

Автоматизация проверки семантической составляющей текстовых ответов обучающихся в цифровой образовательной платформе

А.Г. Леонов^{1, 2, 3, 4}, Н.С. Мартынов², К.А. Машенко^{1, 2, 4}✉,
А.А. Холькина⁵, А.В. Шляхов²

¹ МГУ им. М.В. Ломоносова, г. Москва, 119991, Россия

² ФГУ ФНЦ НИИСИ РАН, г. Москва, 117218, Россия

³ Институт детства Московского педагогического государственного университета (МПГУ), г. Москва, 119435, Россия

⁴ Институт информационных систем Государственного университета управления, г. Москва, 109542, Россия

⁵ Национальный исследовательский технологический университет «МИСиС», г. Москва, 119049, Россия

Ссылка для цитирования

Леонов А.Г., Мартынов Н.С., Машенко К.А., Холькина А.А., Шляхов А.В. Автоматизация проверки семантической составляющей текстовых ответов обучающихся в цифровой образовательной платформе // Программные продукты и системы. 2024. Т. 37. № 3. С. 440–452. doi: 10.15827/0236-235X.142.440-452

Информация о статье

Группа специальностей ВАК: 2.3.5

Поступила в редакцию: 09.04.2024

После доработки: 04.05.2024

Принята к публикации: 14.05.2024

Аннотация. Цель настоящего исследования заключается в разработке и применении системы автоматизированной проверки ответов студентов на открытые текстовые вопросы в рамках цифровой образовательной платформы. Разработанная система предназначена для существенного снижения нагрузки на преподавателя и минимизации ошибок первого и второго рода в процессе проверки ответов студентов. Методология исследования включает использование современных методов извлечения информации, текстового анализа и сопоставления концепций. С помощью построения графа синтаксического разбора предложения формируются синтаксические деревья, после чего с учетом особенностей русского языка проводится анализ отношений между их узлами, например, исключение двойных отрицаний, извлечение именованных сущностей (NER) и распознавание лексических взаимосвязей. Для сравнения деревьев используется поиск оптимального пути редактирования, реализованный с применением динамического программирования. Результатом исследования является разработанная авторами система, способная оценивать ответы студентов с учетом смысловой глубины содержания текстов, что может обеспечить объективность и прозрачность оценки знаний. Кроме того, она позволяет снизить нагрузку на преподавателей, автоматизируя процесс проверки ответов, а возможность параметрической настройки сценария работы системы помогает подстроить ее под поставленные преподавателем специфические требования к ответам. Система способна игнорировать грамматические ошибки, так как ее фокус направлен на отслеживание расхождений или сходств в смысловой составляющей текстов.

Ключевые слова: цифровая образовательная платформа Мирера, смысловой анализ, семантический анализ, система оценки ответов, образовательные технологии, автоматизированная проверка

Благодарности. Работа выполнена в рамках госзадания ФГУ ФНЦ НИИСИ РАН по теме № FNEF-2024-0001 (1023032100070-3-1.2.1)

Введение. Повышение эффективности процесса обучения достигается не только за счет технических и дидактических средств, но и с помощью организации обратной связи с обучаемым, которая обеспечивается контролем знаний в интеллектуальных обучающих системах. Контроль знаний в таких системах, как правило, осуществляется автоматизированными тестирующими компонентами [1]. Тестовые задания, предлагаемые обучающимся, можно разделить на два типа – открытые и закрытые. Первые подразумевают конструируемый ответ на вопрос, а вторые – выбор одного или нескольких ответов из предложенного списка. Основная проблема закрытых вопросов заклю-

чается в том, что они проверяют, насколько хорошо обучающийся запомнил ответ, но не показывают процесс его размышлений. Проверить логику и степень понимания материала таким способом невозможно. Открытые вопросы, напротив, заставляют студента поразмышлять над вопросом, составить конструктивный ответ. Это наиболее полная и естественная форма контроля знаний. Проблемой открытых вопросов является огромная рутинная нагрузка на преподавателя при проверке студенческих ответов. Основная задача, которую в данном процессе должна выполнять интеллектуальная система, заключается в выявлении и проверке смысла в студенческом ответе.

Его правильность определяется путем сравнения этих мыслей с извлекаемыми из эталонного ответа преподавателя. Преимуществом автоматизации проверки является не только снижение нагрузки на преподавателя, но и прозрачность и справедливость результата, поскольку исключается фактор предвзятости.

Обзор существующих автоматизированных систем проверки ответов на открытые вопросы

В ходе анализа существующих автоматизированных систем проверки были выявлены четыре основные категории методов и подходов, на которые опираются архитектурные решения таких систем.

Первая категория использует метод извлечения информации. Он является разновидностью информационного поиска и заключается в выделении структурированной информации из неструктурированного текста. Описанный в работе [2] алгоритм один из самых первых в области проверки свободных ответов на английском языке с помощью извлечения информации. При проверке авторы строят графы разбора ответа обучающегося и эталонного ответа преподавателя, из которых впоследствии извлекается информация и сравнивается между собой. Важным исследованием в данной области является работа [3]. В ней авторы описывают систему, оценивающую ответ по сопоставлению фактов, извлеченных из текстов по определенным грамматическим правилам. Подход к сравнению построенных по извлеченной информации графов изложен в [4]. Рассматривая соединенные подграфы с графовым ядром, авторы получили низкоразмерное векторное представление исходного графа структуры. Затем сравнивались подграфы и измерялась их сопоставимость. Классические подходы текстовой репрезентации в виде графа представлены в [5].

Вторая категория опирается на сопоставление концепций. Метод основан на выделении списка минимальных концепций (ключевых понятий) из ответа обучающегося и эталона и на их последующем сравнении. Именно этим подходом пользовались при создании системы в работе [6]. С помощью фреймворка Generalised Phrase Structure Grammar описываются синтаксис и семантика языка, затем ответ обучающегося и эталонный ответ преподавателя разбиваются на списки минимальных концепций. Получившиеся концепции подсчитыва-

ются, и с учетом веса для каждой из них выставляется оценка.

Третья категория использует при проверке ответов лингвистические корпуса (цельный свод текстов для исследования языка). В работе [7] описываются основанные на корпусах классические подходы к нахождению схожести текстов. При работе с большими текстами преимущественно используются лингвистические корпуса. Для работы с маленькими текстами в качестве корпуса используют эталонные ответы преподавателя. При проверке этим методом составляются N-граммы (последовательности из N элементов) слов. В работе [8], где применялся данный метод, для поиска правильных отрывков из ответа обучающегося сначала используется степень зависимости N-грамм слов эталонного ответа в отрывке ответа обучающегося, а затем SVM-модель (модель, использующая метод опорных векторов) для улучшения ранжирования отрывков, включающая различные меры лексического, синтаксического и семантического сходства.

К числу быстро набирающих популярность относятся системы, использующие машинное обучение. Как правило, этот подход заключается в векторном представлении слов и использовании метрик для естественного языка, определяющих сходство текстов. В работе [9] предлагается подход, использующий различные методы обработки естественного языка и инструменты, такие как SCBOW, Wordnet, Word2vec (для векторного представления слов), косинусное сходство, мультиномиальный наивный Байес (MNB), частота слова и обратная частота документа (TF-IDF) (для определения меры сходства). Особый интерес исследователей в области обработки естественного языка вызывают языковые модели, в которых применяются рекуррентные нейронные сети и трансформеры. За счет своих архитектур эти нейронные сети понимают тексты на семантическом уровне, что позволяет им отвечать на вопросы человека, вести конструктивный диалог. Несмотря на высокие результаты таких языковых моделей, как GPT, BLOOM, LoRA, они регулярно подвергаются критике [10]. В этой связи использование подхода, основанного на языковых моделях, ставится под сомнение в системах, где критически важна цена ошибки. Помимо этого, в русском языке малейшее изменение даже порядка слов может привести к принципиально разным по смыслу ответам, что не обнаруживается моделями, основанными на машинном обучении. В работе [11] авторы

предлагают модель, работающую на графовых сверточных нейронных сетях (GCNS) и использующую деревья синтаксических зависимостей для получения семантической репрезентации предложений. С помощью этой модели слова в предложениях кодируются в семантическое векторное представление.

Зачастую в модели используется смешение обозначенных выше подходов. Например, в работе [12] авторы применяют совмещенную концепцию графа знаний (KG), иллюстрирующего отношения между концепциями, и больших лингвистических корпусов. В работе [13] описана система, совмещающая в себе составление векторных малоразмерных представлений и алгоритм Siamese Manhattan LSTM (MaLSTM) [14]. Работа этой системы была проверена авторами на открытых студенческих ответах в цифровой образовательной платформе. К комплексным относится модель из [15], использующая совместно текстовую сегментацию, определение частей речи, построение семантической матрицы и графа синтаксического разбора. В [16] рассмотрен алгоритм, в котором из текста выделяются ключевые слова с помощью анализа графлетов, а после этого применяется MNB.

В ходе изучения опыта создания систем автоматизированной проверки было отмечено, что только описанная в [3] система создавалась для обработки русскоязычного текста.

Постановка задачи и определение требований

Перед авторами данного исследования стояла задача разработки системы автоматизированной проверки любых по длине ответов на открытые вопросы. Преподавателю необходимо создать набор коротких односложных предложений с мыслями, которые должны присутствовать в ответе студента. Предложения в ответе студента могут быть любой сложности. Разрабатываемая система должна уметь анализировать ответы обучающихся на русском языке и на основе сравнения с эталонным ответом преподавателя выставлять предварительную оценку. При сравнении необходимо учитывать не только полностью совпадающие слова, но и синонимы, антонимы, должны игнорироваться формы и склонения слов, учитываться различные возможные написания дат и имен собственных, а также использование местоимений. Помимо этого, должны учитываться обычные и многоуровневые отрицания

и их влияние на итоговый смысл слов и предложения в целом. За использование антонимов и отрицаний, инвертирующих смысл написанного, при расчете оценки должны начисляться штрафы. Для автоматического выставления «задание зачтено» и «задание не зачтено» преподавателю следует определять пороги отсеечения. Так, если уровень правильности меньше нижнего порога, решение считается неверным. Если доля правильности больше верхнего порога, решение автоматически засчитывается. Если набранная оценка оказалась между двумя порогами, то требуется ручная проверка преподавателем, о чем ему приходит уведомление.

Система развернута на базе цифровой образовательной платформы Мирера, обеспечивающей хранение и сдачу решений, отображение результатов, рассылку уведомлений и пр. [17, 18].

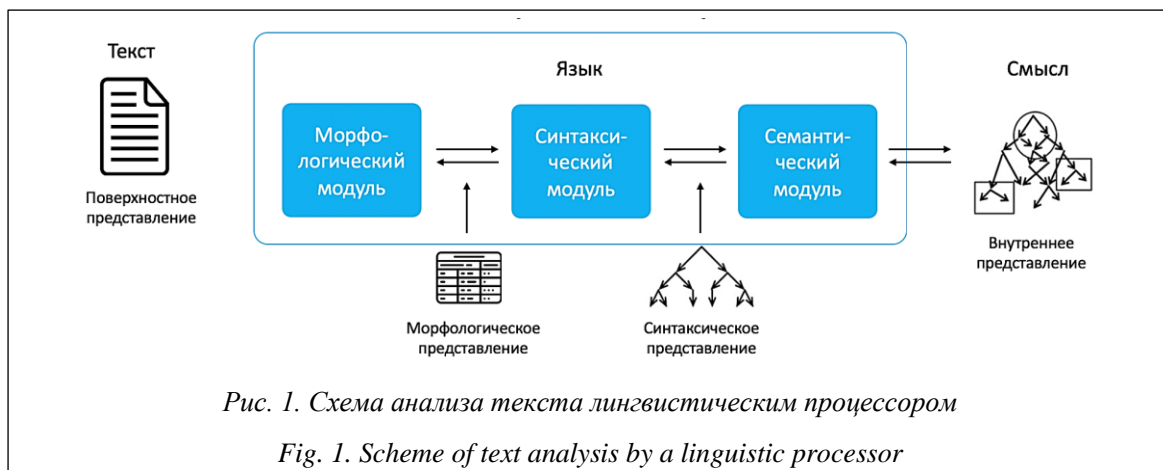
Построение графа синтаксического разбора

Решать поставленную задачу предлагается с помощью информационного поиска, комбинируя подходы построения графов разбора и сопоставления фактов. Таким образом, решение задачи сводится к синтаксическому анализу, результатом которого являются графы синтаксического разбора, а также к построению на основе этих графов деревьев и к их сравнению.

Под синтаксическим анализом в лингвистике и информатике подразумевают процесс сопоставления линейной последовательности лексем (слов, токенов) естественного или формального языка с его формальной грамматикой. Для программной реализации построения графов синтаксического разбора применяются лингвистические процессоры, основными задачами которых являются извлечение смысла из текста и его интерпретация набором формальных средств, понятных компьютеру. В процессе анализа текста лингвистический процессор выполняет морфологический и синтаксический анализ. Результатом работы лингвистического процессора является синтаксическое представление предложения в виде графа зависимостей, которое можно семантически анализировать. Схема работы лингвистического процессора графически представлена на рисунке 1.

При программной реализации компонентов лингвистического процессора используются два подхода – метод, основанный на правилах, и машинное обучение с учителем.

Основная идея первого метода заключается в создании набора правил, которые опреде-



ляют, как проставлять связи в предложении. Русский язык насчитывает огромное количество таких правил. В лингвистические процессоры, основанные на этом методе, например, в Томита-парсер, необходимо вносить эти правила самостоятельно, что требует немалых усилий и отдельной команды лингвистов. Авторы работы [3], использующие этот подход, отмечают, что заранее определить полный набор правил нельзя. По этой причине при анализе сложных предложений могут возникать проблемы.

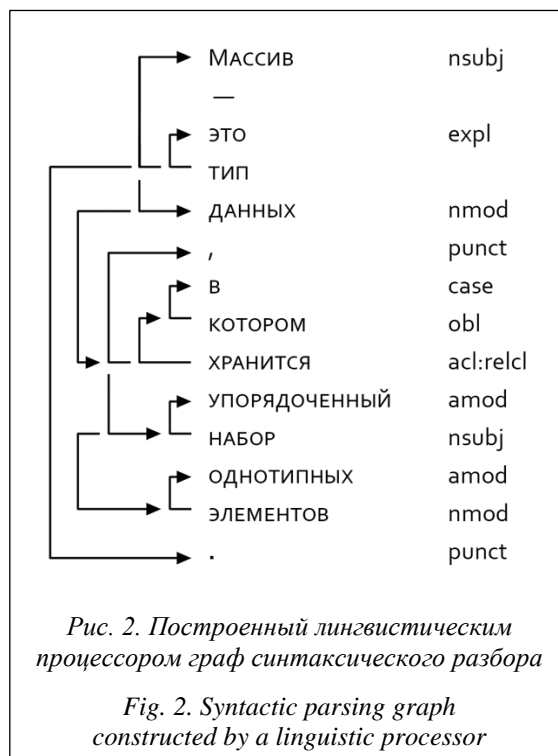
Метод, основанный на машинном обучении, заключается в обучении моделей на лингвистических корпусах, где размечена синтаксическая структура. Один из популярных проектов, содержащий в себе модели для морфологического и синтаксического анализа, добился высоких результатов в автоматическом структурировании текстов на русском языке. Речь идет о проекте Natasha (набор Python-библиотек для обработки текстов на естественном русском языке (<https://natasha.github.io/>)), программные инструменты которого позволяют решать все базовые задачи обработки естественного языка: сегментация на токены и предложения, морфологический и синтаксический анализ, лемматизация, извлечение именованных сущностей (NER) и анализ слов на принадлежность к определенным частям речи (POS-tagging). Эти модели позволили авторам собрать гибкий и эффективный лингвистический процессор, не требующий дополнительных действий от преподавателя по сравнению с инструментами, использующими метод, основанный на правилах. Отмечается проект CoreNLP (<https://stanfordnlp.github.io/CoreNLP/>) для работы с иностранными текстами.

В результате работы лингвистического процессора имеется граф синтаксического разбора

(рис. 2), в котором связи между словами определяются согласно межъязыковой типологии Universal Dependency Relations (<https://universaldependencies.org/u/dep/>).

Построение синтаксического дерева на основе графа синтаксического разбора

Сравнение синтаксических графов является вычислительно сложным алгоритмом. Это обусловлено тем, что в русском языке порядок слов нефиксированный. Так, слова могут идти абсолютно в разной последовательности, при этом смысл сказанного может как не измениться вообще, так и измениться на противоположный. Помимо этого, в синтаксическом



графе предложений русского языка могут быть циклы, а в графе одного предложения может быть несколько компонент связности. Все это приводит к тому, что даже графы похожих по своему устройству предложений могут быть абсолютно разными. Поэтому необходимо, во-первых, очистить исходный граф от частей речи и слов, не влияющих на общий смысл предложения, и, во-вторых, привести граф к дереву, что позволит применять вычислительно более быстрые алгоритмы определения степени их различия. Для работы с графами и, в частности, с деревьями авторы пользуются библиотекой NetworkX – Python-библиотека для работы с графами и сетями (<https://networkx.org/>).

Очистка циклов и объединение в одну компоненту связности

В достаточно сложных и длинных предложениях в русском языке модель построения графа синтаксического разбора из библиотеки Natasha может вернуть граф, в котором будут циклы. Чтобы удалить ребро в цикле наилучшим образом, выделяются части речи, лексически наиболее значимые в русском языке. Такими являются глаголы, существительные, имена собственные и местоимения. Поэтому для устранения циклов в структуре взаимосвязей внутри графа самым приоритетным является удаление ребра, входящего в узел, обладающий наивысшей значимостью. Если оно находится, то происходит разрыв этой взаимосвязи. Однако если такое ребро найти не удается, при возвращении в стартовую вершину удаляется последняя найденная взаимосвязь.

Таким образом, гарантируется исчезновение всех циклов внутри графа. Для предложений со сложной составной структурой несвязный граф, состоящий из нескольких деревьев, преобразуется в односвязное дерево. С этой целью выбирается главный корень, и все остальные корни становятся его наследниками. Впоследствии, если в студенческом ответе составные части сложного предложения будут указаны в другом порядке, алгоритм сравнения деревьев сможет перенаправить это ребро в другую сторону без понижения вероятности совпадения.

Наглядно продемонстрируем преобработку графа с помощью процедур избавления от циклов, соединения корней для получения односвязности и очистки графа от лишних частей речи, которые определены как лексически малозначительные. Рассматривается следующая строка: «Движения макроскопических тел

при скорости, много меньшей скорости света». На рисунке 3а показан предварительно построенный граф для данного текста, а на рисунке 3б – преобразованный исходный граф после применения к нему алгоритма предобработки. Следует отметить, что в данном примере в структуре графа был разорван цикл, соединены между собой две компоненты связности и удалены лексически малозначительные части речи.

Предобработка отрицаний

Важной частью работы являются поиск и анализ лексической отрицательности узлов графа. Отрицание может распространяться как на родительский узел (например, «совсем не яркий» подразумевает, что «не» является наследником у слова «яркий»), так и на узел наследника (например, «нет места» влечет связь, в которой «нет» будет родительским узлом для «места»). Если лексическая отрицательность распространяется от наследника к родителю, то выполняется алгоритм удаления наследника, который заключается в удалении узла отрицания с дальнейшим переприсвоением наследников. После перестроения графовой структуры родителю добавляется метка

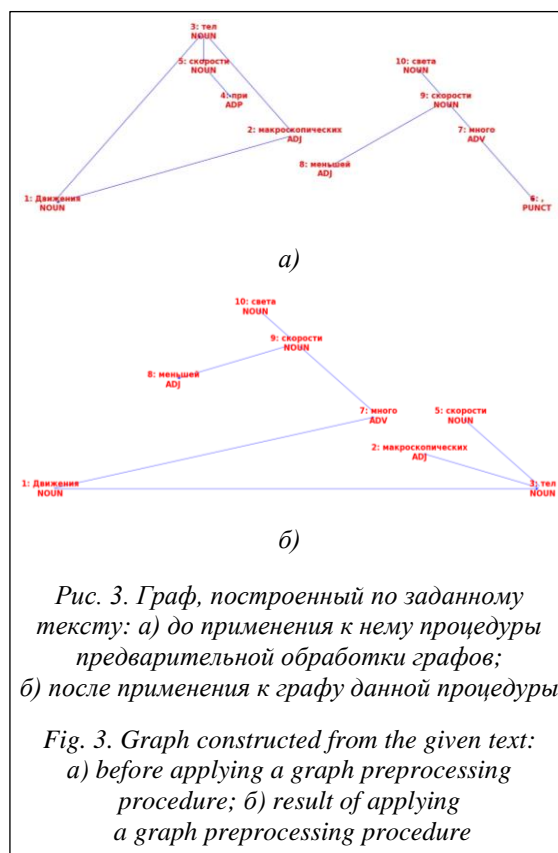


Рис. 3. Граф, построенный по заданному тексту: а) до применения к нему процедуры предварительной обработки графов; б) после применения к графу данной процедуры

Fig. 3. Graph constructed from the given text: а) before applying a graph preprocessing procedure; б) result of applying a graph preprocessing procedure

отрицательного смысла. В случае обратной ситуации, когда отрицание распространяется от родительского узла к наследнику, добавляется метка отрицательного смысла для всех наследников без удаления корневого слова.

Хорошо иллюстрирует работу алгоритма с отрицаниями граф, построенный по следующему предложению: «Я не мог не согласиться, что нет никаких противоречий» (<http://www.swsys.ru/uploaded/image/2024-3/19.jpg>).

Узлы синтаксических деревьев

Чтобы правильно сравнивать узлы синтаксического дерева между собой, требуется корректно определять способы их сравнения. Для достижения этой цели выделим три различных типа узлов, которые получаются в результате построения синтаксического дерева.

1. «Имена». После соответствующей предобработки текста из нескольких элементов дерева в один узел собирается все, что относится к имени, и сохраняется в едином универсальном формате. При сравнении узлов, которым присвоен тип имени, производится поиск частичных совпадений. Например, в случае отсутствия отчества в одном из узлов и его присутствия в другом, а также при условии совпадения остальных частей имени узлы в данной паре признаются совпадающими (узлы «А.С. Пушкин», «Александр Пушкин» и «Пушкин» совпадают).

2. «Даты». Аналогично с обработкой имен все составные части, относящиеся к одной дате, собираются в единый узел с соответствующим присвоением типа узла. Процедура сравнения такого типа узлов определяет даты совпадающими в случае их частичного совпадения («12 декабря 2023», «12.12.2023» и «12 декабря» совпадают).

3. «Текст». В этот тип узлов включаются все остальные вершины графа. Сравнение пары узлов такого типа будет описано детально позднее в силу того, что эта процедура является куда более нетривиальной, чем для остальных типов узлов.

Для наглядности рассмотрим обработку пары следующих предложений: 1) «С сентября 1830 года Михаил Юрьевич Лермонтов учился в Московском университете»; 2) «С 01.09.1830 М.Ю. Лермонтов учился в Московском университете». В данных предложениях даты и имена написаны в разных форматах, однако они в любом случае выделяются в единый узел. Кроме того, при дальнейшей обработке графов

и их последующем сравнении даты и имена в этом примере будут распознаны совпадающими (<http://www.swsys.ru/uploaded/image/2024-3/20.jpg>).

Виды отношений на множестве узлов синтаксических деревьев

Введем отношения на парах узлов.

Назовем узлы совпадающими, если они в точности совпадают после их последовательных преобразований к стандартной форме или являются синонимами и одновременно имеют или не имеют метку отрицательности. Также узлы будут совпадающими, если являются антонимами друг к другу и один из узлов имеет метку отрицательности. Примером таких преобразований для текстовых узлов могут послужить процедуры лемматизации (приведение слов к нормальной форме) и стемминга (нахождение основы слова).

Назовем узлы противоположными, если они в точности совпадают после их последовательных преобразований к стандартной форме или являются синонимами и один из узлов имеет метку отрицательности. Также узлы будут противоположными, если они являются антонимами друг к другу и оба узла одновременно имеют или не имеют метку отрицательности. Для определения того, что слова – синонимы или антонимы, авторы используют открытый репозиторий, в котором синонимы и антонимы представлены в виде графовой структуры (https://github.com/ahmados/rusynonyms/blob/main/ru_synonyms/synonyms.py).

Назовем узлы наложившимися или наложенными друг на друга, если они являются совпадающими или противоположными.

Для узлов, по типу являющихся датами или именами, алгоритмом сравнения будет поиск частичных совпадений. Наибольший интерес представляет алгоритм сравнения текстовых узлов. Следует отметить, что в русском языке местоимения могут употребляться вместо имен существительных, прилагательных или числительных. Поэтому важно при сравнении текстового узла, в котором содержится местоимение, с другим узлом определить его соответствующее отношение к узлу, необязательно являющемуся местоимением. Для таких слов нужно установить совпадение по гендерной принадлежности, определение которой возможно с помощью сравнения текста до применения к нему процедуры стемминга.

Алгоритм сравнения синтаксических деревьев, построенных на основе графов синтаксического разбора

Так как процесс сравнения синтаксических деревьев является вычислительно сложной процедурой, перед его началом необходимо убедиться, что вообще имеет смысл сравнивать графы между собой столь ресурсозатратным алгоритмом. Для этого предварительно определяется степень возможной изоморфности деревьев, построенных на основе эталонного и студенческого ответов, а затем сравниваются все узлы одного графа со всеми узлами другого. В случае получения в качестве результата предварительной проверки процентного соотношения наложившихся узлов меньшего, чем заданная нижняя граница автоматического отклонения, в качестве ответа системы возвращаются заведомо нулевые метрики, означающие несовпадение синтаксических деревьев студенческого и эталонного ответов.

Далее степень изоморфности графов определяется с помощью алгоритма для нахождения оптимального пути между двумя графами с использованием динамического программирования, представленного в работе [19].

1. Инициализируется матрица динамического программирования D значениями $D(i, 0) = i$ и $D(0, j) = j$, где i и j – индексы, представляющие i -й и j -й узлы в двух графах. Объясняется это тем, что первая строка и столбец представляют случай, когда один из графов пуст, поэтому расстояние редактирования между любым узлом непустого графа и пустым графом равно количеству узлов в непустом графе.

2. Вычисляется оптимальный путь редактирования. Для каждой ячейки (i, j) в таблице D вычисляется минимальное расстояние редактирования одного подграфа в другой с помощью возможных элементарных операций над узлами (вставки, удаления, замены). Стоимость каждой операции добавляется к стоимости предыдущего преобразования подграфа, и минимальное значение сохраняется в $D(i, j)$. Оптимальный путь между двумя графами получается обратным проходом через таблицу динамического программирования, начиная с нижней правой ячейки $D(n, m)$, где n и m – количество узлов в двух сравниваемых графах.

3. Применяются оптимизации. Алгоритм включает в себя несколько оптимизаций для повышения его эффективности, таких как сокращение пространства поиска и использование эвристик для управления поиском.

Таким образом, с помощью элементарных преобразований можно искать оптимальные наложения одного графа на другой, содержащие в себе наибольшее количество наложенных узлов. На вход данному алгоритму подается функция, которая определяет узлы, считающиеся наложенными друг на друга. Алгоритм на выходе возвращает пары узлов, оказавшихся наложенными по методу кратчайшего расстояния редактирования.

После получения наилучшего варианта наложения вершин друг на друга проводится анализ отношений на полученных парах узлов. Выявляются сначала совпадающие узлы, а затем распознающиеся как противоположные. В конце обрабатываются вершины, не относящиеся ни к тем, ни к другим, но наложившиеся друг на друга в ходе работы алгоритма, а также не получившие пары вовсе. Для сравнения по смыслу таких аберрантных пар в дальнейшем применяется кросс-проверка с целью обнаружения совпадающих узлов.

Обработка множественных отрицаний

Найденные противоположные вершины подвергаются дополнительной постобработке на предмет выявления и удаления сдвоенных противоположностей. Критерий для их распознавания в рекурсивной обработке – наличие у пары противоположных вершин наследников, которые тоже являются парой противоположных вершин. Анализируется глубина подобной рекурсии, и в дальнейшем четным образом происходит удаление распознанных данных способом пар узлов – остается либо одна самая первая пара противоположных узлов, либо ни одной (<http://www.swsys.ru/uploaded/image/2024-3/13.jpg>). Так, в предложениях «Я не мог не согласиться с этим утверждением» и «Я мог согласиться с этим утверждением» пары противоположностей и их противоположных наследников после обработки выглядят так: «мог (отрицательно) – мог» и «согласиться (отрицательно) – согласиться». Нахождение таких пар отражается при подсчете вероятности в виде совпадений.

Подсчет метрик

При вычислении метрик совпадения деревьев студенческого и эталонного ответов необходимо учитывать различия в найденных парах узлов: совпадающие, противоположные и совпадающие после кросс-проверки вершины.

В работе выделены три информативных показателя.

1. Доля наложения. Она измеряет то, насколько предложения являются подходящими одно другому. В нем учитываются количество наложенных вершин, вычисленное после применения апробационного алгоритма для выявления изоморфности графов, и количество наложений, полученных кросс-проверкой. Все это сравнивается с графом эталонного ответа, в котором из-за простоты предполагается меньшее количество узлов.

2. Доля совпадения. Отличие от доли наложения в том, что вместо количества наложенных вершин рассматривается количество именно совпадающих вершин. Для корректного учета число удаленных двойных противоположностей должно добавляться к количеству совпадающих вершин.

3. Число противоположных вершин. Такие противоположные с эталонным ответом вершины свидетельствуют о наличии лексической ошибки в ответе ученика, поэтому в метрику они поступают в виде штрафа.

Демонстрация работы системы на показательных примерах

Рассмотрим значение анализа синонимов и местоимений на примере сравнения предложений «Изображение не изменится, фото будет менее ярким» и «Картинка не поменяется, оно будет тусклым». На рисунке 4 можно отследить взаимосвязи между дальнейшими наложениями вершин друг на друга. Слова «изображение» и «картинка» являются синонимами, поэтому они отмечаются в виде совпадающих. Аналогично в роли синонимов выступают слова «изменится» и «поменяется», поэтому их отрицания совпадают. Обработка местоимений позволяет соотнести между собой «фото» и «оно» в качестве совпадающих узлов. Наконец, «ярким (отриц.)» и «тусклым» совпадают, так как «ярким» и «тусклым» являются антонимами, а одному из слов приписана метка отрицательности. В конечном счете метрики в виде доли наложения и доли совпадения принимают единичное значение, что свидетельствует о полном совпадении графов студенческого и эталонного решений.

Точно таким же образом обрабатываются обратные случаи, в которых накладываются узлы, являющиеся комбинациями отрицаний, синонимов и антонимов и получающие статус противоположностей.

Продемонстрируем независимость результата от порядка расположения лексически вер-

ных логико-грамматических конструкций внутри предложения. Для этого рассмотрим следующую пару строк: «Они не могли не обратить внимания на эти данные» и «На эти данные не могли не обратить внимание». Графы для данных предложений после предобработки изображены на рисунке 5. Полученные в результате работы системы пары совпадающих вершин и метрики для доли наложения и доли совпадения свидетельствуют о независимости результата ее работы от порядка конструкций внутри предложения. Акцент данного алгоритма сделан на том, чтобы уловить смысловую составляющую текстов. Отметим важность в данном примере кросс-проверки, так как именно с ее помощью пара совпадающих вершин «данные» и «данные» находят друг друга, хотя поиск оптимального пути не поставил их в соответствие из-за отсутствия действующего лица «они» в тексте студенческого ответа.

Аналогично можно рассмотреть пару сложных предложений, в которых основы представлены в разном порядке. Например, «Мы были восхищены, а они едва подавали признаки жизни» и «Они едва подавали признаки жизни, а мы были восхищены» (<http://www.swsys.ru/uploaded/image/2024-3/14.jpg>). Полученные пары совпадающих вершин и единичные метрики свидетельствуют о независимости результата работы системы от порядка расположения основ предложения и связанных с ними оборотов.

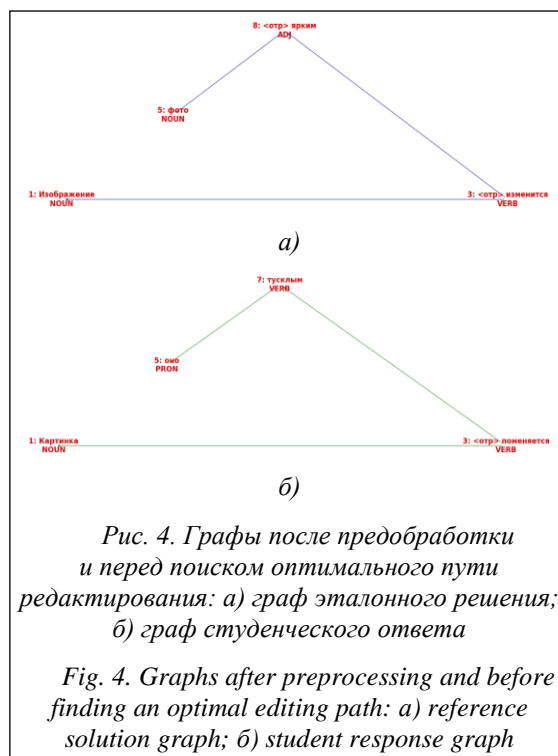
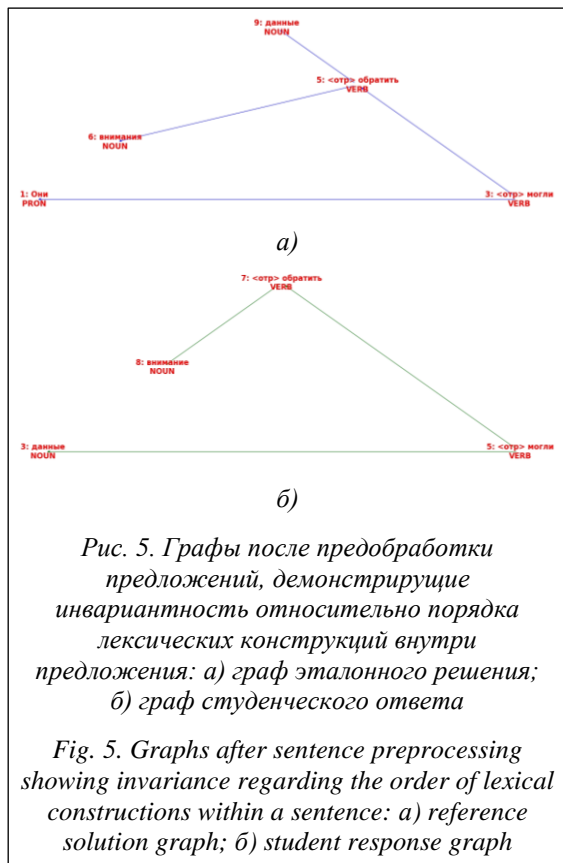


Рис. 4. Графы после предобработки и перед поиском оптимального пути редактирования: а) граф эталонного решения; б) граф студенческого ответа

Fig. 4. Graphs after preprocessing and before finding an optimal editing path: а) reference solution graph; б) student response graph



Научная новизна исследования

В данной работе выделяется разработка системы автоматизированной проверки ответов на основе метода извлечения информации, что позволяет оценивать студенческие ответы со смысловой стороны текста, игнорируя многие грамматические неточности. Это обеспечивает прозрачность, справедливость и исключает предвзятость в оценке результатов, а также позволяет уменьшить нагрузку на преподавателя. Система автоматизированной проверки текстов на основе извлечения информации является более интерпретируемой для пользователей, чем на основе нейросетевых алгоритмов, так как она руководствуется правилами и алгоритмами, специфичными для языка, что делает ее более надежной в обработке студенческих ответов, поскольку проверяется реальный смысл ответа, а не его общая тематика. Отмечается возможность гибкой настройки системы с помощью задаваемых пользователем параметров обработки. Например, настраиваются верхние и нижние границы метрик, по которым проводится классификация итогового результата и которые могут быть специфичными для различных заданий, а также имеется возможность изменять внутреннюю систему штрафов и поощрений для каждой метрики.

Тестирование системы

Для проверки работоспособности системы были проведены контрольные мероприятия для студентов. Их ответы собирались и использовались в качестве тестовой выборки. Всего в выборке участвовали 456 студенческих ответов. После получения ответа система автоматизированной проверки принимала его на вход и, руководствуясь заданными преподавателем порогами отсека по метрике доли наложения, относила к одному из трех классов: правильный ответ, неправильный ответ или неопределенный ответ, нуждающийся в дополнительной ручной проверке преподавателем по причине неуверенности системы после попадания результирующей метрики работы алгоритма внутрь интервала, ограниченного порогами отсека, заданными преподавателем (выше границы признания ответа неверным и ниже границы признания его верным). Далее все те же студенческие ответы отдавались преподавателю, который независимо от решения системы автоматизированной проверки размечал их на правильные и неправильные. В данном тестировании пороги отсека по метрике настраивались вручную и отличались от задания к заданию на основании аналитической оценки преподавателем специфики конкретного задания. В дальнейших исследованиях планируется разработка ассистента автоматической подстройки пороговых значений под статистические особенности каждого отдельно взятого задания. После получения результатов проверок составлялась матрица несоответствий и высчитывались следующие ключевые метрики для полученных результатов: количество истинно положительных, ложноотрицательных, ложноположительных, истинно отрицательных, неопределенных положительных и неопределенных отрицательных результатов (табл. 1).

Отмечается, что пороги отсека настраиваются преподавателем вручную и вопрос их автоматического подбора по текущим статистическим характеристикам является открытым для дальнейших исследований. Важно настраивать их таким образом, чтобы сохранить отсутствие ошибок первого и второго рода, но при этом повысить метрику доли автоматической проверки, что будет выражаться в уменьшении доли ответов, отправленных на дополнительную проверку преподавателем в связи с неуверенностью системы. Стоит упомянуть о невозможности эмпирического под-

бора стандартных порогов, единых для всех заданий, так как задания в общем случае неоднородные и сами метрики являются параметрически варьируемыми. Это означает наличие необходимого требования именно динамической подстройки порогов под статистически значимую выборку для каждого задания.

Таблица 1

Матрица несоответствий для проведенного эксперимента оценки студенческих ответов

Table 1

Inconsistency matrix for the conducted experiment evaluating student responses

Матрица несоответствий		Оценка тестирующей системы		
		Ответ верен	Ответ неверен	Не определено
Оценка преподавателя	Ответ верен	311	0	57
	Ответ неверен	0	11	77

Немаловажной метрикой является доля распознанных ответов с помощью системы автоматической проверки, которая составила 70,61 %. Заметно, что в данном эксперименте система проявляла весьма щадящее по отношению к студенческим ответам поведение, о чем свидетельствует низкое соотношение истинно отрицательных к истинно положительным результатам, равное 0,035. Это достигается за счет ручного выбора достаточно малого нижнего порога отсечения.

Сравнение результатов с современными аналогами

Как уже отмечалось, система из [3] работает с русскоязычными текстами. Приведем для сравнения метрики, полученные в результате ее работы на аналогичном эксперименте со студенческими ответами (табл. 2). Результирующие метрики: точность – 96,64 %, полнота – 83,65 %, F-мера с единичным параметром β – 0,8968.

В силу отсутствия в работе [3] данных по порогам отсечения в целях сравнения можно считать, что пороги настраиваются по наилучшим показателям метрик. В таком случае в качестве преимущества системы, рассмотренной в настоящей работе, можно выделить отсутствие ошибок первого и второго рода. Это достигается благодаря тому, что при возникновении сомнений система отправляет преподава-

телю студенческий ответ на дополнительную проверку. Именно снижение нагрузки на преподавателей при возможности минимизации ошибочных классификаций является значимым преимуществом системы.

Таблица 2

Матрица несоответствий для проведенного эксперимента оценки студенческих ответов

Table 2

Inconsistency matrix for the experiment evaluating student responses conducted

Матрица несоответствий		Оценка тестирующей системы	
		Ответ верен	Ответ неверен
Оценка преподавателя	Ответ верен	747	146
	Ответ неверен	26	526

Заключение

Авторами была разработана система автоматизированной проверки ответов на открытые вопросы на основе метода извлечения информации. Система имеет возможность настройки преподавателем порогов отсечения для оценки. Немаловажную роль играет их корректная подстройка под статистические результаты с учетом минимизации ошибок распознающей системы и адаптации под различные метрики, настраиваемые преподавателем, что является одним из преимущественных направлений для дальнейших исследований.

В подавляющем количестве случаев результат работы системы совпадает с ожиданием преподавателей. Однако подобное соответствие ожиданиям имеет высокую корреляцию с качеством эталонного решения, которое посылается со стороны преподавателя, а также с четкостью формулировки задания. В связи с этим актуальным направлением продолжения работы является автоматизированная помощь преподавателю в составлении правильного эталонного решения.

Акцент в оценивании ответов смещен на анализ смысловой составляющей всего текста, что позволяет игнорировать некоторые грамматические ошибки: словообразовательные, морфологические или синтаксические. Однако нарушение лексических норм отслеживается данной системой, и в случае обнаружения подобного нарушения в результирующую метрику вносится штраф.

Список литературы

1. Леонов А.Г., Дьяченко М.С., Машченко К.А., Орловский А.Е., Райко И.Г., Райко М.В. Новые подходы к автоматизации проверки заданий в цифровых курсах // Информатизация образования и методика электронного обучения: цифровые технологии в образовании: матер. VI Междунар. науч. конф. 2022. Ч. 3. С. 173–178.
2. Mitchell T., Russell T., Broomhead P., Aldridge N. Towards robust computerised marking of free-text responses. Proc. 6th CAA Conf., 2002, pp. 233–249.
3. Кожевников В.А., Сабинин О.Ю. Система автоматической проверки ответов на открытые вопросы на русском языке // Информатика, телекоммуникации и управление. 2018. Т. 11. № 3. С. 57–72.
4. Gao Ju., Gao Ji. A similarity measurement method based on graph kernel for disconnected graphs. Proc. IJCAI, 2019, pp. 6430–6431.
5. Osman A.H., Barukub O.M. Graph-based text representation and matching: A review of the state of the art and future challenges. IEEE Access, 2020, vol. 8, pp. 87562–87583. doi: 10.1109/ACCESS.2020.2993191.
6. Callear D., Jerrams-Smith J., Soh V. CAA of short non-MCQ answers. Proc. 5th Computer Assisted Assessment Conf., 2001, pp. 1–14.
7. Goma W.H., Fahmy A.A. A survey of text similarity approaches. IJCA, 2013, vol. 68, no. 13, pp. 13–18. doi: 10.5120/11638-7118.
8. Othman N., Faiz R. Question answering passage retrieval and re-ranking using N-grams and SVM. Computación y Sistemas, 2016, vol. 20, no. 3, pp. 483–494. doi: 10.13053/CyS-20-3-2470.
9. Bashir F., Arshad H., Javed A. et al. Subjective answers evaluation using machine learning and natural language processing. IEEE Access, 2021, vol. 9, pp. 158972–158983. doi: 10.1109/ACCESS.2021.3130902.
10. Нурутдинов А.Р., Латыпов Р.Х. Перспективы биоинспирированного подхода в разработке систем искусственного интеллекта (обзор тенденций) // Ученые записки Казанского университета. Сер. Физико-математические науки. 2022. Т. 164. № 2-3. С. 244–265. doi: 10.26907/2541-7746.2022.2-3.244-265.
11. Marcheggiani D., Titov I. Encoding sentences with graph convolutional networks for semantic role labeling. Proc. EMNLP, 2017, pp. 1506–1515. doi: 10.18653/v1/D17-1159.
12. Sawant U., Garg S., Chakrabarti S., Ramakrishnan G. Neural architecture for question answering using a knowledge graph and Web corpus. Inf. Retrieval J., 2019, vol. 22, no. 3, pp. 324–349. doi: 10.1007/s10791-018-9348-8.
13. Bahel V., Thomas A. Text similarity analysis for evaluation of descriptive answers. 2021.arXiv preprint arXiv:2105.02935.
14. Othman N., Faiz R., Smaïli K. Manhattan Siamese LSTM for question retrieval in community question answering. In: LNPSE. Proc. OTM, 2019, vol. 11877, pp. 661–677. doi: 10.1007/978-3-030-33246-4_41.
15. Yang J., Li Y., Ga C., Zhang Y. Measuring the short text similarity based on semantic and syntactic information. Future Generation Computer Systems, 2021, vol. 114, pp. 169–180. doi: 10.1016/j.future.2020.07.043.
16. Alqaryouti O., Khwileh H., Farouk T., Nabhan A., Shaalan K. Graph-based keyword extraction. In: SCI. Intelligent Natural Language Processing: Trends and Applications, 2018, vol. 740, pp. 159–172. doi: 10.1007/978-3-319-67056-0_9.
17. Бахтеев О.Ю., Гафаров Ф.М., Гриншкун В.В., Дятлова О.В. и др. Цифровая платформа образования // Вестн. РФФИ. 2022. № 1. С. 87–103. doi: 10.22204/2410-4639-2022-113-01-87-103.
18. Васильев И.А., Кушниренко А.Г., Леонов А.Г., Машченко К.А., Холькина А.А., Шляхов А.В. Цифровая образовательная платформа Мирера – основа цифровой трансформации образовательного процесса // Новые образовательные стратегии в современном информационном пространстве: сб. матер. 2023. С. 140–144.
19. Abu-Aisheh Z., Raveaux R., Ramel J.-Y., Martineau P. An exact graph edit distance algorithm for solving pattern recognition problems. Proc. ICPRAM, 2015, vol. 1, pp. 271–278. doi: 10.5220/0005209202710278.

**Automation of semantic analysis for textual responses of students
in a digital educational platform**

**Aleksandr G. Leonov^{1,2,3,4}, Nikolay S. Martynov², Kirill A. Mashchenko^{1,2,4}✉,
Anna A. Kholkina⁵, Artem V. Shlyakhov¹**

¹M.V. Lomonosov Moscow State University, Moscow, 119991, Russian Federation

²Federal State Institution SRISA RAS, Moscow, 117218, Russian Federation

³Moscow State Pedagogical University, Moscow, 119435, Russian Federation

⁴State University of Management, Moscow, 109542, Russian Federation

⁵National University of Science Research Technological "MISIS",
Moscow, 119049, Russian Federation

For citation

Leonov, A.G., Martynov, N.S., Mashchenko, K.A., Kholkina, A.A., Shlyakhov, A.V. (2024) 'Automation of semantic analysis for textual responses of students in a digital educational platform', *Software & Systems*, 37(3), pp. 440–452 (in Russ.). doi: 10.15827/0236-235X.142.440-452

Article info

Received: 09.04.2024

After revision: 04.05.2024

Accepted: 14.05.2024

Abstract. The purpose of this study is to develop and apply a system of automated checking of students' answers to open-ended text questions within the framework of a digital educational platform. The developed system enables significant reducing of teacher workload and minimizing both type I and type II errors during evaluating students' answers. The research methodology involves modern methods of information extraction, text analysis and concept comparison. Constructing a sentence-parsing graph forms syntactic trees. After that, taking into account the peculiarities of the Russian language, the relations between their nodes are analyzed. For example, there might be double negation elimination, named entity extraction and lexical relationship recognition. To compare trees with each other, the authors use the search for the optimal edit path implemented using dynamic programming. The result of the research is a system that is able to evaluate students' answers taking into account the semantic depth of texts' content, which can ensure objectivity and transparency of knowledge evaluation. In addition, it reduces teacher workload by automating the process of checking answers. The possibility of parametric setting of the system operation scenario helps to adjust it to specific requirements to the answers set by a teacher. The system is able to ignore grammatical errors, as it focuses on tracking differences or similarities in the semantic component of texts. The practical significance of the work is in the possibility of effective and fair control of students' knowledge taking into account a text semantic component, in reducing teachers' routine and accelerating the process of evaluating students' answers. Also, it adapts to teachers' specific requirements and provides prompt feedback. This study is relevant and significant for educational practice, as it corresponds to modern educational trends and technologies.

Keywords: Mirera digital learning platform, meaning analysis, semantic analysis, response evaluation system, educational technology, automated verification

Acknowledgements. The work was carried out within the framework of the state assignment of the Federal State Institution SRISA RAS on topic no. FNEF-2024-0001 (1023032100070-3-1.2.1)

References

1. Leonov, A.G., Dyachenko, M.S., Mashchenko, K.A., Orlovsky, A.E., Raiko, I.G., Raiko, M.V. (2022) 'New approaches to automation of testing tasks in digital courses', *Proc. Int. Sci. Conf. Informatization of Education and E-learning Methods: Digital Technologies in Education*, 3, pp. 173–178 (in Russ.).
2. Mitchell, T., Russell, T., Broomhead, P., Aldridge, N. (2002) 'Towards robust computerised marking of free-text responses', *Proc. 6th CAA Conf.*, pp. 233–249.
3. Kozhevnikov, V.A., Sabinin, O.Yu. (2018) 'System of automatic verification of answers to open questions in Russian', *Computing, Telecommunications and Control*, 11(3), pp. 57–72 (in Russ.).
4. Gao, Ju., Gao, Ji. (2019) 'A similarity measurement method based on graph kernel for disconnected graphs', *Proc. IJCAI*, pp. 6430–6431.
5. Osman, A.H., Barukub, O.M. (2020) 'Graph-based text representation and matching: A review of the state of the art and future challenges', *IEEE Access*, 8, pp. 87562–87583. doi: 10.1109/ACCESS.2020.2993191.
6. Callear, D., Jerrams-Smith, J., Soh, V. (2001) 'CAA of short non-MCQ answers', *Proc. 5th Computer Assisted Assessment Conf.*, pp. 1–14.
7. Gomaa, W.H., Fahmy, A.A. (2013) 'A survey of text similarity approaches', *IJCA*, 68(13), pp. 13–18. doi: 10.5120/11638-7118.
8. Othman, N., Faiz, R. (2016) 'Question answering passage retrieval and re-ranking using N-grams and SVM', *Computación y Sistemas*, 20(3), pp. 483–494. doi: 10.13053/CyS-20-3-2470.
9. Bashir, F., Arshad, H., Javed, A. et al. (2021) 'Subjective answers evaluation using machine learning and natural language processing', *IEEE Access*, vol. 9, pp. 158972–158983. doi: 10.1109/ACCESS.2021.3130902.
10. Nurutdinov, A.R., Latypov, R.Kh. (2022) 'Potentials of the Bio-inspired approach in the development of artificial intelligence systems (trends review)', *Proc. of Kazan University. Phys. and Math. Ser.*, 164(2-3), pp. 244–265 (in Russ.). doi: 10.26907/2541-7746.2022.2-3.244-265.
11. Marcheggiani, D., Titov, I. (2017) 'Encoding sentences with graph convolutional networks for semantic role labeling', *Proc. EMNLP*, pp. 1506–1515. doi: 10.18653/v1/D17-1159.
12. Sawant, U., Garg, S., Chakrabarti, S., Ramakrishnan, G. (2019) 'Neural architecture for question answering using a knowledge graph and Web corpus', *Inf. Retrieval J.*, 22(3), pp. 324–349. doi: 10.1007/s10791-018-9348-8.
13. Bahel, V. and Thomas, A. (2021) 'Text similarity analysis for evaluation of descriptive answers', *arXiv preprint arXiv:2105.02935*.
14. Othman, N., Faiz, R., Smaïli, K. (2019) 'Manhattan Siamese LSTM for question retrieval in community question answering', in *LNPSE. Proc. OTM*, 11877, pp. 661–677. doi: 10.1007/978-3-030-33246-4_41.
15. Yang, J., Li, Y., Ga, C., Zhang, Y. (2021) 'Measuring the short text similarity based on semantic and syntactic information', *Future Generation Computer Systems*, 114, pp. 169–180. doi: 10.1016/j.future.2020.07.043.
16. Alqaryouti, O., Khwileh, H., Farouk, T., Nabhan, A., Shaaan, K. (2018) 'Graph-based keyword extraction', in *SCI. Intelligent Natural Language Processing: Trends and Applications*, 740, pp. 159–172. doi: 10.1007/978-3-319-67056-0_9.
17. Bakhteev, O.Yu., Gafarov, F.M., Grinshkun, V.V., Dyatlova, O.V. et al. (2022) 'Digital Education Platform', *Vestn. RFFI*, (1), pp. 87–103 (in Russ.). doi: 10.22204/2410-4639-2022-113-01-87-103.

18. Vasilev, I.A., Kushniренко, A.G., Leonov, A.G., Mashchenko, K.A., Kholkina, A.A., Shlyakhov, A.V. (2023) 'Mirera digital educational platform – the basis of digital transformation of the educational process', *Proc. NESinODS*, pp. 140–144 (in Russ.).

19. Abu-Aisheh, Z., Raveaux, R., Ramel, J.-Y., Martineau, P. (2015) 'An exact graph edit distance algorithm for solving pattern recognition problems', *Proc. ICPRAM*, 1, pp. 271–278. doi: 10.5220/0005209202710278.

Авторы

Леонов Александр Георгиевич^{1, 2, 3, 4}

к.ф.-м.н., доцент, ведущий научный сотрудник,
зав. кафедрой, профессор,
dr.l@math.msu.su

Николай Сергеевич Мартынов², инженер,

nikolai.martynov@math.msu.ru

Мащенко Кирилл Алексеевич^{1, 2, 4},

младший научный сотрудник,

kirill010399@vip.niisi.ru

Анна Александровна Холькина⁵,

студент, kholkina.a2021@gmail.com

Артем Вячеславович Шляхов²,

младший научный сотрудник,

shlyakhov@vip.niisi.ru

Authors

Aleksandr G. Leonov^{1, 2, 3, 4},

Cand. of Sci. (Physics and Mathematics),
Associate Professor, Professor, Head of Chair,
Leading Researcher, dr.l@math.msu.su

Nikolay S. Martynov², Engineer,

nikolai.martynov@math.msu.ru

Kirill A. Mashchenko^{1, 2, 4},

Junior Researcher,

kirill010399@vip.niisi.ru

Anna A. Kholkina⁵,

Student, kholkina.a2021@gmail.com

Artem V. Shlyakhov¹,

Junior Researcher,

shlyakhov@vip.niisi.ru

¹ МГУ им. М.В. Ломоносова,

г. Москва, 119991, Россия

² ФГУ ФНЦ НИИСИ РАН,

г. Москва, 117218, Россия

³ Институт детства Московского педагогического
государственного университета (МПГУ),

г. Москва, 119435, Россия

⁴ Институт информационных систем

Государственного университета управления,

г. Москва, 109542, Россия

⁵ Национальный исследовательский

технологический университет «МИСиС»,

г. Москва, 119049, Россия

¹ M.V. Lomonosov Moscow State University,
Moscow, 119991, Russian Federation

² Federal State Institution SRISA RAS,
Moscow, 117218, Russian Federation

³ Moscow State Pedagogical University,
Moscow, 119435,

Russian Federation

⁴ State University of Management,

Moscow, 109542,

Russian Federation

⁵ National University of Science

Research Technological "MISIS",

Moscow, 119049, Russian Federation