УДК 004.8

ПРИМЕНЕНИЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ ДЛЯ ЦИФРОВОЙ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ

Жаткина Кристина Николаевна[1], Крейдер Оксана Александровна[2]

[1]Студент;
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: zhatkina-96@mail.ru.

[2]Кандидат технических наук, доцент;
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: kreider.oksana@gmail.com.

В данной статье описывается возможность применения методов интеллектуального анализа данных с целью создания цифровой образовательной платформы. Под понятием цифровой образовательной платформой следует понимать системы организации цифрового взаимодействия производителей и потребителей образовательных услуг, открытую для присоединения новых участников. Для реализации цифровой образовательной платформы предлагается создание агента, который собирает необходимую информацию с сайтов, а также выбор и апробация архитектуры нейронной сети для построения индивидуальной траектории обучающегося с помощью компетентностной модели.

Ключевые слова: агент, нейронная сеть, цифровая образовательная платформа.

APPLICATION OF INTELLECTUAL DATA ANALYSIS METHODS FOR DIGITAL EDUCATIONAL PLATFORM

Zhatkina Kristina[1], Kreider Oksana[2]

[1]Student;
Dubna State University,
Institute of the system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: zhatkina-96@mail.ru.

[2]Candidate of Science in Engineering, associate professor;
Dubna State University,
Institute of the system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: kreider.oksana@gmail.com.

This article describes the possibility of using data mining techniques. In order to join new carpet participants, it is necessary to understand that the system of interaction with them is public educational services. To implement digital educational platforms, it is proposed to create an agent that collects information about sites, and also selects and tests the architecture of the neural network to build an individual trajectory that is trained using the competency-based model.

Keywords: agent, neural network, digital educational platform.

В настоящее время существует множество электронных образовательных платформ, которые предлагают различные учебные курсы по разным направлениям. В связи с возрастающим интересом к электронному обучению, вызванному, в частности, последними событиями всеобщей самоизоляции всё больше и больше появляется открытых бесплатных курсов на образовательных платформах. Любой пользователь может найти необходимый ему курс вне зависимости от уровня подготовки в той или иной области. Зачастую один и тот же пользователь зарегистрирован на нескольких платформах и

одновременно может проходить обучение по нескольким курсам. При анализе популярных образовательных платформ (образовательная онлайн-платформа «*Stepik*», образовательный проект «Лекториум», массовый образовательный проект «*Coursera*», сетевая межуниверситетская площадка «Универсариум», образовательная платформа «Открытое образование», образовательный портал «*Geekbrains*», международная площадка открытого образования «*EdX*», выбор образовательных площадок производился на основании рейтингов агенств *arriva, raex*, а также рейтинговых статьях на сайтах *rbc*) выявлена общая часть всех курсов: название, описание, продолжительность обучения, структура и т.п. В этой связи целесообразно проанализировать возможности интеграции курсов с разных образовательных платформ в одном информационном пространстве для удобства пользователя.

Реализация данного проекта может быть полезна для следующих категорий пользователей:

1. Пользователи образовательных ресурсов. Как правило, один пользователь может быть зарегистрирован на нескольких образовательных платформа и проходить на каждой из них одновременно не один курс. Многие образовательные ресурсы имеют временное ограничение на прохождение курсов, из-за этого другие курсы могут быть отложены пользователем на потом и со временем затеряться. Единое хранилище этих курсов помогло бы решить эту проблему и другие, такие как: необходимость запоминать данные учетных записей или какие аккаунты, где используются для авторизации; переход между платформами для прохождения курсов; поиск интересующих курсов по различным площадкам и т. д.

2. Электронные образовательные ресурсы. Главное, в чем заинтересованы все образовательные платформы – это пользователи, как привлечение новых, так и поддержание интереса действующих. Так как единая цифровая образовательная платформа не предполагает возможность выдачи курсов за собственные, а служит лишь для интеграции существующих с сохранением данных об авторах и площадках, это позволит поднять уровень популярность образовательных ресурсов, привлечь новых пользователей и оптимизировать действия действующих пользователей.

3. Государство и ведомства. Ни для кого не секрет, что государство в настоящее время очень интересуют данные о действиях пользователей в сети Интернет. Цифровые следы, которые оставляют пользователи образовательных платформ так же важны для корректировки и оптимизации образовательного процесса как очного, так и дистанционного. В целом большое внимание уделяется возможности создания индивидуальной образовательной траектории для пользователей с целью более эффективного и углубленного изучения интересующей сферы деятельности.

Таким образом, можно отметить плюсы разработки единой цифровой образовательной платформы для пользователей сети Интернет.

Одним из вариантов реализации интеграции курсов на единой образовательной платформе является создание агента [1], который собирает необходимую информацию с образовательных платформ, таких как: *coursera, stepik* и т. д. Агентами принято считать программный код, заточенный под парсинг сайта с целью получения информации. Парсинг в свою очередь представляет собой процесс сборки и систематизации информации, с помощью специальных программ (агентов), автоматизирующих данные действия. Следует отметить, что в парсинге сайтов нет ничего противозаконного, однако большинство сайтов настраивают дополнительную защиту от автоматизированных действий агента. Однако, собирать можно лишь открытую информацию, которую при необходимости или желании можно собрать вручную, но из-за большого объема таких данных логичным является автоматизация процесса.

Для парсинга данных можно выбрать один из двух форматов:

- воспользоваться специальными программами, которых на рынке существует немало;
- написать их самостоятельно. Для этого может применяться практически любой язык программирования, например, *PHP, C++, Python* и другие. Именно этот вариант был выбран.

Если требуется не вся информация по странице, а только что-то определенное (наименования курса, описание, ссылка на материалы), используется *XPath*.

XPath – это язык, на котором формируются запросы к *XML*-документам и их отдельным элементам. С помощью его команд необходимо определить границы будущего агента, то есть задать как парсить данные с сайта – полностью или выборочно.

Для реализации агента были подробно изучены структурные компоненты образовательных платформ, хранящих в себе данные о курсах. Выходные файлы агента состоят из двух типов: *json* для

дальнейшего размещения информации о курсах на платформах и *csv* для подачи на вход нейронной сети. Оба типа файлов хранят в себе идентичную информацию о курсах: название курса (*name*), краткое описание курса (*description*), ссылка на уроки курса (*url*). Агент реализован на языке *Python*. При реализации агента использовались настройки *proxy* и добавления *user-agent*. При разработке агента использовались библиотеки: *selenium* для отправки запроса на сайт; *time* для тайм-менеджмента, чтобы парсинг сайта не был воспринят как атака; *lxml.html* для работы со структурой *HTML*-документа, *BeautifulSoup* для удобства извлечения нужных данных.

Разработан полный набор операций внутри одного обрабатываемого сайта. На рис. 1 можно увидеть последовательность и цикличность работы с выбранным сайтом с учетом всех переходов между страницами и сбором необходимых данных. Изменение входных параметров по заданной структуре поможет масштабировать программный код и адаптировать его под любой образовательный ресурс.

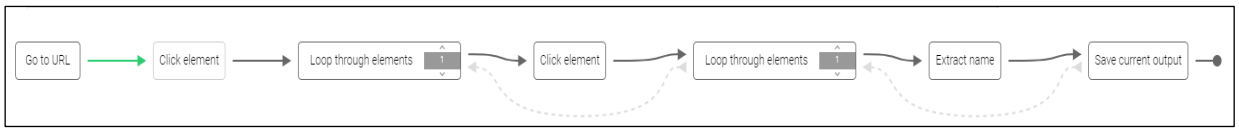

Рис. 1. Общий алгоритм сбора данных с сайта

Процесс организации структуры внутри проекта представлен на рис. 2. После сохранения выходных файлов алгоритм возвращается к верхнему уровню циклов и так повторяется пока все элементы на сайте не будут просмотрены.

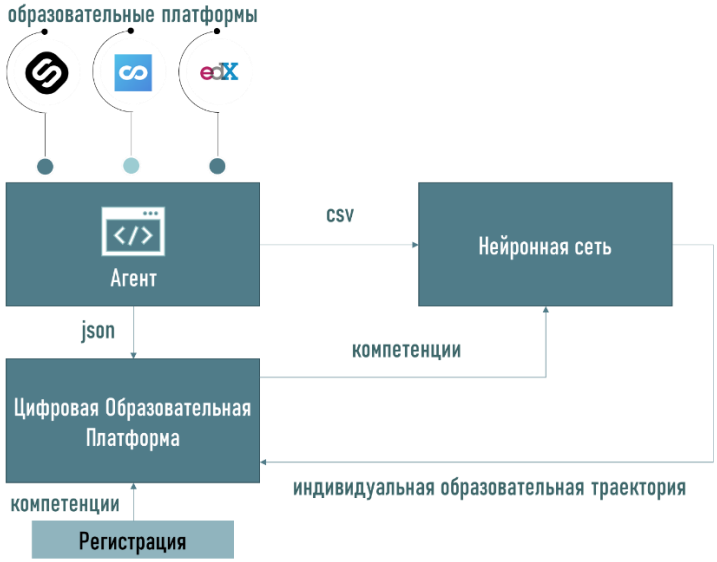

Рис. 2. Схема взаимодействия методов с цифровой образовательной платформой

Для реализации интеграции курсов выполнен этап классификации их по тем или иным направлениям и темам. Интеграция курсов на единой образовательной платформе позволяет рассмотреть возможность построения индивидуальной траектории обучающегося. Этот этап работы включает выбор и апробацию архитектуры нейронной сети. Для проведения анализа архитектур нейронных сетей был выбран новостной *Dataset* от *Keras*, имеющий схожу структуру с данными о курсах, а именно: название (заголовок), описание, класс (выбранные данные уже размечены).

Входными данными для нейронной сети при анализе текста является вектор – текст в виде чисел (применяется метод векторизации). Затем использован метод токенизации – удаление функциональных слов (семантически нейтральных слов, таких как союзы, предлоги, артикли и пр. Для токенизации использовался токен от *Keras*). Далее осуществляется морфологический анализ (производятся разметка по частям речи и стемматизация), что позволяет значительно сократить размерность пространства. Также применяется метод извлечения признаков из текста «мешок слов» (*bag of words*, множество всех слов). В нейронных сетях плотное векторное представление слов (каждому токену сопоставляется вектор, размерность вектора ниже, чем у *one hot encoding*) определяется в процессе обучения. На

первом этапе элементы векторов инициализируются случайными числами, а изменение значений векторов происходит с помощью метода обратного распространения ошибки. Как итог, подготовка данных к подаче на вход нейронной сети является самым долгим и трудозатратным процессом.

Выбор архитектур производился на основании изученных научных статей, и рекомендуемым типом нейронных сетей для анализа текста являются рекуррентные нейронные сети (сети с циклом) [2]. Однако из-за минусов рекуррентных сетей (таких как длительный период обучения, исчезающий градиент, ограниченная «длительность» запоминания предыдущей информации) используются более совершенные архитектуры, а именно: *LSTM* (*Long Short-Term Memory* – долгая краткосрочная память) и *GRU* (*Gated Recurrent Unit* – закрытый рекуррентный блок) [3], основанная на тех же принципах, что и *LSTM*, но использует меньше фильтров и операций для вычисления [4, 5], а также сверточная нейронная сеть. Подбор оптимальных параметров производился с помощью *GridSearchCV* от *sklearn*. Основными были два критерия: минимальная сложность архитектуры нейронной сети при максимальной точности распознавания.

После анализа трех архитектур (*LSTM, GRU* и сверточная нейронная сеть) была выбрана архитектура *GRU*. В связи с тем, что изначальные архитектуры, выбранные для анализа, являлись самыми простыми, необходима доработка выбранной архитектуры. Первым этапом доработки архитектуры стало добавление дополнительных сверточных слоев и тестирование на возможность переобучения. При увеличении количества *batch* и *epoch* происходил процесс переобучения (*overtraining*), что реализовано добавлением слоя *dropout*. Устройство этого слоя позволяет отключать произвольные нейроны на каждом скрытом слое, уменьшая тем самым возможность переобучения. После этого были изучены возможности применения *transfer learning* – это проблема исследования в машинном обучении, которая сосредоточена на хранении знаний, полученных при решении одной проблемы, и применении их к другой, но связанной проблеме и *triplet loss* – это функция потерь для искусственных нейронных сетей, где базовый (якорный) вход сравнивается с положительным (достоверным) входом и отрицательным (ложным) входом, для улучшения точности классификации и уменьшения функции потерь соответственно.

После доработки архитектуры нейронной сети был выбран размеченный датасет с сайта *Kaggle*, который имеет следующую структуру: *course_id, course_title, url, is_paid, price, num_subscribers, Дnum_reviews, num_lectures, level, content_duration, published_timestamp, subject*. Для обучения нейронной сети задействована выборка, состоящая из *course_title, url, subject*, так как эти атрибуты максимально приближены к данным, полученным после работы агента (выходной файл самого агента не использовался для обучения нейронной сети, так как не был размечен). Такие параметры помогут наиболее полно отразить возможность составления компетентностной модели для студентов и использованы в учебном процессе университета «Дубна». Для внедрения возможностей выбранной архитектуры нейронной сети необходимо, чтобы датасет для ее обучения был размечен специалистами.

В результате проделанной работы изучены, подобраны и апробированы методы интеллектуального анализа данных и определено их место и роль для применения в области образования. А также изучены возможности применения нейронных сетей и агентов в электронном обучении и выбраны инструменты реализации цифровой образовательной среды на основе методов интеллектуального анализа данных. Выбранные методы интеллектуального анализа данных апробированы на конкретных площадках и датасетах, и могут быть использованы для создания единой цифровой образовательной платформы, в части интеграции в едином информационном пространстве электронных образовательных ресурсов и построения компетентностной модели обучающегося за счет обработки данных через нейронную сеть.

В рамках реализации агента, позволяющего собирать информацию о курсах с популярных образовательных платформ, определена архитектура нейронной сети *GRU* для анализа данных о курсах с целью формирования предложений пользователю по перечню учебных курсов, соответствующих запросу, предпочтениям и заданным компетенциям. Такая возможность позволяет построить индивидуальную траекторию обучения и может быть внедрена в учебный процесс университета «Дубна». Проведенные исследования имеют дальнейшие перспективы по изучению вопросов построения единого образовательного пространства и индивидуальной образовательной траектории.

Список литературы

1. IBM Cloud Application Performance Management. Конфигурирование агента Python agent. — 2019.
2. Рекуррентные нейронные сети (RNN) с Keras / Общество IT специалистов Habr. — [Электронный ресурс]. URL: https://habr.com/ru/post/487808/, свободный (дата обращения: 18.06.2020).
3. GRU и LSTM: современные рекуррентные нейронные сети // Научный журнал «Молодой Ученый». — [Электронный ресурс]. URL: https://moluch.ru/archive/95/21426/, свободный (дата обращения: 18.06.2020).
4. Николенко С. Глубокое обучение. Погружение в мир нейронных сетей. — 2018. — С. 480.
5. Черняк Е. Глубинное обучение в обработке и анализе текстов // Технологии. 2019.