Разработка рекомендательной системы для подбора тренировок

А.В. Зубков, С.В. Степанов, В.В. Носкин, Н.Д. Сибирный, Ю.А. Орлова

Волгоградский Государственный Технический Университет

Аннотация: В статье рассматриваются разработанные авторами методы и подходы для рекомендательной системы, которые направлены на повышение качества реабилитации пациента во время выполнения дыхательных тренировок. С целью описания тренировок, был разработан собственный язык для конкретной предметной области, а также его грамматика и синтаксический анализатор. Благодаря данному языку можно построить дерево, описывающее конкретную тренировку пациента. К полученному дереву применяются два основных метода, рассматриваемых в статье: «Метод для анализа проблемных участков в ходе прохождения тренировок пациентами» и «Метод нечеткого поиска схожих участков в тренировке». С помощью данных методов в работе предлагается анализировать проблемные участки тренировок пациентов во время реабилитации и искать похожие затруднительные участки пациента для подбора похожих упражнений с целью поддержания уровня разнообразности заданий и вовлечения пациента в процесс.

Ключевые слова: рекомендательная система, система управления обучением, реабилитация, медицина, дыхательная тренировка, маркерная система, предметно-ориентированный язык, расстояние Левенштейна.

Введение

В реабилитации многие используют цифровые комплексы, которые позволяют повысить интерес пациента к тренировке, и тем самым непосредственно улучшить качество реабилитации [1-3]. В процессе реабилитации пациент погружается в виртуальную среду, перед ним ставится игровая цель, которой необходимо достичь и тем самым пациент, проходя игру за игрой, выполняет набор упражнений, относящихся к реабилитации [4]. В современных процессах реабилитации наблюдается одна проблема – когда у пациента не получается выполнить одно упражнение (пройти стадию игры), он теряет мотивацию и интерес к игровому реабилитационному процессу. Данную проблему также можно наблюдать и в системах управления обучением (Learning Management System - LMS) во время тестирования студентов [5]. Для ее решения используют рекомендательные системы, которые подбирают пользователю наиболее подходящие тренировки, позволяющие отработать набор выявленных проблем. Другими словами, система проводит поиск схожих элементов в базе тренировок и

выбирает тренировку, которая имеет схожие элементы, вызвавшие у пользователя наибольшие проблемы. Целью рекомендательной системы является повышение разнообразия упражнений, направленных на проработку проблемных элементов пройденных тренировок.

Из поставленной цели вытекают следующие задачи:

- разработать метод для анализа проблемных участков в ходе прохождения пациентами тренировки;
- разработать метод нечеткого поиска схожих участков в тренировке.

В рамках данной статьи разработана рекомендательная система тренировок на базе игрового тренажера для дыхательных тренировок.

Система дыхательной тренировки

Дыхательная реабилитация является важной составляющей комплексного лечения пациентов с различными заболеваниями легких, так как позволяет раскрыть плохо вентилируемые участки легких и улучшить их функциональность [6]. Кроме того, основной задачей дыхательной реабилитации является восстановление оптимального паттерна дыхания, что позволит увеличить дыхательный потенциал функций легких и улучшить качество жизни пациентов.

На данный момент врачи выделяют три основных типа дыхания:

- Грудное дыхание - это тип дыхания, при котором вдох происходит за счет расширения грудной клетки и подъема ребер вверх. В этом типе дыхания активно задействованы мышцы грудной клетки, а брюшная область почти не участвует в процессе вдоха.

- Брюшное дыхание - это тип дыхания, при котором вдох происходит за счет расширения брюшной полости и опускания диафрагмы вниз. В этом типе дыхания задействованы мышцы брюшной стенки и диафрагмы, а грудная клетка практически не участвует в процессе вдоха.

- Смешанное дыхание - это комбинация грудного и брюшного дыхания, при котором в процессе вдоха задействованы как мышцы грудной клетки, так и мышцы брюшной стенки, диафрагмы. В этом типе дыхания вдох происходит за счет расширения как грудной, так и брюшной полостей, что позволяет легким лучше вентилировать.

Основной целью тренажера дыхательных тренировок является развитие смешанного типа дыхания у пациента. При реализации тренажера были решены следующие задачи:

- Разработан метод для автоматизированного определения типа дыхания человека;

- Разработан метод для обеспечения биологической обратной связи между пациентом и тренажером;

- Разработан игровой пользовательский сценарий для тренажера;

- Разработана модель промежуточного представления для описания тренировок;

- Разработана рекомендательная система подбора тренировок;

- Разработаны программно-алгоритмические средства для проведения дыхательных тренировок.

Механика тренажера представляет собой аркадный платформер с видом сбоку, где пользователь управляет сферой, которой необходимо собрать как можно больше звезд, не выходя за жирные линии. Пользователю предлагается в тренажере пройти разные участки пути, просто вдыхая и выдыхая воздух легкими (Рис. 1). Биологическая обратная связь обеспечивается системой захвата движения на базе трех маркеров. Маркеры крепятся на мечевидном отростке грудины, между 9 и 11 грудными позвонками и над пупком пациента. Данное размещение позволяет отследить динамику расширения и сужения при грудном, брюшном и смешанном типах дыхания, представив это, как расстояние между двумя маркерами. При вдохе

шарик, которым управляет пользователь, поднимается, а при выдохе, соответственно, опускается.

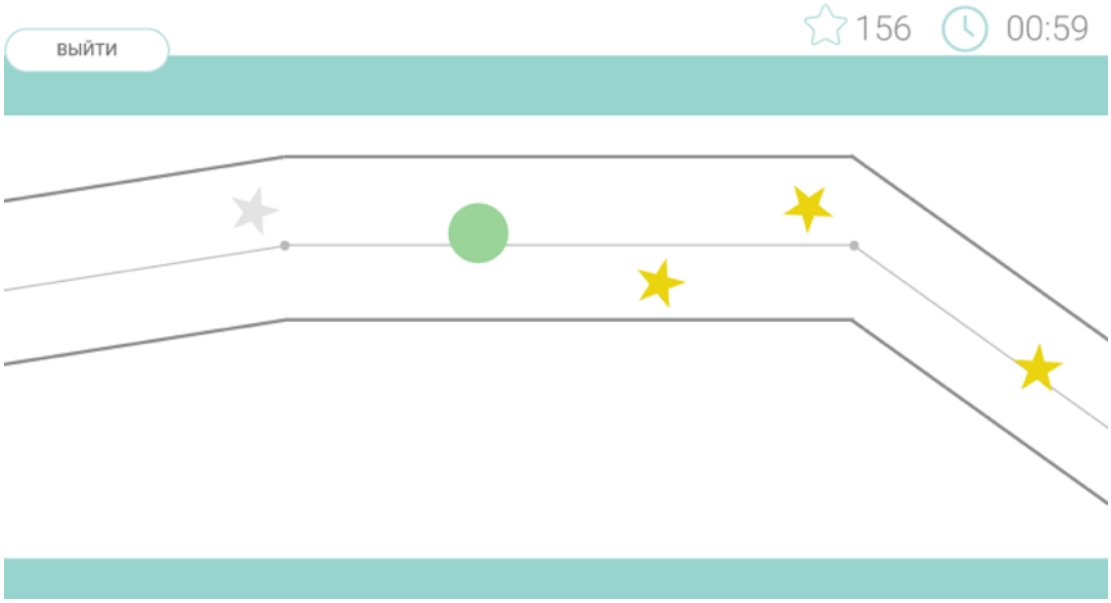

Рис. 1. – Пример работы программы.

Для построения уровня необходимы следующие характеристики:

- id – тренировки (сущность, которая хранит в себе геометрию уровня);
- доминирующий тип дыхания человека в повседневной жизни (характеристика, показывающая какие мышцы и какой паттерн будут использоваться, в рамках тренировки);
- максимальные расстояния между датчиками, для паттерна дыхания, который подвергается тренировке;
- минимальное расстояние между датчиками, для паттерна дыхания, который подвергается тренировке.

После того, как пациент попадает в тренажер в первый раз, ему предлагается список упражнений с их кратким описанием. После выбора упражнения, тренажер замеряет доминирующий тип дыхания, который зависит от того, каким расстоянием будет управляться шарик (Рис. 2). Если у пользователя доминирует брюшной тип дыхания, тогда расстояние в тренировке будет учитываться между грудным и спинным датчиком. Если грудное дыхание доминирует, то расстояние учитывается между брюшным и

грудным датчиком. А если смешанное, то будет браться среднее расстояние, с использованием всех трёх маркеров (где коэффициент K направлен на уравновешивание двух расстояний). Далее пациент проходит калибровку, где ему необходимо максимально вдохнуть указанным способом и максимально возможно выдохнуть. После вычисления всех основных характеристик, необходимых для построения уровня, запускается основная часть тренажера.

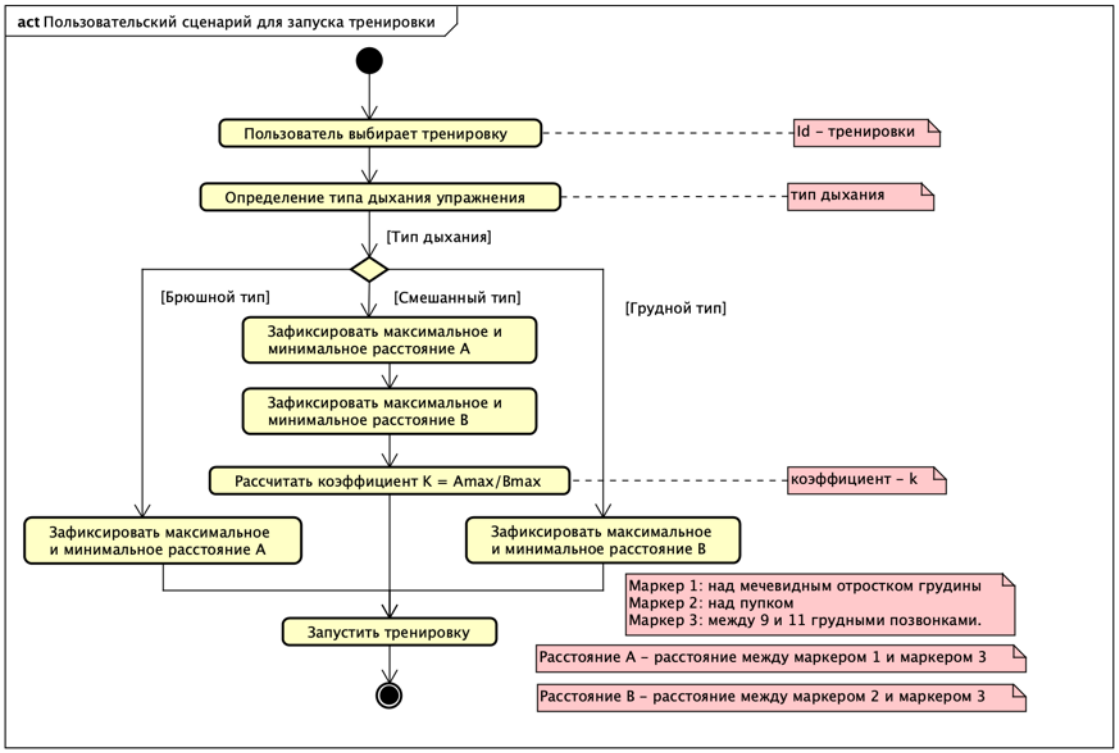

Рис.2. Сценарий запуска программы.

Язык для описания игровых тренировок

Так как современный мир не стоит на месте, на данный момент не только изучаются разнообразные аспекты, влияющие на качество реабилитации, но и происходит постоянное улучшение методик проведения самих упражнений. Поэтому, в качестве точки расширения, было принято решение по расширению набора упражнений с помощью промежуточного представления. Промежуточное представление тренировки представляет из себя описательный язык игровых задач, которые ставятся перед игроком в определенный момент времени тренировки.

Каждый оператор в языке представляет собой часть лабиринта

Грамматика языка представлена ниже:

```
program : commands EOF ;
commands: command* ;
commandParams: command (',' command)*;
command: ('hold' OP (NUMBER) CP) |
        ('inhale' OP (NUMBER ',' NUMBER) CP) |
        ('exhale' OP (NUMBER ',' NUMBER) CP) |
        ('inhaleForce' OP (NUMBER ',' NUMBER) CP) |
        ('inhaleFading' OP (NUMBER ',' NUMBER) CP) |
        ('exhaleForce' OP (NUMBER ',' NUMBER) CP) |
        ('exhaleFading' OP (NUMBER ',' NUMBER) CP) |
        ('repeat' OP NUMBER ',' commandParams CP) |
        'end';
IDENTIFIER : (LETTER (LETTER | DIGIT)*) ;
OP : '(';
CP : ')';
NUMBER : (DIGIT)+ ;
WHITESPACE : [ \r\n\t] + -> channel (HIDDEN);
DIGIT : '0'..'9';
LETTER : LOWER | UPPER ;
LOWER : ('a'..'z') ;
UPPER : ('A'..'Z') ;
```

На основе описанной грамматики генерирует LL-парсер с помощью генератора нисходящих анализаторов для формальных языков ANTLR (ANother Tool for Language Recognition) [7], где «LL» означает, что строки анализируется слева направо (left to right) и используется левосторонний вывод (leftmost derivation). Используя сгенерированный парсер, можно получить дерево, описывающее порядок действия тренировки для дальнейшего анализа и преобразований. На Рис. 3 изображен пример

разобранного дерева с помощью ANTLR- парсера на основе представленной грамматики для примера с вложенными циклами: repeat(2, exhale(1, 30), repeat(2, inhale(1, 20), hold(1), exhale(3, 20))).

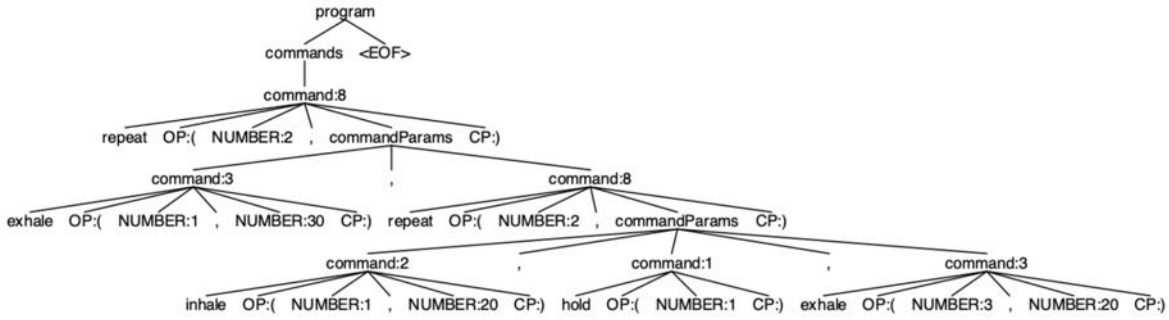

Рис. 3. Дерево, описывающее тренировку, после парсинга на основе ANTLR-парсера.

Таблица № 1.

Описание операторов языка

Оператор	Описание
inhale (duration, volume)	команда для выполнения линейного вдоха, где duration - продолжительность, volume - конечный объем
exhale(duration,volume)	команда для выполнения линейного выдоха, где duration - продолжительность, volume - конечный объем
inhaleFading (duration, volume)	команда для выполнения горизонтально-параболического вдоха, где duration - продолжительность, volume - конечный объем
exhaleFading (duration, volume)	команда для выполнения горизонтально-параболического выдоха, где duration - продолжительность, volume - конечный объем
inhaleForce (duration, volume)	команда для выполнения параболического вдоха, где duration - продолжительность, volume - конечный объем
exhaleForce (duration, volume)	команда для выполнения параболического выдоха, где duration - продолжительность, volume - конечный объем

hold (duration)	команда для удержания дыхания на указанную продолжительность, с начальным объемом Vs. Формула: V(t) = Vs
repeat (times, commands)	команда для повторения указанных команд (commands) заданное количество (times) раз
pause(duration)	команда для паузы на указанную duration - продолжительность
end	команда для окончания программы (сценария)

Значения операторов представлены в таблице 1. Операторы типа inhale и exhale в своей реализации являются взаимозаменяемыми, данное решение позволяет пользователю языка визуально отделять вдохи и выдохи и контролировать текущий объем пользователя.

Для расчета текущей вертикальной координаты положения шарика, было введено понятие объема - V_p. Оно высчитывается, с использованием максимальных и минимальных нормализованных значений расстояний, рассчитываемых ранее и увеличенных в 100 раз.

Операторы inhale и exhale представляют собой прямые участки пути (Формула 1), которые проводятся через следующие две точки $A(t_s, V_s)$ и $B(t_s + duration, volume)$.

$$V(t) = \frac{(volume - V_s) * (t - t_s - duration)}{duration} + volume, \quad (1)$$

где $V(t)$ - вертикальная координата положения центральной точки уровня на момент t, t - текущее время от начала выполнения упражнения , $volume$ - целевой показатель объема, t_s - начальное время, V_s - вертикальная координата положения центральной точки уровня на момент ts, $duration$ - продолжительность.

Операторы суффиксом “Fading” представляют участок пути, напоминающих степенную квадратичную функцию. Для построения данного участка, используется следующая формула (2):

$$V(t) = \frac{(volume - V_s)}{duration^2} + (t - t_s)^2, \quad (2)$$

где *V(t)* - вертикальная координата положения центральной точки уровня на момент t, t - текущее время от 0 , *volume* - целевой показатель объема, t_s - начальное время, V_s - вертикальная координата положения центральной точки уровня на момент t_s, *duration* - продолжительность.

Операторы суффиксом “Force” представляют участок пути, напоминающих степенную функцию со степенью 0.5 (т.е. квадратный корень). Для построения данного участка, используется следующая формула (3):

$$(t) = \frac{volume - V_s}{duration^2} * \sqrt{t - t_s}, \quad (3)$$

где *V(t)* - вертикальная координата положения центральной точки уровня на момент t, t - текущее время от начала упражнения , *volume* - целевой показатель объема, t_s - начальное время, V_s - вертикальная координата положения центральной точки уровня на момент t_s, *duration* - продолжительность.

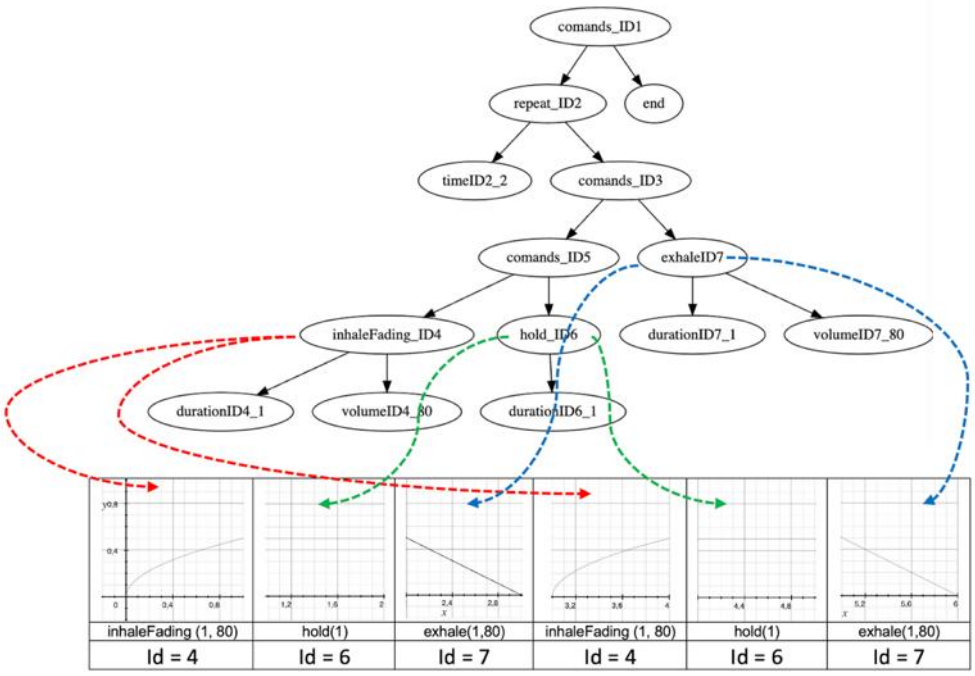

Рис. 4. Пример описания ломанной функции, описывающей уровень по коду “repeat(2, inhaleFading(1, 80), hold(1), exhale (1, 80)) end ”.

На Рис. 4 продемонстрирована ломаная функция, которая описывает уровень, необходимый для прохождения пользователем.

Для построения верхней стенки уровня, строится функция $Vtop(t)= V(t) + n$, а для нижней $Vbottom(t) = V(t) - n$. Значение n зависит от сложности уровня. Чем меньше n, тем уже участок, соответственно, тем уровень сложнее.

Метод для анализа проблемных участков в ходе прохождения тренировки пациентов

Для того, чтобы система могла предложить пользователю наиболее подходящий уровень, который позволил бы отработать основные элементы, которые вызывают у пользователя наибольшие проблемы, необходимо разработать метод, позволяющий выделить эти участки (Рис. 5).

На формуле (4) показана метрика, использующаяся для расчета показателя суммы расстояний отклонения пользователя на данный момент времени.

$$Error = \sum_{t=-m}^{m} \begin{cases} |V_p(t) - V(t) + n|, \text{ если } V_p(t) > V(t) + n \\ 0, \text{ если } V(t) + n \geq V_p(t) \leq V(t) \\ |V_p(t) - V(t) - n|, \text{если } V_p(t) < V(t) - n \end{cases}, \quad (4)$$

Были выделены совместно со специалистами основные характеристики для данных участков:

- допустимое количество элементов в участке от 2 до 5;
- допустимое среднеквадратичное значение отклонения ошибки пользователем между участками должно составлять до 40%.

Для выявления проблемных участков, где пользователь совершил наибольшее количество ошибок, используется абстрактное синтаксическое дерево, в котором в узлах дополнительно хранится следующая информация: среднеквадратичное отклонение ошибки и среднее арифметическое значение ошибки для дочерних узлов дерева [8].

Изначально строится абстрактное синтаксическое дерево без узлов repeat, pause (Рис. 6). Для каждого узла дерева операторов, описывающих уровень, записывается значение ошибки. Если оператор описывает несколько участков уровня (те используется циклично), то для него высчитывается среднее арифметическое значение ошибки и среднее квадратичное отклонение для каждой попытки. Далее операция повторяется для узлов типа commands. Затем все узлы типа commands ранжируются по записанной в них функции ошибки и отбрасываются те узлы, которые не удовлетворяют условию, выделенному совместно со специалистами (количество элементов в участке от 2 до 5 и допустимое среднеквадратичное значение отклонения ошибки пользователем между участками должно составлять до 40%).

где $V(t)$ - вертикальная координата положения центральной точки уровня на момент t, $V_p(t)$ - вертикальная координата положения пользовательского объекта на момент t.

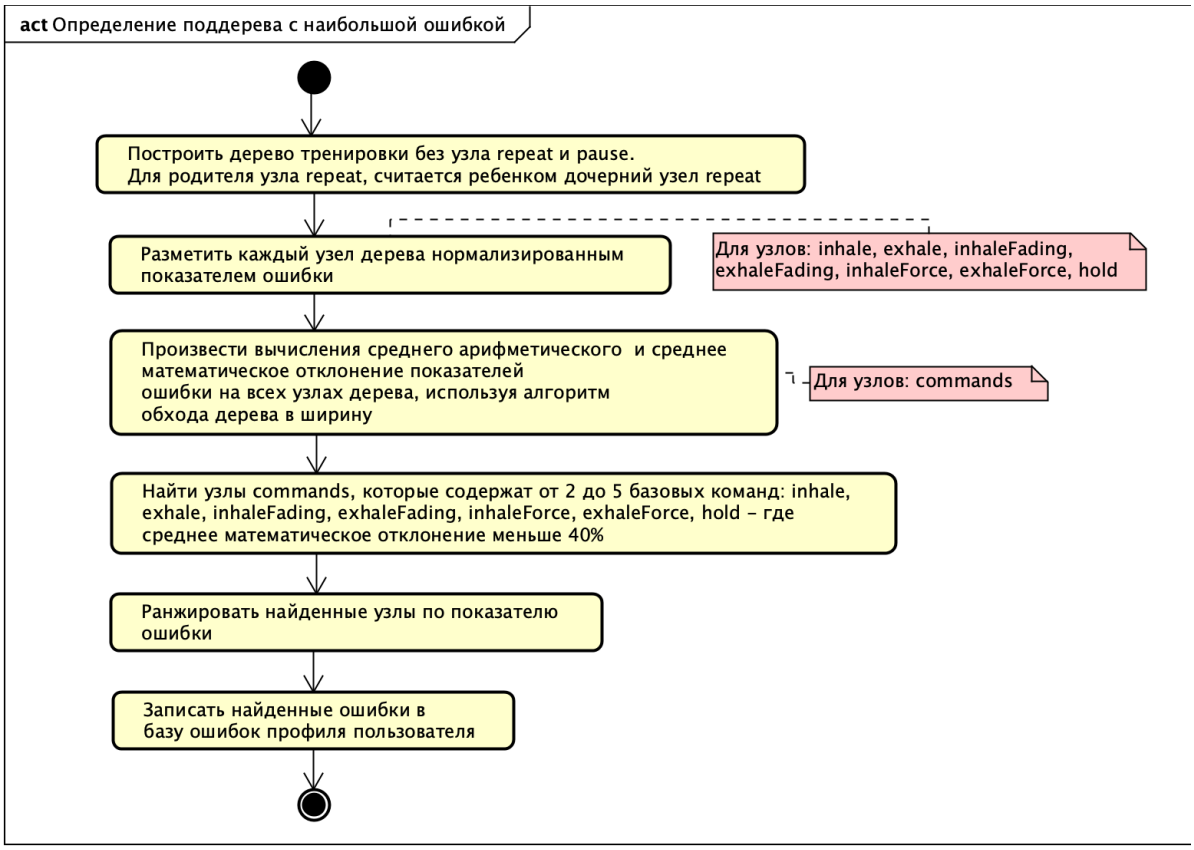

Рис. 5. Метод поиска и ранжирования по важности проблемных участков.

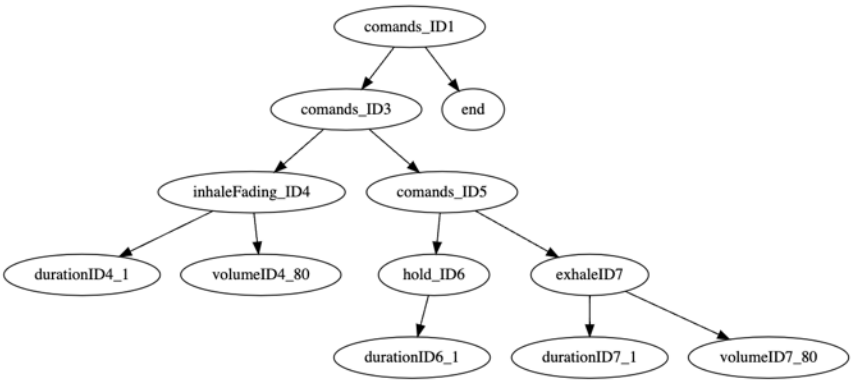

Рис. 6. Пример результирующего дерева, после удаления лишних узлов метода анализа проблемных участков в ходе прохождения тренировки пациентами.

Полученные первые три поддерева записываются в профиль пациента, как проблемные участки.

Метод нечеткого поиска схожих участков в тренировке.

Для реализации метода нечеткого поиска был выбран путь сведения задачи к нечеткому поиску подстроки в строке. В качестве метрики схожести [9] был выбран метод Дамерау — Левенштейна [10], а для представления деревьев в виде строки использовалась прямая польская нотация [11]. Чтобы уравновесить расстояния для алгоритма с использованием числовых показателей, был выбран путь представления их также в виде строк. Для *volume*, использовался нетерминальный символ V для каждого десятка. Для times, использовался нетерминальный символ T для каждой единицы и аналогично для *duration* использовался нетерминальный символ D. В качестве разделителя использовался пробел. А узел commands был преобразован к виду, когда он может принимать сразу более двух операторов и иметь одно значение - количество входящих узлов N, преобразованных аналогично с times. Пример строки кода:

repeat(2, inhaleFading(1, 80), hold(1), exhale(1, 80)) end

Пример в виде дерева представлен на Рис. 7. Пример в прямой польской нотации:

end exhale D VVVVVVVV hold D inhaleFading D VVVVVVVV commands NNN repeat TT commands NN

После преобразования синтаксического дерева к строке сравниваются все поддеревья с корнем commands используя метрику схожести алгоритма Дамерау — Левенштейна. Для каждого поддерева записывается id и расстояние. Далее происходит сортировка списка по убыванию и фильтрация по пороговому значению метрики схожести. Метод представлен на Рис. 8.

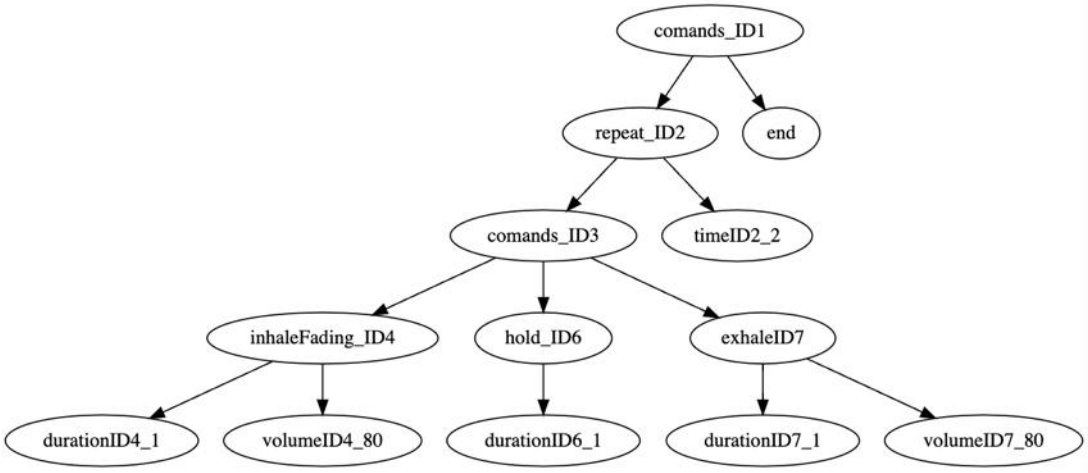

Рис. 7. Пример результирующего дерева, после модификации нетерминала commands.

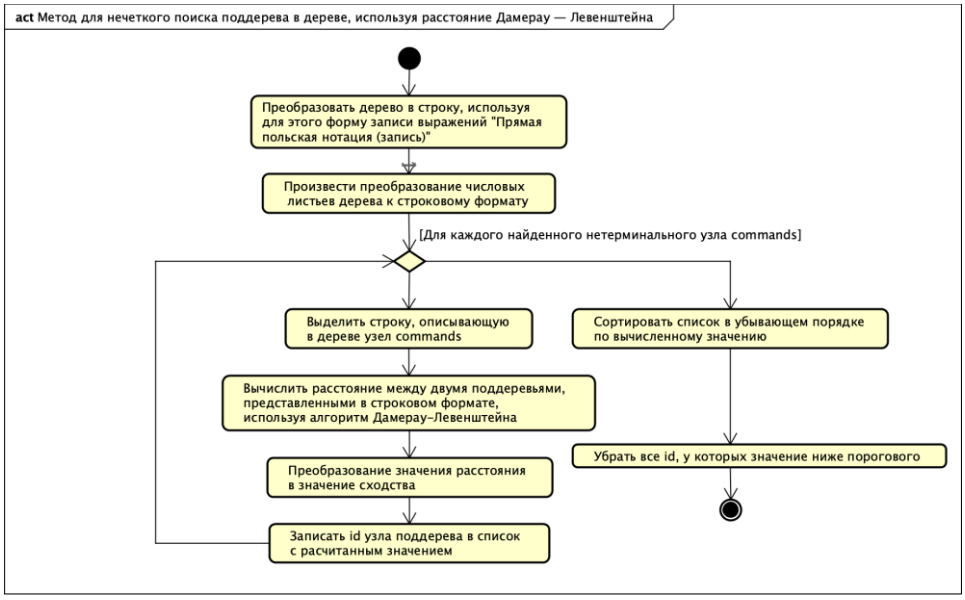

Рис. 8. Метод для нечеткого поиска поддерева в дереве, на базе метрики схожести расстояния Дамерау — Левенштейна.

Для поиска наиболее подходящего уровня вводятся коэффициент значимости проблемы в уровне (формула 5) и коэффициент значимости уровня (формула 6).

Метод представляет собой полный перебор (исключая повторы) каждого уровня (тренировки) из БД и каждого проблемного участка из профиля пользователя. Для каждой пары применяется метод для нечеткого поиска поддерева в дереве, на базе метрики схожести расстояния Дамерау — Левенштейна и рассчитывается для каждого проблемного участка в уровне коэффициент значимости проблемы и для каждого уровня - коэффициент значимости уровня. Метод представлен на Рис. 9.

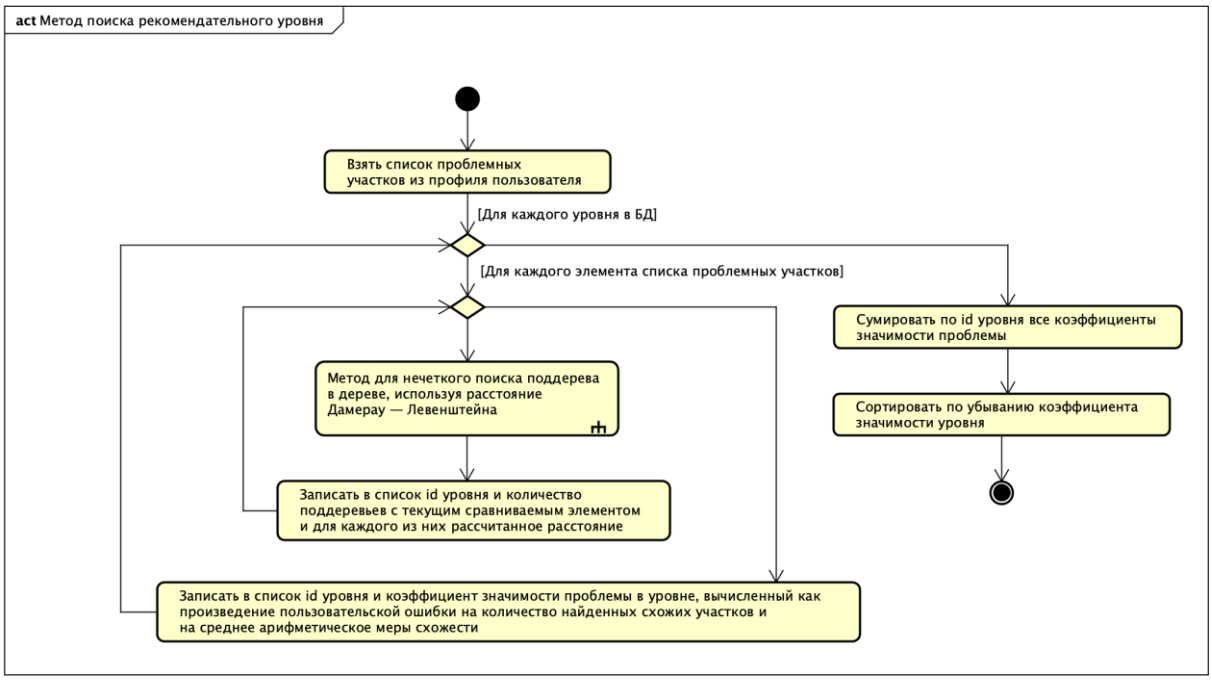

Рис. 9. Метод ранжирования уровня по значимости.

В результате чего, пользователь получает ранжированный список уровней, которые он может пройти, отработав участки, наиболее похожие на то, что у него не получалось ранее.

$$Kpi = \sum_{i=0}^{n} E_i * c_i * avg(me), \quad (5)$$

$$Ki = \sum_{i=0}^{n} Kpi_i, \quad (6)$$

где E_i – пользовательская ошибка для проблемы тренировки, c_i – количество найденных участков, которые удовлетворяют условию, *me* – мера близости.

Заключение.

В результате исследования, была разработана рекомендательная система для тренажера дыхательной реабилитации, использующая следующие два реализованных метода:

- метод для анализа проблемных участков в ходе прохождения тренировки пациентов;
- метод нечеткого поиска схожих участков в тренировке.

На данной системе был проведен ряд ручных тестов, которые продемонстрировали работоспособность системы.

Литература

1. Fekete M., Fazekas-Pongor V., Balazs P., Tarantini S., Nemeth AN., Varga JT. Role of new digital technologies and telemedicine in pulmonary rehabilitation: Smart devices in the treatment of chronic respiratory diseases // Wien Klin Wochenschr. 2021. №133. С. 1201-1207.

2. Леонова А.В., Синютин С.А., Шпаковская О.Ю. Разработка портативного пневмотренажера для тренировки дыхательной системы в борьбе с последствиями от перенесенного COVID-19 // Инженерный вестник Дона, 2022, №4. URL: ivdon.ru/ru/magazine/archive/n4y2022/7575/.

3. Стбеаков И.Н., Шутин Д.В., Марахин Н.А. Машинное обучение в реабилитационной медицине и пример классификатора движений пальцев для кистевого тренажера // Инженерный вестник Дона, 2020, №6. URL: ivdon.ru/ru/magazine/archive/N6y2020/6514/.

4. Зубков А.В., Сибирный Н.Д., Самоходкина И.А., Орлова Ю.А. Разработка цифровой модели реабилитации пациентов после инсульта для

восстановления бытовых навыков // Программная инженерия: современные тенденции развития и применения (ПИ-2021). 2021. С. 119-121.

5. Murad D. F., Heryadi Y., Wijanarko B. D., Isa S. M., Budiharto W. Recommendation System for Smart LMS Using Machine Learning: A Literature Review // International Conference on Computing, Engineering, and Design (ICCED). 2018. С. 113-118.

6. Yang F., Liu N., Hu J.Y., Wu L.L., Su G.S., Zhong N.S., Zheng Z.G Pulmonary rehabilitation guidelines in the principle of 4S for patients infected with 2019 novel coronavirus (2019-nCoV) // Zhonghua Jie He He Hu Xi Za Zhi. 2020. С. 180-182.

7. Kathleen F., Terence P. LL(*): the foundation of the ANTLR parser generator. // Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). 2011. С. 425-436.

8. Fluri B., Wuersch M., PInzger M., Gall H. Change Distilling: Tree Differencing for Fine-Grained Source Code Change Extraction // IEEE Transactions on Software Engineering. 2007. №11. С. 725–743.

9. Levandowsky M., Winter D. Distance between sets // Nature. 1971. №234. С. 34-35.

10. Introduction to Algorithms / Cormen, Leiserson, Rivest, Stein , 3 изд. The MIT Press, 2009. 1292 с.

11. Лозовский В. В. Алгоритмические основы обработки данных. М.: РТУ МИРЭА, 2022. 337 с.

References

1. Fekete M, Fazekas-Pongor V, Balazs P, Tarantini S, Nemeth AN, Varga JT. Wien Klin Wochenschr. 2021. №133. pp. 1201-1207.

2. Leonova A.V., Sinyutin S.A., Shpakovskaya O.Y. Inzhenernyj vestnik Dona, 2022, №4. URL: ivdon.ru/ru/magazine/archive/n4y2022/7575/.

3. Stbeakov I.N., Shutin D.V., Marakhin N.A. Inzhenernyj vestnik Dona, 2020, №6. URL: ivdon.ru/ru/magazine/archive/N6y2020/6514/.

4. Zubkov A.V., Sibirnyy N.D., Samokhodkina I.A., Orlova Yu.A. Programmnaya inzheneriya: sovremennye tendentsii razvitiya i primeneniya (PI-2021). 2021. pp. 119-121.

5. Murad D. F., Heryadi Y., Wijanarko B. D., Isa S. M., Budiharto W. International Conference on Computing, Engineering, and Design (ICCED). 2018. pp. 113-118.

6. Yang F., Liu N., Hu J.Y., Wu L.L., Su G.S., Zhong N.S., Zheng Z.G. Zhonghua Jie He He Hu Xi Za Zhi. 2020. pp. 180-182.

7. Kathleen F., Terence P., Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). 2011. pp. 425-436.

8. Fluri B., Wuersch M., PInzger M., Gall H., IEEE Transactions on Software Engineering. 2007. №11. pp. 725–743.

9. Levandowsky M., Winter D. Nature. 1971. №234. pp. 34-35.

10. Introduction to Algorithms. Cormen, Leiserson, Rivest, Stein , 3 rd. The MIT Press, 2009. P. 1292.

11. Lozovskiy V. V. Algoritmicheskie osnovy obrabotki dannykh [Algorithmic foundations of data processing]. RTU MIREA, 2022. P. 337.