Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

Генерация вопросов для проверки знаний по изучаемым текстам

Студент группы ИУ5-23М _____ Наседкин И.А.

Руководитель курсовой работы _____ Гапанюк Ю.Е.

2020 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУ5_____
(Индекс)
_____ Чёрненький В.М.
« _____ » _____ 2020 г.

З А Д А Н И Е
на выполнение курсового проекта

по дисциплине Обработка и анализ данных_____

Студент группы Наседкин И.А.

_____
(Фамилия, имя, отчество)

Тема курсового проекта Генерация вопросов для проверки знаний по изучаемым текстам____
_____
_____

Направленность КП (учебный, исследовательский, практический, производственный, др.)
_____
Источник тематики (кафедра)

График выполнения проекта: 25% к 4 нед., 50% к 8 нед., 75% к 12 нед., 100% к 16 нед.

Задание_____
_____
_____
_____
_____

Оформление курсового проекта:

Расчетно-пояснительная записка на __39_ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
_____
_____
_____
_____

Дата выдачи задания «01» сентября 2020 г.

Руководитель курсового проекта _____Гапанюк Ю.Е.
(Подпись, дата)

Студент _____Наседкин И.А.
(Подпись, дата)

Оглавление

Анализ

Введение

Учебный план

Основной целью системы высшего образования является профессиональная подготовка специалистов высшей квалификации в соответствии с социальным заказом. Поэтому именно профессиональная деятельность специалистов задает и определяет цели изучения всех учебных дисциплин, а следовательно, и содержание, и структуру, и формы соответствующей учебной деятельности студентов, готовящихся к этой профессиональной работе. Вот почему особое значение в настоящее время, когда происходит переход к новым государственным образовательным стандартам по всем учебным специальностям, приобретают исследования по прогнозированию последствий реформирования учебного процесса.

Наиболее распространенные из существующих методик управления основаны либо на принципе модульного представления учебных дисциплин, либо на основе дерева целей подготовки специалиста, однако методики управления учебным процессом вуза на основе формализованного подхода к построению учебной дисциплины разработаны недостаточно.

Одним из направлений работ в области управления учебным процессом вуза является составление учебных планов вузов на основе дерева целей подготовки специалиста. Основные цели обучения – определение списка знаний и умений, которыми должен обладать выпускник вуза. Каждой цели ставится в соответствие одна или несколько дисциплин учебного плана. Входными данными являются коэффициенты относительной важности целей учебного процесса. Далее происходит набор необходимых тем учебных дисциплин, наиболее важных для формирования специалиста. При таком алгоритме работы не учитываются связи между модулями. Связи между модулями, попавшими в учебный план, оцениваются после отбора

содержания, поэтому может проявиться информационная недостаточность для изучения некоторых модулей, т. к. элементы-предки, необходимые для них в качестве информационной базы, могут иметь недостаточно высокий групповой вес.

Другим направлением работ в области управления учебным процессом вуза является методика модульного обучения. Сущность модульного обучения заключается в том, чтобы максимально обособить отдельные блоки (модули) учебного материала. При модульном построении обучения строится граф логической структуры предмета, в котором указываются не только внутрипредметные, но и межпредметные связи. Затем в отдельные учебные элементы, составляющие структуру модуля, выбираются полностью те темы из графа логической структуры, которые необходимы для изучения конкретного учебного элемента, причём эти темы могут выбираться из различных учебных дисциплин, в зависимости от междисциплинарных связей.

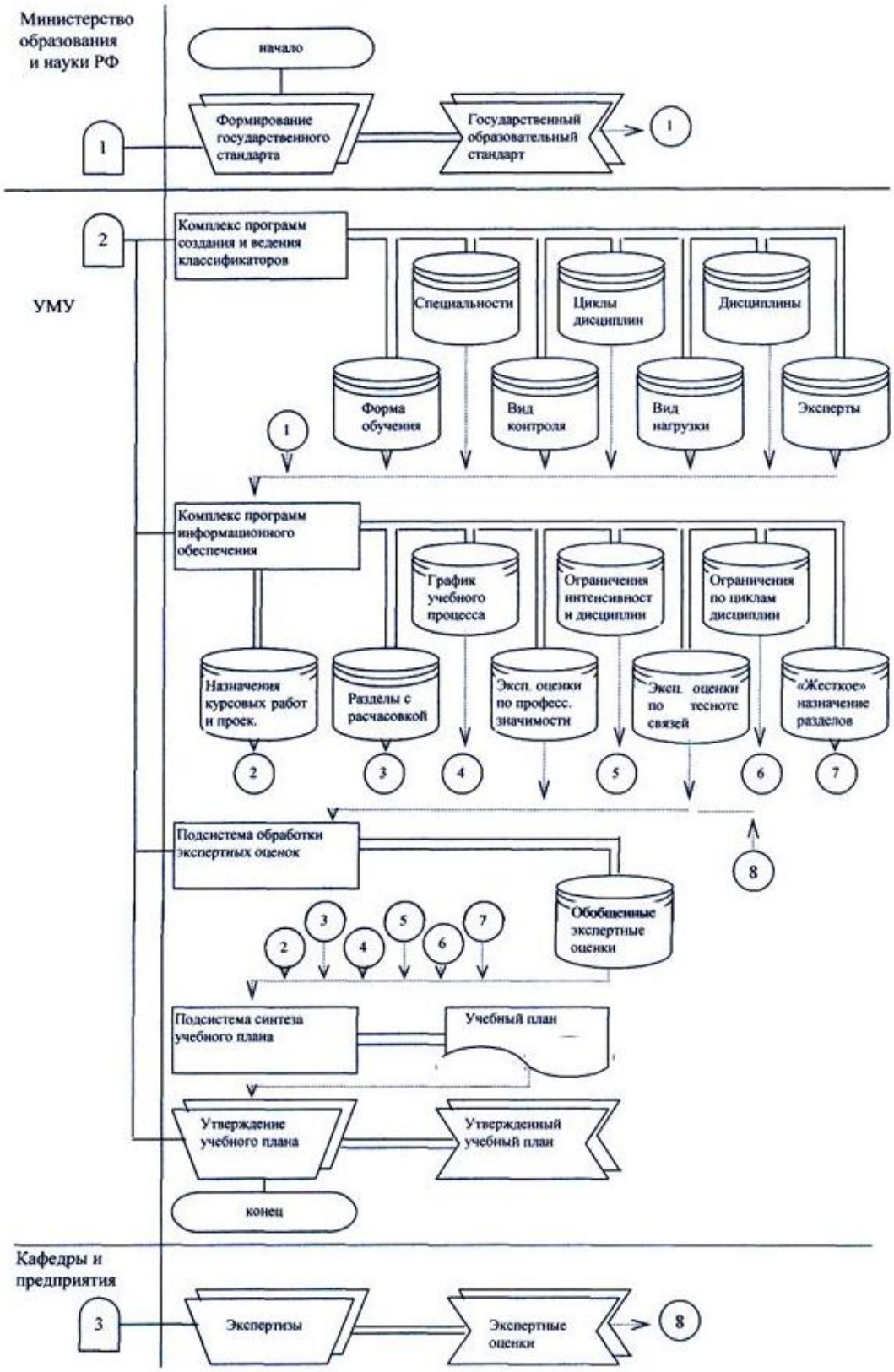
Некоторым недостатком такого подхода является то, что в модули помещается информация, не относящаяся непосредственно к изучаемой дисциплине. Причем информация фундаментальных наук для данной специальности (в частности, для инженерного образования – математика, физика и другие естественнонаучные дисциплины) может дублироваться несколько раз в различных модулях. Это, конечно же, положительно влияет на качество усвоения материала, но значительно сокращает общий объем учебного материала, который можно преподнести студенту за срок его обучения в вузе.

Кроме этого, недостаток модульного подхода проявляется в том, что при отборе наиболее связанных и часто используемых модулей для формирования учебного плана в него могут не войти модули, которые имеют малое количество связей, но при этом имеют большое значение для формирования знаний выпускника.

Нахождение и количественная оценка взаимосвязей между учебными дисциплинами позволяют устранить недостатки методики, основанной на построении дерева целей подготовки специалиста. Построение базы учебных дисциплин и нахождение места каждой учебной дисциплины в учебном плане, основанного на степени важности каждого модуля для вклада в формирование бакалавра, позволяют избежать дублирования изучаемой информации и перераспределить время на изучение наиболее важных модулей, что устраняет недостаток методики управления учебным процессом на основе модульного обучения.

Кроме того, в связи с переходом к новым образовательным стандартам, состав дисциплин стал более вариативным, происходит переход от почти статического набора дисциплин к динамическому.

Технологический процесс синтеза учебных планов вузов.



LDA

Латентное размещение Дирихле — применяемая в машинном обучении и информационном поиске порождающая модель, позволяющая

объяснять результаты наблюдений с помощью неявных групп, благодаря чему возможно выявление причин сходства некоторых частей данных. Например, если наблюдениями являются слова, собранные в документы, утверждается, что каждый документ представляет собой смесь небольшого количества тем и что появление каждого слова связано с одной из тем документа. LDA является одним из методов тематического моделирования и впервые был представлен в качестве графовой модели для обнаружения тематик Дэвидом Блеем, Эндрю Ыном и Майклом Джорданом в 2003 году.

В LDA каждый документ может рассматриваться как набор различных тематик. Подобный подход схож с вероятностным латентно-семантическим анализом (pLSA) с той разницей, что в LDA предполагается, что распределение тематик имеет в качестве априори распределения Дирихле. На практике в результате получается более корректный набор тематик.

К примеру, модель может иметь тематики классифицируемые как «относящиеся к кошкам» и «относящиеся к собакам», тематика обладает вероятностями генерировать различные слова, такие как «мяу», «молоко» или «котёнок», которые можно было бы классифицировать как «относящиеся к кошкам», а слова, не обладающие особой значимостью (к примеру, служебные слова), будут обладать примерно равной вероятностью в различных тематиках.

Научение

Итеративное научение, как обучение в строго повторяющихся условиях – одна из простейших разновидностей научения, имеет место в широком классе явлений: формирование разнообразных навыков, усвоение информации человеком, научение животных (выработка условных рефлексов) и обучение технических и кибернетических систем. Различные аспекты итеративного научения исследуются в педагогике, психологии и физиологии человека и животных, в теории управления и в других науках.

Многочисленные экспериментальные данные свидетельствуют, что важнейшей общей закономерностью итеративного научения в живых системах (человек, группы людей, животные) и неживых системах (системы распознавания образов, вероятностные автоматы с переменной структурой, нейронные сети и др.) является замедленно-асимптотический характер кривых научения: они монотонны, скорость изменения критерия уровня научения со временем уменьшается, а сама кривая асимптотически стремится к некоторому пределу. В большинстве случаев кривые итеративного научения аппроксимируются экспоненциальными кривыми.

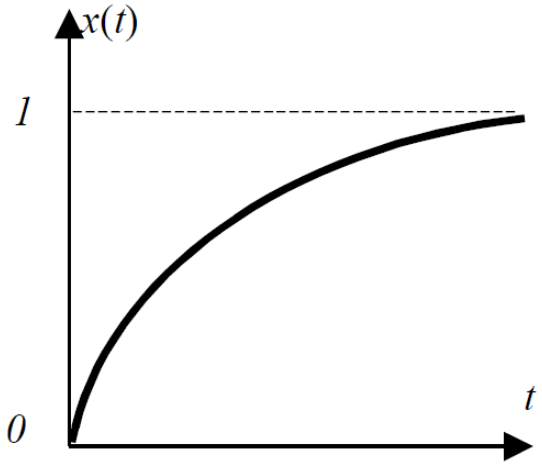


Рис 2.2а

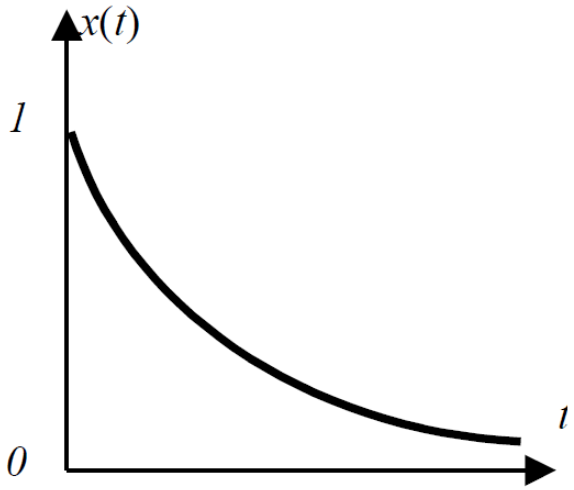


Рис 2.2а

Нормированные кривые итеративного научения.

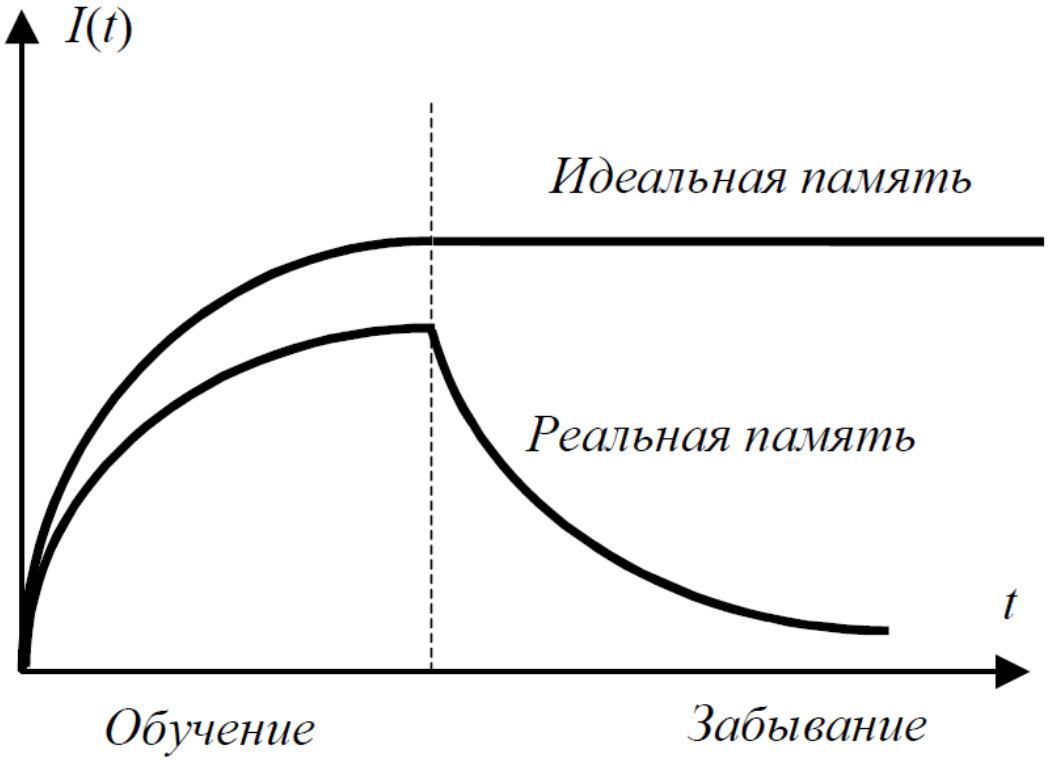


Рис. 6.1. Количество запомненной информации

Закон итеративного научения:

Если число элементов научаемой системы достаточно велико и/или внешние и внутренние условия ее функционирования стационарны, то кривая научения, соответствующая его результативным характеристикам, будет примерно экспоненциальной.

Адаптивные тесты – IRT

Адаптивные тесты

Достоинства тестовых методов – объективный характер оценок, сопоставимость и возможность их проверки – обеспечиваются не автоматически, а благодаря выполнению определенных требований к качеству заданий и тестов в целом. Плохой тест, не отвечающий по тем или иным характеристикам определенным критериям качества, может послужить источником искаженной информации о знаниях учеников. Причем скорректировать ее в процессе тестирования никак нельзя, если, конечно, процесс тестирования носит массовый характер, а не организован в

адаптивном режиме. Отсюда проистекает необходимость научного обоснования качества тестовых материалов.

Очевидно, что в настоящее время проблема адаптивных тестов чрезвычайно актуальна. У истоков адаптивного тестирования лежало стремление к повышению эффективности тестовых измерений, что, как правило, связывалось с уменьшением числа заданий, времени, стоимости тестирования и с повышением точности оценок, полученных испытуемыми по результатам выполнения теста.

Исследователи видели возможность повышения эффективности в адаптации тестов, трудность которых учитывала диапазон подготовленности тестируемых.

Преимущества применения адаптивного тестирования для контроля знаний в математике:

1. Контроль знаний осуществляется проверкой решения задач, что не всегда осуществимо в других дисциплинах;

2. Задачи можно расположить по уровню трудности (например по количеству этапов решения), это и создает предпосылки для адаптивного тестирования и объективного оценивания в автоматизированном режиме.

Настройки алгоритма адаптации могут варьироваться в зависимости от общего уровня группы, предпочтения преподавателя или студента, стоящей задачи.

Настройки адаптации задает либо преподаватель, либо студент.

Условия начала тестирования:

1. Тест существует в системе, в нем содержится необходимое количество тем (может быть как одна, так и несколько тем);

2. Преподавателем или студентом выбран начальный уровень тестирования. Если уровень выбран преподавателем, у студента нет возможности выбора уровня;

3. Выбрано количество вопросов, на которые необходимо ответить, чтобы перейти на следующий уровень сложности и на следующую тему.

Рекомендуемые настройки:

• если тема в тесте одна — количество вопросов на уровне сложности равно n;

• если тем в тесте несколько, но меньше 7 — количество вопросов больше одного и меньше n;

• если тем в тесте больше 7 — количество вопросов равно 1.

Данные цифры обоснованы научными исследования о когнитивных возможностях мозга. Преподаватель может менять настройки, однако это не рекомендуется, так как при большом количестве тем в тесте большое количество вопросов вызывает утомляемость и путаницу, и тестирование не даст адекватной оценки уровня знаний.

Условия перехода к следующей теме:

1. Несколько раз достигнут один и тот же уровень сложности

2. Переход на следующий уровень сложности, при текущем уровне «5»

Условия выхода из теста (при тестировании по одной теме):

1. Несколько раз достигнут один и тот же уровень сложности

2. Переход на следующий уровень сложности, при текущем уровне сложности «5». Итоговая оценка — 5 баллов.

3. Переход на уровень сложности «2». Итоговая оценка — 2 балла.

4. Не осталось вопросов заданного уровня сложности. Преподавателю выводится статистика ответов на вопросы, после чего он принимает решение о том, какую оценку ставить.

Психологические особенности адаптивного тестирования:

1. В основу проведения тестирования положен принцип четкого понимания испытуемыми, что необходимо делать.

2. Адаптивный тест может определить уровень знаний тестируемого с помощью меньшего количества вопросов. При выполнении одного и того же адаптивного теста тестируемые с высоким уровнем подготовки и тестируемые с низким уровнем подготовки увидят совершенно разные наборы вопросов: первый увидит большее число сложных вопросов, а

последний - легких. Доли правильных ответов у обоих могут совпадать, но так как первый отвечал на более сложные вопросы, то он наберет большее количество баллов.

3. Студент проходит тест спокойно, никто сторонний его не подгоняет.

Система выключится автоматически, как только истечет время тестирования.

4. После прохождения теста студент может попросить индивидуальной беседы с преподавателем, если результаты теста его не утроили. Преподаватель, в свою очередь, может дать свой ответ на основе статистики прохождения студентом теста.

Разумеется, тестирование не заменяет и не отменяет традиционных форм педагогического контроля, основанных на непосредственном общении преподавателя со студентом. Такой контроль выполняет важные обучающие функции, вооружает преподавателей информацией об уровне знаний учащихся.

Однако традиционные формы педагогического контроля носят во многом субъективный характер и не позволяют получить сопоставимые данные, необходимые для управления процессом образования.

IRT

Под современной теорией тестов понимается распространенная, на западе Item Response Theory (IRT), нацеленная на оценивание латентных качеств личности и параметров заданий теста на, основе математико-статистических моделей измерения IRT является частью более общей теории латентно-структурного анализа, хотя каждое из направлений имеет свои характерные особенности. В частности, в теории латентно-структурного анализа оцениваемые – значения параметров рассматриваются как некоторые дискретные точки на оси латентной переменной, в то время как в Item Response Theory распределения переменных предполагаются непрерывными.

К наиболее значимым преимуществам IRT обычно относят:

- устойчивые, объективные оценки параметра, характеризующего уровень знаний испытуемых. Устойчивость можно считать наиболее важным преимуществом IRT. Источником ее является относительная инвариантность оценок параметра испытуемых от трудности заданий, включенных в тест;
- устойчивые, объективные оценки параметра трудности заданий, не зависящие от свойств выборки испытуемых, выполняющих тест;
- измерение значений параметров испытуемых и заданий теста в одной и той же шкале, имеющей свойства интервальной шкалы.

Первоначально в IRT вводится основное предположение о существовании некоторой взаимосвязи между наблюдаемыми результатами тестирования и латентными (скрытыми от непосредственного наблюдения) качествами испытуемых, выполняющих тест. Обычно эти латентные качества трактуются как способности испытуемых или как уровни знаний по предмету в зависимости от целей измерения, которые выдвигаются при создании педагогического теста.

Предполагается, что каждому испытуемому ставится в соответствие только одно значение латентного параметра определяющего наблюдаемые результаты выполнения теста.

Очевидно, что логика должна быть такова: латентные параметры, вернее, взаимодействие двух множеств их значений порождает наблюдаемые результаты выполнения теста. Элементы первого множества — это значения латентного параметра, определяющие уровни знаний *N* испытуемых Θ_i, где i=1, 2,..., *N*. Второе множество образуют значения латентного параметра β_j, j=I, 2,..., *n,* равные трудностям *n* заданий теста. Однако на практике всегда ставится обратная задача: по ответам испытуемых на задания теста оценить значения латентных параметров Θ и β. Для ее решения нужно ответить по меньшей мере на два вопроса.

Первый связан с выбором вида соотношения между латентными параметрами Θ и β. Датский математик Georg Rasch предложил ввести это соотношение в виде разности Θ-β, предполагая, что параметры Θ и β оцениваются в одной и той же шкале.

Значение параметра Θ_i можно рассматривать как положение i-го испытуемого, а значение β_j - как положение j-го задания на одной и той же оси переменных Θ, β. В таком случае идея введения разности параметров получает интересную геометрическую интерпретацию. Абсолютная величина разности |Θ-β| - это расстояние, на котором находится испытуемый с уровнем знаний Θ от задания с трудностью β. Если эта разность велика по модулю и отрицательна, то задание бесполезно для измерения уровня знаний i-го студента. Обучаемый наверняка не сможет выполнить его правильно. С другой стороны, большие положительные значения этой разности тоже не представляют интереса ни для процесса контроля, ни для обучения f-го испытуемого. Задание такой трудности давно им освоено. С точки зрения подхода, предлагаемого в IRT, такие задания неэффективны для оценивания данного значения Θ.

Ответ на второй вопрос, который является центральным в IRT, связан с выбором математической модели. Следуя основному, предположению IRT, можно утверждать, что есть некоторая математическая модель взаимосвязи между эмпирическими результатами тестирования и значениями латентных переменных Θ и β.

Можно рассматривать условную вероятность правильного выполнения i-ым испытуемым с уровнем знаний Θ_i различных по трудности заданий теста, считая Θ_i параметром i-го студента а β_j - независимой переменной. В этом случае условная вероятность будет функцией латентной переменной β.

$$P_i\{x_{ij}=1|\Theta_i\}=f(\Theta_i - \beta)\ i=1..N$$

Аналогично вводится условная вероятность правильного выполнения j-го задания, трудностью β_j, различными испытуемыми группы. Здесь независимой переменной является Θ, а β_j - параметр, определяющий трудность j-го задания теста:

$$P_i\{x_{ij}=1|\beta_j\}=f(\Theta-\beta_j)\ i=1..N,$$

где $x_{ij}=\{0,1\}$, 1, если ответ *i*-го испытуемого на j-е задание теста правильный; и 0, если ответ *i*-го испытуемого на j-е задание, теста неправильный.

N — число испытуемых;

n — количество заданий в тесте.

В теории IRT функции *f(*β*)* и φ(Θ) получили название "Item response functions" (IRF). Специальное название имеют и их графики: график функции P_j - это характеристическая кривая j-го задания (ICC), а график функции P_i - индивидуальная кривая i-го испытуемого (PCC).

В предположении нормального распределения значений латентных переменных Θ и β предлагается две таких функции. Одна из них, обычно обозначаемая символом Ψ(*x*), относится к семейству логистических кривых; другая, Φ(*x*) является интегральной функцией нормированного нормального распределения.

Наиболее сильный аргумент в пользу логистической функции связан не с качеством измерений, а с относительной простотой ее аналитического задания, выгодной при оценивании параметров Θ и β. Поэтому в практических приложениях предпочтение обычно отдают функции Ψ(l,7*x*). Число параметров, входящих в аналитическое задание функций, является основанием для подразделения семейства IRF на классы. Среди логистических функций различают: однопараметрическую модель G.Rasch,

двухпараметрическую модель A.Birnbaum, трехпараметрическую модель A.Birnbaum.

Байесовская классификация

Байесовский классификатор — широкий класс алгоритмов классификации, основанный на принципе максимума апостериорной вероятности. Для классифицируемого объекта вычисляются функции правдоподобия каждого из классов, по ним вычисляются апостериорные вероятности классов. Объект относится к тому классу, для которого апостериорная вероятность максимальна.

Байесовский подход к классификации основан на теореме, утверждающей, что если плотности распределения каждого из классов известны, то искомый алгоритм можно выписать в явном аналитическом виде. Более того, этот алгоритм оптимален, то есть обладает минимальной вероятностью ошибок.

На практике плотности распределения классов, как правило, не известны. Их приходится оценивать (восстанавливать) по обучающей выборке. В результате байесовский алгоритм перестаёт быть оптимальным, так как восстановить плотность по выборке можно только с некоторой погрешностью. Чем короче выборка, тем выше шансы подогнать распределение под конкретные данные и столкнуться с эффектом переобучения.

Байесовский подход к классификации является одним из старейших, но до сих пор сохраняет прочные позиции в теории распознавания. Он лежит в основе многих достаточно удачных алгоритмов классификации.

К числу байесовских методов классификации относятся:

- Наивный байесовский классификатор
- Линейный дискриминант Фишера

- Квадратичный дискриминант
- Метод парзеновского окна
- Метод радиальных базисных функций (RBF)
- Логистическая регрессия

Марковские цепи

Цепь Маркова — последовательность случайных событий с конечным или счётным числом исходов, где вероятность наступления каждого события зависит от состояния, достигнутого в предыдущем событии. Характеризуется тем свойством, что, говоря нестрого, при фиксированном настоящем будущее независимо от прошлого. Названа в честь А. А. Маркова (старшего), который впервые ввёл это понятие в работе 1906 года.

Одно из свойств, сильно упрощающее исследование случайного процесса — это «марковское свойство». Если объяснять очень неформальным языком, то марковское свойство сообщает нам, что если мы знаем значение, полученное каким-то случайным процессом в заданный момент времени, то не получим никакой дополнительной информации о будущем поведении процесса, собирая другие сведения о его прошлом. Более математическим языком: в любой момент времени условное распределение будущих состояний процесса с заданными текущим и прошлыми состояниями зависит только от текущего состояния, но не от прошлых состояний (свойство отсутствия памяти). Случайный процесс с марковским свойством называется марковским процессом.

Марковское свойство обозначает, что если мы знаем текущее состояние в заданный момент времени, то нам не нужна никакая дополнительная информация о будущем, собираемая из прошлого.

На основании этого определения можно сформулировать определение «однородных цепей Маркова с дискретным временем» (в дальнейшем для простоты будем называть «цепями Маркова»). Цепь Маркова — это

марковский процесс с дискретным временем и дискретным пространством состояний. Итак, цепь Маркова — это дискретная последовательность состояний, каждое из которых берётся из дискретного пространства состояний (конечного или бесконечного), удовлетворяющее марковскому свойству.

Математически можно обозначить цепь Маркова так:

$$X = (X_n)_{n \in \mathbb{N}} = (X_0, X_1, X_2, ...)$$

где в каждый момент времени процесс берёт свои значения из дискретного множества E, такого, что

$$X_n \in E \qquad \forall n \in \mathbb{N}$$

Тогда марковское свойство подразумевает, что у нас есть

$$\mathbb{P}(X_{n+1} = s_{n+1} | X_n = s_n, X_{n-1} = s_{n-1}, X_{n-2} = s_{n-2}, ...) = \mathbb{P}(X_{n+1} = s_{n+1} | X_n = s_n)$$

Благодаря марковскому свойству динамику цепи Маркова определить довольно просто. И в самом деле нужно определить только два аспекта: исходное распределение вероятностей (то есть распределение вероятностей в момент времени n=0), обозначаемое

$$\mathbb{P}(X_0 = s) = q_0(s) \qquad \forall s \in E$$

и матрицу переходных вероятностей (которая даёт вероятности того, что состояние в момент времени n+1 является последующим для другого

состояния в момент n для любой пары состояний), обозначаемую

$$\mathbb{P}(X_{n+1} = s_{n+1} | X_n = s_n) = p(s_n, s_{n+1}) \qquad \forall (s_{n+1}, s_n) \in E \times E$$

Если два этих аспекта известны, то полная (вероятностная) динамика процесса чётко определена.

Свойства марковских цепей: разложимость, периодичность, невозвратность и возвратность

Марковское свойство цепей Маркова сильно облегчает изучение этих процессов и позволяет вывести различные интересные явные результаты (среднее время возвратности, стационарное распределение…)

Мощным инструментом являются цепи Маркова при моделировании задач, связанных со случайной динамикой. Благодаря их хорошим свойствам они используются в различных областях, например, в теории очередей (оптимизации производительности телекоммуникационных сетей, в которых сообщения часто должны конкурировать за ограниченные ресурсы и ставятся в очередь, когда все ресурсы уже заняты), в статистике (хорошо известные методы Монте-Карло по схеме цепи Маркова для генерации случайных переменных основаны на цепях Маркова), в биологии (моделирование эволюции биологических популяций), в информатике (скрытые марковские модели являются важными инструментами в теории информации и распознавании речи), а также в других сферах.

Seq2Seq

Seq2Seq модели - наиболее часто используемая архитектура в машинном переводе и нейросетевых вопросно-ответных системах. Наибольшее количество памяти в таких моделях расходуется на хранение матрицы представлений, содержащей представление каждого слова из словаря. Для пословной генерации согласованного ответа требуется иметь

наряду со стандартной формой слова еще и все его словоформы. В некоторых языках у слов имеется лишь небольшое количество словоформ (например, единственное и множественное число). Тем не менее, в таких языках как русский, у многих слов присутствует большое число словоформ, получающихся изменением рода, числа, падежа и времени.

Модели со словарями, достаточно полно покрывающими множество всех словоформ, превышают разумные ограничения как по времени, так и по памяти. Во многих работах для обхода этой проблемы переходят к посимвольным моделям. В таких моделях размер словаря соответствует размеру используемого алфавита. Посимвольная генерация позволяет избегать хранения словоформ, однако из-за увеличения длины последовательности в несколько раз, модель быстро забывает начало предложения. Альтернативное решение хранение в словаре лишь стандартной формы слов. Такая модель будет генерировать несогласованный текст и не может быть использована в рабочей системе.

Одной из наиболее популярных архитектур в машинном переводе являются модели sequence to sequence (Seq2Seq). Такие модели состоят из двух рекуррентных сетей: кодировщика и декодировщика. Кодировщик строит представление входной последовательности слов. Далее полученное представление (последние выход и значение ячейки сети) копируются в декодировщик. По полученному представлению декодировщик пытается восстановить целевую последовательность слов. В задачах машинного перевода входной и выходной последовательностями являются предложения на разных языках. В вопросно-ответных и диалоговых системах вопрос и ответ.

Для преобразования слов во входные вектора используется так называемая матрица представлений (embedding matrix). Количество строк этой матрицы равно размеру словаря, а число столбцов размеру ячейки LSTM. Каждая строка соответствует векторному представлению

соответствующего слова. Каждое слово перед подачей на вход LSTM сети заменяется на соответствующую строку матрицы представлений.

На вход декодировщику на первом такте подается специальный символ <GO>, затем на каждом такте подается сгенерированное в предыдущую итерацию слово. Генерация ответа продолжается до тех пор, пока не будет сгенерировано специальное слово – маркер конца строки <EOL> (end of line). Во время обучения в качестве сгенерированного символа на следующий такт передается целевой символ, а распределение на предсказанных символах передается в функцию потерь.

Во время предсказания требуется найти наиболее вероятное предложение с точки зрения модели. Сделать это напрямую невозможно, так как модель позволяет вычислять только наилучшее слово при фиксированных предыдущих. Компромиссным решением между жадным выбором слов и полным перебором является Beam Search. При использовании этого метода на каждой итерации выбирается небольшое количество лучших кандидатов, а остальные гипотезы отбрасываются.

После применения Beam Search сеть часто начинает отвечать при помощи наиболее часто встречающихся в выборке ответов: да, нет, не знаю. Для борьбы с этим можно обучить две модели: предсказывающую ответ по вопросу и вопрос по ответу. Предложения, сгенерированные beam search первой модели, переранжируются согласно выпуклой комбинации логарифмов правдоподобий двух моделей:

$$\lambda \log P(A|Q) + (1 - \lambda) \log P(Q|A), \text{ где } Q \text{ — вопрос, } A \text{ — ответ, } \lambda \in [0, 1]$$

- гиперпараметр. Теперь частотные ответы будут иметь низкую вероятность, так как по ним редко возможно восстановить вопрос. Для генерации длинных предложений к функции ранжирования добавляется штраф, поощряющий генерацию большого числа слов.

Обычно в качестве такой фукнции выбирается $\gamma|A|$, где $|A|$ - число сгенерированных слов, а ɣ - гиперпараметр.

Декодер на выходе должен преобразовать последовательность на входе encoder в последовательность выходе декодера (ответ) неопределенной длины. Поэтому модель называется — seq2seq. В общем случае, длина последовательности на выходе декодера не равна длине последовательности на входе encoder. Например, на входе слово на русском, а на выходе на китайском. Очевидно, что длины предложений будут совпадать далеко не всегда.

При этом на этапе обучения длина выходной последовательности известна, поэтому обучать можно обычным способом, выровняв длины предложений добавлением 0-ей.

Декодер:

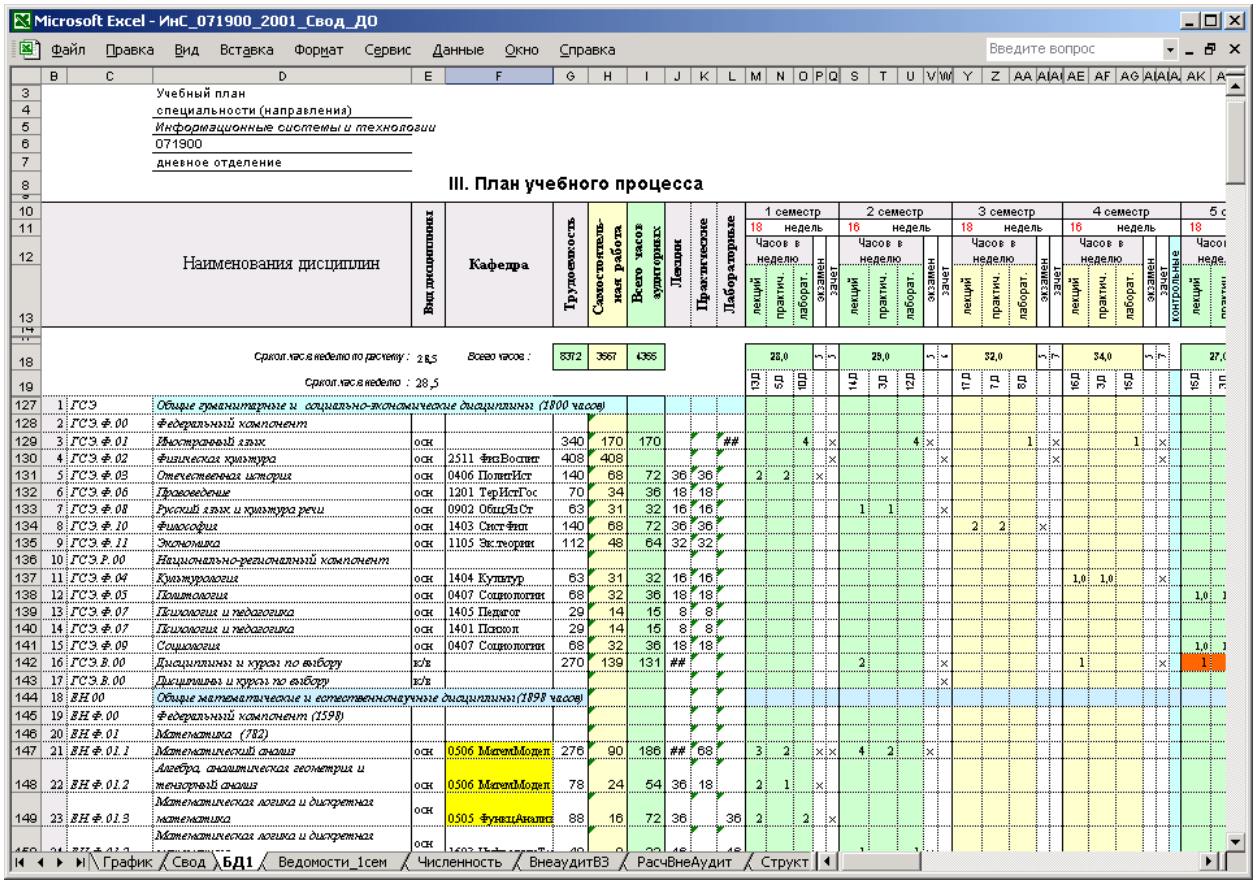
- Принимает на вход состояние («thought vector») с encoder-a через параметр initial_state рекуррентной сети.
- Последовательность в которую происходит преобразование дополняется специальными тегами:
    - Начало последовательности подаваемой на вход декодера в начале дополняется тегом <start> (может быть использован любой тег, например, <BOS>, гарантированно не встречающийся в тексте).
    - Конец последовательности подаваемой на выход декодера в конце дополняется тегом <end> (может быть использован любой тег, например, <EOS>, гарантированно не встречающийся в тексте).

- Последовательность дополненная тегом <start> преобразуется в пространство векторов с помощью embedding и подается на вход декодера.
- Последовательность индексов дополненная индексом тега <end> подается на выход декодера (dense слой) без преобразования в embedding. Модель обучается категоризации, т.е. выдавать индекс слова из словаря с некоторой вероятностью. Соответственно, в качестве активационной функции используется activation=’softmax’. Подавать на выход индексы вместо OHE можно, поскольку используется loss=’sparse_categorical_crossentropy’.
- Encoder и decoder обучаются в общей модели.

Теория + исследования

Учебный план

Примеры учебного плана, оформленного в виде таблицы в Excel:



Копия не для утверждения **Рабочий учебный план на 2019/2020 учебный год** Специальность/Направление 09.04.01/03 Кафедра ИУ5

Группа	Количество Студентов	Семестр	Учебных недель (Т+ГА)	Теоретические занятия (Т)*	Аттестация(Э)*	НИР(Н)*	Практика(П)*	ГИА(ГА)*	Каникулы(К)
ИУ5-11М ИУ5-12М ИУ5-13М ИУ5-14М	43 (5)	1	17	02.09.2019 - 30.12.2019	31.12.2019 - 24.01.2020	02.09.2019 - 30.12.2019			25.01.2020 - 06.02.2020
ИУ5-21М ИУ5-22М ИУ5-23М ИУ5-24М	43 (5)	2	17	07.02.2020 - 06.06.2020	08.06.2020 - 28.06.2020	07.02.2020 - 06.06.2020	07.02.2020 - 06.06.2020 29.06.2020 - 19.07.2020		20.07.2020 - 31.08.2020

* занятия в нерабочие праздничные дни не проводятся

п	1 семестр Дисциплины	Каф	Общая труд-ть З.е.	Общая труд-ть Ак.час	Виды учебной работы и трудоемкость, час: Контактная работа Итого	Лек	Упр	Лаб	Проч	Аттестация Зач	Дзач	Экз	Прочее Дзач	Оценка, баллы	График выполнения контрольных мероприятий: Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Т/Н	Э	Э	Э	Э	К
1	Аналитические модели автоматизированных систем обработки информации и управления	ИУ5	4	144	41	34	0	0	7	зач	-	-	КуР	50 50			%			РК	ДЗ	М %			%			ДЗ %	РК	М					
2	Иностранный язык	Л2	2	72	38.9	0	34	0	4.9	зач	-	-	-	33 33 34					РК	М		РК %		М			РК		М					
3	Методология научного познания	СГН4	2	72	36.4	34	0	0	2.4	зач	-	-	-	50 50							РК	М %							ДЗ	М				
4	Научно-исследовательская работа	ИУ5	3	108	108	0	0	0	108	-	-	-	НИРС				%					%			%									
5	Объектно-ориентированное проектирование автоматизированных систем обработки информации и управления	ИУ5	6	216	59.4	17	0	34	8.4	-	-	экз	КуР	35 35 30			%		ДЗ			РК	М %			ДЗ %			%	РК	М				
6	Теоретические основы реляционной алгебры	ИУ5	4	144	58.3	34	17	0	7.3	-	-	экз	-	35 35 30					ДЗ		ДЗ	РК	М					ДЗ	РК		М				
7	Теория планирования экспериментов	ИУ5	5	180	63.1	17	34	0	12.1	-	-	экз	КуР	35 35 30			%					РК %	М		%			%	РК	М				
8	Технологии разработки программного обеспечения	ИУ5	4	144	55.4	17	0	34	4.4	-	-	экз	-	35 35 30					РК		ДЗ		ДЗ	М				ДЗ	РК	М				
	Итого:		30	1080	460.4	153	85	68	154.4	3	0	4	4	Недели - числ -ням	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Трудоемкость в неделю - 51.4 академических часов Контактная нагрузка в неделю - 21.9 часов

п	2 семестр Дисциплины	Каф	Общая труд-ть З.е.	Общая труд-ть Ак.час	Виды учебной работы и трудоемкость, час: Контактная работа Итого	Лек	Упр	Лаб	Проч	Аттестация Зач	Дзач	Экз	Прочее Дзач	Оценка, баллы	График выполнения контрольных мероприятий: Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Т/П/Н	Э	Э	Э	П	П
1	Иностранный язык	Л2	2	72	38.9	0	34	0	4.9	зач	-	-	-	33 33 34					РК	М			РК	М				РК		М					
2	Методы машинного обучения	ИУ5	3	108	70.8	34	0	34	2.8	-	-	р.экз	-	40 60							РК	М						РК	ДЗ	М				
3	Научно-исследовательская работа	ИУ5	4	144	144	0	0	0	144	-	-	-	НИРС				%					%			%			%						
4	Основы предпринимательства	ИБМ6	2	72	36.4	34	0	0	2.4	зач	-	-	-	30 30 40					РК	М			РК	М				РК		М				
5	Педагогическая практика	ИУ5	1	36	36	0	0	0	36	-	-	-	Пр																					
6	Постреляционные базы данных	ИУ5	4	144	55	17	0	34	4	-	-	экз	-	35 35 30							ДЗ		М					ДЗ	РК	М				
7	Проектирование интеллектуальных систем	ИУ5	4	144	55.1	17	0	34	4.1	-	-	экз	-	35 35 30							РК		ДЗ	М				ДЗ	РК	М				
8	Проектно-технологическая практика	ИУ5	2	72	72	0	0	0	72	-	-	-	Пр																					
9	Структурное проектирование автоматизированных систем обработки информации и управления	ИУ5	3	108	55.5	34	0	17	4.5	-	-	экз	-	35 35 30									РК	М					РК	М				
10	Технологии разработки программного обеспечения	ИУ5	2	72	4.3	0	0	0	4.3	-	-	-	КуР				%					%			%			%						
11	Эксплуатационная практика	ИУ5	3	108	108	0	0	0	108	-	-	-	П																					
	Итого:		30	1080	675.9	136	34	119	386.9	2	0	4	5	Недели - числ -ням	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Трудоемкость в неделю - 48.6 академических часов Контактная нагрузка в неделю - 29.4 часов

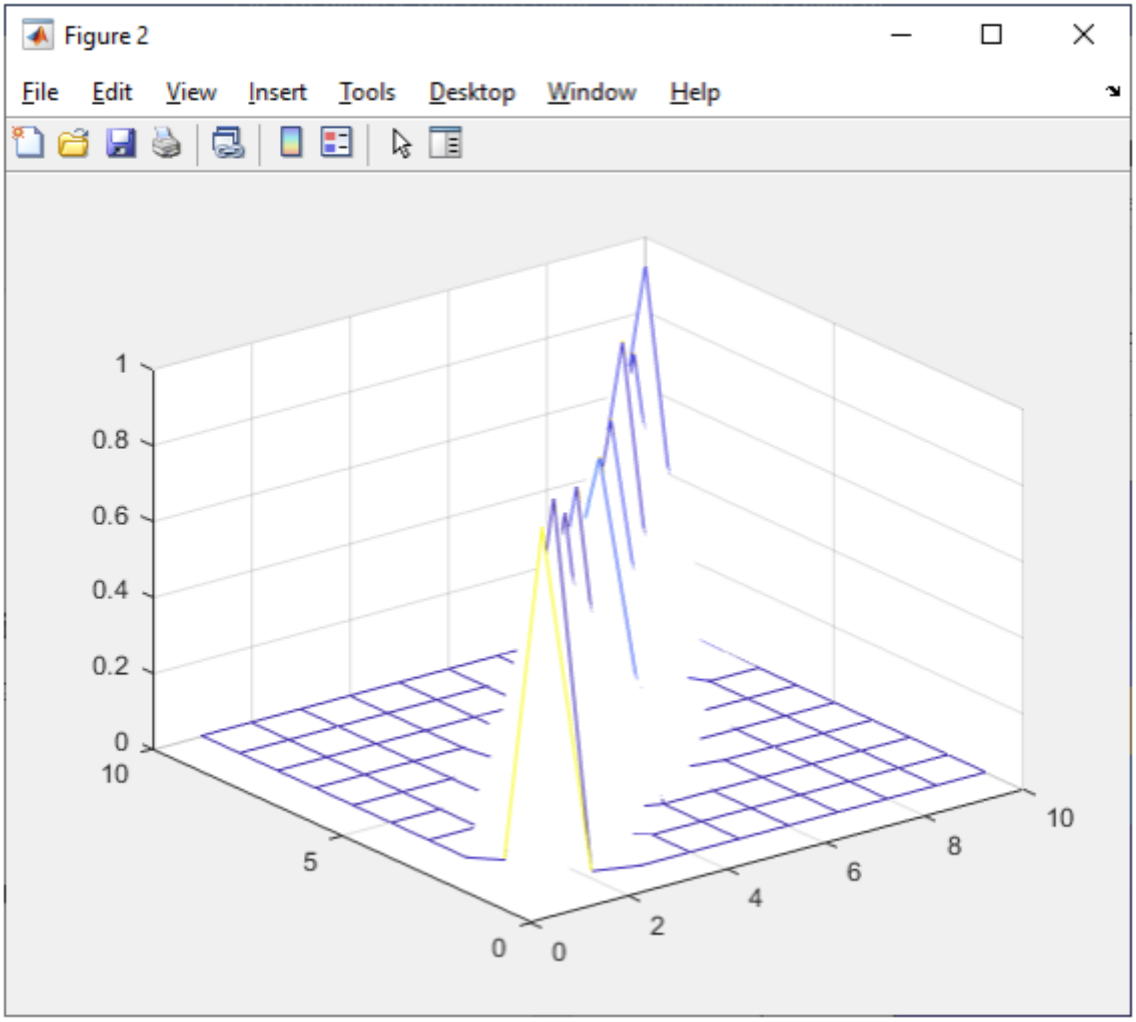
Ошибочные классификации

Ниже представлены матрицы ошибочных классификаций, полученные в ходе исследовательской работы при моделировании прохождения статических и адаптивных тестов, и их различные визуализации.

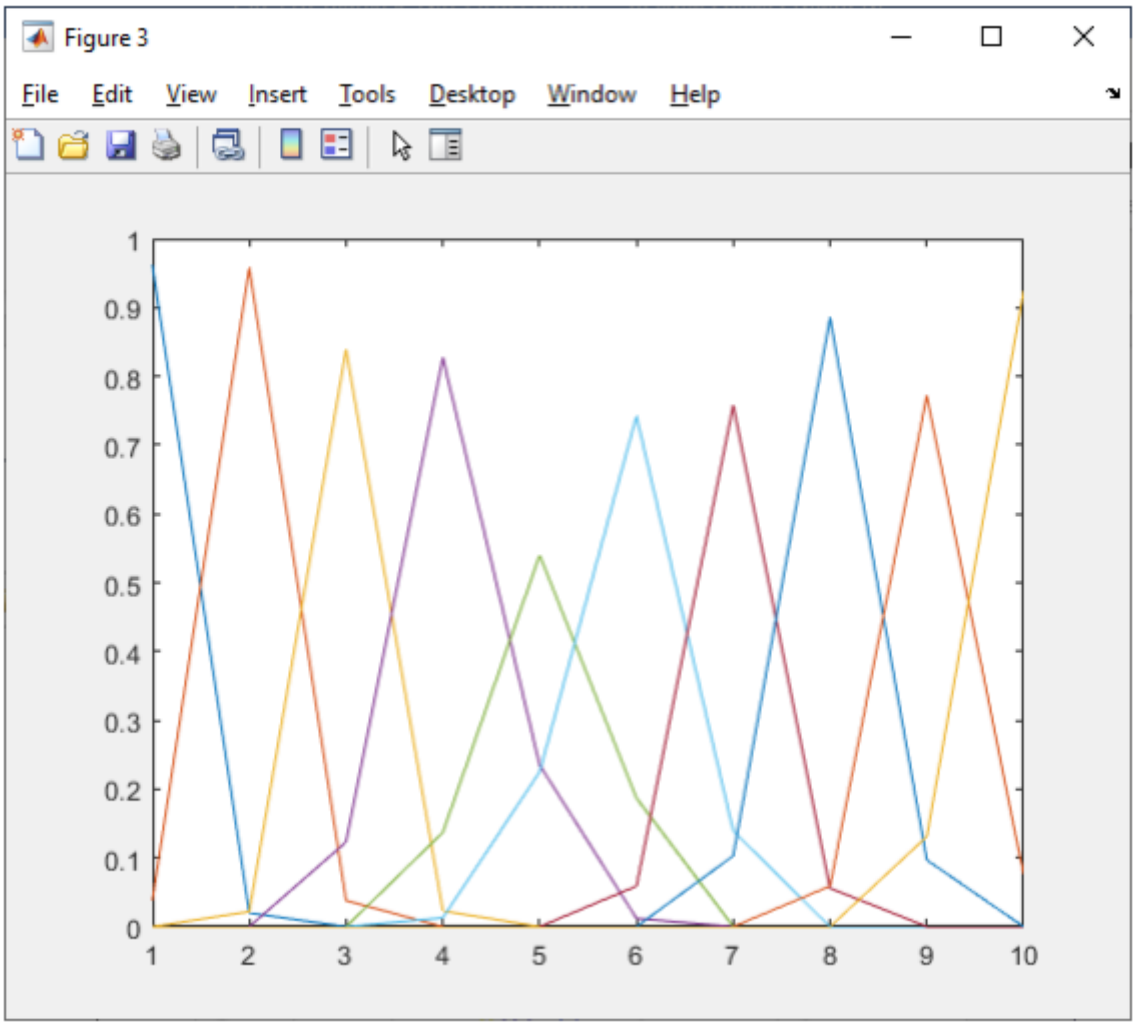
Матрица ошибочной классификации для **простого статического тестирования**

0.961	0.039	0	0	0	0	0	0	0	0
0.02	0.958	0.022	0	0	0	0	0	0	0
0	0.038	0.839	0.123	0	0	0	0	0	0
0	0	0.023	0.827	0.137	0.013	0	0	0	0
0	0	0	0.235	0.54	0.225	0	0	0	0
0	0	0	0.012	0.187	0.742	0.059	0	0	0
0	0	0	0	0	0.139	0.758	0.103	0	0
0	0	0	0	0	0	0.056	0.885	0.059	0
0	0	0	0	0	0	0	0.097	0.772	0.131
0	0	0	0	0	0	0	0	0.076	0.924

Трехмерная визуализация матрицы



Двумерная визуализация

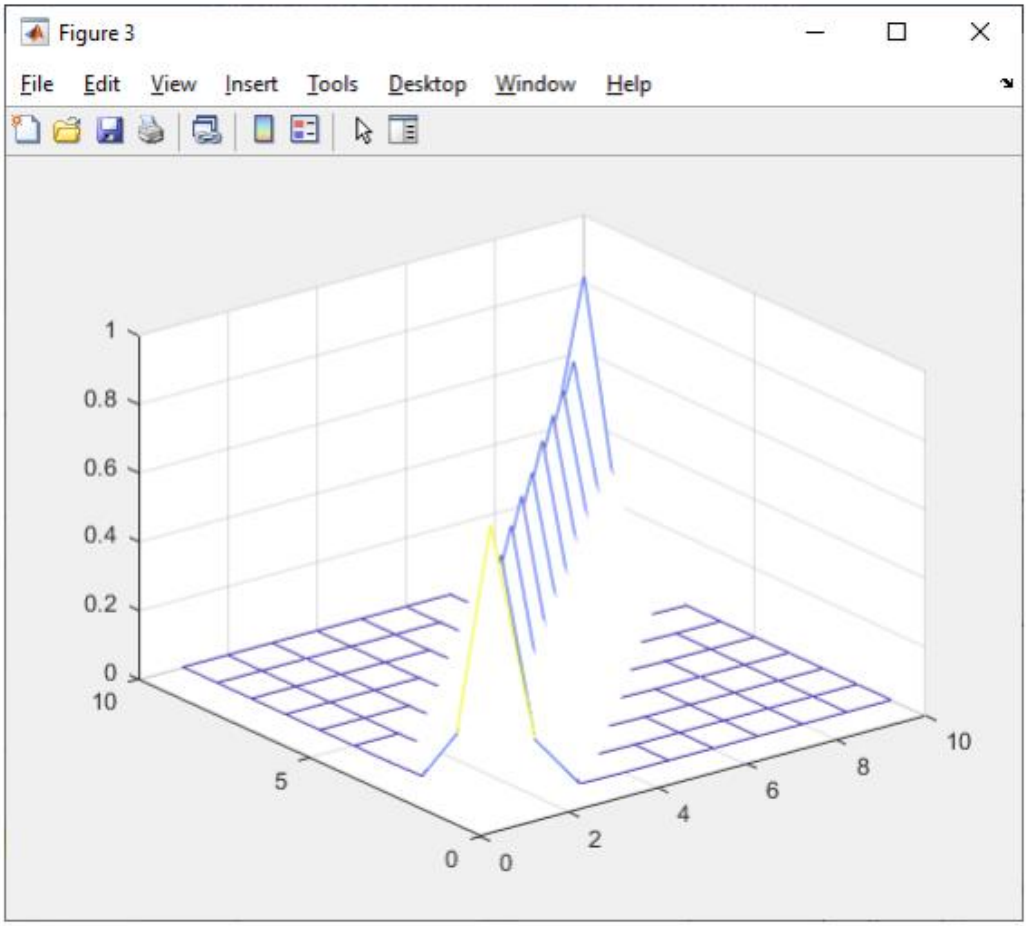


Адаптивный тест

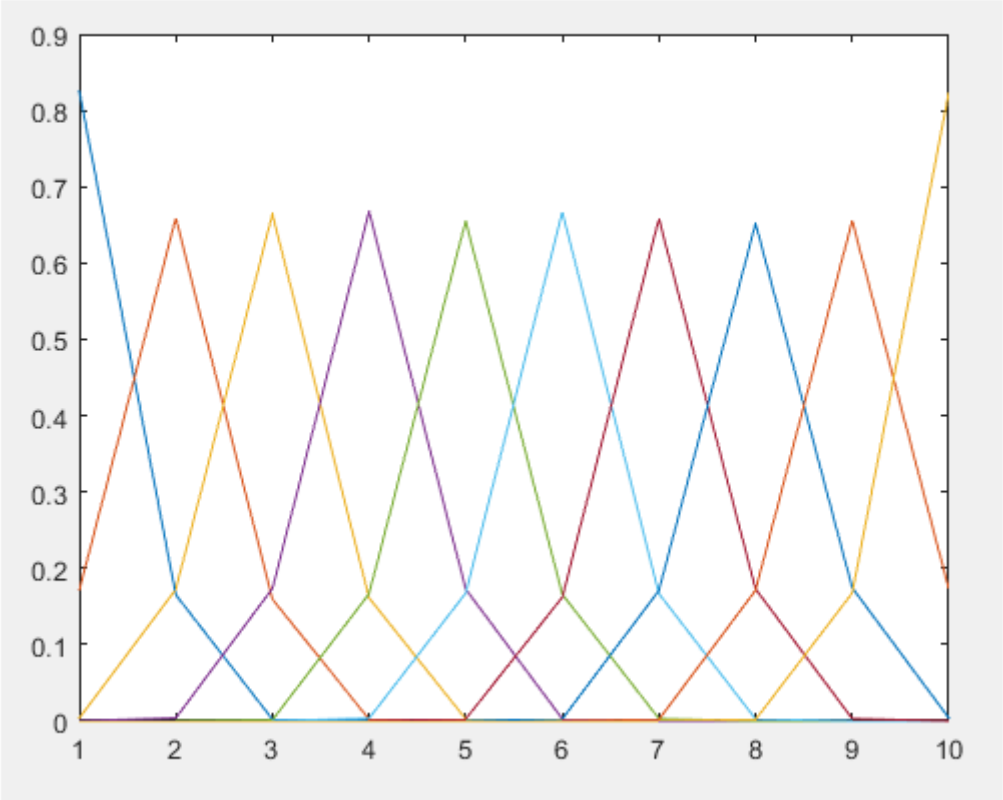
Матрица ошибочной классификации

0.8208	0.1773	0.0019	0	0	0	0	0	0	0
0.1753	0.6478	0.1741	0.0028	0	0	0	0	0	0
0.0021	0.154	0.6719	0.1695	0.0025	0	0	0	0	0
0	0.0019	0.1652	0.6609	0.1696	0.0024	0	0	0	0
0	0	0.0013	0.1695	0.648	0.1792	0.002	0	0	0
0	0	0	0.0021	0.1618	0.6705	0.1634	0.0022	0	0
0	0	0	0	0.0033	0.1706	0.6504	0.1732	0.0025	0
0	0	0	0	0	0.0024	0.1721	0.6497	0.174	0.0018
0	0	0	0	0	0	0.0021	0.1767	0.6525	0.1687
0	0	0	0	0	0	0	0.0035	0.1738	0.8227

Трехмерная визуализация



Двумерная визуализация



СГВ на основе Seq2Seq

Умение задавать релевантные и умные вопросы всегда было неотъемлемой частью человеческого обучения, так как позволяло оценить понимание текста(например, при аудировании, чтении статей).

Автоматизированные системы генерации вопросов могут смягчить эту проблему, эффективно обучаясь на больших массивах данных.

Система генерации вопросов(СГВ) может иметь много областей применения таких, как создание ответов на популярные вопросы, интеллектуальные системы обучения, автоматизация понимания прочитанного, виртуальные помощники/собеседники.

Задачей СГВ является создание синтаксически согласованных, семантически корректных и естественных вопросов по тексту.

Основанные на нейронных сетях модели Seq2Seq представляют современный подход к проблеме генерации вопросов. Большинство таких

моделей принимают в качестве входных данных одно предложение, ограничивая таким образом свою полезность в реальном мире.

Подав на вход отрывок текста пользователи могут вручную выбрать участки текста из автоматически сгенерированного набора субстантивных словосочетаний и именованных сущностей. Вопросы затем создаются посредством сочетания инновационной Seq2Seq модели с динамическими словарями, копировальным механизмом для расширения сети и механизмом избирательного внимания.

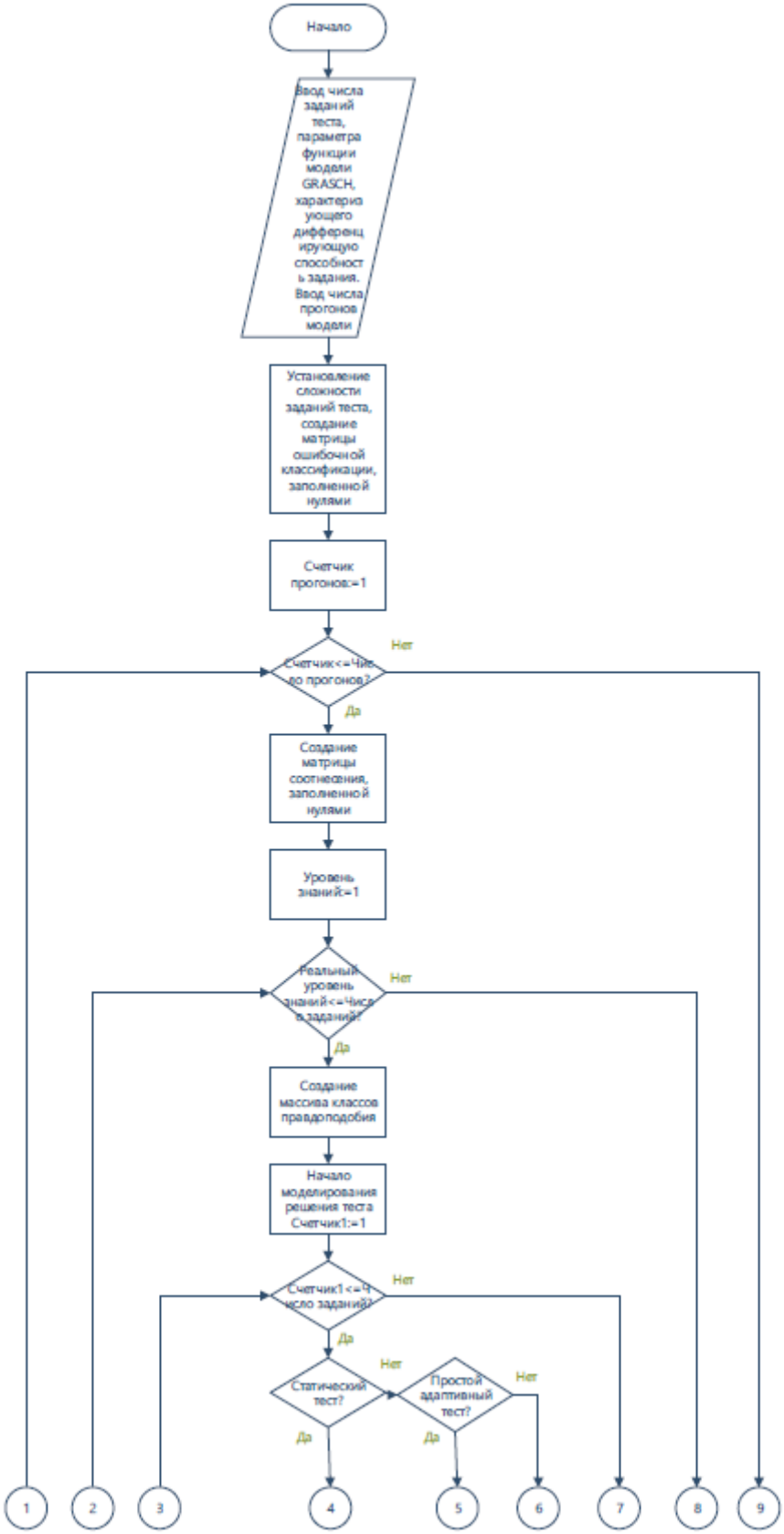
Новая техника фильтрации вопросов основана на предобученном BERT для удаления вопросов, на которые невозможно дать ответ.

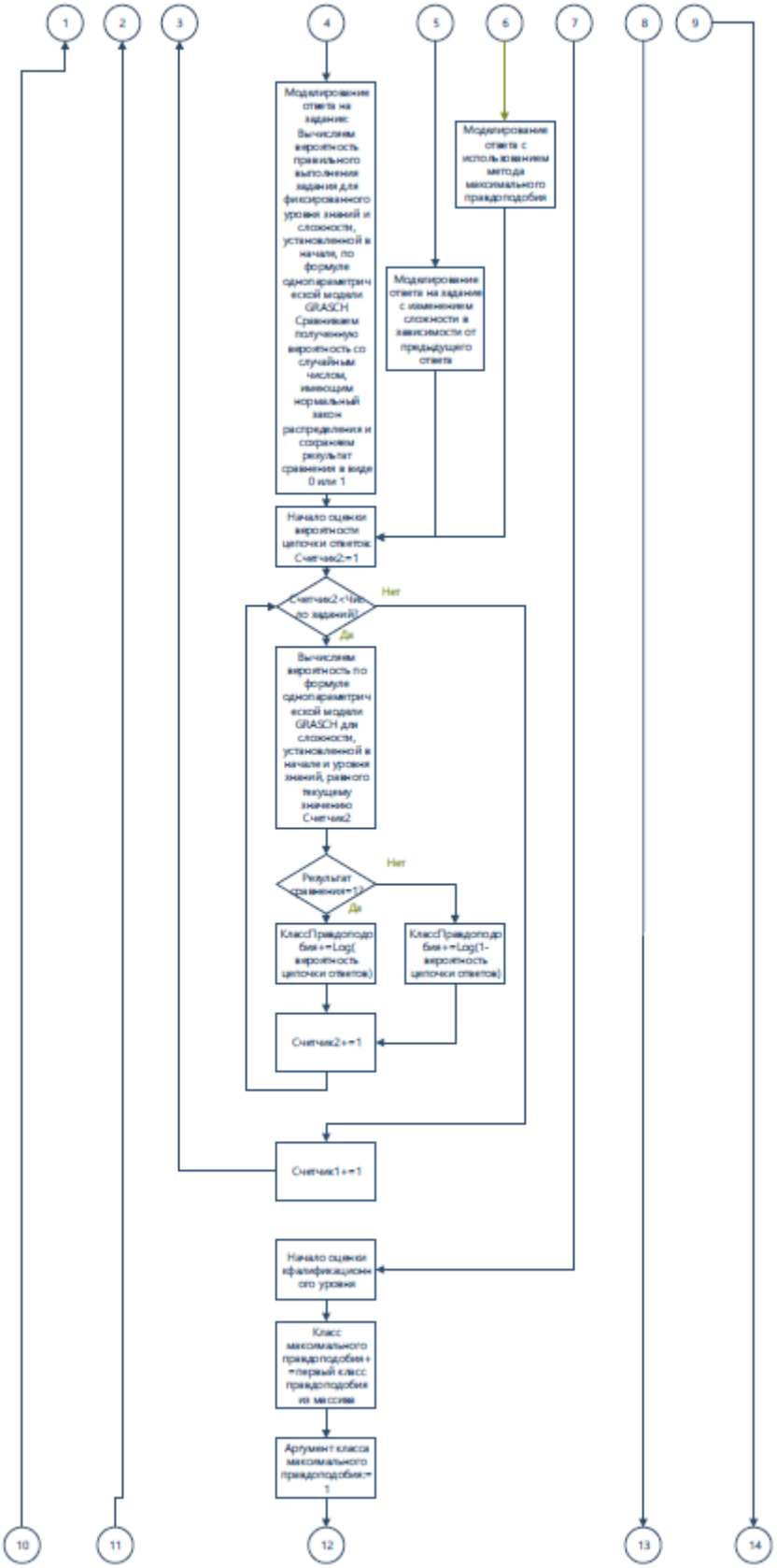
Архитектура системы, её принцип работы выглядят следующим образом:

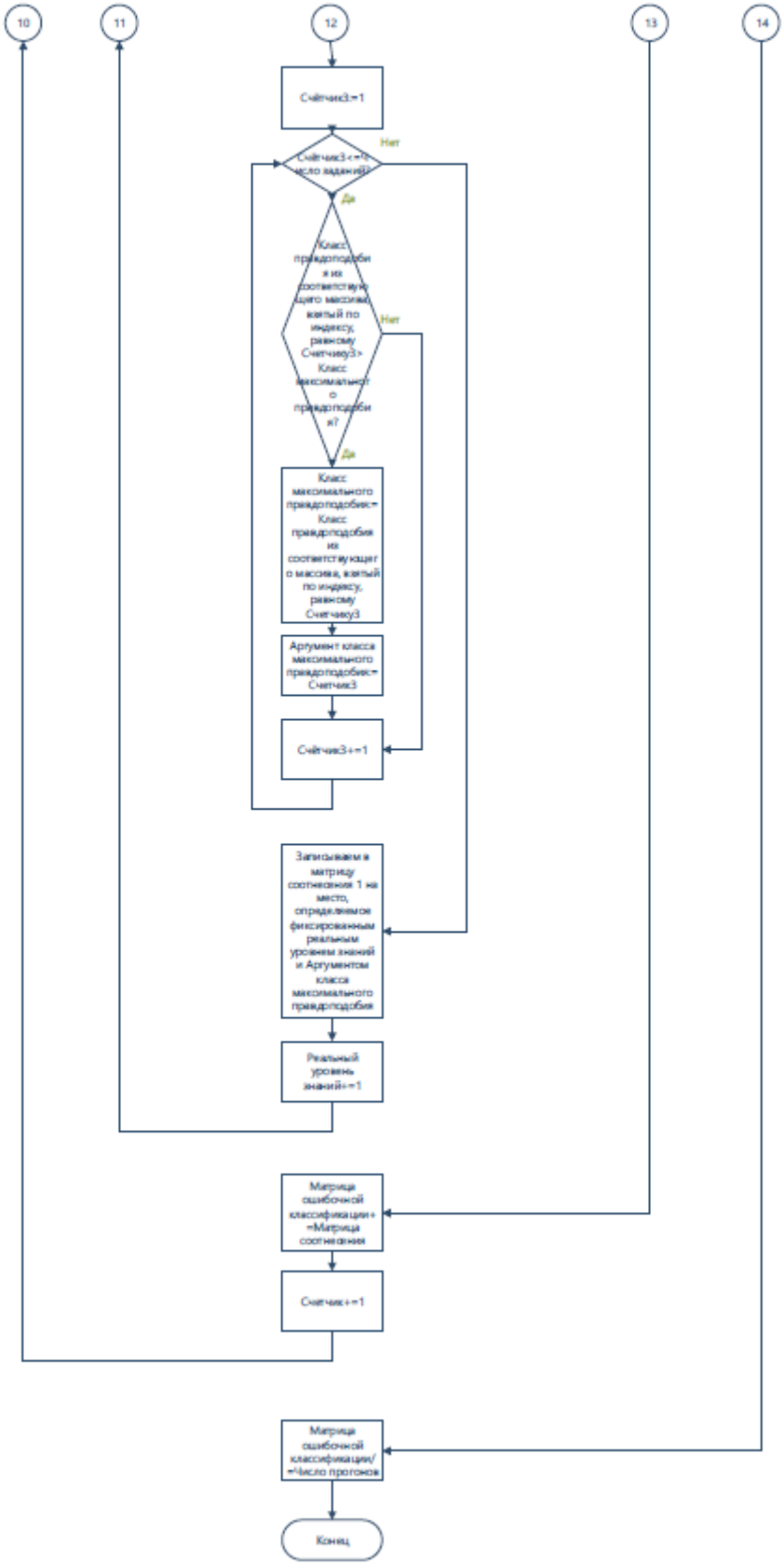
При получении абзаца текста СГВ сначала автоматически просматривает содержимое, отмечая символы, которые не получается обработать, которые пользователю предлагается редактировать или удалить. Затем пользователю предлагается выбрать область текста, которая в целом содержит ответ на будущий вопрос. В качестве альтернативы пользователь сам может создать такой объект. В качестве третьего шага выбранные участки помечаются в отрывке и подаются на вход модуля генерации вопросов. Модуль генерации вопросов представляет собой seq2seq модель с динамическими словарями, многоразовым копированием внимания и глобальным механизмом максимально избирательного внимания. Модуль пытается автоматически сгенерировать наиболее релевантные и вместе с тем синтаксически и семантически корректные вопросы на основе выбранных кусочков текста. В качестве последнего шага происходит фильтрации вопросов без ответов с помощью модуля фильтрации. Основанном на BERT. Оставшиеся вопросы демонстрируются пользователю, сгруппированные по ответам. Каждая группа ответов соответствует уникальной стем-форме этих ответов.

Программная поддержка

Алгоритм программы, моделирующей статический и адаптивный тесты в MATLAB


Начало
Ввод числа заданий теста, параметра функции модели GRASCH, характеризующего дифференцирующую способность задания. Ввод числа прогонов модели
Установление сложности заданий теста, создание матрицы ошибочной классификации, заполненной нулями
Счетчик прогонов:=1
Счетчик<=Число прогонов?
Нет
Да
Создание матрицы соотнесения, заполненной нулями
Уровень знаний:=1
Реальный уровень знаний<=Число заданий?
Нет
Да
Создание массива классов правдоподобия
Начало моделирования решения теста Счетчик1:=1
Счетчик1<=Число заданий?
Нет
Да
Статический тест?
Нет
Да
Простой адаптивный тест?
Нет
Да
1
2
3
4
5
6
7
8
9


1
2
3
4
5
6
7
8
9
Моделирование ответа на задание: Вычисляем вероятность правильного выполнения задания для фиксированного уровня знаний и сложности, установленной в начале, по формуле однопараметрической модели GRASCH Сравниваем полученную вероятность со случайным числом, имеющим нормальный закон распределения и сохраняем результат сравнения в виде 0 или 1
Моделирование ответа с использованием метода максимального правдоподобия
Моделирование ответа на задание с изменением сложности в зависимости от предыдущего ответа
Начало оценки вероятности цепочки ответов: Счетчик2:=1
Счетчик2<Чис-ло заданий?
Нет
Да
Вычисляем вероятность по формуле однопараметрической модели GRASCH для сложности, установленной в начале и уровня знаний, равного текущему значению Счетчик2
Результат сравнения=1?
Нет
Да
КлассПравдоподобия+=Log(вероятность цепочки ответов)
КлассПравдоподобия+=Log(1- вероятность цепочки ответов)
Счетчик2+=1
Счетчик1+=1
Начало оценки квалификационного уровня
Класс максимального правдоподобия+=первый класс правдоподобия из массива
Аргумент класса максимального правдоподобия:=1
10
11
12
13
14


10
11
12
13
14
Счётчик3:=1
Счётчик3<=Число заданий?
Нет
Да
Класс предпочтения соответствующего массива, взятый по индексу, равному Счётчику3 = Класс максимального предпочтения?
Нет
Да
Класс максимального предпочтения:= Класс предпочтения соответствующего массива, взятый по индексу, равному Счётчику3
Аргумент класса максимального предпочтения:= Счётчик3
Счётчик3:=1
Записываем в матрицу соотношения 1 на место, определяемое фиксированным реальным уровнем знаний и Аргументом класса максимального предпочтения
Реальный уровень знаний:=1
Матрица ошибочной классификации:= Матрица соотношения
Счетчик:=1
Матрица ошибочной классификации/ =Число прогонов
Конец

Алгоритм работы СГВ на основе Seq2Seq

1. Просматривание абзаца

Поскольку каждое предложение/слово в абзаце не может быть целью вопросов, важно отфильтровать неподходящие. Учитывая текст абзаца, система автоматически просматривает его содержимое, чтобы проверить, содержит ли абзац какие-либо символы, отличные от ASCII, URL-адреса и т.д., и помечает их для редактирования пользователями.

2. Выбор ответа

СГВ позволяет пользователям выбрать любую именованную сущность или субстантивное словосочетание, присутствующее в абзаце, в качестве ключевого ответа. Как уже упоминалось ранее, пользователю предоставляется список всех именованных сущностей и субстантивных фраз, извлеченных с помощью теггера Stanford CoreNLP для выбора ключевых ответов. Кроме того, пользователи могут вручную выбрать набор фраз из отрывка в качестве ключевых. Выбранные фразы помечаются в исходных предложениях с помощью BIO (Begin, Inside, Outside) нотации.

3. Генерация вопросов

Кодируем ключевые интервалы ответов в отрывке с помощью BIO-нотации и обучаем модель seq2seq, дополненную динамическим словарем, механизмом копирования и глобальным максимально разреженным вниманием.

Модуль генерации вопросов состоит из кодера абзацев и декодера вопросов.

Кодер получает на вход абзац в виде одного непрерывного вектора фиксированной длины. Это векторное представление абзаца передается декодеру с многоразовым механизмом копирования и максимально разреженным вниманием для генерации вопросов.

4. Фильтрация вопросов на основе BERT

Мы используем модель BERTbase, чтобы отфильтровать вопросы, на которые нет ответов, генерируемые нашей моделью. мы точно настраиваем BERT на SQuAD 2.0.

SQuAD 2.0 расширяет состав SQUAD более чем на 50000 вопросов, на которые нет ответов. Неразрешимые вопросы помечаются атрибутом невозможно.

Мы представляем входной вопрос (вопрос, сгенерированный нашей моделью генерации вопросов) и отрывок в одной упакованной последовательности токенов, используя при этом специальный токен [SEP] для отделения вопроса от прохода.

Аналогично мы используем специальный классификационный маркер [CLS] в начале каждой последовательности. Обозначим конечное скрытое представление токена [CLS] через C и конечное скрытое представление для I-го входного токена через Ti.

Для каждого «невозможного» вопроса мы представляем индекс начала и конца ответа с использованием токена [CLS], так как он не имеет никакого индекса начала и конца ответа.

Аналогично мы сравниваем оценку «невозможного» вопроса со счетом лучшего ненулевого кусочка с ответом чтобы предсказать возможность ответа на вопрос. Оценка интервала отсутствия ответа вычисляется следующим образом: s_{null} = S.C + E.C, где S € $\mathbb{R}^h$ - векторное представление индекса начала ответа и E € $\mathbb{R}^h$ - векторное представление конечного индекса ответа. Оценка ненулевого интервала ответов определяется как $s_{i,j} = \max_{j>=i}$ {S.Ti + E.Tj}, если оценка $s_{null} - s_{i,j} > V$, где V-порог вычисляется с использованием проверочного набора, то вопрос не отвечает с помощью абзаца.

5. Групповые / Фасетные представления вопросов

Мы группируем вместе все ответы и соответствующие им вопросы, которые имеют одну и ту же стем-форму. Например, два возможных диапазона ответов "переключение" и "переключатели" имели бы одну и ту же

стем-форму "переключатель". Таким образом, фразы "переключение" и "переключатели" и связанные с ними вопросы были бы сгруппированы вместе под одной и той же стем-формой "переключать". Короче говоря, каждая такая группа вопросов дает сетку вопросов. Внутри каждой группы вопросы сортируются в порядке убывания их вероятностей.

Мы вычисляем доверительный балл вопроса(внутривопросная вероятность), нормализуя балл луча x как: $e^x/(1 + e^x)$. Конечная межвопросовая вероятность пары вопрос-ответ вычисляется из вопроса с максимальной внутривопросовой вероятностью p как: p-min(P)/ max(P)-min(P), где P-набор максимальных вероятностных оценок по всем ответам.

Список литературы:

[1] Vishwajeet Kumar, Sivaanandh Muneeswaran , Ganesh Ramakrishnan , and Yuan-Fang Li. 2019 ParaQG: A System for Generating Questions and Answers from Paragraphs. arXiv preprint arXiv:1909.01642v1

[2] Ш. И. Цыганов, Математические методы педагогических измерений., Вестник Башкирского университета., 2009. Т. 14. №3, – с.1263-1270

[3] Новиков Д.А., Закономерности итеративного научения., М.: Институт проблем управления РАН, 1998. – 77 с.