph with a dedicated initial state, all other states of which are reachable from the initial one, taking into account orientation. At the same time, if the network transitions are marked, the reachability graphs also receive the corresponding marks of all arcs. At the same time, the concept of isomorphism of marked graphs arises, but this publication deals only with issues for networks without markings. Even for this simpler case, some questions remain open. The paper proves that any finite directed graph is modeled by a suitable Petri net, that is, it is isomorphic to the reachability graph of the network. For infinite graphs, examples of non-modeled graphs are given.

Keywords: Petri nets; network reachability graph; network coverage graph; graph isomorphism

INFORMATION ABOUT THE AUTHORS

Yuriy Anatol'yevich Belov
correspondence author

orcid.org/0000-0002-4221-6470. E-mail: belov45@yandex.ru
PhD in Mathematics.

Funding: This work was supported by P. G. Demidov Yaroslavl State University Project № VIP-016.

For citation: Y. A. Belov, "Remarks on the Reachability Graphs of Petri Nets", *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 366-371, 2022.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 29, № 4, 2022
сайт журнала: www.mais-journal.ru

Замечания о графах достижимости сетей Петри

Ю. А. Белов[1]

DOI: 10.18255/1818-1015-2022-4-366-371

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, д. 14, г. Ярославль, 150003 Россия.

УДК 681.3
Научная статья
Полный текст на русском языке

Получена 7 октября 2022 г.
После доработки 28 ноября 2022 г.
Принята к публикации 30 ноября 2022 г.

Рассматривается вопрос – какие графы изоморфны графам достижимости сетей Петри. Графы достижимости, или множества достижимых состояний, представляют множества всевозможных различных состояний сети, получающихся из данного начального состояния s_0 конечной цепочкой допустимых переходов. Они имеют естественную структуру ориентированного графа с выделенным начальным состоянием, все другие состояния которого достижимы из начального с учётом ориентации. При этом, если переходы сети снабжены пометками, графы достижимости также получают соответствующие пометки всех дуг. При этом возникает понятие изоморфизма размеченных графов, но в данной публикации рассматриваются лишь вопросы для сетей без разметок. Даже для этого более простого случая некоторые вопросы остаются открытыми.
В заметке доказывается, что любой конечный ориентированный граф моделируется подходящей сетью Петри, то есть он изоморфен графу достижимости сети. Для бесконечных графов приводятся примеры не моделируемых графов. Ставятся некоторые открытые вопросы по теме.

Ключевые слова: сети Петри; граф достижимости сети; граф покрытия сети; изоморфизм графов

ИНФОРМАЦИЯ ОБ АВТОРАХ

Юрий Анатольевич Белов автор для корреспонденции	orcid.org/0000-0002-4221-6470. E-mail: belov45@yandex.ru доцент, кандидат физ.-мат. наук.

Финансирование: Работа выполнена в рамках инициативной НИР ЯрГУ им. П. Г. Демидова № VIP-016.

Для цитирования: Y. A. Belov, “Remarks on the Reachability Graphs of Petri Nets”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 366-371, 2022.

Введение

Для анализа функционирования распределённых систем часто используется аппарат сетей Петри см. [1—3]. При этом теоретическое рассмотрение свойств сети обычно приводит к необходимости изучения так называемого дерева или графа достижимости данной сети. Свойства соответствующего графа достижимости, конечно, однозначно определяются исходной сетью. Однако вопросы, связанные, например, с тем, какие свойства сети обеспечивают наличие определённых свойств графа достижимости, чаще всего не имеют удовлетворительных ответов.

Данная заметка как раз касается этих задач.

1. Основные понятия

Определения. Сеть Петри $D(P, T)$ есть двудольный ориентированный граф со взвешенными дугами [2]. Вершины доли P называются позициями, вершины доли T – переходами, $P \cap T = \emptyset$. Размерность сети есть $|P| = d$. Множество дуг – это $F \subset (P \times T) \cup (T \times P)$. $w : F \longrightarrow N = \{0, 1, 2, ...\}$ – весовая функция, определяющая кратность дуг. Если кратность дуги равна нулю, то считаем, что соответствующая дуга отсутствует. Остальные дуги, т. е. дуги с положительными кратностями, образуют для каждого перехода t множество входных дуг: $(p, t) \in (P \times T)$ и соответственно, множество выходных дуг $(t, p) \in (T \times P)$ с соответствующими весами. Вектор $in(t) = (w(p, t)|p \in P)$ кратностей входящих дуг называется вектором ресурсов, потребляемых данным переходом t, вектор $out(t) = (w(t, p)|p \in P)$ – вектором производимых ресурсов. Вектор $\delta(t_i) = out(t_i) - in(t_i)$ называется вектором сдвига данного перехода t_i.

Состоянием сети D называется целочисленный неотрицательный вектор $s = (p_1, p_2, ...p_d)$, задающий количество абстрактных неименованных фишек, располагаемых в каждой позиции, то есть одно из отображений $P \longrightarrow N$.

2. Моделируемые графы

Переход t_i называется активным (допустимым) в данном состоянии s, если выполняется неравенство $s \geq in(t_i)$ при покомпонентном сравнении [4]. Другими словами, количество имеющихся фишек в текущем состоянии сети не менее количества потребляемых ресурсов данного перехода (для каждой позиции). Каждый переход, активный в данном состоянии, может привести к изменению состояния, как говорят, может «сработать». Измение состояния сети происходит по правилу: $s' = s - in(t_i) + out(t_i) = s + \delta(t_i)$, что часто обозначается следующим образом: $s \xrightarrow{t_i} s'$. Таким образом, активные переходы могут перераспределять ресурсы (фишки) по позициям. Если в данном состоянии допустимы (активны) несколько переходов, то сеть может перейти в любое из допустимых состояний. Рассматривая всевозможные конечные цепочки допустимых переходов, начинающиеся из состояния s_0, получаем множество $Re(s_0)$ всевозможных состояний, достижимых из данного начального состояния s_0.

Хотя данное множество может быть и бесконечным, оно имеет естественную структуру ориентированного графа с выделенным начальным состоянием.

При этом все вершины достижимы из начальной с учётом ориентации, и все вершины можно считать помеченными неотрицательными векторами, указывающими текущее «распределение ресурсов по позициям».

Можно считать, что такой конечный граф является подграфом некоторого полного ориентированного графа порядка d без петель и параллельных дуг с выделенной вершиной. При этом полустепень захода и полустепень исхода каждой вершины равна $d - 1$. Возникает естественный вопрос: какие графы изоморфны, как ориентированные графы, графам $Re(s_0)$ достижимости подходящих сетей Петри. Такие графы будем называть моделируемыми.

Можно проверить, что конечный полный ориентированный граф, отмеченный ранее, является моделируемым. При этом он моделируется даже неизоморфными сетями, например, сетями различных размерностей. Вообще, для конечных графов многое ясно.
Однако графы $Re(s_0)$ могут быть и бесконечными, и здесь на основные вопросы пока нет приемлемых ответов.

3. Результаты

Теорема 1. *Всякий конечный ориентированный граф G порядка d без петель и параллельных дуг моделируется некоторой сетью Петри порядка d, то есть изоморфен как ориентированный граф графу достижимости $Re(s_0)$ некоторой подходящей сети Петри. При этом сеть может быть выбрана даже автоматной [2] с одной циркулирующей фишкой.*

Доказательство. Пусть имеется вершинная разметка исходного графа, то есть просто произвольная нумерация вершин. Для доказательства первого утверждения сопоставим каждой вершине 0-1-вектор размерности d с одной единицей на месте номера вершины и остальными нулями. Далее определяются переходы, потребляемыми ресурсами которых становятся одна фишка и производимыми ресурсами – также одна фишка. Например, первой вершине соответствует вектор $(1, 0, ...0)$, третьей вершине – вектор $(0, 0, 1, ...0)$. Тогда переход t_{13} из первой вершины в третью определится так: $\delta(t_{13}) = (-1, 0, 1, ...0)$. Таким образом получится некоторая автоматная сеть Петри с начальным состоянием $(1, 0, ...0)$. Можно проверить, что граф достижимых состояний данной сети изоморфен исходному заданному ориентированному графу. □

В связи с первым предложением можно рассматривать вопросы моделируемости различных классов графов различными типами сетей Петри. Кроме того, интересно указать минимум размерности сети, моделирующей данный граф. Здесь можно увидеть некоторый аналог алгоритма минимизации диаграммы, реализующей данный формальный регулярный язык [5].

Теорема 2. *Для любого $k > 1$ существует не моделируемое ориентированное бесконечное счётное выходящее корневое дерево [6], в котором вершин k-го уровня имеется имеется ровно одна или две.*

Доказательство. Рассмотрим сначала бесконечную возрастающую цепь из начальной вершины s_0.

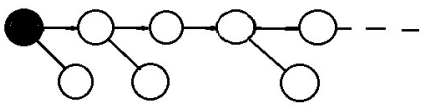


Fig. 1. Non-modeled graph with a limited number of vertices of all levels

Рис. 1. Не моделируемый граф с ограниченным количеством вершин всех уровней

Далее, к соответствующим вершинам цепи присоединяем по одной дуге и одной концевой вершине, назовём это «отросток». Присоединяем отросток к первой вершине, затем к третьей вершине

цепи, далее к шестой, потом к десятой и т. д., увеличивая на каждом шаге построения расстояние между отростками, примерно, как указано на рис. 1.

Полученное дерево не моделируемо. Предположим противное. Пусть дерево моделируется сетью размерности d. Тогда каждой вершине цепи будет соответствовать неотрицательный целочисленный вектор длины d, описывающий текущее состояние сети.

Из любой бесконечной последовательности неотрицательных целочисленных векторов можно выделить бесконечную строго возрастающую подпоследовательность [2, 4]. Отметим ещё свойство монотонности активности: если некоторый переход сети активен в данном состоянии, то он останется активным и в любом состоянии большем, чем данное [4]. Рассмотрим теперь хотя бы первую вершину возрастающей подпоследовательности. На некотором расстоянии от неё, например, через q шагов, имеется отросток. Через какое-то фиксированное расстояние m от этой вершины имеется следующий элемент возрастающей подпоследовательности. Он строго больше, чем первый элемент, поэтому все переходы, активные в первом случае, остаются активными и сейчас. Все последующие элементы наследуют это свойство активности, поэтому через q шагов от второго элемента снова будет находиться отросток. Таким образом, как угодно далеко вдоль цепи будут находиться отрезки фиксированной длины m, содержащие два отростка, что противоречит строению дерева. □

По поводу второго предложения можно ещё отметить, что построить не моделируемый бесконечный граф можно проще, используя ограничение на скорость роста количества вершин данного уровня в любом моделируемом графе.

Способ построения не моделируемого графа, опирающийся на эти соображения, автор узнал от В. А. Башкина, за что ему искренне благодарен.

Действительно, в таком графе количество вершин данного уровня (см. [6]) может быть не более, чем степенным, что довольно просто доказать. Это следует из того, что в любом графе достижимости, т. е. моделируемом, каждое достижимое состояние определяется только множеством переходов, приводящих к этому состоянию, и не зависит от порядка выполнения данных переходов. Поэтому привести пример соответствующего не моделируемого графа, нарушающего данное условие нетрудно – например, свободное однородное (бесконечное) дерево степени 3 (см. рис 2).

Но в предложении 2 приводится, возможно, более интересный пример, в котором количество вершин данного уровня вообще не растёт, оно всегда равно 1 или 2, однако граф, в данном случае даже дерево, не моделируем. Это означает, что ограничение на рост не является достаточным условием моделируемости. Никаких достаточных условий или критериев моделируемости, видимо, пока нет, во всяком случае, автору об этом ничего неизвестно.

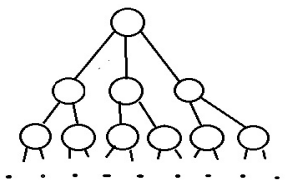


Fig. 2. An example of a non-modeled graph with too many vertices of the current level

Рис. 2. Пример не моделируемого графа со слишком большим количеством вершин текущего уровня

Заключение

Другой интересный вопрос, как представляется, указать также условия, при которых две сети Петри имеют изоморфные графы достижимости.

В случае конечных графов достижимости вопрос решается положительно с использованием конструкции дерева (или графа) покрытия [6], а для бесконечных графов достижимости позитивных результатов, видимо, нет. Граф покрытия всегда конечен, его можно считать некоторой аппроксимацией (точного) графа достижимости и этот граф совпадает с графом достижимости в конечном случае.

Ясно только, что для данного достижимого графа можно построить бесконечную серию сетей, размерности которых уходят в бесконечность, но все сети серии имеют графы, изоморфные данному.

Другими словами, уточняя основной вопрос, если для двух сетей графы покрытия изоморфны, при каких дополнительных условиях будут изоморфны и графы достижимости, видимо, является открытым.

References

[1] W. Reisig, *Petri Nets. An Introduction*. Springer-Verlag, New York, 1985.

[2] V. E. Kotov, *Petri Nets*, in Russian. Moscow, Nauka, 1984.

[3] M. Diaz, *Petri Nets: Fundamental Models, Verification and Applications*. J. Wiley, Inc. USA, 2009.

[4] H.-C. Yen, “Introduction to Petri Net Theory”, *Recent Advances in Formal Languages and Applications*, vol. 25, pp. 343–373, 2006.

[5] E. V. Kuzmin and V. A. Sokolov, *Automata Counter Machines*, in Russian. P.G. Demidov Yaroslavl State University, 2012.

[6] V. A. Evstigneev and V. N. Kasianov, *Teoria Grafov. Algoritmy obrabotki dereview*, in Russian. Nauka, Novosibirsk, 1984.

MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 29, NO. 4, 2022
journal homepage: www.mais-journal.ru

ALGORITHMS

The Polynomial Algorithm of Finding the Shortest Path in a Divisible Multiple Graph

A. V. Smirnov[1]

DOI: 10.18255/1818-1015-2022-4-372-387

[1]P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

MSC2020: 05C38, 05C65
Research article
Full text in Russian

Received August 23, 2022
After revision November 7, 2022
Accepted November 9, 2022

In this paper, we study undirected multiple graphs of any natural multiplicity $k > 1$. There are edges of three types: ordinary edges, multiple edges and multi-edges. Each edge of the last two types is a union of k linked edges, which connect 2 or $(k + 1)$ vertices, correspondingly. The linked edges should be used simultaneously. If a vertex is incident to a multiple edge, it can be also incident to other multiple edges and it can be the common end of k linked edges of some multi-edge. If a vertex is the common end of some multi-edge, it cannot be the common end of another multi-edge.
Divisible multiple graphs are characterized by a possibility to divide the graph into k parts, which are adjusted on the linked edges and which have no common edges. Each part is an ordinary graph. As for an ordinary graph, we can define the integer function of the length of an edge for a multiple graph and set the problem of the shortest path joining two vertices. Any multiple path is a union of k ordinary paths, which are adjusted on the linked edges of all multiple and multi-edges. In the article, we show that the problem of the shortest path is polynomial for a divisible multiple graph. The corresponding polynomial algorithm is formulated. Also we suggest the modification of the algorithm for the case of an arbitrary multiple graph. This modification has an exponential complexity in the parameter k.

Keywords: multiple graph; divisible graph; multiple path; shortest path; reachability set; polynomial algorithm

INFORMATION ABOUT THE AUTHORS

Alexander Valeryevich Smirnov
correspondence author

orcid.org/0000-0002-0980-2507. E-mail: alexander_sm@mail.ru
PhD, Associate Professor, Department of Theoretical Computer Science.

Funding: This work was supported by P.G. Demidov Yaroslavl State University Project № VIP-016.

For citation: A. V. Smirnov, "The Polynomial Algorithm of Finding the Shortest Path in a Divisible Multiple Graph", *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 372-387, 2022.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 29, № 4, 2022
сайт журнала: www.mais-journal.ru

Полиномиальный алгоритм поиска кратчайшего пути в делимом кратном графе

А. В. Смирнов[1]

DOI: 10.18255/1818-1015-2022-4-372-387

[1]Ярославский государственный университет им. П. Г. Демидова, ул. Советская, д. 14, г. Ярославль, 150003 Россия.

УДК 519.17
Научная статья
Полный текст на русском языке

Получена 23 августа 2022 г.
После доработки 7 ноября 2022 г.
Принята к публикации 9 ноября 2022 г.

В статье рассматриваются неориентированные кратные графы произвольной натуральной кратности $k > 1$. Кратный граф содержит ребра трех типов: обычные, кратные и мультиребра. Ребра последних двух типов представляют собой объединение k связанных ребер, которые соединяют 2 или $(k + 1)$ вершину соответственно. Связанные ребра могут использоваться только согласованно. Если вершина инцидентна кратному ребру, то она может быть инцидентна другим кратным ребрам, а также она может быть общим концом k связанных ребер мультиребра. Если вершина является общим концом мультиребра, то она не может быть общим концом никакого другого мультиребра.

Делимые кратные графы характеризуются возможностью выделения k частей, согласованных на связанных ребрах и не содержащих общих ребер. Каждая часть представляет собой обычный граф. Как и для обычного графа, для кратного графа можно ввести целочисленную функцию длины ребра и поставить задачу о кратчайшем пути между двумя вершинами. Кратный путь является объединением k обычных путей, согласованных на связанных ребрах кратных и мультиребер. В статье показано, что задача о кратчайшем пути в делимом кратном графе является полиномиальной. Сформулирован соответствующий полиномиальный алгоритм. Также предложена модификация алгоритма для случая произвольного кратного графа. Эта модификация имеет экспоненциальную по параметру k трудоемкость.

Ключевые слова: кратный граф; делимый граф; кратный путь; кратчайший путь; множество достижимости; полиномиальный алгоритм

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Валерьевич Смирнов
автор для корреспонденции

orcid.org/0000-0002-0980-2507. E-mail: alexander_sm@mail.ru
канд. физ.-мат. наук, доцент, кафедра теоретической информатики.

Финансирование: Работа выполнена в рамках инициативной НИР ЯрГУ им. П.Г. Демидова № VIP-016.

Для цитирования: A. V. Smirnov, “The Polynomial Algorithm of Finding the Shortest Path in a Divisible Multiple Graph”, *Modeling and analysis of information systems*, vol. 29, no. 4, pp. 372-387, 2022.

Введение

В данной статье мы рассмотрим задачу о *кратчайшем пути* в кратном графе. Кратные графы содержат три типа ребер (обычные, кратные и мультиребра) и являются обобщением обычных графов – по сути, обычный граф имеет кратность $k = 1$. Определения кратного графа кратности $k > 1$ и делимого кратного графа были сформулированы в статье [1]. Там же была поставлена задача о кратчайшем кратном пути между двумя вершинами, дано определение связности кратного графа, получены полиномиальные алгоритмы ее проверки. В отличие от обычного графа, в связном кратном графе путь между двумя вершинами существует не всегда, и в работе [1] рассмотрен критерий существования кратного пути между двумя вершинами. Проверить условие критерия для любых двух вершин можно с помощью быстрых полиномиальных алгоритмов.

Среди других известных обобщений графов наиболее близкими нам концепциями являются мультиграфы, гиперграфы (см., например, [2, 3]), а также метаграфы (см. [4, 5]). Действительно, как и в мультиграфах, в кратных графах допускается наличие нескольких ребер между парой вершин (набор таких ребер мы будем в дальнейшем называть *кратным ребром*), однако в случае кратного графа количество таких ребер должно быть строго равным k. В кратных графах присутствуют *мультиребра*, соединяющие между собой $(k + 1)$ вершину. Но в отличие от гиперребер гиперграфа, мультиребро представляется в виде k связанных ребер, имеющих один общий конец, причем все эти k ребер должны использоваться согласованно. По сути, понятие мультиребра близко понятию ребра между вершиной и метавершиной в метаграфе. При этом в метаграфе, напомним, метапуть между двумя метавершинами фактически моделирует причинно-следственные связи в некоторой предметной области. Однако в кратном графе используется принципиально иной подход к определению пути: *кратный путь* должен состоять ровно из k обычных путей, проходящих по обычным ребрам, а также по связанным ребрам кратных и мультиребер; при этом пути должны быть согласованы (одинаковы) на кратных и мультиребрах. Поэтому кратный граф нельзя считать частным случаем метаграфа.

Отметим также, что частным случаем кратного графа является кратная сеть (см. [6, 7]). Задача о наибольшем потоке в кратной сети обобщает классическую задачу (см. [8]) и имеет ряд приложений в сфере экономики, управления, финансов. В частности, кратные сети и потоки используются для поиска решения NP-трудной задачи целочисленного сбалансирования трех- и четырехмерной матрицы (см., например, [9, 10]).

В работе [1] предложен экспоненциальный алгоритм нахождения кратного пути. В данной статье мы обоснуем полиномиальность данной задачи в случае делимого кратного графа и построим алгоритм, который будет решать ее за полиномиальное время. Делимые кратные графы характеризуются возможностью выделения k частей, согласованных на связанных ребрах и не содержащих общих ребер.

Затем мы рассмотрим модификацию алгоритма для случая произвольного графа. Во многих случаях модифицированный алгоритм будет работать достаточно быстро, однако в общем случае его трудоемкость будет экспоненциальной по параметру k (кратность графа).

1. Кратные графы и деревья: необходимые определения. Постановка задачи о кратчайшем кратном пути

Напомним несколько определений, связанных с кратными графами и путями, которые ранее были сформулированы в статьях [1, 11].

Определение 1. Кратный граф G *произвольной натуральной кратности* $k > 1$ *– это граф, вершины которого могут соединяться ребрами одного из 3 видов:*

1. Обычное ребро e^o*; множество обычных ребер обозначим через* E^o.

2. Кратное ребро e^k *между двумя вершинами, которое состоит из* k *одинаковых связанных ребер; связанные ребра кратного ребра могут использоваться только согласованно; множество кратных ребер обозначим через* E^k.
3. Связанное ребро e *между двумя вершинами, имеющее один общий конец с другим* $(k-1)$ *ребром (у любых двух из* k *связанных ребер только один конец является общим); множество связанных общей вершиной ребер будем называть* мультиребром e^m*; связанные ребра мультиребра могут использоваться только согласованно; множество мультиребер обозначим через* E^m.

Если вершина инцидентна какому-либо кратному ребру, то она может быть инцидентна другим кратным ребрам, а также она может быть общим концом какого-либо мультиребра.

Если вершина является общим концом какого-либо мультиребра, то она не может быть общим концом никакого другого мультиребра.

Если вершина является отдельным концом мультиребра или инцидентна обычному ребру, то она не может быть общим концом мультиребра и не может быть инцидентна кратному ребру.

Множества вершин и ребер графа G обозначим через V и E соответственно. Заметим, что $E = E^o \cup E^k \cup E^m$.

В данной статье рассматриваются только неориентированные кратные графы.

Рис. 1 и 2 иллюстрируют определение 1. В левой части рис. 1 кратное ребро представлено в виде объединения k одинаковых ребер между двумя вершинами, что показано штрихами. Равенство (или согласованность) связанных ребер предполагает, что все характеристики этих ребер (например, длина) одинаковы, и эти ребра могут использоваться только одновременно. Так, если осуществляется проход в определенном направлении по одному из связанных ребер, то одновременно с этим все остальные ребра проходятся в том же самом направлении. Кратное ребро может включаться в какие-либо новые структуры только целиком. В дальнейшем мы будем обозначать кратные ребра жирными линиями, как в правой части рис. 1.

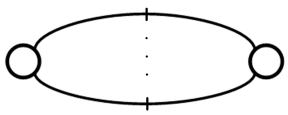


Fig. 1. Multiple edge

Рис. 1. Кратное ребро

В левой части рис. 2 мультиребро $\{x_0, \{x_1, \ldots, x_k\}\}$ представлено в виде объединения k одинаковых ребер, связывающих общую вершину x_0 с k разными вершинами $x_1, \ldots, x_k$. Как и на рис. 1, равенство ребер показано штрихами. Согласованность связанных ребер имеет тот же смысл, что и для кратных ребер. В дальнейшем мультиребра мы будем изображать при помощи расщепляющихся на k частей линий, как в правой части рис. 2.

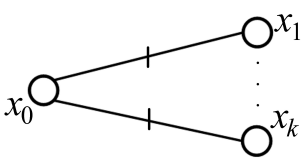


Fig. 2. Multi-edge

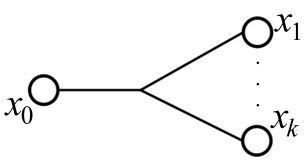


Рис. 2. Мультиребро

Определение 2. Обычной вершиной *назовем вершину, которая инцидентна обычному ребру или является отдельным концом мультиребра.*

Кратной вершиной *назовем вершину, которая инцидентна кратному ребру или является общим концом мультиребра.*

Из определения 1 следует, что множества обычных и кратных вершин не пересекаются. При этом кратная вершина может быть соединена с обычными только посредством мультиребра.

Определение 3. Делимым кратным графом *назовем такой граф, в котором между двумя концами одного мультиребра не существует пути, проходящего только по обычным ребрам.*

При удалении всех мультиребер делимый граф распадется на n компонент связности (связность здесь понимается в том же смысле, что и для обычных графов), каждая из которых содержит только кратные ребра либо только обычные ребра. При этом связанные ребра каждого мультиребра можно пронумеровать от 1 до k таким образом, что каждой компоненте связности, содержащей только обычные ребра, будут инцидентны связанные ребра мультиребер с одинаковыми номерами.

Определение 4. Частью G_i ($i \in \overline{1,k}$) *делимого графа* $G(V,E)$ *назовем подграф, содержащий связанные ребра с номером i всех кратных и мультиребер, а также компоненты связности, состоящие из обычных ребер и инцидентные i-ым связанным ребрам всех мультиребер.*

Каждая часть G_i является обычным графом. При этом возможность выделения частей G_i является особенностью делимых графов. В общем случае получить части G_i не удастся.

Пример 1. Рассмотрим кратный граф G кратности 2 со следующими множествами обычных, кратных и мультиребер:

$$E^k = \big\{\{x_1,x_3\},\ \{x_1,x_4\},\ \{x_2,x_3\},\ \{x_4,x_5\},\ \{x_{13},x_{14}\},\ \{x_{13},x_{15}\},\ \{x_{14},x_{15}\}\big\};$$

$$E^m = \big\{\{x_2,\{x_7,x_{10}\}\},\ \{x_3,\{x_6,x_9\}\},\ \{x_4,\{x_6,x_9\}\},\ \{x_5,\{x_8,x_9\}\},\ \{x_{13},\{x_8,x_{10}\}\}\big\};$$

$$E^o = \big\{\{x_6,x_7\},\ \{x_6,x_8\},\ \{x_7,x_8\},\ \{x_9,x_{10}\},\ \{x_9,x_{11}\},\ \{x_9,x_{12}\},\ \{x_{10},x_{12}\},\ \{x_{11},x_{12}\}\big\}.$$

Этот граф представлен на рис. 3. Для лучшей читаемости рисунка на нем подписаны только номера вершин без указания буквы “x”.

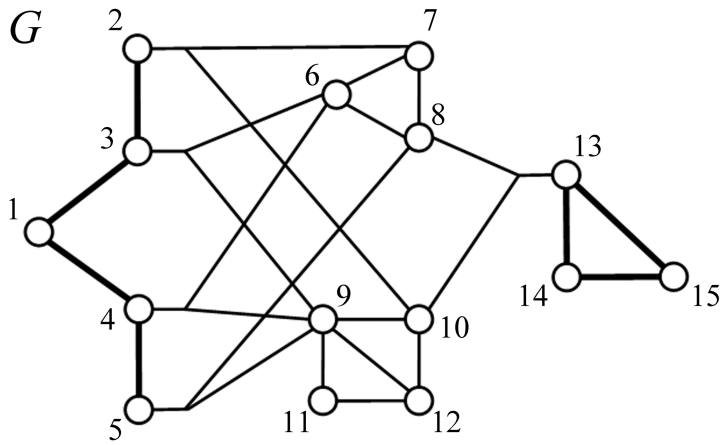


Fig. 3. Divisible graph of multiplicity 2

Рис. 3. Делимый граф кратности 2

Граф G является делимым. Части G_1 и G_2 этого графа показаны на рис. 4, связанные ребра всех кратных и мультиребер изображены пунктирными линиями.

Заметим, что граф перестанет быть делимым, если добавить в него обычное ребро между любой парой вершин из множеств $\{x_6,x_7,x_8\}$ и $\{x_9,x_{10},x_{11},x_{12}\}$.

Определим теперь путь в кратном графе.

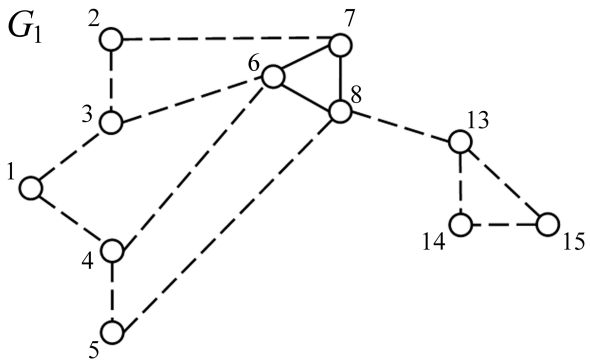


Fig. 4. Partition of a divisible graph

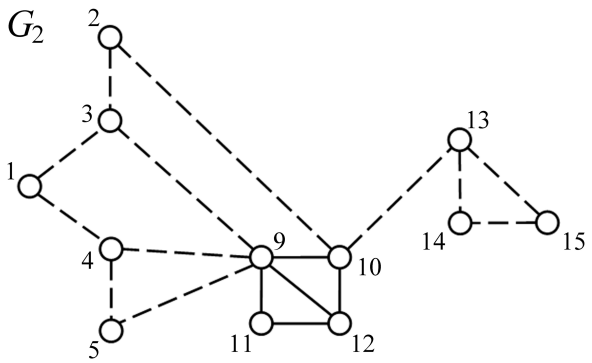


Рис. 4. Части делимого графа

Определение 5. *$S(x, y) = \cup_{i=1}^{k} S^i(x, y)$ является* кратным путем из вершины x в вершину y *в графе $G(V, E)$, если выполнены следующие условия:*

1. *$S^i(x, y) = \left(\{x, v_1^i\}, \{v_1^i, v_2^i\}, \ldots, \{v_{l_i-1}^i, v_{l_i}^i\}, \{v_{l_i}^i, y\}\right)$, где $l_i \geqslant 0$, – последовательность ребер, представляющая собой обычный (некратный) путь из x в y, где каждое ребро $\{a, b\}$ является либо обычным ребром в графе $G(V, E)$, либо i-ым связанным ребром кратного или мультиребра. Значения l_i и l_j $(i \neq j)$ не согласовываются и могут быть как равными, так и различными. Если в путь $S(x, y)$ не входит ни одного кратного или мультиребра, то $S^2(x, y) = S^3(x, y) = \ldots = S^k(x, y) = \varnothing$.*
2. *Любая обычная вершина может встретиться в $S^i(x, y)$ несколько раз, то есть $S^i(x, y)$ может содержать циклы.*
3. *Никакая кратная вершина не может встретиться в $S^i(x, y)$ дважды.*
4. *Любое обычное ребро может встречаться в $S^i(x, y)$ несколько раз, причем направления, в которых оно проходится в разных вхождениях, могут не совпадать.*
5. *Обычное ребро, входящее в $S^i(x, y)$, может также входить в любой $S^j(x, y)$, $j \neq i$.*
6. *Все пути $S^i(x, y)$ согласованы (одинаковы) на общей части. Это условие означает, что если связанное ребро какого-то кратного или мультиребра входит в некоторый путь $S^i(x, y)$, то остальные связанные ребра должны входить во все $S^j(x, y)$, $j \neq i$ (по одному связанному ребру в каждый $S^j(x, y)$). При этом порядок вхождения всех кратных и мультиребер во все $S^i(x, y)$ одинаков.*

   Фактически это значит, что если e_1 и e_2 – это два ребра пути $S(x, y)$, каждое из которых либо кратное, либо мультиребро, и в проекции $S^i(x, y)$ связанное ребро из e_1 проходится раньше связанного ребра из e_2, то во всех остальных проекциях $S^j(x, y)$ связанные ребра из e_2 могут проходиться только после связанных ребер из e_1.
7. *Если $S(x, y)$ содержит мультиребро $\{x_0, \{x_1, \ldots, x_k\}\}$, проходимое в направлении от общего конца, то он не может содержать никакого другого мультиребра $\{y_0, \{x_1, \ldots, x_k\}\}$, проходимого в том же направлении. Аналогичное условие должно выполняться и в случае движения к общему концу.*

Определение 6. *Кратный путь $S(x, y)$ является* кратным циклом, *если $x = y$ и $S(x, y) \neq \varnothing$.*

Пример 2. Проиллюстрируем определение кратного пути. Для этого рассмотрим граф, показанный на рис. 3, и построим в нем кратный путь $S(x_1, x_{15})$ из вершины x_1 в вершину x_{15}, состоящий из двух обычных путей:

$$S^1(x_1, x_{15}) = \left(\underline{\{x_1, x_4\}}, \underline{\{x_4, x_5\}}, \underline{\{x_5, x_8\}}, \underline{\underline{\{x_8, x_7\}}}, \underline{\{x_7, x_2\}},\right.$$
$$\left.\underline{\{x_2, x_3\}}, \underline{\{x_3, x_6\}}, \{x_6, x_7\}, \underline{\underline{\{x_7, x_8\}}}, \underline{\{x_8, x_{13}\}}, \underline{\{x_{13}, x_{15}\}}\right);$$

$$S^2(x_1, x_{15}) = \big(\{x_1, x_4\},\ \{x_4, x_5\},\ \{x_5, x_9\},\ \{x_9, x_{12}\},\ \{x_{12}, x_{10}\},\ \{x_{10}, x_2\},$$
$$\{x_2, x_3\},\ \{x_3, x_9\},\ \{x_9, x_{12}\},\ \{x_{12}, x_{10}\},\ \{x_{10}, x_{13}\},\ \{x_{13}, x_{15}\}\big).$$

Связанные ребра кратных и мультиребер отмечены подчеркиванием. Двойным подчеркиванием в пути $S^1(x_1, x_{15})$ отмечено обычное ребро $\{x_7, x_8\}$, которое в этом пути проходится дважды, но в противоположных направлениях. Соответственно, в пути $S^2(x_1, x_{15})$ двойным подчеркиванием отмечены обычные ребра $\{x_9, x_{12}\}$ и $\{x_{12}, x_{10}\}$, которые также проходятся дважды, но в одном и том же направлении. Таким образом, в пути $S^1(x_1, x_{15})$ содержится обычный цикл $(x_7, x_2, x_3, x_6, x_7)$, а в пути $S^2(x_1, x_{15})$ содержится обычный цикл $(x_9, x_{12}, x_{10}, x_2, x_3, x_9)$, однако кратный путь $S(x_1, x_{15})$ не содержит в себе кратных циклов, как и должно быть. Полученный кратный путь $S(x_1, x_{15})$ показан на рис. 5.

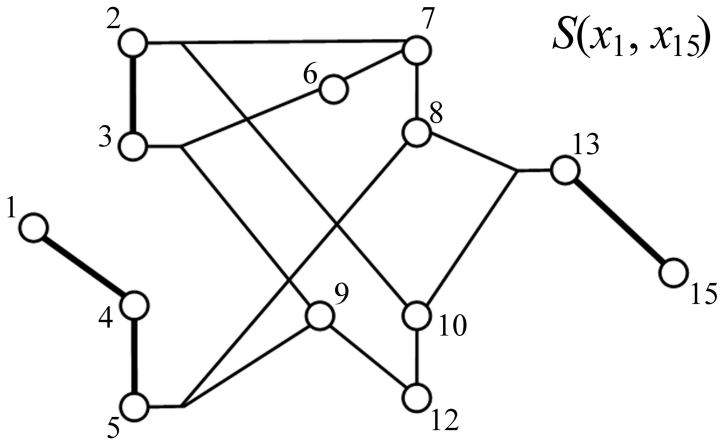


Fig. 5. Multiple path in the multiple graph

Рис. 5. Кратный путь в делимом графе

Отметим, что при замене в кратном пути $S(x_1, x_{15})$ последовательности ребер $\big(\{x_4, x_5\},\ \{x_5, \{x_8, x_9\}\},\ \{x_8, x_7\}\big)$ на последовательность $\big(\{x_4, \{x_6, x_9\}\},\ \{x_6, x_7\}\big)$ мы снова получим два обычных пути, согласованных на связанных ребрах. Однако объединение этих обычных путей не даст кратного пути, поскольку будет нарушено условие 7 из определения 5. Действительно, здесь будет два мультиребра $\{x_3, \{x_6, x_9\}\}$ и $\{x_4, \{x_6, x_9\}\}$ с одинаковым набором обычных вершин – концов мультиребра, и оба мультиребра будут проходиться в направлении от кратной вершины.

Определение 7. *Кратный граф $G(V, E)$ является* связным, *если одновременно выполнены два условия:*

1. *Кратный путь $S(x, y)$ существует для любых двух кратных вершин $x \in V$, $y \in V$.*
2. *Невозможно выделить такой подграф $G' \subset G$, который будет содержать только обычные ребра, и при этом подграфы G' и $G \setminus G'$ не будут соединены ни одним ребром (обычным ребром или связанным ребром мультиребра).*

В отличие от обычных графов, связность кратного графа не предполагает наличие кратных путей из каждой вершины в каждую. Фактически в связном кратном графе между каждой парой вершин должен существовать обычный (некратный) путь, использующий связанные ребра кратных и мультиребер несогласованно, а кратные пути обязательно должны существовать только для пар кратных вершин.

Для делимого кратного графа определение связности может быть переписано в более простой форме, что обусловлено структурой графа.

Определение 8. *Делимый кратный граф $G(V, E)$ является* связным, *если одновременно выполнены два условия:*

1. *Кратный путь $S(x, y)$ существует для любых двух кратных вершин $x \in V$, $y \in V$.*
2. *Каждая из частей G_i является связным (некратным) графом.*

Определение 9. *Целочисленная функция $l(e)$, определенная для всех ребер $e \in E$, является* длиной (весом) ребра *в кратном графе $G(V, E)$, если выполнено следующее:*

1. *$l(e) > 0$ для любого ребра e.*
2. *Если e является кратным или мультиребром, то $l(e_1) = l(e_2) = \ldots = l(e_k)$ и $l(e) = k \cdot l(e_1)$, где $e_1, \ldots, e_k$ – это связанные ребра данного ребра e.*

Тогда *длина кратного пути $S(x, y)$* будет определяться по формуле

$$l(S(x, y)) = \sum_{i=1}^{k} l(S^i(x, y)) = \sum_{i=1}^{k} \sum_{e_j \in S^i(x,y)} l(e_j),$$

при этом может оказаться $e_j = e_p$ $(j \neq p)$, то есть в сумме учитывается каждое повторное вхождение обычного ребра в $S^i(x, y)$.

Задача 1 (кратчайший кратный путь)**.** *В кратном графе $G(V, E)$ требуется найти* кратчайший путь из вершины x в вершину y, *то есть такой путь $S_{\min}(x, y)$, что для любого пути $S(x, y)$ выполнено*

$$l(S_{\min}(x, y)) \leqslant l(S(x, y)).$$

2. Алгоритм нахождения кратчайшего кратного пути в делимом графе

При построении алгоритма мы будем использовать множества достижимости по обычным и кратным ребрам (см. [1]). Напомним соответствующие определения.

Определение 10. Множеством достижимости по кратным ребрам *для некоторой кратной вершины x назовем множество R_x^k всех вершин y таких, что существует путь из x в y, проходящий только по кратным ребрам.*

Определение 11. Множеством достижимости по обычным ребрам *для некоторой обычной вершины x назовем множество R_x^o всех вершин y таких, что существует путь из x в y, проходящий только по обычным ребрам.*

Очевидно, что $x \in R_x^k$, $x \in R_x^o$. Если $y \in R_x^k$, то $R_y^k = R_x^k$. Если $y \in R_x^o$, то $R_y^o = R_x^o$.

Определение 12. *Множества достижимости R_x^k и R_y^k являются* смежными, *если для произвольных вершин $a \in R_x^k$, $b \in R_y^k$ существует соединяющий их кратный путь $S(a, b)$.*

Сформулируем теперь алгоритм решения задачи 1 для делимого кратного графа. На отдельных шагах этого алгоритма для нахождения минимальных участков пути, содержащих только обычные ребра, мы будем использовать известный алгоритм Дейкстры (см. [12]).

Пусть имеется взвешенный делимый кратный граф $G(V, E)$ кратности k. Требуется найти кратчайший кратный путь $S_{\min}(x, y)$ между двумя выбранными вершинами x и y. Через e_a^m будем обозначать мультиребро, инцидентное кратной вершине a. В алгоритме мы будем использовать следующие структуры данных:

- множества достижимости R_a^k и R_b^o;
- множества индексов I_a, ассоциированные с каждым мультиребром e_a^m;

- $G^{ord}(V^{ord}, E^{ord})$ – обычный граф, минимальному пути $S^{ord}_{\min}(x, y)$ в котором будет соответствовать кратчайший кратный путь $S_{\min}(x, y)$ той же длины в исходном графе;
- $S_{mo}(a, b)$ – кратный путь между двумя кратными вершинами, состоящий из мультиребер e^m_a и e^m_b, а также из минимальных обычных путей между соответствующими парами обычных вершин – концов этих мультиребер.

Алгоритм 1 (кратчайший кратный путь в делимом графе).

1. Найдем все множества достижимости по кратным и обычным ребрам R^k_a и R^o_b с помощью полиномиальных алгоритмов 1, 2 из статьи [1]. Пронумеруем все найденные множества R^o_b в произвольном порядке от 1 до t и обозначим их через $R_1, \ldots, R_t$.

2. Проверим выполнение критерия существования кратного пути между вершинами x и y (теорема 3 из статьи [1]). Напомним, что это можно сделать за полиномиальное время, проверяя смежность соответствующих множеств достижимости по кратным ребрам (алгоритм 3 той же статьи). Если критерий не выполнен, выходим из алгоритма.

3. Если x и y – обычные вершины, то $y \in R^o_x$ и кратчайший кратный путь $S_{\min}(x, y)$ проходит только по обычным ребрам, инцидентным вершинам из R^o_x (следует из теоремы 3 из статьи [1]). Находим этот путь с помощью алгоритма Дейкстры и выходим из алгоритма 1. Иначе переходим на шаг 4.

4. Для каждого мультиребра $e^m_a = \{a, \{a_1, \ldots, a_k\}\}$ сформируем множество индексов $I_a = \{i_1, \ldots, i_k\}$ таким образом, что каждый i_p равен номеру множества достижимости, в которое попадает a_p: $a_p \in R_{i_p}$. Поскольку граф G – делимый, $i_p \neq i_q$, если $p \neq q$. Далее для удобства будем считать, что вершины $a_1, \ldots, a_k$ мультиребра e^m_a пронумерованы в порядке возрастания значений i_p (если это не так, их можно быстро перенумеровать).

5. Будем строить обычный граф $G^{ord}(V^{ord}, E^{ord})$ следующим образом.

5.1. Для каждой кратной вершины $v \in V$ создадим обычную вершину v и поместим ее в V^{ord}.

5.2. Для каждого кратного ребра $\{u, v\} \in E$ создадим соответствующее обычное ребро $\{u, v\}$ той же длины и поместим его в E^{ord}.

5.3. Для каждой кратной вершины a, инцидентной мультиребру e^m_a, создадим дополнительную обычную вершину a' и поместим ее в V^{ord}. Создадим также обычное ребро $\{a, a'\}$ длины 1 и поместим его в E^{ord}.

5.4. Рассмотрим все пары мультиребер из E^m. Для каждой такой пары $\{e^m_a, e^m_b\}$ ($e^m_a = \{a, \{a_1, \ldots, a_k\}\}$, $e^m_b = \{b, \{b_1, \ldots, b_k\}\}$) проверим равенство $I_a = I_b$. Если оно выполнено, с помощью алгоритма Дейкстры найдем кратчайшие обычные пути $S_{\min}(a_p, b_p)$ ($p \in \overline{1, k}$), каждый из которых будет проходить только по вершинам из соответствующего множества R_{i_p} (какие-то из этих путей могут иметь длину 0, если $a_p = b_p$). Сформируем и запомним кратный путь

$$S_{mo}(a, b) = e^m_a \cup S_{\min}(a_1, b_1) \cup \ldots \cup S_{\min}(a_k, b_k) \cup e^m_b,$$

который будет кратчайшим кратным путем без кратных ребер между вершинами a и b. Добавим в E^{ord} обычное ребро $\{a', b'\}$ длины $l(\{a', b'\}) = l(S_{mo}(a, b)) - 2$.

6. Найдем кратчайший путь $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ с помощью алгоритма Дейкстры.

7. Построим теперь искомый кратчайший кратный путь $S_{\min}(x, y)$ в графе $G(V, E)$. Для этого будем последовательно просматривать ребра пути $S^{ord}_{\min}(x, y)$ и в зависимости от типа ребер выполнять одно из следующих действий.

7.1. Если в путь $S^{ord}_{\min}(x, y)$ входит ребро $\{u, v\}$, где обе вершины без штриха, то включаем в кратный путь $S_{\min}(x, y)$ соответствующее кратное ребро $\{u, v\}$.

7.2. Если в путь $S^{ord}_{\min}(x, y)$ входит цепь вида $(\{a, a'\}, \{a', b'\}, \{b', b\})$, то включаем в кратный путь $S_{\min}(x, y)$ найденный на шаге 5.4 кратный путь $S_{mo}(a, b)$ (если на шаге 5.4 был найден путь $S_{mo}(b, a)$, то путь $S_{mo}(a, b)$ получается из него простым обращением).

Алгоритм 1, очевидно, является полиномиальным, при этом наибольшую трудоемкость имеет шаг 5.4. В худшем случае на этом шаге придется $\frac{k}{2} \cdot (|E^m|^2 - |E^m|)$ раз применить полиномиальный алгоритм Дейкстры (поиск k кратчайших путей для каждой пары мультиребер). При этом, если нужно в одном графе найти кратчайшие кратные пути между различными парами вершин, для этого будет использоваться один и тот же граф $G^{ord}(V^{ord}, E^{ord})$, повторно выполнять шаг 5 не требуется.

Отметим также, что поиск множеств R_a^m на шаге 1 и проверка критерия на шаге 2 позволяют существенно сократить вычисления в том случае, если пути не существует. Однако, если известно, что x и y – кратные вершины, а граф связен, то данную проверку можно не проводить.

Шаг 3 обусловлен свойствами делимых графов. Если граф не будет делимым, этот шаг нужно будет пропустить, и тогда потребуются дополнительные действия на шагах 5 и 7.

Наконец, заметим, что вершины a' создаются на шаге 5.3 для того, чтобы по ребрам пути $S^{ord}_{\min}(x, y)$ можно было однозначно определить, какой переход надо осуществить в исходном графе G между двумя кратными вершинами a и b: по единственному кратному ребру $\{a, b\}$ или же по кратному пути $S_{mo}(a, b)$, содержащему мультиребра e_a^m и e_b^m.

Пример 3. Продемонстрируем работу алгоритма 1. Для этого рассмотрим граф кратности 2, показанный на рис. 6. Как и ранее, на рисунке подписаны только номера вершин без буквы “x”. На ребрах шрифтом Courier New отмечены длины.

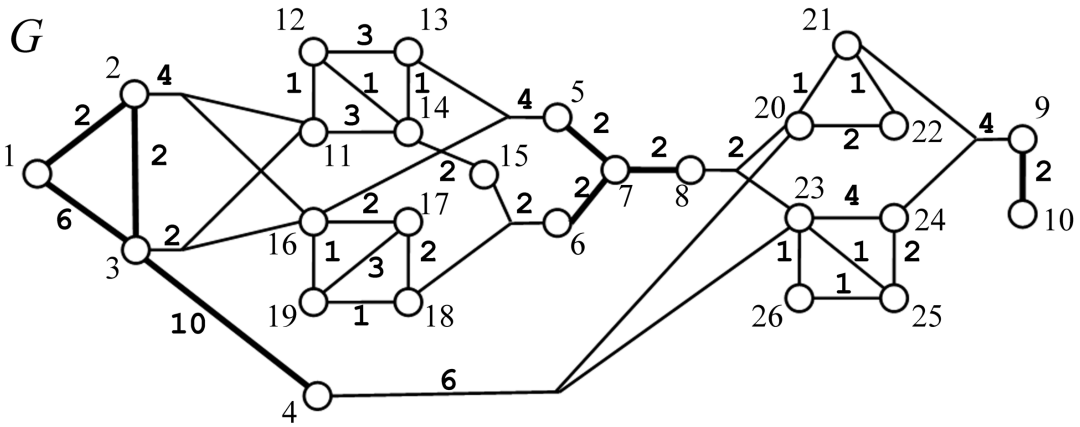


Fig. 6. Weighted divisible graph of multiplicity 2

Рис. 6. Взвешенный делимый граф кратности 2

Будем искать кратчайший кратный путь $S_{min}(x_1, x_{10})$. Вершины x_1 и x_{10} кратные и граф $G(V, E)$ связен, поэтому такой путь существует. Сначала определим и пронумеруем множества достижимости по обычным ребрам:

$$R_1 = R^o_{x_{11}} = \{x_{11}, x_{12}, x_{13}, x_{14}, x_{15}\}, \quad R_2 = R^o_{x_{16}} = \{x_{16}, x_{17}, x_{18}, x_{19}\},$$

$$R_3 = R^o_{x_{20}} = \{x_{20}, x_{21}, x_{22}\}, \quad R_4 = R^o_{x_{23}} = \{x_{23}, x_{24}, x_{25}, x_{26}\}.$$

Множества индексов I_a для мультиребер тогда определяются так:

$$I_{x_2} = \{1, 2\}, \quad I_{x_3} = \{1, 2\}, \quad I_{x_4} = \{3, 4\}, \quad I_{x_5} = \{1, 2\}, \quad I_{x_6} = \{1, 2\}, \quad I_{x_8} = \{3, 4\}, \quad I_{x_9} = \{3, 4\}.$$

Построим теперь кратные пути $S_{mo}(a, b)$ для подходящих пар мультиребер:

$$S_{mo}(x_2, x_3) = e^m_{x_2} \cup \varnothing \cup \varnothing \cup e^m_{x_3}, \quad l\,(S_{mo}(x_2, x_3)) = 6;$$

$$S_{mo}(x_2, x_5) = e^m_{x_2} \cup \big(\{x_{11}, x_{12}\},\ \{x_{12}, x_{14}\},\ \{x_{14}, x_{13}\}\big) \cup \varnothing \cup e^m_{x_5}, \quad l\,(S_{mo}(x_2, x_5)) = 11;$$

$$S_{mo}(x_2, x_6) = e^m_{x_2} \cup \left(\{x_{11}, x_{12}\},\ \{x_{12}, x_{14}\},\ \{x_{14}, x_{15}\}\right) \cup \left(\{x_{16}, x_{19}\},\ \{x_{19}, x_{18}\}\right) \cup e^m_{x_6},\quad l\left(S_{mo}(x_2, x_6)\right) = 12;$$

$$S_{mo}(x_3, x_5) = e^m_{x_3} \cup \left(\{x_{11}, x_{12}\},\ \{x_{12}, x_{14}\},\ \{x_{14}, x_{13}\}\right) \cup \varnothing \cup e^m_{x_5},\quad l\left(S_{mo}(x_3, x_5)\right) = 9;$$

$$S_{mo}(x_3, x_6) = e^m_{x_3} \cup \left(\{x_{11}, x_{12}\},\ \{x_{12}, x_{14}\},\ \{x_{14}, x_{15}\}\right) \cup \left(\{x_{16}, x_{19}\},\ \{x_{19}, x_{18}\}\right) \cup e^m_{x_6},\quad l\left(S_{mo}(x_3, x_6)\right) = 10;$$

$$S_{mo}(x_5, x_6) = e^m_{x_5} \cup \left(\{x_{13}, x_{14}\},\ \{x_{14}, x_{15}\}\right) \cup \left(\{x_{16}, x_{19}\},\ \{x_{19}, x_{18}\}\right) \cup e^m_{x_6},\quad l\left(S_{mo}(x_5, x_6)\right) = 11;$$

$$S_{mo}(x_4, x_8) = e^m_{x_4} \cup \varnothing \cup \varnothing \cup e^m_{x_8},\quad l\left(S_{mo}(x_4, x_8)\right) = 8;$$

$$S_{mo}(x_4, x_9) = e^m_{x_4} \cup \left(\{x_{20}, x_{21}\}\right) \cup \left(\{x_{23}, x_{25}\},\ \{x_{25}, x_{24}\}\right) \cup e^m_{x_9},\quad l\left(S_{mo}(x_4, x_9)\right) = 14;$$

$$S_{mo}(x_8, x_9) = e^m_{x_8} \cup \left(\{x_{20}, x_{21}\}\right) \cup \left(\{x_{23}, x_{25}\},\ \{x_{25}, x_{24}\}\right) \cup e^m_{x_9},\quad l\left(S_{mo}(x_8, x_9)\right) = 10.$$

Применив шаг 5 алгоритма 1, получим обычный граф $G^{ord}(V^{ord}, E^{ord})$, показанный на рис. 7. Вершины кратчайшего пути $S^{ord}_{\min}(x_1, x_{10})$ (шаг 6) показаны серым. Его длина $l\left(S^{ord}_{\min}(x_1, x_{10})\right) = 29$.

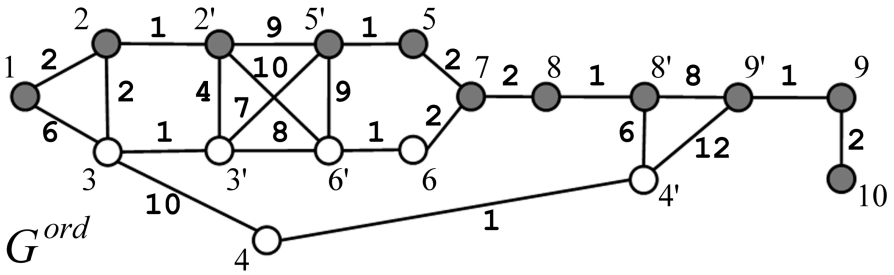


Fig. 7. Ordinary graph G^{ord} and the shortest path

Рис. 7. Обычный граф G^{ord} и кратчайший путь

Применяя шаг 7, получим искомый кратчайший кратный путь $S_{min}(x_1, x_{10})$ (рис. 8):

$$S_{min}(x_1, x_{10}) = \left(\{x_1, x_2\},\ S_{mo}(x_2, x_5),\ \{x_5, x_7\},\ \{x_7, x_8\},\ S_{mo}(x_8, x_9),\ \{x_9, x_{10}\}\right),$$

$$l\left(S_{\min}(x_1, x_{10})\right) = l\left(S^{ord}_{\min}(x_1, x_{10})\right) = 29.$$

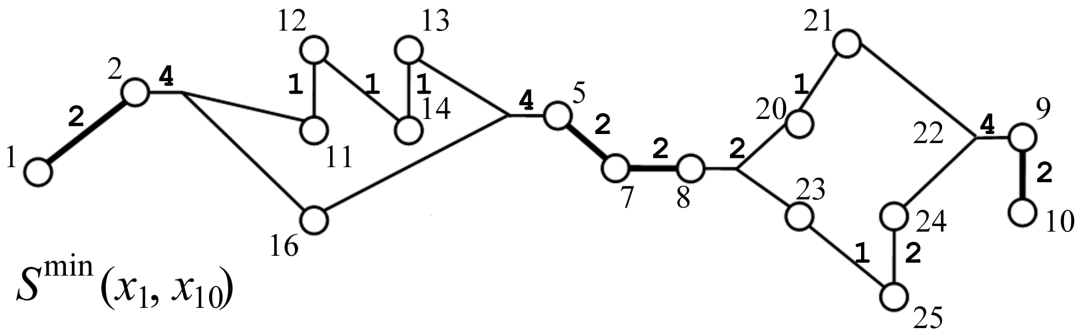


Fig. 8. The shortest path in the multiple graph G

Рис. 8. Кратчайший путь в кратном графе G

Отметим, что, заменив в пути $S_{min}(x_1, x_{10})$ мультиребро e^m_2 на последовательность из кратного ребра $\{x_2, x_3\}$ и мультиребра e^m_3, мы снова получим кратчайший кратный путь от x_1 до x_{10}.

3. Обоснование полиномиальности задачи для делимого графа

Теорема 1. *Пусть в делимом кратном графе $G(V, E)$ существует путь между вершинами x и y. Тогда кратчайший кратный путь $S_{\min}(x, y)$ между этими вершинами может быть найден за полиномиальное время с помощью алгоритма 1.*

Доказательство. Если x и y – обычные вершины, утверждение теоремы очевидно.

Пусть x и y – кратные вершины. Применим полиномиальный алгоритм 1 и получим кратчайший путь $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$. Покажем, что путь $S^{ord}_{\min}(x, y)$ всегда будет соответствовать кратчайшему кратному пути $S_{\min}(x, y)$ в исходном графе. Справедливы следующие утверждения.

1. Любому кратному пути $S(x, y)$ соответствует обычный путь $S^{ord}(x, y)$, если в кратном пути все части, не содержащие кратных ребер, можно представить в виде объединения путей $S_{mo}(a, b)$ (следует из правил построения графа $G^{ord}(V^{ord}, E^{ord})$).
2. В любом минимальном кратном пути все части, не содержащие кратных ребер, можно представить в виде объединения путей $S_{mo}(a, b)$ (иначе длину пути можно уменьшить, заменив цепи из обычных ребер на более короткие).
3. Следовательно, любому минимальному кратному пути в графе $G(V, E)$ соответствует обычный путь в графе $G^{ord}(V^{ord}, E^{ord})$.

Однако обратное неверно: не всякому обычному пути в графе $G^{ord}(V^{ord}, E^{ord})$ соответствует кратный путь в графе $G(V, E)$. Такими путями являются те, в которых встречается два и более ребра вида $\{a', b'\}$ подряд (невозможно установить соответствие с обычными и мультиребрами в графе $G(V, E)$), либо те, в которых два и более ребра вида $\{a, a'\}$ приводят к нарушению условия 7 из определения 5 при переходе к кратному пути (два и более мультиребра с одинаковым набором обычных вершин проходятся в одинаковом направлении).

Покажем, что для любого минимального пути $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ ни одна из отмеченных двух аномалий не возникает.

Сначала предположим, что в путь $S^{ord}_{\min}(x, y)$ входит цепь $P^{ord} = (\{a', b'\}, \{b', c'\})$.

Вершины a, a', b, b', c, c' графа $S^{ord}_{\min}(x, y)$ порождены мультиребрами $e^m_a = \{a, \{a_1, \dots, a_k\}\}$, $e^m_b = \{b, \{b_1, \dots, b_k\}\}$, $e^m_c = \{c, \{c_1, \dots, c_k\}\}$ кратного графа $G(V, E)$ в результате действий шага 5 алгоритма 1. Поэтому длина цепи P^{ord} может быть выражена так:

$$l\left(P^{ord}\right) = l\left(S_{mo}(a, b)\right) - 2 + l\left(S_{mo}(b, c)\right) - 2,$$

$$l\left(P^{ord}\right) = l(e^m_a) + l\left(S_{\min}(a_1, b_1)\right) + \dots + l\left(S_{\min}(a_k, b_k)\right) + l(e^m_b) - 2 + l(e^m_b) + l\left(S_{\min}(b_1, c_1)\right) + \dots + l\left(S_{\min}(b_k, c_k)\right) + l(e^m_c) - 2,$$

$$l\left(P^{ord}\right) = l(e^m_a) + l(e^m_c) + l\left(S'(a_1, c_1)\right) + \dots + l\left(S'(a_k, c_k)\right) + 2l(e^m_b) - 4,$$

где $S'(a_i, c_i)$ $(i \in \overline{1, k})$ – это какой-то, не обязательно минимальный, путь между вершинами a_i и c_i в графе $G(V, E)$, проходящий только по обычным ребрам.

При этом ребра вида $\{a', b'\}$ образованы для всех мультиребер с совпадающими множествами индексов $I_a = I_b$. Это значит, что если в графе $G^{ord}(V^{ord}, E^{ord})$ есть ребра $\{a', b'\}$ и $\{b', c'\}$, то в нем обязательно есть и ребро $\{a', c'\}$. Его длина выражается так:

$$l\left(\{a', c'\}\right) = l\left(S_{mo}(a, c)\right) - 2 = l(e^m_a) + l(e^m_c) + l\left(S_{\min}(a_1, c_1)\right) + \dots + l\left(S_{\min}(a_k, c_k)\right) - 2.$$

Заметим, что $l\left(S'(a_i, c_i)\right) \geqslant l\left(S_{\min}(a_i, c_i)\right)$ $(i \in \overline{1, k})$. Кроме того, $l(e^m_b) \geqslant 2$ и $2l(e^m_b) - 2 > 0$. Отсюда получаем, что

$$l\left(P^{ord}\right) > l\left(\{a', c'\}\right),$$

а значит, замена цепи $P^{ord} = (\{a', b'\}, \{b', c'\})$ на ребро $\{a', c'\}$ в пути $S^{ord}_{\min}(x, y)$ приведет к пути меньшей длины. Следовательно, $S^{ord}_{\min}(x, y)$ не является кратчайшим. Полученное противоречие доказывает отсутствие аномалии первого вида в кратчайшем пути $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$.

Теперь предположим, что в минимальный путь $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ вошли два ребра $\{a, a'\}$, $\{b, b'\}$, соответствующие мультиребрам $e^m_a = \{a, \{x_1, \dots, x_k\}\}$, $e^m_b = \{b, \{x_1, \dots, x_k\}\}$ с одинаковым набором обычных вершин, и эти два ребра проходятся в одном направлении. Для определенности будем считать, что это направление от кратной вершины (обратная ситуация обосновывается аналогично).

Наличие в пути $S^{ord}_{\min}(x, y)$ ребер $\{a, a'\}, \{b, b'\}$, проходимых в направлении к вершине со штрихом, влечет обязательное наличие в этом пути цепей $(\{a, a'\},\ \{a', c'\},\ \{c', c\})$ и $(\{b, b'\},\ \{b', d'\},\ \{d', d\})$, а также цепи $S^{ord}(c, b)$, соединяющей вершины c и b. Таким образом, цепь

$$P^{ord} = \left(\{a, a'\},\ \{a', c'\},\ \{c', c\}\right) \cup S^{ord}(c, b) \cup \left(\{b, b'\},\ \{b', d'\},\ \{d', d\}\right) \subseteq S^{ord}_{\min}(x, y).$$

Заметим, что вершины c, c', d, d' в графе $G^{ord}(V^{ord}, E^{ord})$ порождены соответствующими мультиребрами e^m_c и e^m_d исходного графа. Наличие ребер $\{a', c'\}$ и $\{b', d'\}$ в E^{ord} говорит о том, что $I_c = I_d = I_a = I_b$. Следовательно, вершины a', b', c', d' попарно смежны и цепь

$$P^{ord}_0 = \left(\{a, a'\},\ \{a', d'\},\ \{d', d\}\right) \subset G^{ord}(V^{ord}, E^{ord}).$$

Оценим длину цепей P^{ord} и P^{ord}_0. Будем обозначать через L_{ac} суммарную длину кратчайших обычных путей в кратном графе $G(V, E)$ между обычными вершинами – концами мультиребер e^m_a и e^m_c (L_{ad} и L_{bd} определяются аналогичным образом). Мультиребра e^m_a и e^m_b имеют одинаковый набор обычных вершин, поэтому $L_{ad} = L_{bd}$. Тогда длины цепей P^{ord} и P^{ord}_0 выражаются так:

$$l\left(P^{ord}\right) = l\left(S_{mo}(a, c)\right) + l\left(S^{ord}(c, b)\right) + l\left(S_{mo}(b, d)\right) = l(e^m_a) + L_{ac} + l(e^m_c) + l\left(S^{ord}(c, b)\right) + l(e^m_b) + L_{bd} + l(e^m_d),$$

$$l\left(P^{ord}_0\right) = l\left(S_{mo}(a, d)\right) = l(e^m_a) + L_{ad} + l(e^m_d) = l(e^m_a) + L_{bd} + l(e^m_d).$$

Таким образом,

$$l\left(P^{ord}\right) = l\left(P^{ord}_0\right) + L_{ac} + l(e^m_c) + l\left(S^{ord}(c, b)\right) + l(e^m_b) > l\left(P^{ord}_0\right).$$

Заменяя в пути $S^{ord}_{\min}(x, y)$ цепь P^{ord} на цепь P^{ord}_0, мы получим путь меньшей длины. Соответственно, путь $S^{ord}_{\min}(x, y)$ не является кратчайшим. Полученное противоречие доказывает отсутствие аномалии второго вида в кратчайшем пути $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$.

Следовательно, любому кратчайшему пути $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ соответствует кратный путь в исходном делимом графе $G(V, E)$, для которого соблюдены все условия из определения 5 (условия 1–6 обеспечиваются правилами построения графа $G^{ord}(V^{ord}, E^{ord})$, а выполнение условия 7 обосновано выше). Из свойств, определенных в начале доказательства, следует, что указанный кратный путь будет кратчайшим.

Теорема доказана.

4. Модификация алгоритма для произвольного кратного графа

Алгоритм 1 сформулирован для делимых кратных графов, однако он может быть легко трансформирован для случая произвольного кратного графа.

Пусть имеется произвольный взвешенный кратный граф $G(V, E)$ кратности k. Требуется найти кратчайший кратный путь $S_{\min}(x, y)$ между двумя выбранными вершинами x и y.

Алгоритм 2 (кратчайший путь в произвольном кратном графе).

1. Найдем все множества достижимости по кратным и обычным ребрам R^k_a и R^o_b с помощью полиномиальных алгоритмов 1, 2 из статьи [1]. Пронумеруем все найденные множества R^o_b в произвольном порядке от 1 до t и обозначим их через $R_1, \dots, R_t$.

2. Проверим выполнение критерия существования кратного пути между вершинами x и y (теорема 3 и полиномиальный алгоритм 3 из статьи [1]). Если критерий не выполнен, выходим из алгоритма.

3. Для каждого мультиребра $e_a^m = \{a, \{a_1, \dots, a_k\}\}$ сформируем мультимножество индексов $I_a = \{i_1, \dots, i_k\}$ таким образом, что каждый i_p равен номеру множества достижимости, в которое попадает a_p: $a_p \in R_{i_p}$. Если граф G не является делимым, хотя бы для одного мультиребра обязательно найдутся индексы $p \neq q$ такие, что $i_p = i_q$. Далее для удобства будем считать, что вершины $a_1, \dots, a_k$ мультиребра e_a^m пронумерованы в порядке возрастания значений i_p (если это не так, их можно быстро перенумеровать).

4. Будем строить обычный граф $G^{ord}(V^{ord}, E^{ord})$ следующим образом.

4.1. Для каждой кратной вершины $v \in V$ создадим обычную вершину v и поместим ее в V^{ord}.

4.2. Для каждого кратного ребра $\{u, v\} \in E$ создадим соответствующее обычное ребро $\{u, v\}$ той же длины и поместим его в E^{ord}.

4.3. Для каждой кратной вершины a, инцидентной мультиребру e_a^m, создадим дополнительную обычную вершину a' и поместим ее в V^{ord}. Создадим также обычное ребро $\{a, a'\}$ длины 1 и поместим его в E^{ord}.

4.4. Рассмотрим все пары мультиребер из E^m. Для каждой такой пары $\{e_a^m, e_b^m\}$ ($e_a^m = \{a, \{a_1, \dots, a_k\}\}$, $e_b^m = \{b, \{b_1, \dots, b_k\}\}$) проверим равенство $I_a = I_b$. Если оно выполнено, будем поочередно рассматривать все подмножества I_a^j совпадающих индексов из I_a.

Если подмножество I_a^j содержит только один элемент i_p ($I_a^j = \{i_p\}$), с помощью алгоритма Дейкстры найдем кратчайший обычный путь $S_{\min}(a_p, b_p)$, проходящий только по вершинам из R_{i_p}. Обозначим $c_p = b_p$.

Если подмножество I_a^j содержит q одинаковых элементов ($I_a^j = \{i_p, i_{p+1}, \dots, i_{p+q-1}\}$, $i_r = i_s$ для всех $r \in \overline{p, p+q-1}$, $s \in \overline{p, p+q-1}$), найдем с помощью алгоритма Дейкстры q^2 кратчайших обычных путей $S_{\min}(a_r, b_s)$ ($r \in \overline{p, p+q-1}$, $s \in \overline{p, p+q-1}$), проходящих только по вершинам из R_{i_p}. Выберем среди них q путей, попарно непересекающихся в начальных и конечных вершинах и имеющих при этом минимальную суммарную длину (соответствует выбору одного из $q!$ множеств путей). Упорядочим выбранные пути по возрастанию номера начальной вершины, конечные вершины обозначим через c_r ($r \in \overline{p, p+q-1}$).

Просмотрев все подмножества, сформируем и запомним кратный путь

$$S_{mo}(a, b) = e_a^m \cup S_{\min}(a_1, c_1) \cup \dots \cup S_{\min}(a_k, c_k) \cup e_b^m,$$

который будет кратчайшим кратным путем без кратных ребер между вершинами a и b. Добавим в E^{ord} обычное ребро $\{a', b'\}$ длины $l(\{a', b'\}) = l(S_{mo}(a, b)) - 2$.

4.5. Если x – обычная вершина, скопируем ее в V^{ord}. Далее рассмотрим все мультиребра $e_a^m = \{a, \{a_1, \dots, a_k\}\}$ с $I_a = \{p, \dots, p\}$, где $R_p = R_x^o$. Для каждого такого мультиребра с помощью алгоритма Дейкстры найдем кратчайшие пути $S_{\min}(x, a_r)$ ($r \in \overline{1, k}$), проходящие только по вершинам из R_x^o. сформируем и запомним кратный путь

$$S_{mo}(x, a) = S_{\min}(x, a_1) \cup \dots \cup S_{\min}(x, a_k) \cup e_a^m,$$

который будет кратчайшим кратным путем без кратных ребер между вершинами x и a. Добавим в E^{ord} обычное ребро $\{x, a'\}$ длины $l(\{x, a'\}) = l(S_{mo}(x, a)) - 2$.

4.6. Если y – обычная вершина, скопируем ее в V^{ord}. Далее рассмотрим все мультиребра $e_a^m = \{a, \{a_1, \dots, a_k\}\}$ с $I_a = \{p, \dots, p\}$, где $R_p = R_y^o$. Для каждого такого мультиребра с помощью алгоритма Дейкстры найдем кратчайшие пути $S_{\min}(a_r, y)$ ($r \in \overline{1, k}$), проходящие только по вершинам из R_y^o. сформируем и запомним кратный путь

$$S_{mo}(a, y) = e_a^m \cup S_{\min}(a_1, y) \cup \dots \cup S_{\min}(a_k, y),$$

который будет кратчайшим кратным путем без кратных ребер между вершинами a и y. Добавим в E^{ord} обычное ребро $\{a', y\}$ длины $l(\{a', y\}) = l(S_{mo}(a, y)) - 2$.

4.7. Если обе вершины x и y обычные и $y \in R^o_x$, с помощью алгоритма Дейкстры найдем кратчайший путь $S'_{\min}(x, y)$, проходящий только по вершинам из R^o_x, и запомним его. Добавим в E^{ord} обычное ребро $\{x, y\}$ длины $l(S'_{\min}(x, y))$.

5. Найдем кратчайший путь $S^{ord}_{\min}(x, y)$ в графе $G^{ord}(V^{ord}, E^{ord})$ с помощью алгоритма Дейкстры.

6. Построим теперь искомый кратчайший кратный путь $S_{\min}(x, y)$ в графе $G(V, E)$. Для этого будем последовательно просматривать ребра пути $S^{ord}_{\min}(x, y)$ и в зависимости от типа ребер выполнять одно из следующих действий.

6.1. Если в путь $S^{ord}_{\min}(x, y)$ входит ребро $\{u, v\}$, где обе вершины без штриха, то включаем в кратный путь $S_{\min}(x, y)$ соответствующее кратное ребро $\{u, v\}$.

6.2. Если в путь $S^{ord}_{\min}(x, y)$ входит цепь вида $(\{a, a'\}, \{a', b'\}, \{b', b\})$, то включаем в кратный путь $S_{\min}(x, y)$ найденный на шаге 4.4 кратный путь $S_{mo}(a, b)$ (если на шаге 4.4 был найден путь $S_{mo}(b, a)$, то путь $S_{mo}(a, b)$ получается из него простым обращением).

6.3. Если в путь входит цепь вида $(\{x, a'\}, \{a', a\})$, то включаем в кратный путь $S_{\min}(x, y)$ найденный на шаге 4.5 кратный путь $S_{mo}(x, a)$.

6.4. Если в путь входит цепь вида $(\{a, a'\}, \{a', y\})$, то включаем в кратный путь $S_{\min}(x, y)$ найденный на шаге 4.6 кратный путь $S_{mo}(a, y)$.

6.5. Если путь $S^{ord}_{\min}(x, y)$ состоит из единственного ребра $\{x, y\}$ и x, y – обычные вершины в $G(V, E)$, то $S_{\min}(x, y) = S'_{\min}(x, y)$ ($S'_{\min}(x, y)$ найден на шаге 4.7).

Применимость алгоритма 2 для задачи 1 в случае произвольного кратного графа обосновывается так же, как в теореме 1 обосновывалась применимость алгоритма 1 для задачи 1 в случае делимого кратного графа. В доказательстве нужно сделать лишь незначительные изменения, учитывающие ситуации, когда x или y – обычная вершина.

Отметим, что в большинстве случаев алгоритм 2 будет достаточно быстрым, однако в общем случае он экспоненциален по параметру k (кратность графа). Действительно, на шаге 4.4 для каждого подмножества I^j_a происходит перебор $q!$ вариантов выбора путей, где q может принимать значение от 1 до k. В худшем случае придется перебрать $\frac{k!}{2} \cdot (|E^m|^2 - |E^m|)$ вариантов. Этот случай реализуется, если все обычные вершины находятся в одном множестве достижимости R_1.

Однако для графов небольшой кратности алгоритм 2 будет выполняться за приемлемое время даже в указанном худшем случае. Кроме того, в случае когда нужно найти несколько кратчайших путей в одном и том же графе, шаг 4.4 повторно выполнять не нужно: если ищутся пути между различными парами кратных вершин, можно для этого использовать один и тот же граф $G^{ord}(V^{ord}, E^{ord})$; если же начало или конец очередного пути – обычная вершина, потребуется лишь небольшая перестройка графа, выполняемая за полиномиальное время (добавление вершин с помощью шагов 4.5–4.7 алгоритма либо же их удаление, если обычные вершины были концами предыдущего найденного пути).

References

[1] A. V. Smirnov, “The Shortest Path Problem for a Multiple Graph”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 625–633, 2018. DOI: 10.3103/S0146411618070234.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd. The MIT Press, McGraw-Hill Book Company, 2009.

[3] C. Berge, *Graphs and Hypergraphs*. North-Holland Publishing Company, 1973.

[4] A. Basu and R. W. Blanning, “Metagraphs in workflow support systems”, *Decision Support Systems*, vol. 25, no. 3, pp. 199–208, 1999. DOI: 10.1016/S0167-9236(99)00006-8.

[5] A. Basu and R. W. Blanning, *Metagraphs and Their Applications*, ser. Integrated Series in Information Systems. Springer US, 2007, vol. 15.

[6] V. S. Rublev and A. V. Smirnov, "Flows in Multiple Networks", *Yaroslavsky Pedagogichesky Vestnik*, vol. 3, no. 2, pp. 60–68, 2011.

[7] A. V. Smirnov, "The Problem of Finding the Maximum Multiple Flow in the Divisible Network and its Special Cases", *Automatic Control and Computer Sciences*, vol. 50, no. 7, pp. 527–535, 2016. DOI: 10.3103/S0146411616070191.

[8] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.

[9] V. S. Roublev and A. V. Smirnov, "The Problem of Integer-Valued Balancing of a Three-Dimensional Matrix and Algorithms of Its Solution", *Modeling and Analysis of Information Systems*, vol. 17, no. 2, pp. 72–98, 2010.

[10] A. V. Smirnov, "Network Model for the Problem of Integer Balancing of a Four-Dimensional Matrix", *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 558–566, 2017. DOI: 10.3103/S0146411617070185.

[11] A. V. Smirnov, "Spanning tree of a multiple graph", *Journal of Combinatorial Optimization*, vol. 43, no. 4, pp. 850–869, 2022. DOI: 10.1007/s10878-021-00810-5.

[12] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. DOI: 10.1007/BF01386390.