

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПРОКТОРИНГА ДЛЯ ПЛАТФОРМЫ GOOGLE ФОРМЫ

А.Л. Голованов

аспирант, e-mail: agolovanov@yahoo.com

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В работе представлено описание разработанных автором программных модулей для организации системы автоматизированного прокторинга при онлайн-тестировании. Приводится анализ созданного с помощью разработанных программ датасета о прохождении тестирований и описание использованных методов для автоматического определения нечестности прохождения тестирования.

Ключевые слова: онлайн прокторинг, программное обеспечение, сбор данных, анализ данных.

Введение

В современном мире в связи с широким развитием скорости и доступности Интернета ему находят всё новые применения. Сейчас всё чаще различные мероприятия, такие как лекции, встречи, выступления, практические занятия, проходят онлайн. Такими же привычными стали тестирования, собеседования, соревнования по программированию, проходящие удалённо.

Однако в удалённом формате большой проблемой оказывается проверка того, выполнял ли тестируемый задания самостоятельно, не пользовался ли он посторонним материалом. Наблюдение за тестируемым во время выполнения заданий не даёт гарантий того, что попыток списывания не будет или того, что человек сможет их заметить. Также такое наблюдение может оказывать психологическое давление на сдающего, что может сказаться на итоговом результате работы [1]. Проверка записанного сеанса сдачи в свою очередь влечёт за собой большие затраты по времени и также имеет человеческий фактор. Одним из вариантов контроля, который поможет избежать озвученных проблем, является некоторая автоматизированная система прокторинга.

Настоящая работа посвящена реализации нескольких программных модулей, предназначенных для отслеживания и сбора данных о действиях пользователя во время онлайн-теста и последующий их анализ для определения вероятности списывания.

В качестве платформы для проведения тестов в работе используется сервис «Google Формы». Такой выбор обусловлен сразу несколькими его преимуществами:

- **Обширная функциональность.** Несмотря на название, Google Формы предоставляют широкие возможности для кастомизации тестовых заданий. Тест может содержать поля для краткого и подробного письменного ответа, вопросы с выбором одного или нескольких правильных ответов, сетку множественного выбора ответа, поля для загрузки файлов. Также тестовое задание может включать в себя изображения и видео, которые могут использоваться для формулировки вопроса, есть возможность создать тест из нескольких разделов. Администраторская функциональность также включает возможность делать тест доступным только на определённый период времени, дать или запретить возможность повторных попыток, просматривать присланные ответы и получать статистические данные по ним;
- **Регистрация.** Для создания форм достаточно иметь Google-аккаунт, а заполнять их можно даже без аккаунта;
- Google не взимает никакой оплаты за проведение тестирования на их платформе и не показывает какой-либо рекламы;

В дальнейшем, к разработанной системе прокторинга можно добавлять новые платформы, на которых может происходить тестирование.

1. Программный сбор данных о действиях пользователя

Первым компонентом программного комплекса для прокторинга является модуль сбора данных о действиях пользователя. Так как сервис «Google Формы» доступен для работы с ним в любом веб-браузере, модулем сбора данных будет являться разработанное расширение для браузера. Расширения для браузера позволяют расширить функциональность браузера или сайта. С их помощью можно изменять содержимое сайта, получать доступ к информации о браузере, а также некоторой информации о компьютере пользователя. Для разработки расширений используются стандартные веб-технологии, такие как JavaScript, HTML и CSS. Таким образом, с помощью специального расширения можно реализовать сбор данных и отправку их на сервер.

Перед его реализацией были составлен список признаков, которые будут максимально информативны и смогут показать наиболее подробно те действия, которые совершал пользователь во время тестирования. Список событий и собираемая информация выглядит следующим образом:

1. Открытие и обновление информации во вкладке браузера:

- URL-адрес страницы;
- название вкладки;
- порядковый номер вкладки;
- дата и время случившегося события;

- флаг того, что работает ли сейчас со вкладкой пользователь;
 - флаг того, что вкладка открыта в режиме «инкогнито»;
 - статус загрузки вкладки;
2. Переключение на какую-либо вкладку браузера:
- URL-адрес страницы;
 - название вкладки;
 - порядковый номер вкладки;
 - дата и время переключения;
3. Перемещение курсора мыши за пределы окна браузера:
- дата и время выхода;
4. Возвращение курсора мыши в окно теста:
- дата и время входа;
5. Выбор ответа в вопросе с одним вариантом ответа:
- текст вопроса;
 - текст ответа;
 - номер ответа;
 - дата и время ответа;
6. Выбор ответа в вопросе с несколькими вариантами ответов:
- текст вопроса;
 - текст ответа;
 - номер ответа;
 - дата и время ответа;
7. Побуквенное заполнение текста в поле краткого или подробного ответа на вопрос:
- текст вопроса;
 - текст ответа;
 - дата и время ответа;
8. Завершение теста:
- дата и время завершения;
 - название теста;
 - данные о пользователе (имя тестируемого);

9. Очистка формы от заполненных ответов:

- дата и время события;

10. Переход на страницу новой попытки решения:

- дата и время события;

2. Серверная часть программного комплекса

Для реализации серверного приложения был использован фреймворк Flask, написанный на языке Python. Выбор этого языка и фреймворка позволил избежать проблем с вызовом Python-программы для анализа собранных данных из другого языка программирования.

Для хранения данных о прохождении тестов использована база данных MongoDB, так как она позволяет сохранять данные в формате JSON документов без лишней обработки и получать их по запросу.

3. Сбор данных для датасета

Для реализации методов автоматизированного определения списывания необходим специальный датасет, в котором будут присутствовать все выбранные признаки. Датасет был собран с помощью студентов Новосибирского Государственного Технического Университета, проходивших на курсе по операционным системам периодическое тестирование с целью выявить качество остаточных знаний после лекции. Для этих целей были применены разработанные программные модули по сбору и сохранению данных. Перед массовым объявлением о тестировании двое студентов проверили работоспособность подготовленных программ и качество подготовленной инструкции.

В начале лекции студентам предоставлялось время на прохождение теста по теме предыдущей лекции. Тем, кто согласился на участие в эксперименте, было установлено разработанное расширение для сбора данных, автоматически запускавшееся при открытии формы с нужным тестом. По завершению теста отслеживание останавливалось. В ходе эксперимента были собраны данные о прохождении 4 тестов группой примерно из 20 человек.

В базе данных содержится больше 40 различных сессий с записью действий тестируемых студентов, собранных во время проводимых экспериментов. Действия содержат либо исключительно выполнение заданий, либо не связанные с прямым выполнением теста. Это не означает, что любые события, не относящиеся к отправке ответов, являются недопустимыми, так как тестируемый теоретически может совершать любые действия, не направленные на списывание.

Очевидно, нечестным выполнением теста будет являться сессия, содержащая запись с поисками похожей темы, что и просматриваемый сейчас вопрос. Ниже продемонстрирован пример того, каким будет сообщение при поисковом запросе во время прохождения тестирования.

```

{
  "_id": "6217305dec45c7bcac5c2ea0",
  "testSessionId": "H6xSYwyjtvvjA0LKq8hIq",
  "testId": "...",
  "createdAt": "2022-02-24T07:14:36.003Z",
  "type": "TAB_EVENT",
  "event": {
    "url": "https://yandex.ru/search/?clid=2341036-605...",
    "title": "Критическая секция - Яндекс: нашёлся 1 млн рез...",
    "active": true,
    "incognito": false,
    "index": 4,
    "status": "complete"
    "audible" false,
    "selected" true
  }
}

```

Для лучшего понимания устройства датасета и его содержимого, будет приведён подробный пример записи одного из сеансов по прохождению теста. Идентификатор рассматриваемой записи: Z4oSpPxLXwtv17Vt1WDvJ.

Время начала выполнения теста считается с момента отправления первого сообщения о событии. На листинге ниже продемонстрировано это событие об обновлении вкладки браузера с тестом на сайте «Google Формы», описания содержимого сообщений приведено ниже.

```

{
  "_id": "62172f69ec45c7bcac5c2a4c",
  "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
  "testId": "...",
  "createdAt": "2022-02-24T07:10:34.482Z",
  "type": "TAB_EVENT",
  "event": {
    "url": "https://docs.google.com/forms/d/e/.../viewform",
    "title": "Тест №2",
    "active": true,
    "incognito": false,
    "index": 42,
    "status": "complete"
    "audible" false,
    "selected" true
  }
},
{
  "_id": "62172f69ec45c7bcac5c2a4e",
  "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
  "testId": "...",
  "createdAt": "2022-02-24T07:10:34.595Z",
  "type": "MOUSE_ENTER_EVENT",
  "event": {
    "isTrusted": true
  }
},

```

В данном случае следующим событием оказалось событие о том, что курсор мыши пользователе вошёл в границы экрана браузера. В дальнейшем, все события о входе и выходе курсора считаются парными, то есть после выхода курсора за пределы экрана курсор обязательно вернётся, то есть будет отправлено сообщение о входе в экран.

Когда экзаменуемый отвечает на вопрос, имеющий один или несколько вариантов ответа, и отмечает выбранный вариант, формируется сообщение. В нём будет содержаться информация о вопросе и ответе, аналогичное сообщение будет отправлено при изменении выбранного ответа. Пример сообщений приведён ниже.

```
{
  "_id": "62172f69ec45c7bcac5c2aa6",
  "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
  "testId": "...",
  "createdAt": "2022-02-24T07:11:04.674Z",
  "type": "SOLUTION_CHECKBOX_EVENT",
  "event": {
    "taskNumber": 1,
    "taskTitle": "1. Критическая секция",
    "taskType": "checkbox",
    "answerNumber": 3,
    "answerTitle": "предотвращает одновременное использование..."
  }
},
{
  "_id": "62172f69ec45c7bcac5c2afd",
  "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
  "testId": "...",
  "createdAt": "2022-02-24T07:11:21.522Z",
  "type": "SOLUTION_CHECKBOX_EVENT",
  "event": {
    "taskNumber": 2,
    "taskTitle": "2. В режиме ядра",
    "taskType": "checkbox",
    "answerNumber": 1,
    "answerTitle": "выполняется код ядра ОС"
  }
},
```

В какой-то момент тестируемый столкнулся с вопросом, на который он не знает ответа. Это видно по тому, что в фрагменте записи прохождения теста, показанной на рисунке, появилось сообщение о поисковом запросе в Google и открытая страница, где он хотел найти верный ответ.

```
{
  "_id": "62172f69ec45c7bcac5c329d",
  "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
  "testId": "...",
  "createdAt": "2022-02-24T07:19:31.591Z",
  "type": "TAB_EVENT",
  "event": {
```

```

        "url": "https://www.google.com/search?q=%D0%BF%D1%80%D0...",
        "title": "промежуточная таблица записей процессов - Поиск...",
        "active": true,
        "incognito": false,
        "index": 3,
        "status": "complete"
        "audible" false,
        "selected" true
    }
},
{
    "_id": "62172f69ec45c7bcac5c32cc",
    "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
    "testId": "...",
    "createdAt": "2022-02-24T07:19:42.053Z",
    "type": "TAB_EVENT",
    "event": {
        "url": "https://docstore.mik.ua/bach/glava_11.htm",
        "title": "2.2 ВВЕДЕНИЕ В ОСНОВНЫЕ ПОНЯТИЯ СИСТЕМЫ",
        "active": true,
        "incognito": false,
        "index": 3,
        "status": "complete"
        "audible" false,
        "selected" true
    }
},

```

Тест завершается, когда итоговая форма была отправлена, о чём также приходят сразу несколько сообщений: уведомление о завершении, название теста. Также в данном случае было отправлено сообщение об автоматической перезагрузке вкладки с тестом. Это произошло из-за того, что на стороне расширения завершение тестируемого до конца не успело обработаться.

```

{
    "_id": "62172f69ec45c7bcac5c3273",
    "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
    "testId": "...",
    "createdAt": "2022-02-24T07:29:54.416Z",
    "type": "FORM_SOLUTION_SENT_EVENT",
    "event": {
        "isTrusted" true
    }
},
{
    "_id": "62172f69ec45c7bcac5c3276",
    "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",
    "testId": "...",
    "createdAt": "2022-02-24T07:29:54.417Z",
    "type": "FORM_TEST_TITLE_EVENT",
    "event": {
        "testTitle": "Тест №2",
        "testDescription": "основные понятия ОС Unix",
    }
}

```

```
    },  
    {  
      "_id": "62172f69ec45c7bcac5c3c75",  
      "testSessionId": "Z4oSpPxLXwtv17Vt1WDvJ",  
      "testId": "...",  
      "createdAt": "2022-02-24T07:29:54.420Z",  
      "type": "TAB_EVENT",  
      "event": {  
        "url": "https://docs.google.com/forms/d/e/.../viewform",  
        "title": "Тест №2",  
        "active": true,  
        "incognito": false,  
        "index": 0,  
        "status": "complete"  
        "audible" false,  
        "selected" true  
      }  
    }  
  },  
}
```

4. Определение списывания

Разработка модуля по определению списывания производилась с помощью языка программирования Python 3.8 и множества различных библиотек и фреймворков для машинного обучения и нейронных сетей, о которых более подробно будет написано ниже.

Задача по определению списывания была разделена на несколько подзадач по типу событий и тому, как можно воспользоваться полученной информацией. Это позволило применить разные подходы и алгоритмы для получения нужного результата.

События по выходу курсора за пределы окна браузера и его возвращению объединены в один список. Это позволяет построить последовательность, отсортированную по времени с начала теста для пользователя и до его завершения. Первым элементом в списке данных всегда является событие о выходе за пределы окна, так как при старте теста курсор находится внутри окна. Последним элементом — событие о возвращении курсора в окно браузера.

Для определения подозрительной активности тестируемого было решено рассмотреть данную задачу как задачу о кластеризации промежутков времени, во время которых курсор мыши был вне теста. Для этого список событий был разделён на события о выходе и возвращении курсора. Строковое представление даты и времени в формате ISO 8601 в обоих списках заменяется на целочисленную сумму минут, секунд и миллисекунд с нужными коэффициентами. В качестве выборки для кластеризации взят модуль разницы между двумя промежутками времени.

Полученная выборка передаётся в алгоритм кластеризации, разбивающий все точки на два кластера: «нормальные» и «подозрительные». В кластер «нормальные» должны попадать точки близкие к координатной прямой, что говорит о краткосрочности потери фокуса. В кластер «подозрительные» должны попасть

самые большие промежутки, находящиеся на самом значительном расстоянии от координатной прямой.

Проверка того, сколько и какие точки попадали классифицировались как «нормальные» и «подозрительные», проводилась с помощью графиков с изображениями точек классов и их принадлежности данным в датасете. Для этого бралась случайная сессия из датасета, её события о выходе и входе курсора и время появления этого события.

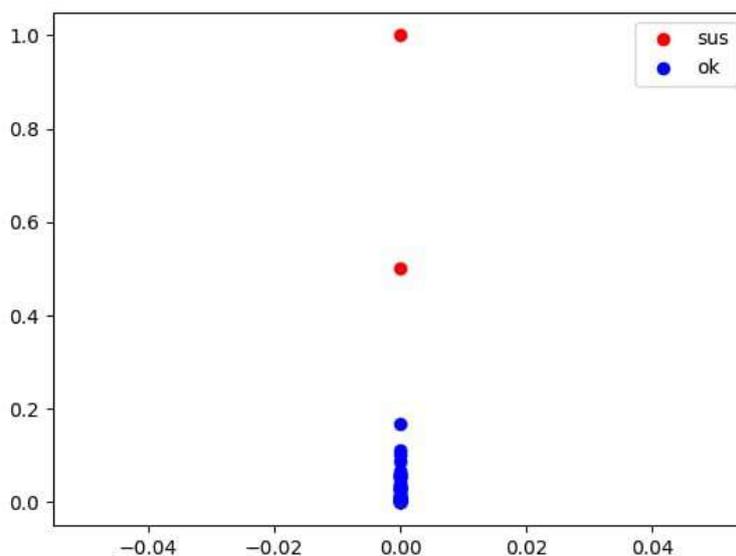


Рис. 1. Пример кластеризации промежутков отвлечения, когда тестируемый отводил курсор более двух раз

В примере на рисунке 1 видно, что всего две точки были определены в класс «подозрительные» (отмечены красным цветом на рисунке) из-за наибольшей продолжительности отвлечения от теста. Остальные же были определены в класс «нормальные» (отмечены синим цветом) из-за крайне короткого промежутка отсутствия курсора внутри браузера.

В записи сеанса тестирования, проверенного на рисунке 2, количество выходов курсора за пределы браузера уже намного больше. Особенно тех, которые были приняты за подозрительные. Здесь, как и в прошлом примере, важнейшим фактором является размер промежутка отсутствия, что влияет на итоговую оценку честности.

Следующими данными для обработки были события по обновлению вкладок браузера и данные о выполненных заданиях. В этих данных проверялся следующий факт: если пользователь выполнял задание честно, то в истории его вкладок не должны присутствовать формулировки вопросов заданий или формулировки ответов.

Делать такую проверку «наивно» нельзя, поиск полной фразы из формулировки задания не сможет распознать тот факт, что тестируемый мог перефразировать вопрос, же искать какую-то часть фразы из вопроса.

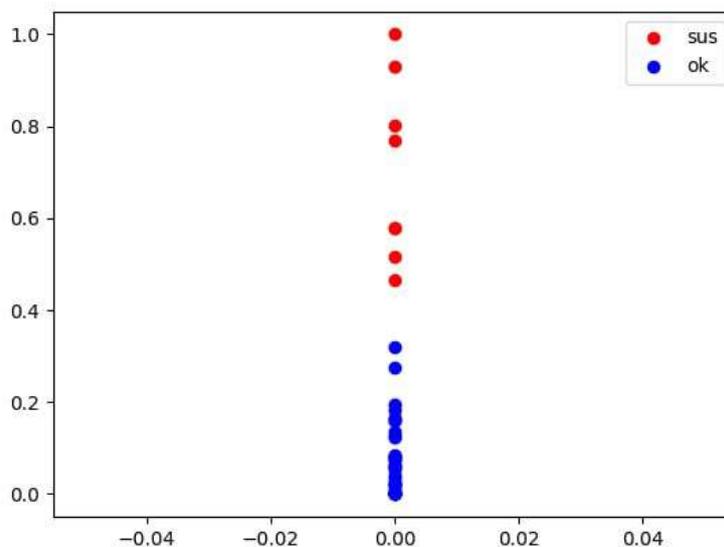


Рис. 2. Пример кластеризации промежутков отвлечения, когда тестируемый часто убирал курсор за пределы браузера

Правильная проверка должна заключаться в проверке названия поискового запроса или названия вкладки браузера на схожесть с текстом вопроса. Для этого необходимо реализовать или же воспользоваться методом проверки строк на схожесть.

Для реализации проверки на схожесть названия вкладки с формулировкой задания была использована библиотека `sklearn` и имеющийся в ней модуль векторизации данных `TfidfVectorizer`. В нём реализован достаточно эффективный метод векторизации слов, основанный на преобразовании слов в эмбединги — числовые вектора. Существуют разные способы построения таких векторов, но `TfidfVectorizer` основан на частоте повторения слов в переданном словаре (исходном тексте) [3].

Этот способ векторизации был применён таким образом, что из формулировок вопросов, на которые ответил тестируемый, строится словарь. Затем, пошагово проверяются тексты всех открываемых в момент теста вкладок на схожесть с имеющимся словарём. В тех случаях, когда результат сравнения выдаёт высокое значение схожести, найденный запрос сохраняется отдельно для запоминания его соответствия с вопросом теста (см. табл. 1).

Однако, некоторые испытуемые могут искать ответ на необходимый вопрос на английском языке. Возникновение такой ситуации крайне вероятно при прохождении тестов на технические темы, когда значительная часть актуальной и точной информации написана на английском языке. На такой случай проверка в проверку на схожесть формулировок заданий и запросов был добавлен второй этап. Во время него формулировки вопросов и ответов сначала переводятся на английский язык с помощью NLP библиотеки `translate` для языка Python, а затем проверка на схожесть с помощью `TfidfVectorizer` проводится ещё раз.

Последним рассмотренным типом событий было событие о вписывании от-

Таблица 1. Пример одной итерации сравнения поискового запроса с частью формулировок вопросов и ответов теста

No	texts	similar
1	1. Критическая секция процесса	0.816497
2	доступна только тем процессам, созданным ядром ОС	0.000000
3	предотвращает одновременное использование неск...	0.000000
4	2. В режиме ядра	0.000000
5	выполняется код ядра ОС	0.000000
6	3. Механизм квантования времени реализует	0.000000
7	подсистема управления процессами	0.000000
8	4. Смена контекста выполняется	0.000000
9	при переходе из режима ядра в заблокированное...	0.000000
10	при переходе из заблокированного состояния в...	0.000000

вета в область краткого или развёрнутого ответа. Было выдвинуто предположение, что участник, выполняющий задания нечестно, может воспользоваться подходом «копировать-вставить», когда ответ на вопрос не набирается на клавиатуре посимвольно, а вставляется сразу полностью из буфера обмена компьютера. Частое присутствие такого типа событий среди хронологии выполнения теста может говорить о том, что тестируемый каким-то образом смог найти подходящий, по его мнению, ответ и выдать его за свой.

Так как расширение для браузера для вопросов с письменным ответом отправляет сообщения на каждое нажатие клавиши клавиатуры, определить одномоментно появившуюся строку с ответом можно с помощью комбинирования проверки на последовательное изменение текста ответа и проверки нового значения на схожесть со старым при резком изменении ответа.

При нахождении не поэтапного набора текстового ответа на вопрос и его значительном отличии от предыдущего, вопрос, на котором это произошло, а также введённый ответ сохраняются.

После проверки всего списка событий, которые были сгенерированы проверяемым во время онлайн-тестирования, создаётся отчёт в виде JSON объекта, в который включаются данные о проводимом тесте и тех действиях, которые были восприняты разработанной системой как «подозрительные». Такой вид результата обработки позволит защитить тестируемого на появление каких-либо непредвиденных ошибок или возможных ложных срабатываний, которых всё же полностью невозможно избежать. Проверяющий же в таком случае сможет сам увидеть, на каких вопросах какое поведение или ответ посчитала подозрительным и вынести свой окончательный ответ о полученном результате.

Заключение

Результатом работы является разработанный программный комплекс, позволяющий внешне дополнить функциональность сервиса «Google Формы» средством автоматизированного прокторинга на основе действий пользователя во время прохождения онлайн-тестирования.

ЛИТЕРАТУРА

1. Conijn R., Kleingeld A., Matzat U., Snijders, C. The fear of big brother: The potential negative side-effects of proctored exams // Journal of Computer Assisted Learning. 2022. P. 1521-1534. doi: 10.1111/jcal.12651.
2. Воронцов К.В. Лекции по алгоритмам кластеризации и многомерного шкалирования. URL: <http://www.ccas.ru/voron/download/Clustering.pdf> (дата обращения: 25.05.2022).
3. TfidfVectorizer — scikit-learn 1.1.1 documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (дата обращения 25.05.2022).

DEVELOPMENT OF AN AUTOMATED PROCTORING SYSTEM FOR GOOGLE FORMS

A.L. Golovanov

Ph.D. Student, e-mail: agolovanov@yahoo.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. The paper presents a description of software modules developed by the author to create a system of automated proctoring for online testing. The analysis of the data set created with the help of the developed programs and the description of the methods used to automatically determine the dishonesty of testing is given.

Keywords: online proctoring, software, data collection, data analysis.

Дата поступления в редакцию: 20.11.2022