DOI: 10.18721/JCSTCS.11306
УДК 004.8

СИСТЕМА АВТОМАТИЧЕСКОЙ ПРОВЕРКИ ОТВЕТОВ НА ОТКРЫТЫЕ ВОПРОСЫ НА РУССКОМ ЯЗЫКЕ

В.А. Кожевников, О.Ю. Сабинин

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Рассмотрены системы автоматической проверки открытых ответов на естественном языке и описаны требования, предъявляемые к подобным системам. Предложена функциональная схема системы и изучены вопросы ее реализации. В качестве лингвистического процессора разработанной системы выбрана система извлечения информации из текстов Томита-парсер компании Яндекс. Написаны грамматические правила извлечения сущностей из текстов на русском языке и предложен алгоритм анализа ответов. При оценивании каждого ответа можно задать веса для каждой сущности и для каждого из имеющихся эталонных ответов, что позволяет производить настройку системы. Система реализована и протестирована при проверке ответов студентов.

Ключевые слова: система тестирования, обработка текста на русском языке, компьютерная лингвистика, извлечение информации, открытые вопросы, краткие развернутые ответы, Томита-парсер.

Ссылка при цитировании: Кожевников В.А., Сабинин О.Ю. Система автоматической проверки ответов на открытые вопросы на русском языке // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2018. Т. 11. № 3. С. 57–72. DOI: 10.18721/JCSTCS.11306

SYSTEM OF AUTOMATIC VERIFICATION OF ANSWERS TO OPEN QUESTIONS IN RUSSIAN

V.A. Kozhevnikov, O.Yu. Sabinin

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

The paper considers systems of automatic verification of open answers in natural language and describes the requirements for such systems. The article proposes a functional scheme of such a system and discusses its implementation. The Tomita-parser by Yandex for extracting information from texts was chosen as a linguistic processor of the developed system. Grammatical rules for extracting entities from texts in Russian are written and an algorithm for analyzing the answers is proposed. Weights can be set for each entity and for each of the available reference responses when evaluating each response, allowing to customize the system. The system is implemented and tested by checking students' answers.

Keywords: testing system, text processing in Russian language, computational linguistics, information extraction, open question, short answer, Tomita parser, Natural Language Processing.

Citation: Kozhevnikov V.A., Sabinin O.Yu. System of automatic verification of answers to open questions in Russian. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems, 2018, Vol. 11, No. 3, Pp. 57–72. DOI: 10.18721/JCSTCS.11306

Введение

Процесс обучения в любом образовательном учреждении подразумевает проверку усвоения учащимися полученных знаний. Такая проверка важна для обратной связи и последующей корректировки процесса обучения.

Одним из эффективных видов проверки знаний студентов является тестирование, играющее очень важную роль при оценивании результатов обучения. Если при классическом («аудиторном») образовании тестирование может быть как основным, так и вспомогательным инструментом проверки знаний, то при набирающем все больший вес дистанционном обучении компьютерное тестирование – практически единственный способ контроля усвоения знаний.

Как показывают наша практика и различные исследования (например, [1, 2]), наиболее эффективный метод тестирования – тестирование с помощью открытых вопросов (т. е. вопросов, где отвечающий дает развернутый или короткий ответ на естественном языке). При таком виде тестирования практически полностью исключается возможность угадывания правильного ответа, а от студента требуется умение не просто вспомнить термин или определение, но и корректно сформулировать свои мысли.

Однако на практике компьютерное тестирование ограничивается вопросами, в которых формат ответов жестко задан, а вопросы, требующие ответа на естественном языке, считаются непригодными для автоматизированного тестирования из-за сложностей обработки. Так, для использующейся во многих вузах системы тестирования (например, на базе платформы Moodle) возможны вопросы, требующие ввода текста на естественном языке, но система просто проверяет посимвольное совпадение введенного текста с эталонным ответом, введенным преподавателем. И такие ответы приходится проверять вручную преподавателям.

Преимущества систем автоматизированного оценивания ответов на открытые вопросы по сравнению с классической проверкой преподавателем:

- уменьшение времени проверки – система позволяет сразу оценить результаты тестирования,
- увеличение объективности оценивания – автоматизированное оценивание позволяет снизить ошибки из-за усталости или невнимательности и убрать возможные претензии студентов из-за предвзятого отношения.

Анализ существующих систем тестирования с помощью открытых вопросов

Существующие системы тестирования с помощью открытых вопросов можно разделить на несколько категорий, в зависимости от используемых методов работы с текстом [1].

К первой категории относятся системы, основанные на сопоставлении концепций (Concept Mapping). В таких системах, как правило, ответы тестируемого и эталонные ответы разбиваются на некоторые списки ключевых понятий (так называемых минимальных концепций), и либо подсчитывается общее количество концепций для выставления оценки, либо во время оценивания рассматривается только одна каким-то образом выбранная концепция. Каждой концепции можно приписать некий вес, и тогда в итоговой оценке учитываются эти веса.

Одной из первых таких работ была система [3], в которой вопросы рассматривались как гипотезы, и отвечающим необхо-

димо было дать несколько объяснений для данных гипотез, каждая из которых могла соответствовать или не соответствовать одному из ответов учителя. Тестируемых просили написать до 15 ответов (гипотез) для объяснения одного явления (длиной до 15 слов). Каждый ответ рассматривался как отдельная концепция. Применялся метод LCS (Lexical Conceptual Structure) из [4], в котором основанная на концепциях лексика и общая грамматика получаются из заранее подготовленного перед тестированием эталонного множества ответов (было подготовлено 172 ответа).

В [5] описана система ATM (Automatic Text Marker), написанная на Prolog. Она использует фреймворк Generalised Phrase Structure Grammar (GPSG) для описания синтаксиса и семантики языка. ATM разбивает эталонные ответы преподавателя и ответы тестируемого на некоторые списки минимальных концепций (так называемые "smallest viable unit of concepts") и подсчитывает общее количество концепций для выставления оценки. Каждой такой концепции приписан некий вес, и в итоговой оценке эти веса суммируются.

В работе [6] сопоставлено максимальное количество концепций в эталонных преподавательских ответах и ответах тестируемого. Для сопоставления использован классификатор TLC (Topical/Local Classifier), в основе которого Байесовский подход, и сопоставление основано на множестве правил и канонических представлений текста. При этом рассматриваются синтаксические и морфологические вариации слов, анафоры и синонимы. Эталонные ответы преподавателя рассматриваются как отдельные высказывания для каждой концепции. Во время оценивания рассматривается только одна концепция, что упрощает оценивание. Методы, применяемые в системе Concept Rater (c-rater), позднее использовались другими исследователями.

Метод сопоставления концепций в дальнейшем совершенствовался, но в современных системах мало используется.

Вторую категорию образуют системы, основанные на методах извлечения информации (Information Extraction Systems). Здесь, как правило, происходит сопоставление неким шаблонам, при этом используются регулярные выражения или деревья разбора. В результате из неструктурированных текстов извлекается информация в виде структурированных данных. Ответ может разбиваться на сегменты, и происходит оценивание каждого сегмента. Для каждого вопроса можно сформулировать более одного шаблона.

Одна из первых работ этой категории – [7]. В ней по ответам преподавателя и студентов строилось дерево разбора, и информация извлекалась поиском по шаблону по такому дереву. Причем использовались два подхода: полностью автоматизированный (blind) и с вмешательством человека (moderated). Преимущество второго метода – возможность пересмотра модели после этого вмешательства.

В [8] описана система WebLAS (Web-based Language Assessment System), написанная на Perl. Она разбивает эталонный ответ на сегменты, выбирает важные и просит преподавателя подтвердить их и назначить каждому свой вес. Также преподаватель может принять или отклонить семантически схожие альтернативы. Затем с помощью регулярных выражений обнаруживается наличие или отсутствие каждого сегмента в ответе студента. Соответственно, возможно оценивание не всего ответа, а его отдельных сегментов.

Метод извлечения информации – один из самых популярных в подобных системах. Предлагаемая в данной работе система также принадлежит к этой категории.

Третью категорию составляют системы, основанные на использовании корпусов (корпусом в лингвистике называется собранный и обработанный по определенным правилам набор текстов, используемый в качестве базы для исследования языка). Обычно корпуса применяются для работы с большими текстами, но подобный метод можно применять и при анализе коротких ответов, как правило, с использованием эталонного ответа в качестве словаря для ограничения правильных ответов. Часто при этом используются *N*-граммы (*N*-грамма – это последователь-

ность из *N* элементов, например, для слова «тест» 3-граммами будут «тес» «ест» и т. д.) и некоторые инвертированные индексы структуры данных.

Одна из первых работ этой категории – [9]. В ней для оценивания использована метрика BLEU (BiLingual Evaluation Understudy). Метод основан на совпадении *N*-грамм и на нормализованной длине выборки. Авторы усовершенствовали оригинальный алгоритм BLEU и назвали его ERB (Evaluating Responses with Bleu). На основе этого метода сделана онлайн система Atenea. Позднее к данному методу добавили другой метод, основанный на корпусах, – LSA (Latent Semantic Analysis), и стала использоваться комбинация методов BLEU и LSA с весами.

Авторы системы SAMText (Short Answer Measurementof TEXT) [10] применяют метод, являющийся вариантом LSA, основанный на некотором инвертированном индексе структуры данных. Их алгоритм семантической связанности требует наличия выделенного доменного специфичного индекса или собрания тематически ориентированных документов (т. е. корпуса), который создается с помощью автоматического механизма обхода web-контента, собирающего документы, основанные на описательных ключевых слов домена. Авторы утверждают, что инвертированный индекс и идея обхода документов больше подходят для кратких ответов, чем для больших.

Этот метод, хотя и несколько менее популярный, чем предыдущий, в последнее время в связи с разработкой доступных корпусов становится активно используемым.

В четвертую категорию входят системы, применяющие машинное обучение. При этом обычно используются разные метрики, взятые из методов обработки естественного языка. Они либо комбинируются, либо происходит оценивание с помощью одной из имеющихся классификационных или регрессионных моделей.

Одной их первых в данной категории была система [11], использовавшая для машинного обучения метрики ROUGE из [12] и линейную регрессию. Метрики определяли статистики совпадения *N*-грамм, сравнивали самую длинную общую подпоследовательность (LSC) слов, взвешенную LSC, статистику использования скип-биграмм и т. д. между оцениваемым и эталонным ответами. Архитектура этой системы имела гибкий дизайн, что позволяло ей работать либо автономно, либо как компонент другой системы, например, обучающей платформы.

Следует отметить работу [13]. В отличие от остальных систем проверки ответов, здесь к вопросам, ответам учащихся и эталонным ответам преподавателей добавляется четвертая компонента – читаемые тексты. По мнению авторов, это полезно, т. к. ответ студентов может относиться только к одной части прочитанных текстов. Поэтому нужно анализировать не только пару «ответ учащегося – эталонный ответ», но и пары «ответ учащегося – читаемый текст» и «эталонный ответ – читаемый текст».

Методы машинного обучения в системах анализа текстов набирают популярность в последние годы.

К пятой категории относятся системы, не попадающие под предыдущие категории, или в которых используются комбинированные методы. Многие из этих систем были представлены на коммерческих исследовательских конкурсах Automated Student Assessment Prize [14, 15], Recognizing Textual Entailment Challenge [16] и подобных им.

Проанализировано около сорока систем тестирования из всех категорий и установлено, что подавляющее большинство из них может работать только с английским языком, четыре – с испанским и две – с немецким. Ни одна из систем тестирования не может работать с русским языком. Из-за особенностей русского языка (некоторые из которых будут описаны ниже) модели, применяемые в разработанных системах тестирования, пришлось бы существенно перерабатывать и/или использовать русскоязычные корпусы.

В связи со сказанным выше создание системы тестирования с автоматическим анализом открытых ответов на русском языке является актуальной задачей.

Основа любой подобной системы – лингвистический процессор (ЛП) – приложение, осуществляющее лингвистический анализ системы. Разработка хорошего ЛП для русского языка – чрезвычайно трудоемкая задача. Поэтому нами был выбран уже существующий и хорошо зарекомендовавший себя ЛП для работы с русскими текстами. Он принадлежит к категории извлечения информации, и поэтому создаваемая нами система анализа открытых ответов тоже принадлежит к этой категории.

Требования, накладываемые на систему, функциональная схема и сценарий использования системы преподавателем

При разработке системы автоматической проверки ответов на открытые вопросы учитывалось, что система должна удовлетворять следующим требованиям:

- Уметь анализировать ответы на русском языке.
- Клиентская часть системы должна быть максимально облегчена, чтобы тестирование можно было проходить на обычных компьютерах без специально установленного программного обеспечения (ПО) как в компьютерной аудитории вуза, так и дома (в противном случае тестирование должно проводиться на компьютерах с установленным отдельно клиентским приложением, что могло бы сузить применимость такой системы).
- Обладать возможностью создавать т. н. курсы (как, например, на портале дистанционных образовательных технологий СПбПУ [17]). Курс – это набор учебных материалов (лекций, самостоятельных заданий и т. д.) и оценивающих знания тестов по какой-либо дисциплине. Сами тесты должны находиться внутри курсов. У одного курса может быть несколько тестов.
- Система должна быть многопользовательским приложением. В ней должны существовать группы преподавателей и студентов с разными правами (преподаватели составляют тесты, вопросы и эталонные ответы на них, студенты проходят тестирование). Пользователи объединяются в группы. Причем существует иерархия групп (одни группы могут входить в другие). Один пользователь может состоять в разных группах. Пользователи записываются на курс (как слушатели курса или преподаватели).
- В системе должен существовать банк вопросов, использующихся в тестах, и база эталонных ответов (ответов, данных преподавателями). Причем для более корректной работы системы важна возможность наличия нескольких правильных ответов на один и тот же вопрос, которые может дать как автор вопроса, так и другие преподаватели. Для удобства составления тестов вопросы тоже объединяются в группы. Группы вопросов также имеют иерархическую вложенную структуру. Тесты формируются на основе вопросов из этого банка. Это позволяет использовать один и тот же вопрос в разных тестах. При этом тесты можно создавать по-разному. Например, можно указать, какие вопросы из банка включать в тест, или указать, что в тест войдет какое-то количество случайных вопросов из этой группы вопросов, а какое-то – из другой и т. д. Разумеется, оба способа составления тестов можно комбинировать при составлении одного теста. Система анализирует текст ответа студента и сравнивает его со всеми эталонными ответами, что позволяет более корректно оценить ответ студента.
- Поскольку существует вероятность того, что один и тот же пользователь может проходить один и тот же тест более одного раза, то должна храниться информация о попытках сдачи теста, об оценке, которую поставила система, и если этот ответ оценивал еще и преподаватель, то и об оценке преподавателя за ответ.
- Для более корректной работы системы необходим большой словарь, в котором содержатся как общеупотребительные слова, так и узкоспециализированные термины из предметной области, по которой происходит тестирование.

Также нужен словарь синонимов, которым пользуется система проверки при оценивании ответа. Можно использовать готовые словари или разработать свои.

- Тонкая настройка системы при оценивании каждого ответа. Преподаватель может задать ценность каждого извлечен-

ного из ответа факта (об этом ниже) путем задания весов для каждого факта. Должна существовать возможность сделать это по-разному для каждого эталонного ответа. Используя эти веса, система вычисляет итоговую оценку ответа.

Как уже отмечалось, основной частью подобной системы является ЛП, и нами было решено использовать одну из существующих систем анализа текстов на русском языке, а не разрабатывать собственную.

Как правило, в ЛП анализ текста включает в себя следующие этапы [18]:

1. Графематический анализ (сегментация, токенизация и т. д.).
2. Морфологический анализ (нормализация, стемминг, частеречная разметка и т. д.).
3. Предсинтаксический анализ.
4. Синтаксическая сегментация.
5. Синтаксический анализ.
6. Семантический анализ (хотя до сих пор нет универсальных математических моделей и вообще формальных средств описания смысла слов).

Анализ текста на русском языке осложняется тем, что он считается одним из самых сложных для изучения: в нем, например, свободный порядок слов (иногда без контекста или специальных интонаций в речи даже носителю русского языка трудно правильно истолковать фразу), большое количество правил (и множество исключений из них, и даже исключений из исключений), сложная пунктуация и некоторые другие особенности. В отличие, например, от английского языка, русский язык – синтетический, а не аналитический, т. е. слова в данном языке состоят в основном из нескольких морфем (морфемой называют минимальную единицу языка, которая еще имеет некоторый смысл).

Русский язык считается языком с сильной морфологией: для большинства существительных есть по шесть падежей единственного и множественного числа, три склонения с различающимися окончаниями, три рода; существуют свои особенности образования падежей у одушевленных и неодушевленных существительных, для некоторых падежей существует несколько вариантов форм. В русском языке имеется гигантское, по сравнению с тем же английским языком, количество форм для прилагательных, существует деление глаголов на совершенный и несовершенный виды, и т. д. Ко всему этому существует множество исторически сложившихся исключений.

Русский язык – язык фузионный, а не агглютинативный, в нем морфемы не просто механически приклеиваются (агглютинация – приклеивание) друг к другу, а как бы спаиваются (фузия – сплав) друг с другом. В русском языке абсолютно разные по смыслу слова могут отличаться только ударением, он вообще очень богат формами словоизменения и моделями словообразования по сравнению с тем же английским языком. Все это приводит к тому, что подход к анализу текста в рассмотренных в предыдущем разделе системах для русского языка работать не будет.

В качестве ЛП для системы тестирования было рассмотрено более сорока приложений для работы с русским языком. Большая часть из них – коммерческие или академические системы с закрытым кодом. Из бесплатных и открытых продуктов был выбран Томита-парсер компании Яндекс [19], обладающий, на наш взгляд, наибольшей функциональностью и имеющий подробную документацию. Для Томита-парсера можно создавать свои грамматики и словари, описывать свои факты, что естественно будет необходимым для построения корректной системы автоматической проверки ответов на узкоспециализированные вопросы. В состав Томита-парсера входят три модуля: сегментатор (отвечающий за разбиение текста на предложения), токенизатор (отвечающий за разбиение на слова) и морфологический анализатор mystem.

Исходя из перечисленных требований, предложена и реализована функциональная схема системы тестирования, основные модули которой изображены на рис. 1.

Через браузер с помощью web-интерфейса в БД заносятся вопросы и ответы, происходит непосредственно процесс тестирования, там же выводятся результаты тестирования.

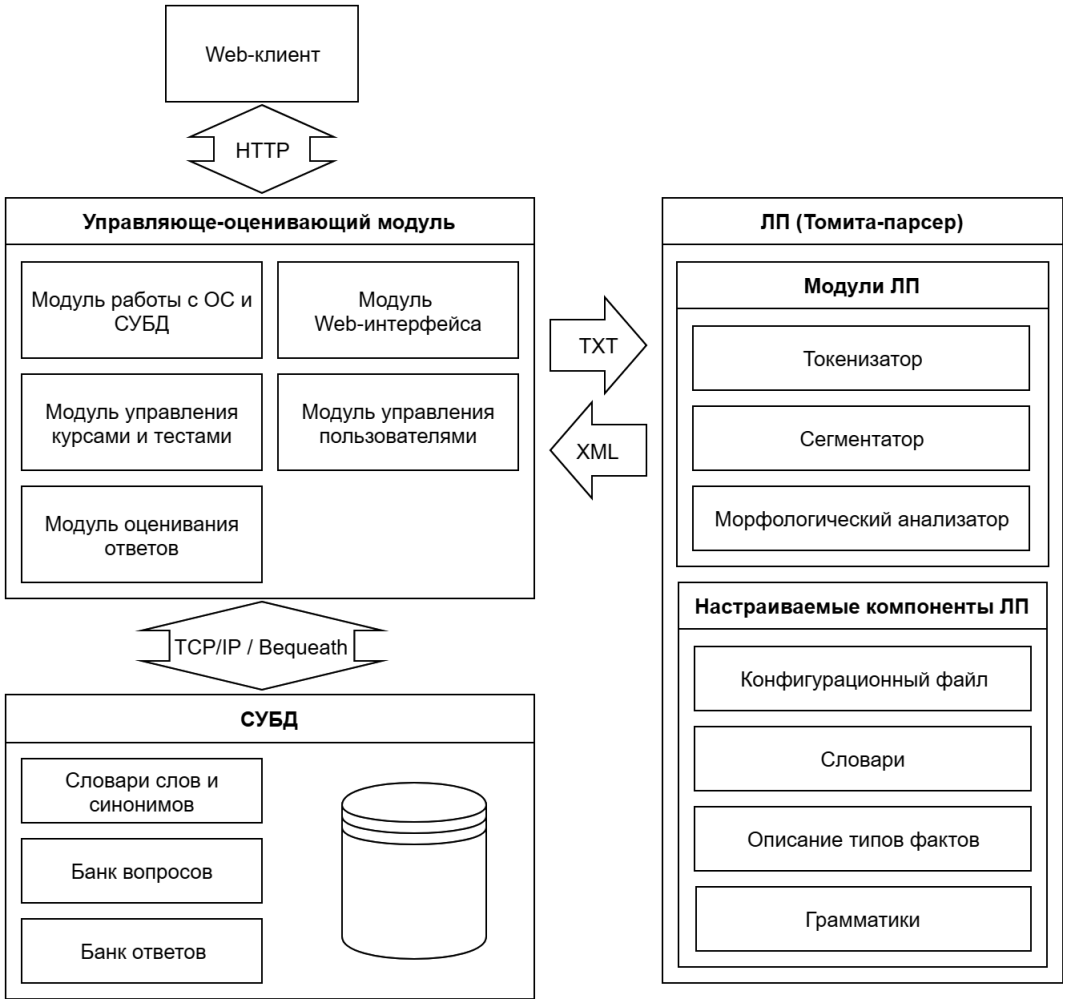

Рис. 1. Функциональная схема системы тестирования

Fig. 1. Functional diagram of the testing system

В БД хранятся вопросы и ответы, а также вспомогательные словари слов и синонимов.

Модуль управления пользователями отвечает за создание пользователей и разделяет их на группы. Модуль управления курсами и тестами создает и редактирует тесты и содержащие их курсы. Задачи модуля работы с ОС и СУБД следующие:

занести в базу данных (БД) вопросы и ответы, а также всю вспомогательную информацию (оценки за ответы, даты сдачи тестов и т. д.);

получить из БД текущий ответ тестируемого и записать его в текстовый файл для анализа Томита-парсером;

найти в БД все эталонные ответы на этот текущий ответ и каждый из них также записать в соответствующий текстовый файл;

запустить Томита-парсер для текущего ответа и всех соответствующих ему эталонных ответов, указав парсеру соответствующие файлы для анализа;

распарсить получившиеся после анализа парсера файлы (формата XML) для передачи фактов модулю оценивания ответов.

Модуль оценивания ответов оценивает ответ тестируемого по алгоритму, описанному в следующем разделе.

БД состоит из 18 таблиц, основные из которых:

- course – таблица, содержащая инфор-

мацию о курсах;

• course_groups – таблица, содержащая информацию о группах пользователей (студентов и преподавателей), записанных на курс;

• course_tests – таблица, содержащая тесты курса;

• question – таблица, содержащая банк вопросов;

• question_answer – таблица, содержащая эталонные ответы на вопросы;

• question_group – таблица, содержащая группы вопросов;

• test – таблица, содержащая тесты;

• test_try – таблица, содержащая попытки сдачи теста;

• user – таблица, содержащая информацию о пользователях;

• words – таблица, содержащая словарь слов;

• word_synonyms – таблица, соединяющая слова с их синонимами (словарь синонимов).

Для правильной работы Томита-парсера нужно создать следующие файлы:

конфигурационный файл;

файл(ы) с описанием типов фактов, которые извлекает парсер;

газеттир(ы) (словарь ключевых слов для парсера, использующихся при анализе грамматиками);

файл(ы) с набором грамматик.

Про факты и грамматики будет сказано в следующем разделе.

Приведем краткий сценарий использования системы преподавателем.

После входа в систему у преподавателя имеется возможность выбора одного из существующих курсов или создания нового. При выборе существующего курса можно зайти в банк вопросов этого курса, составить новый тест или посмотреть ответы на уже существующие тесты для этого курса. Если зайти в банк вопросов, то можно составить новый вопрос (указав иерархическую группу этого вопроса и минимум один эталонный ответ) или написать еще один эталонный ответ на существующий вопрос, или задать веса для фактов существующего эталонного ответа. При составлении теста указывается количество вопросов в тесте, откуда они берутся (случайно или указать конкретные), критерии оценивания (процент правильных ответов для соответствующей оценки за тест). При выборе уже существующего теста появляется возможность его изменить или самостоятельно оценить ответы студентов на вопросы этого теста. Также имеется возможность запустить оценивание теста заново, например, если добавлены новые эталонные ответы к вопросам или новые веса.

Алгоритм анализа ответов в системе

Анализ ответов в системе происходит по следующему алгоритму.

Из базы данных берется очередной ответ тестируемого на какой-нибудь вопрос, он переводится в текстовый файл и подается на вход Томита-парсера. Томита-парсер анализирует его и записывает результаты (список фактов) в XML-файл. Этот XML-файл затем парсится, и получается список фактов в удобном для программы виде (рис. 2).

Томита-парсер позволяет выделять из текста на русском языке сущности, которые называются *фактами*. Выделение происходит с помощью написанных пользователем шаблонов (или грамматических правил), наборы таких правил называют *грамматиками*. Факты можно представить как некие таблицы с колонками, которые называются *полями фактов* (например, факт «собрание» может иметь поля «место», «время» и «тема»).

Вот пример простейшего грамматического правила:

S ->AdjWord<h-reg1>+.

По этому правилу будет выделяться прилагательное или причастие, или порядковое числительное (Adj), после которого следует один или несколько (оператор +) слов из букв русского или латинского алфавита (Word), у каждого из которых первая буква должна стоять в верхнем регистре (<h-reg1>).

Нами было написано 41 грамматическое правило (которые Томита-парсер затем перевела в 138 правил) для извлечения 11 фактов (шести фактов, описывающих

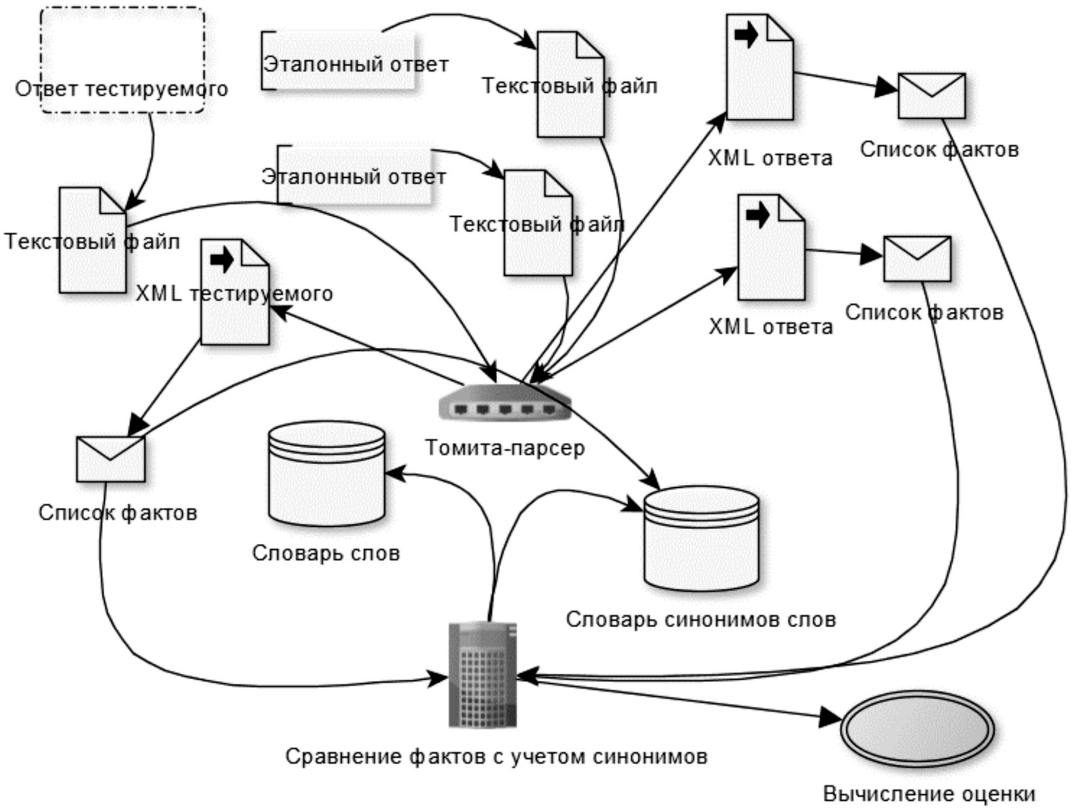

Рис. 2. Схема проверки ответа

Fig. 2. Response verification scheme

части речи (имя существительное, глагол, имя прилагательное, причастие, частица, предлоги и союзы), трех фактов, описывающих члены предложения (подлежащее, сказуемое, дополнение), и двух фактов, описывающих специфичные для предметной области тестирования термины и названия). К сожалению, невозможно заранее придумать полный набор грамматических правил, пригодный для анализа произвольно сколь угодно сложного предложения: чем лингвистически сложнее анализируемое предложение, тем сложнее должны быть соответствующие правила.

По нашим грамматикам системой был получен список списков фактов ответа (*список списков*, т. к. может существовать несколько экземпляров одного и того же факта).

Например, если в предложении четыре существительных и два глагола, то у нас будет список «Существительное» из четырех фактов и список «Глагол» из двух фактов.

Далее, для того же самого вопроса в базе данных находятся все эталонные ответы. Для них проводится такая же процедура по получению списка списков фактов.

Пусть A, B, C, ..., K – извлекаемые нами факты (например, если A – это «Существительное», а B – «Глагол», то для рассмотренного выше предложения из четырех существительных и одного глагола получатся списки $\{A_1, A_2, A_3, A_4\}$ и $\{B_1, B_2\}$).

В результате работы парсера в общем случае у нас есть список списков фактов (обозначим этот список ξ) проверяемого ответа $\{\{A_1, A_2, ..., A_{p1}\}, \{B_1, B_2, ..., B_{q1}\}, \{C_1, C_2, ..., C_{r1}\}, ...\{K_1, K_2, ..., K_{z1}\}\}$ (какие-то из списков фактов могут быть пустыми) и набор списков фактов соответствующих эталонных ответов:

$\{\{A'_1, A'_2, ..., A'_{p2}\}, \{B'_1, B'_2, ..., B'_{q2}\}, \{C'_1, C'_2, ..., C'_{r2}\}, ...\{K'_1, K'_2, ..., K'_{z2}\}\}$ (обозначим этот список α),

{{A''$_1$, A''$_2$, ..., A''$_{p3}$}, {B''$_1$, B''$_2$, ..., B''$_{q3}$}, {C''$_1$, C''$_2$, ..., C''$_{r3}$}, ...{K''$_1$, K''$_2$, ..., K''$_{z3}$}} (обозначим этот список β) и т. д.

Далее запускается алгоритм сравнения: список ξ поочередно сравнивается с каждым списком из набора списков фактов соответствующих эталонных ответов (с α, с β и т. д.).

Сравнение происходит так: в списке ξ для каждого из N фактов нужно найти долю правильности f. Доля правильности считается следующим образом: берется список, соответствующий данному факту (например, {A_1, A_2, ..., A_{p1}}), и для каждого члена этого списка вначале ищется совпадение с членами списка этого же факта в списке α (то есть {A'$_1$, A'$_2$, ..., A'$_{p2}$}). Если совпадения не нашлось, то в словаре синонимов находятся все синонимы для этого члена, и потом определяется, нет ли среди этих синонимов членов списка этого же факта в списке α ({A'$_1$, A'$_2$, ..., A'$_{p2}$}). Каждому члену списков фактов A_i можно сопоставить пару чисел (m, s) (от match и synonym), где $m = 1$, если A_i совпадает с одним из {A'$_1$, A'$_2$, ..., A'$_{p2}$}; $m = 0$, если A_i не совпадает ни с одним из {A'$_1$, A'$_2$, ..., A'$_{p2}$}; $s = 1$, если есть синоним для A_i, совпадающий с одним из {A'$_1$, A'$_2$, ..., A'$_{p2}$}; $s = 0$ – нет синонима для A_i, совпадающего с одним из {A'$_1$, A'$_2$, ..., A'$_{p2}$} (возможные значения пар (m, s) – (1,0), (0,1) и (0,0)).

Суммируем все явные совпадения (Σm) и совпадения с учетом синонимов (Σs) – это будет числитель для f. В знаменателе будет стоять количество членов списка этого же факта в списке α (обозначим это количество через σ – для списка {A'$_1$, A'$_2$, ..., A'$_{p2}$} будет $\sigma = p2$). Таким образом, f будет равно единице, если для каждого члена списка данного факта списка α эталонного ответа найдется либо точное совпадение в списке данного факта списка ξ, либо синоним. В общем случае, f всегда принимает значения в интервале [0;1] (естественно, мы обрабатываем случаи, когда данного факта вообще нет в эталонном ответе: когда числитель равен нулю, или когда тестируемый ввел несколько синонимов для одного факта в своем ответе, и знаменатель стал больше числителя).

Запишем математически формулу для доли правдивости данного i-го факта:

$$f_i = \frac{\sum_k m_k + \sum_k s_k}{\sigma}. \quad (1)$$

Приведем простейший пример. Пусть есть вопрос «Как изменится на фотографии вид полной Луны, если закрыть правую половину объектива телескопа?» и один из эталонных ответов «Изображение не изменится, фотография станет менее яркой». Система оценивала ответ студента «Вид не поменяется, фото будет тусклее».

Парсер извлек список фактов (это список ξ) для проверяемого ответа {«Существительное» {«вид», «фото»}, «Глагол» {«поменяться», «будет»}, «Прилагательное» {«тусклый»} и т. д.} и список фактов (это список α) для эталонного ответа {«Существительное» {«изображение», «фотография»}, «Глагол» {«измениться», «стать»}, «Прилагательное» {«яркая»} и т. д.}.

Проверяем список факта «Существительное» – для слова «вид» совпадения в списке α нет, но в словаре синонимов есть информация, что «вид» и «изображение» – это синонимы, поэтому для этого слова будем иметь $m = 0$ и $s = 1$. Аналогично, для слова «фото» нет точного совпадения, но есть синоним «фотография», поэтому также $m = 0$ и $s = 1$. Таким образом, доля правдивости факта «Существительное» в списке ξ для списка α будет равна $f = (1+1)/2 = 1$. Далее проверяем список для факта «Прилагательное» (в данном случае список состоит из одного элемента и в ξ, и в α) – для слова «тусклый» $m = 0$ и $s = 1$ и доля правдивости факта «Прилагательное» в списке ξ для списка α будет равна $f = 1/1 = 1$. Такая проверка происходит для всех фактов в списке ξ.

При задании эталонного ответа каждому i-му факту можно приписать свой вес w_i (для каждого ответа веса задаются независимо). Далее считается оценка – суммируются все веса (Σw_i) для всех фактов для данного эталонного ответа (это будет нормировочный знаменатель) и суммируется произведение веса факта на его долю прав-

дивости (это числитель), формулой оценки ответа $mark_j$ при сравнении с данным j-м эталонным ответом будет:

$$mark_j = \frac{\sum_{i=1}^{N} w_i f_i}{\sum_{i=1}^{N} w_i}. \quad (2)$$

Оценка ответа также лежит в интервале [0;1]. А далее нужно проделать точно такую же процедуру для следующего эталонного ответа (для списка β) и т. д. Итоговая оценка будет максимальной из всех оценок для всех эталонных ответов (ее можно умножить на 100 % для наглядности) – это следующая формула:

$$mark = \max\{mark_j\}. \quad (3)$$

Для иллюстрации формул (2), (3) предположим, что преподаватель дал еще один эталонный ответ «Луна останется полной», и что он для обоих эталонных ответов приписал веса, равные единице, для фактов «Существительное» и «Прилагательное», и равные нулю для всех остальных фактов. По формуле (2) ответ студента по сравнению с первым эталонным ответом дает оценку $mark_1 = (1 \cdot 1 + 1 \cdot 1) / (1 + 1) = 1$, а со вторым эталонным ответом (поскольку доли правдивости фактов «Существительное» и «Прилагательное» теперь равны нулю) дает оценку $mark_2 = (1 \cdot 0 + 1 \cdot 0) / (1 + 1) = 0$. В результате по формуле (3) ответ студента будет оценен на 100 %.

Реализация и тестирование системы

При реализации системы в качестве СУБД была выбрана СУБД Oracle Database 12c (но тестировалась система и на версии 11g). Все модули, кроме web-интерфейса, были написаны на языке PL/SQL, а для последнего была использована Java (с HTML/CSS). Для тестирования взяли ответы студентов на вопросы по дисциплинам «Администрирование СУБД Oracle» и «Физика». Использование вопросов из столь разных проблемных областей позволило более широко проанализировать поведение системы.

Для системы применялся словарь общеупотребительных слов, который мы пополнили узкоспециализированными терминами по физике и администрированию СУБД Oracle. В основу этого словаря положены готовые бесплатные словари, найденные в Интернете (например, [20]). На момент тестирования словарь состоял примерно из 200 000 слов. Практически каждое слово из этого словаря было и в словаре синонимов размером около 440 000 строк.

Для тестирования системы было собрано и проверено (независимо системой и преподавателем) 1445 ответов студентов 1, 2 и 4 курсов. Отметим лингвистическое разнообразие ответов: на некоторые вопросы тестируемые давали ответ, состоящий из одного слова, на другие – из более 50 слов. Примеры вопросов и ответов приведены в табл. 1. Результаты тестирования даны в табл. 2. Указан процент ответов, которые система и преподаватель оценивали как правильные или неправильные.

Видно, что результаты системы очень хорошие: корректное определение – 88,10 %, некорректное определение – 11,90 %.

Также количественной мерой результатов согласия оценивания системы и преподавателя может служить коэффициент согласия каппа Коэна, который очень часто используется в работах по системам автоматической проверки [21]. Для вычисления коэффициента согласия каппа на основе полученных результатов может быть составлена таблица сопряженности (табл. 3).

Тогда коэффициент каппа будет считаться по формуле:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \quad (4)$$

где p_o – наблюдаемая (чистая) согласованность; p_e – случайная согласованность, которые для нашей таблицы сопряженности вычисляются по формулам:

$$p_o = \frac{747 + 526}{747 + 146 + 26 + 526};$$

$$p_e = \frac{(747 + 26) \cdot (747 + 146) +}{(747 + 146 + 26 + 526) \times}$$
$$\frac{+ (526 + 26) \cdot (526 + 146)}{\times (747 + 146 + 26 + 526)}.$$

Результаты вычислений дают для

Таблица 1

Примеры вопросов и ответов

Table 1

Examples of questions and answers

Категории ответов	Вопрос	Эталонный ответ	Ответ студента
Ответы, которые преподаватель оценил как правильные, и система тоже оценила как правильные	Кому доступны данные, добавленные во временную таблицу в течение сессии?	Пользователю сессии	Пользователю, создавшему эту сессию
	Как изменится на фотографии вид полной Луны, если закрыть правую половину объектива телескопа?	*а*) Изображение не изменится, станет менее ярким. *б*) Луна останется полной	Вид не поменяется, фото будет тусклее
	Что изучает классическая механика?	*а*) Медленные движения макроскопических тел. *б*) Движения макроскопических тел при скорости, много меньшей скорости света	Движение макроскопических тел со скоростями малыми, по сравнению со скоростью света
Ответы, которые преподаватель оценил как неправильные, а система оценила как правильные	При дифракции Френеля на круглом отверстии дифракционная картина будет иметь вид чередующихся светлых и темных концентрических колец, в центре которой будет светлое пятно, если отверстие открывает...	Четное число зон Френеля	Нечетное число зон Френеля
Ответы, которые преподаватель оценил как правильные, а система оценила как неправильные	Поясните смысл нижеприведенного состояния ограничения: enable novalidate	Будет проводиться проверка новых данных, а введенные ранее данные останутся без проверки	Ограничение активируется без проверки старых данных
	Что изучает классическая механика?	*а*) Медленные движения макроскопических тел. *б*) Движения макроскопических тел при скорости, много меньшей скорости света	Изучает законы движения макроскопических тел, скорости которых малы по сравнению со скоростью света в вакууме

Ответы, которые преподаватель оценил как неправильные, и система тоже оценила как неправильные	Что такое кластер? Для чего он используется?	Кластер – это способ хранения в одном сегменте группы таблиц, имеющих один или более общих столбцов. Используется для ускорения доступа к совместно используемым данным из нескольких таблиц	Вид таблиц, хранящихся в одном сегменте
	Что получится, если от кинетической энергии системы материальных точек отнять кинетическую энергию той же системы в ее относительном движении по отношению к поступательно движущейся системе координат с началом в центре масс?	Кинетическая энергия всей массы системы, сосредоточенной в ее центре масс	Кинетическая энергия движущейся системы

Таблица 2

Результаты тестирования

Table 2

Test results

Категории оценки ответов	Общее количество (%)
Ответы, которые преподаватель оценил как правильные, и система тоже оценила как правильные	747 (51,70)
Ответы, которые преподаватель оценил как неправильные, а система оценила как правильные	26 (1,80)
Ответы, которые преподаватель оценил как правильные, а система оценила как неправильные	146 (10,10)
Ответы, которые преподаватель оценил как неправильные, и система тоже оценила как неправильные	526 (36,40)

Таблица 3

Таблица сопряженности

Table 3

Contingency Table

		Система	
		Ответ верен	Ответ неверен
Преподаватель	Ответ верен	747	146
	Ответ неверен	26	526

коэффициента каппа значение 0,76, что считается достаточно высоким результатом. Например, у описанных выше систем c-rater [6] и SAMText [10] средние значения коэффициентов каппа равны соответственно 0,74 и 0,73, а у лучших работ ASAP SAS [14] в 2012 г. коэффициент каппа был от 0,73 до 0,75.

Скорость оценивания системы намного превосходит скорость оценивания преподавателем: на оценивание одного теста с 30 ответами уходит времени порядка 40 с. При этом, в отличие от преподавателя, система может без устали проверить сколько угодно тестов.

Анализ результатов тестирования показал, что система отлично справляется с короткими ответами (в смысле совпадения результатов оценки системы и преподавателя, если и исходный эталонный ответ не велик), но испытывает некоторые сложности при оценивании больших (порядка 40–50 слов) предложений.

Существует несколько возможностей по совершенствованию системы:

1. Экстенсивные – добавление новых грамматик извлечения фактов и добавление новых слов в словари слов и синонимов.

2. Добавление в словари слов и синонимов грамматически ошибочных версий слов (например, «програма», «праграмма» наряду с «программа»). Это позволит обрабатывать логически правильные ответы, в которых сделаны описки и грамматические ошибки.

3. Наряду с системой весов разных фактов, можно ввести систему проверки ключевых слов. Например, автор вопроса может добавить к ответу список ключевых слов. Тогда система начнет анализ ответа с этого списка – если ни одного из ключевых слов (или их синонимов) в ответе не найдено, то ответ сразу можно считать неправильным.

Заключение

В статье предложена функциональная схема системы автоматического оценивания ответов на открытые вопросы на русском языке. Для лингвистического процессора системы, относящегося к категории извлечения информации, предложен алгоритм анализа ответов. Введена возможность настройки системы при оценивании каждого ответа путем задания весов для каждой сущности и для каждого из имеющихся эталонных ответов.

Система была реализована и протестирована. По оцениванию результатов тестов по дисциплинам «Администрирование СУБД Oracle» и «Физика» можно заключить, что полученная система удовлетворяет предъявленным к ней требованиям и выполняет проверку ответов по крайней мере в данных предметных областях, с высокой степенью согласия с оцениванием преподавателем. Получившаяся система достаточно универсальна и может использоваться для тестирования по любым предметам.

В настоящее время продолжается работа по дальнейшему усовершенствованию системы.

Кроме реализации описанных выше возможностей по совершенствованию, для поиска синонимов предлагается использовать систему Word2vec, обученную на русской Wikipedia и Национальном корпусе русского языка. Word2vec – программа для анализа семантики естественных языков, и использование больших корпусов русского языка представляется перспективным для дополнительного поиска семантических синонимов.

СПИСОК ЛИТЕРАТУРЫ

1. **Burrows S., Gurevych I., Stein B.** The eras and trends of automatic short answer grading // Internat. Journal of Artificial Intelligence in Education. 2015. Vol. 25. Pp. 60–117.

2. **Mödritscher F., Sindler A.** Quizzes are not enough to reach high-level learning objectives! // Proc. of the World Conf. on Educational Multimedia, Hypermedia and Telecommunications 2005. Montreal, Canada, 2005. Pp. 3275–3278.

3. **Burstein J., Kaplan R., Wolff S., Lu C.** Using lexical semantic techniques to classify free-responses // Proc. of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons. 1996. Pp. 20–29.

4. **Dorr B., Hendler J., Blanksteen S., Migdaloff B.** On beyond syntax: use of lexical conceptual structure for intelligent tutoring // Intelligent Language Tutors. Mahwah, New Jersey: Lawrence Erlbaum Publishers, 1995. Pp 289–311.

5. **Callear D., Jerrams-Smith J., Soh V.** CAA of Short Non-MCQ Answers // Proc. of the 5th Computer Assisted Assessment Conf. Loughborough, United Kingdom, 2001. Pp 1–14.

6. **Leacock C., Chodorow M.** C-rater: automated scoring of short-answer questions // Computers and the Humanities. 2003. No. 37(4). Pp. 389–405.

7. **Mitchell T., Russell T., Broomhead P., Aldridge N.** Towards robust computerised marking of free-text responses // Proc. of the 6th Computer Assisted Assessment Conf. 2002. Pp. 233–249.

8. **Bachman L.F., Carr N., Kamei G., Kim M., Pan M.J., Salvador C., Sawaki Y.** A reliable approach to automatic assessment of short answer free responses // Proc. of the 19th Internat. Conf. on Computational Linguistics. Taipei, 2002. Pp. 1–4.

9. **Alfonseca E., Perez D.** Automatic assessment of open ended questions with a BLEU-Inspired algorithm and shallow NLP // Lecture Notes in Computer Science. Advances in Natural Language Processing. 2004. Vol. 3230. Pp. 25–35.

10. **Bukai O., Pokorny R., Haynes J.** An automated short-free-text scoring system: development and assessment // Proc. of the 20th Interservice. Industry Training, Simulation, and Education Conf. National Training and Simulation Association. 2006. Pp 1–11.

11. **Gütl C.** e-Examiner: Towards a fully-automatic knowledge assessment tool applicable in adaptive E-Learning systems // Proc. of the 2nd Internat. Conf. on Interactive Mobile and Computer Aided Learning. Amman, Jordan, 2007. Pp. 1–10.

12. **Lin C.-Y.** ROUGE: A Package for automatic evaluation of summaries // Proc. of the 1st Text Summarization Branches out Workshop at ACL. Barcelona, Spain, 2004. Pp. 74–81.

13. **Horbach A., Palmer A., Pinkal M.** Using the text to evaluate short answers for reading comprehension exercises // Proc. of the 2nd Joint Conf. on Lexical and Computational Semantics. Atlanta, USA: Association for Computational Linguistics, 2013. Vol. 1. Pp. 286–295.

14. Automated Student Assessment Prize. The Hewlett Foundation: Short Answer Scoring // URL: https://www.kaggle.com/c/asap-sas

15. Automated Student Assessment Prize. The Hewlett Foundation: Automated Essay Scoring // URL: https://www.kaggle.com/c/asap-aes

16. Past TAC Data // URL: https://tac.nist.gov//data/

17. Портал дистанционных образовательных технологий СПбПУ // URL: https://dl.spbstu.ru/

18. **Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В.** Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. М.: МИЭМ, 2011.

19. Томита-парсер. Документация. Руководство разработчика // URL: https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage/

20. MySQL БД синонимов русского языка // URL: https://www.mindcollapse.com/blog/177.html

21. Cohen'skappa // URL: https://en.wikipedia.org/wiki/Cohen's_kappa.

Статья поступила в редакцию 31.03.2017.

REFERENCES

1. **Burrows S., Gurevych I., Stein B.** The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 2015, Vol. 25, Pp. 60–117.

2. **Mödritscher F., Sindler A.** Quizzes are not enough to reach high-level learning objectives! *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005,* Montreal, Canada, 2005, Pp. 3275–3278.

3. **Burstein J., Kaplan R., Wolff S., Lu C.** Using lexical semantic techniques to classify free-responses. *Proceedings of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons*, 1996, Pp. 20–29.

4. **Dorr B., Hendler J., Blanksteen S., Migdaloff B.** On beyond syntax: use of lexical conceptual structure for intelligent tutoring. *Intelligent Language Tutors.* Mahwah, New Jersey: Lawrence Erlbaum Publishers, 1995, Pp 289–311.

5. **Callear D., Jerrams-Smith J., Soh V.** CAA of Short Non-MCQ Answers. *Proceedings of the 5th Computer Assisted Assessment Conference,* Loughborough, United Kingdom, 2001, Pp 1–14.

6. **Leacock C., Chodorow M.** C-rater: automated scoring of short-answer questions. *Computers and the Humanities*, 2003, No. 37(4), Pp. 389–405.

7. **Mitchell T., Russell T., Broomhead P., Aldridge N.** Towards robust computerised marking of free-text responses. *Proceedings of the 6th Computer Assisted Assessment Conference*, 2002, Pp. 233–249.

8. **Bachman L.F., Carr N., Kamei G., Kim M., Pan M.J., Salvador C., Sawaki Y.** A reliable approach to automatic assessment of short answer free responses. *Proceedings of the 19th International Conference on Computational Linguistics.* Taipei, 2002, Pp. 1–4.

9. **Alfonseca E., Perez D.** Automatic assessment of open ended questions with a BLEU-Inspired

algorithm and shallow NLP. *Lecture Notes in Computer Science. Advances in Natural Language Processing*, 2004, Vol. 3230, Pp. 25–35.

10. **Bukai O., Pokorny R., Haynes J.** An automated short-free-text scoring system: development and assessment. *Proceedings of the 20th Interservice. Industry Training, Simulation, and Education Conference. National Training and Simulation Association*, 2006, Pp 1–11.

11. **Gütl C.** e-Examiner: Towards a fully-automatic knowledge assessment tool applicable in adaptive E-Learning systems. *Proceedings of the Second International Conference on Interactive Mobile and Computer Aided Learning.* Amman, Jordan, 2007, Pp. 1–10.

12. **Lin C.-Y.** ROUGE: A package for automatic evaluation of summaries. *Proceedings of the 1st Text Summarization Branches out Workshop at ACL.* Barcelona, Spain, 2004, Pp. 74–81.

13. **Horbach A., Palmer A., Pinkal M.** Using the text to evaluate short answers for reading comprehension exercises. *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics.* Atlanta, USA: Association for Computational Linguistics, 2013, Vol. 1, Pp. 286–295.

14. Automated Student Assessment Prize. The Hewlett Foundation: Short Answer Scoring. Available: *https://www.kaggle.com/c/asap-sas*

15. Automated Student Assessment Prize. The Hewlett Foundation: Automated Essay Scoring. Available: *https://www.kaggle.com/c/asap-aes*

16. Past TAC Data. Available: *https://tac.nist.gov//data/*

17. Portal of distance learning technologies SPbPU. Available: *https://dl.spbstu.ru/*

18. **Bolshakova Ye.I., Klyshinskiy E.S., Lande D.V., Noskov A.A., Peskova O.V., Yagunova Ye.V.** *Avtomaticheskaya obrabotka tekstov na yestestvennom yazyke i kompyuternaya lingvistika* [*Automatic Natural Language Processing and Computational Linguistics*]. Moscow: MIEM Publ., 2011. (rus)

19. Tomita parser Documentation. Developer's Guide. Available: *https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage/*

20. MySQL database of Russian synonyms. Available: *https://www.mindcollapse.com/blog/177.html*

21. Cohen's kappa. Available: *https://en.wikipedia.org/wiki/Cohen's_kappa.*

Received 31.03.2017.

СВЕДЕНИЯ ОБ АВТОРАХ / THE AUTHORS

КОЖЕВНИКОВ Вадим Андреевич
KOZHEVNIKOV Vadim A.
E-mail: vadim.kozhevnikov@gmail.com

САБИНИН Олег Юрьевич
SABININ Oleg Yu.
E-mail: olegsabinin@mail.ru