

Глава 1

Введение

Автоматическая постановка и дополнение обучающих задач востребованное направление в сфере образования. Алгоритмические методы позволяют разрешать классические проблемы образования, включающие составление методической литературы, пресечение недобросовестной кооперации обучающихся при индивидуальном контроле знаний, формирование индивидуальной образовательной траектории.

Работы также показывают успешное применение автоматической генерации для формирования индивидуальной образовательной траектории тематически однородного, но разноуровневого по сложности. Тем не менее предложенные подходы требуют значительных временных экспертов для создания новых методических курсов.

Стремительное развитие генеративного моделирования в областях естественного языка [radford2019language] и машинного зрения [rombach2022highresolution][song2020ge определили новые подходы в известных задачам нотариального консультирования, .

Задача работы применить методы генеративного моделирования для решения задачи. В дополнение работы выпускается кодовая база и обучающие данные для воспроизведения эксперимента и проведения позволяет На практике

В работах ... уже

Появляются возможности генерации задач тематических задач по интересам обучающегося, обращаясь к предметному опыту и интуиции.

объективного оценивания не на

Работа состоит из 5 частей.

В первой части работы будут описаны

1.1 Педагогическая

В секции будут описаны требования к педагогической задаче и ее типы.

1.1.1 Введение

Педагогическая задача является основой образовательного процесса и играет ключевую роль в достижении учебных целей. Её цель состоит в том, чтобы обеспечить студентам определённые образовательные возможности и помочь им развить необходимые знания, умения и навыки.

При создании педагогической задачи важно учитывать не только содержание обучения, но и индивидуальные особенности студентов, их уровень знаний и способности. Педагогическая задача должна быть четко сформулирована, чтобы студенты могли понять, что от них требуется, и чувствовать уверенность в выполнении задания.

Важным аспектом педагогической задачи является её реалистичность и актуальность. Задача должна иметь практическую ценность и быть связанной с реальными жизненными ситуациями или профессиональными задачами. Это поможет стимулировать интерес и мотивацию студентов к изучению материала.

Педагогическая задача также должна предоставлять возможность для развития критического мышления и применения знаний на практике. Она должна быть структурированной и обеспечивать возможность оценки выполнения студентами поставленной задачи. Критерии оценки должны быть ясными и объективными, чтобы обеспечить справедливую оценку достижения учебных целей.

Реализация педагогической задачи может включать использование различных методов обучения и оценки, таких как групповая работа, проектная деятельность, обсуждения, решение проблемных ситуаций и другие. Это позволит стимулировать активное участие студентов в образовательном процессе и способствовать их полноценному развитию.

В академическом контексте условие задачи должно быть представлено в понятной и ясной формулировке, чтобы студенты могли полностью понять, что от них требуется. Это важно для обеспечения эффективного обучения и достижения учебных целей. При этом условие задачи должно быть достаточно простым, чтобы студенты могли легко освоить материал и выполнить задание, но при этом содержательным, чтобы оно имело академическую ценность и было связано с обучающей программой других предметов.

Параллели с обучающей программой других предметов могут быть важны для того, чтобы показать студентам связь между различными областями знаний и помочь им понять, как полученные знания применяются на практике. Например, задача по математике может быть сформулирована таким образом, чтобы студенты могли увидеть её применение в других предметах, таких как физика или экономика.

Кроме того, условие задачи должно быть структурированным и логически последовательным, чтобы студенты могли легко следовать указаниям и выполнять задание без лишних затруднений. Это позволит им сконцентрироваться на освоении материала и достижении желаемых результатов.

Таким образом, условие задачи должно быть простым и понятным, иметь параллели с обучающей программой других предметов и быть структурированным и логически последовательным, чтобы обеспечить эффективное обучение и достижение учебных целей.

1.1.2 Виды задач

Типы задач В русской математической школе существует несколько подходов к составлению задач, которые характеризуются своими особенностями и методами:

1. **Классический подход**: Этот подход базируется на традиционных математических задачах, которые имеют четкую формулировку, логическую структуру и обычно решаются с использованием стандартных методов и приемов. Такие задачи часто используются для развития навыков решения и анализа математических задач.

2. **Геометрический подход**: Этот подход акцентируется на задачах, связанных с геометрией, и требует умения визуализировать и рисовать геометрические фигуры, а также применять геометрические концепции и теоремы для их решения. Геометрические задачи обычно развивают пространственное мышление и геометрическую интуицию.

3. **Алгебраический подход**: Этот подход основан на задачах, связанных с алгеброй и алгебраическими выражениями. Они могут включать в себя решение уравнений и неравенств, работы с многочленами, факторизацию и раскрытие скобок. Алгебраические задачи развивают навыки работы с алгебраическими структурами и алгебраическими методами решения.

4. **Комбинаторный подход**: Этот подход фокусируется на задачах, связанных с комбинаторикой и теорией вероятностей. Он включает в себя задачи о перестановках, комбинациях, размещениях, задачи на сочетания, вероятностные модели и другие комбинаторные конструкции. Комбинаторные задачи развивают навыки анализа и оценки различных комбинаторных структур и их свойств.

5. **Алгоритмический подход**: Этот подход связан с задачами на программирование и разработку алгоритмов. Он включает в себя задачи на программирование на различных языках программирования, алгоритмические задачи на графах, строках, матрицах и т. д. Алгоритмические задачи развивают навыки алгоритмизации, программирования и решения задач с использованием компьютера.

Эти подходы могут использоваться как самостоятельно, так и в комбинации для создания разнообразных и интересных математических задач, способствующих развитию различных аспектов математического мышления и навыков решения задач.

1.1.3 Структура методического материала

Задачник - это учебный ресурс, который обычно организован в виде последовательного набора задач, объединенных по общей теме или концепции. Он служит для систематизации и структурирования материала, предоставляя студентам или исследователям возможность практического применения знаний и навыков.

Структура задачника обычно состоит из разделов или тематических блоков, которые охватывают определенные аспекты изучаемой области. Каждый блок начинается с введения в тему, где обозначаются ключевые понятия и основные принципы, а также могут даваться краткие объяснения теоретических аспектов.

Задачи внутри блоков обычно распределены в порядке возрастания сложности или последовательности углубления в изучаемую тему. Начальные задачи могут быть более простыми и базовыми, позволяя студентам получить предварительное понимание концепции, после чего следуют более сложные задачи, требующие более глубокого анализа и применения полученных знаний.

Каждая задача обычно сопровождается пояснениями или инструкциями, которые помогают студентам понять, как решить задачу, и обычно включает в себя краткое описание цели задачи и указания на соответствующие теоретические материалы.

Важно, чтобы задачник обеспечивал разнообразие заданий, таких как теоретические задачи, практические задания и примеры, а также давал возможность студентам проверить свои знания и навыки через различные виды задач.

В конце задачника обычно предоставляются дополнительные ресурсы, такие как дополнительная литература, ссылки на онлайн-ресурсы или рекомендации по дальнейшему обучению, что помогает студентам расширить свои знания и глубже понять изучаемую тему.

1.1.4 Сложность задачи

Креативное мышление, характеризующееся способностью видеть и понимать свойства объектов, которые не всегда могут быть строго описаны или выражены формально, играет ключевую роль в решении нестандартных задач. Выдающиеся студенты, обладающие креативным мышлением, способны обнаруживать такие свойства объектов, как симметрии, непрерывность и подобие, что позволяет им искать необычные подходы к решению проблем.

Кроме того, креативное мышление способствует умению обобщать. Сложные задачи часто требуют применения знаний из различных областей или дисциплин, а также способности видеть связи между различными аспектами проблемы. Студенты, обладающие креативным мышлением, могут успешно переносить знания и методы из одной области на другую, что помогает им сформулировать новые идеи и подходы к решению задач.

Важным аспектом является также умение работать с литературой и другими ис-

точниками информации. Сложные задачи могут потребовать дополнительного изучения и анализа, а также использования материалов из смежных дисциплин. Креативное мышление позволяет студентам не только эффективно работать с этими источниками, но и интегрировать полученные знания в свои решения задач.

Таким образом, креативное мышление играет важную роль в решении нестандартных задач, позволяя студентам гибко подходить к проблеме, видеть скрытые аспекты и находить новые пути решения. Это способствует развитию критического мышления, инновационности и адаптивности, что является важным аспектом в современном образовании. В рамках секции будут описаны методы, применяемые в генеративном моделировании для решения задачи генерации задач.

1.2 Использование нейросетевых подходов

Нейронные сети представляют собой вычислительные модели, состоящие из узлов, называемых нейронами, организованных в слои. Каждый нейрон взвешивает входные сигналы, представленные как вектор $\mathbf{x} = (x_1, x_2, \dots, x_n)$, с весами $\mathbf{w} = (w_1, w_2, \dots, w_n)$ и смещением b , где n - количество входов, x_i - i -й входной сигнал, w_i - весовой коэффициент i -го входа, b - смещение (bias). На выходе нейрона производится линейная комбинация входов с весами и смещением:

$$z = \sum_{i=1}^n w_i x_i + b$$

Полученная сумма z затем подвергается нелинейному преобразованию при помощи функции активации $f(z)$, которая определяет активацию нейрона:

$$y = f(z)$$

Функция активации обычно вводится для добавления нелинейности в модель, что позволяет нейронной сети моделировать сложные нелинейные зависимости в данных. Некоторые из распространенных функций активации включают в себя сигмоидальную функцию (σ), гиперболический тангенс (\tanh), ReLU (Rectified Linear Unit) и их вариации.

В случае многослойной нейронной сети, выходы нейронов одного слоя становятся входами для следующего слоя, образуя цепочку преобразований. Процесс передачи данных через нейроны последовательных слоев называется прямым распространением (forward propagation).

Нейронные сети обучаются путем настройки весов \mathbf{w} и смещений b с использованием алгоритмов оптимизации, таких как градиентный спуск. Во время обучения модель минимизирует функцию потерь L , которая оценивает разницу между предсказанным результатом и истинным значением:

$$L = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{y}_i)$$

где N - количество обучающих примеров, y_i - истинное значение, \hat{y}_i - предсказанное значение.

1.2.1 Модель Хопфилда

Модель Хопфилда - это нейронная сеть, предложенная Джоном Хопфилдом в 1982 году, которая является одним из первых примеров рекуррентных нейронных сетей и используется для моделирования ассоциативной памяти и ассоциативного запоминания. Эта модель вдохновлена концепцией обработки информации в мозге и базируется на принципе ассоциативного запоминания, который подразумевает возможность восстановления целевого образа (или паттерна) по части информации.

В модели Хопфилда каждый нейрон представляет собой двоичный (или биполярный) элемент, который может находиться в одном из двух состояний: активном (1) или неактивном (0). Нейроны соединены сетью симметричных связей, где каждая связь имеет ассоциативную силу между соответствующими нейронами. Эта сеть связей может быть представлена в виде симметричной матрицы весов, где элементы w_{ij} обозначают силу связи между нейронами i и j .

Динамика модели Хопфилда определяется обновлением активации каждого нейрона в соответствии с функцией активации и правилом Хопфилда:

1. ****Инициализация****: Начнем с заданного начального состояния нейронов.
2. ****Обновление нейронов****: Нейроны обновляются параллельно и асинхронно на основе правила Хопфилда:

$$s_i(t+1) = \text{sign} \left(\sum_j w_{ij} s_j(t) \right)$$

3. ****Повторение шага 2****: Процесс обновления нейронов повторяется до тех пор, пока система не стабилизируется в некотором состоянии или пока не произойдет схождение к сохраненным паттернам.

Одно из ключевых свойств модели Хопфилда - это способность к ассоциативному запоминанию. После обучения модель способна восстанавливать сохраненные образы при предъявлении искаженных или неполных версий этих образов. Это происходит благодаря тому, что система восстанавливается к ближайшему сохраненному образу в пространстве состояний.

Модель Хопфилда была активно изучена как теоретически, так и экспериментально, и она оказала влияние на развитие нейроинформатики и ассоциативного запоминания. Она также служит основой для более сложных и глубоких моделей нейронных сетей, используемых в современных исследованиях машинного обучения.

и искусственного интеллекта.

1.3 Обработка естественного языка

Глава 2

Методы обработки естественного языка

Раздел имеет две части представление языка и его генерация.

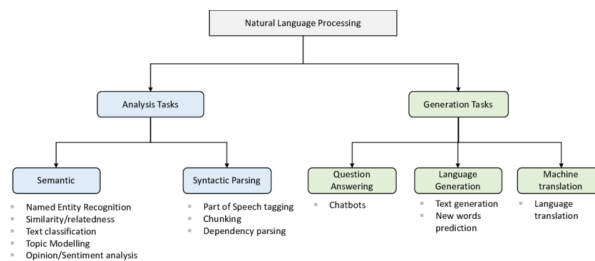


Рис. 2.1: Таксономия современных подходов обработки естественного языка

Анализ естественного языка это междисциплинарная дисциплина.

2.0.1 Представление

2.0.2 Лемматизация

Лемматизация представляет собой процесс нормализации текста, целью которого является приведение слов к их базовой форме или лемме. В контексте обработки естественного языка (Natural Language Processing, NLP), лемматизация является важным этапом предварительной обработки текста, который позволяет уменьшить размер словаря и улучшить качество анализа.

Формально, лемматизация выражается в преобразовании слова w в его лемму $\text{lemma}(w)$ с использованием правил и алгоритмов, учитывающих морфологические особенности языка. Лемма представляет собой каноническую, нормализованную форму слова, которая может быть использована для обобщения различных грамматических форм одного и того же слова.

Математически, процесс лемматизации может быть представлен как отображение слова w в его лемму $\text{lemma}(w)$:

$$\text{lemma}(w) = \text{лемма}$$

Например, для слова "бегу его леммой будет слово "бежать". Лемматизация помогает уменьшить словарь слов и снизить размерность пространства признаков в текстовых данных, что положительно влияет на производительность алгоритмов обработки текста, таких как классификация или кластеризация.

Применение лемматизации часто сопровождается предварительным шагом токенизации, в котором текст разбивается на отдельные слова или токены. Это позволяет применить лемматизацию к каждому слову в тексте независимо от контекста. Лемматизация часто используется в различных областях NLP, включая информационный поиск, анализ тональности, машинный перевод и другие.

2.0.3 Векторное представление

Практически востребованной оказалась дистрибутивная гипотеза [Schutze], легшая в основу алгоритма [NIPS2013_9aa42b31].

В генеративном моделировании естественного языка, встает задача представления слов в виде векторов в многомерном пространстве, что позволяет моделировать семантические и синтаксические аспекты текста в компактной форме. Это представление, известное как "векторное вложение" или "embedding" позволяет выразить смысловые и лингвистические свойства слов, используемых в языке.

Формально, векторное вложение \mathbf{e}_w слова w представляет собой векторное представление этого слова в многомерном пространстве:

$$\mathbf{e}_w = (e_{w1}, e_{w2}, \dots, e_{wd})$$

где d - размерность пространства вложения (число измерений), e_{wj} - j -ая компонента вектора вложения \mathbf{e}_w .

Эти векторные представления обычно изучаются и извлекаются из больших корпусов текстов с использованием различных алгоритмов, таких как word2vec, GloVe (Global Vectors for Word Representation), FastText и другие. Они обладают свойством сохранения семантической близости слов в пространстве вложения: слова, которые часто встречаются в похожих контекстах, имеют близкие векторные представления.

Векторные вложения слов играют важную роль в генеративном моделировании естественного языка, так как они позволяют моделям представлять слова в виде непрерывных числовых значений, которые могут быть использованы как входные данные для алгоритмов машинного обучения. Это позволяет моделям эффективно изучать зависимости между словами и генерировать тексты семантически богатые и лингвистически осмысленные.

2.0.4 Генерация

2.0.5 N-граммы

N-граммы представляют собой последовательности из n элементов в тексте или последовательности символов, где n обозначает количество элементов в последовательности. Элементы могут быть символами, словами или более крупными фрагментами текста в зависимости от контекста применения. Анализ n-грамм является важным методом в обработке естественного языка (Natural Language Processing, NLP) для изучения частотности последовательностей слов или символов в текстовых данных.

Формально, n-грамма ngram_n длины n в тексте T определяется как последовательность n элементов, где каждый элемент x_i может быть символом, словом или другими единицами текста:

$$\text{ngram}_n = (x_1, x_2, \dots, x_n)$$

Для примера, если текст T представляет собой строку "мама мыла раму то бигramмы (2-граммы) этого текста будут:

$$\text{ngram}_2 = ("мама, "мыла), ("мыла, "раму)$$

Использование n-грамм в анализе текста позволяет оценивать частотность последовательностей слов или символов и изучать лингвистические характеристики текста, такие как структура, стиль и тематика. Кроме того, n-граммы могут использоваться в задачах моделирования языка, предсказания следующего слова в предложении, а также в машинном переводе и других приложениях обработки естественного языка.

2.0.6 Авторегрессионная модель

Авторегрессионные модели в генеративном моделировании естественного языка представляют собой класс статистических моделей, которые моделируют вероятностное распределение последовательности слов или символов в тексте. Эти модели базируются на предположении о зависимости текущего элемента последовательности от предыдущих элементов, что позволяет им учитывать контекст и последовательность в текстовых данных.

Формально, в авторегрессионной модели вероятность появления последовательности $W = (w_1, w_2, \dots, w_T)$ слов или символов определяется как произведение вероятностей каждого слова при условии предыдущих:

$$P(W) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \dots \cdot P(w_T|w_1, w_2, \dots, w_{T-1})$$

где $P(w_t|w_1, w_2, \dots, w_{t-1})$ - вероятность появления слова w_t при условии всех предыдущих слов w_1, w_2, \dots, w_{t-1} .

Авторегрессионные модели могут быть реализованы с использованием различных подходов, включая марковские модели, рекуррентные нейронные сети и модели с авторегрессионными свойствами, такие как GPT (Generative Pre-trained Transformer) и LSTM (Long Short-Term Memory). Они находят применение в широком спектре задач обработки естественного языка, включая генерацию текста, машинный перевод, синтез речи и другие.

Одной из ключевых особенностей авторегрессионных моделей является их способность учитывать контекст и последовательность слов в тексте, что позволяет им генерировать качественные тексты с учетом структуры и семантики. Тем не менее, выбор подходящей модели и обучение ее требуют значительных вычислительных ресурсов и экспертных знаний в области машинного обучения и обработки естественного языка.

2.0.7 Attention

Механизм внимания (attention) в генеративном моделировании естественного языка является ключевым компонентом в нейронных сетях, позволяющим модели фокусироваться на определенных частях входных данных при выполнении задач обработки текста. Этот механизм позволяет модели адаптироваться к различным контекстам и динамически выделять важные элементы во входных последовательностях.

Формально, предположим, что у нас есть входные данные $X = (x_1, x_2, \dots, x_T)$ и контекст C , а также текущее состояние скрытого слоя модели h_t . Механизм внимания вычисляет вектор внимания α , который определяет важность каждого элемента входной последовательности на текущем временном шаге:

$$\alpha_t = \text{softmax}(f(h_t, X))$$

где f - функция, которая вычисляет важность каждого элемента входной последовательности, а softmax применяется для получения нормированных весов внимания.

С использованием вектора внимания α , взвешенная сумма контекста C вычисляется как:

$$\text{context}_t = \sum_{i=1}^T \alpha_{ti} x_i$$

Полученный контекст используется в дальнейших вычислениях модели для выполнения задач, таких как генерация текста или классификация.

Механизм внимания позволяет модели сосредоточиться на наиболее значимых частях входных данных в каждый момент времени, что делает его особенно полезным для задач, требующих адаптивности и контекстного понимания, таких как ма-

шинный перевод, генерация текста и вопросно-ответные системы. Этот механизм стал ключевым инструментом в области генеративного моделирования естественного языка, позволяя моделям эффективно работать с различными типами данных и контекстами.

2.1 Обработка изображений

Глава 3

Методы обработки естественного языка

Раздел имеет две части представление языка и его генерация.

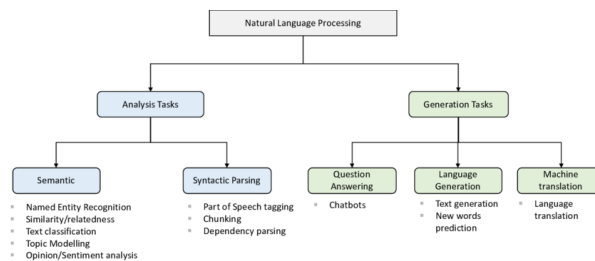


Рис. 3.1: Таксономия современных подходов обработки естественного языка

Анализ естественного языка это междисциплинарная дисциплина.

3.0.1 Представление

3.0.2 Лемматизация

Лемматизация представляет собой процесс нормализации текста, целью которого является приведение слов к их базовой форме или лемме. В контексте обработки естественного языка (Natural Language Processing, NLP), лемматизация является важным этапом предварительной обработки текста, который позволяет уменьшить размер словаря и улучшить качество анализа.

Формально, лемматизация выражается в преобразовании слова w в его лемму $\text{lemma}(w)$ с использованием правил и алгоритмов, учитывающих морфологические особенности языка. Лемма представляет собой каноническую, нормализованную форму слова, которая может быть использована для обобщения различных грамматических форм одного и того же слова.

Математически, процесс лемматизации может быть представлен как отображение слова w в его лемму $\text{lemma}(w)$:

$$\text{lemma}(w) = \text{лемма}$$

Например, для слова "бегу его леммой будет слово "бежать". Лемматизация помогает уменьшить словарь слов и снизить размерность пространства признаков в текстовых данных, что положительно влияет на производительность алгоритмов обработки текста, таких как классификация или кластеризация.

Применение лемматизации часто сопровождается предварительным шагом токенизации, в котором текст разбивается на отдельные слова или токены. Это позволяет применить лемматизацию к каждому слову в тексте независимо от контекста. Лемматизация часто используется в различных областях NLP, включая информационный поиск, анализ тональности, машинный перевод и другие.

3.0.3 Векторное представление

Практически востребованной оказалась дистрибутивная гипотеза [Schutze], легшая в основу алгоритма [NIPS2013_9aa42b31].

В генеративном моделировании естественного языка, встает задача представления слов в виде векторов в многомерном пространстве, что позволяет моделировать семантические и синтаксические аспекты текста в компактной форме. Это представление, известное как "векторное вложение" или "embedding" позволяет выразить смысловые и лингвистические свойства слов, используемых в языке.

Формально, векторное вложение \mathbf{e}_w слова w представляет собой векторное представление этого слова в многомерном пространстве:

$$\mathbf{e}_w = (e_{w1}, e_{w2}, \dots, e_{wd})$$

где d - размерность пространства вложения (число измерений), e_{wj} - j -ая компонента вектора вложения \mathbf{e}_w .

Эти векторные представления обычно изучаются и извлекаются из больших корпусов текстов с использованием различных алгоритмов, таких как word2vec, GloVe (Global Vectors for Word Representation), FastText и другие. Они обладают свойством сохранения семантической близости слов в пространстве вложения: слова, которые часто встречаются в похожих контекстах, имеют близкие векторные представления.

Векторные вложения слов играют важную роль в генеративном моделировании естественного языка, так как они позволяют моделям представлять слова в виде непрерывных числовых значений, которые могут быть использованы как входные данные для алгоритмов машинного обучения. Это позволяет моделям эффективно изучать зависимости между словами и генерировать тексты семантически богатые и лингвистически осмысленные.

3.0.4 Генерация

3.0.5 N-граммы

N-граммы представляют собой последовательности из n элементов в тексте или последовательности символов, где n обозначает количество элементов в последовательности. Элементы могут быть символами, словами или более крупными фрагментами текста в зависимости от контекста применения. Анализ n-грамм является важным методом в обработке естественного языка (Natural Language Processing, NLP) для изучения частотности последовательностей слов или символов в текстовых данных.

Формально, n-грамма ngram_n длины n в тексте T определяется как последовательность n элементов, где каждый элемент x_i может быть символом, словом или другими единицами текста:

$$\text{ngram}_n = (x_1, x_2, \dots, x_n)$$

Для примера, если текст T представляет собой строку "мама мыла раму то бигramмы (2-граммы) этого текста будут:

$$\text{ngram}_2 = ("мама, "мыла), ("мыла, "раму)$$

Использование n-грамм в анализе текста позволяет оценивать частотность последовательностей слов или символов и изучать лингвистические характеристики текста, такие как структура, стиль и тематика. Кроме того, n-граммы могут использоваться в задачах моделирования языка, предсказания следующего слова в предложении, а также в машинном переводе и других приложениях обработки естественного языка.

3.0.6 Авторегрессионная модель

Авторегрессионные модели в генеративном моделировании естественного языка представляют собой класс статистических моделей, которые моделируют вероятностное распределение последовательности слов или символов в тексте. Эти модели базируются на предположении о зависимости текущего элемента последовательности от предыдущих элементов, что позволяет им учитывать контекст и последовательность в текстовых данных.

Формально, в авторегрессионной модели вероятность появления последовательности $W = (w_1, w_2, \dots, w_T)$ слов или символов определяется как произведение вероятностей каждого слова при условии предыдущих:

$$P(W) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \dots \cdot P(w_T|w_1, w_2, \dots, w_{T-1})$$

где $P(w_t|w_1, w_2, \dots, w_{t-1})$ - вероятность появления слова w_t при условии всех предыдущих слов w_1, w_2, \dots, w_{t-1} .

Авторегрессионные модели могут быть реализованы с использованием различных подходов, включая марковские модели, рекуррентные нейронные сети и модели с авторегрессионными свойствами, такие как GPT (Generative Pre-trained Transformer) и LSTM (Long Short-Term Memory). Они находят применение в широком спектре задач обработки естественного языка, включая генерацию текста, машинный перевод, синтез речи и другие.

Одной из ключевых особенностей авторегрессионных моделей является их способность учитывать контекст и последовательность слов в тексте, что позволяет им генерировать качественные тексты с учетом структуры и семантики. Тем не менее, выбор подходящей модели и обучение ее требуют значительных вычислительных ресурсов и экспертных знаний в области машинного обучения и обработки естественного языка.

3.0.7 Attention

Механизм внимания (attention) в генеративном моделировании естественного языка является ключевым компонентом в нейронных сетях, позволяющим модели фокусироваться на определенных частях входных данных при выполнении задач обработки текста. Этот механизм позволяет модели адаптироваться к различным контекстам и динамически выделять важные элементы во входных последовательностях.

Формально, предположим, что у нас есть входные данные $X = (x_1, x_2, \dots, x_T)$ и контекст C , а также текущее состояние скрытого слоя модели h_t . Механизм внимания вычисляет вектор внимания α , который определяет важность каждого элемента входной последовательности на текущем временном шаге:

$$\alpha_t = \text{softmax}(f(h_t, X))$$

где f - функция, которая вычисляет важность каждого элемента входной последовательности, а softmax применяется для получения нормированных весов внимания.

С использованием вектора внимания α , взвешенная сумма контекста C вычисляется как:

$$\text{context}_t = \sum_{i=1}^T \alpha_{ti} x_i$$

Полученный контекст используется в дальнейших вычислениях модели для выполнения задач, таких как генерация текста или классификация.

Механизм внимания позволяет модели сосредоточиться на наиболее значимых частях входных данных в каждый момент времени, что делает его особенно полезным для задач, требующих адаптивности и контекстного понимания, таких как ма-

шинный перевод, генерация текста и вопросно-ответные системы. Этот механизм стал ключевым инструментом в области генеративного моделирования естественного языка, позволяя моделям эффективно работать с различными типами данных и контекстами.

Глава 4

Проделанная работа

Работа по созданию ассистента выполнялась поэтапно.

Исходным этапом работы являлось создание корпуса педагогических задач, извлеченных из открытых источников российских учебников. Процесс сбора данных осуществлялся при помощи технологий оптического распознавания символов (OCR), включая методы, разработанные в рамках данного исследования. В дальнейшем полученный корпус

Глава разделена на части по

В главе приведено описание поставленных по трем направлениям. Генерация текста задачи, сопровождающей иллюстрации и

4.1 Формирование Иллюстрации

Данные были собраны из открытых источников [libmipt][mathedu].

В состав датасета входит более десятка тысяч аннотированных изображений. Результат моделирования предоставлены на открытых ресурсах¹

4.2 Заключение

Работа представляет

4.2.1 Благодарности

Автор благодарит кафедру инновационной педагогики за предоставление консультационной информации в области права и . Отдельная благодарность моему научному руководителю Щербакову Дмитрию Евгеньвичу за возможность работы в актуальной и современной тематике генеративного моделирования

¹<https://github.com/NMashalov/Generative-modeling-appliance-for-creating-educational-tasks> и https://huggingface.co/datasets/NMashalov/task_illustrations_dataset