

Знакомство с DeepSeek и его применение в образовании

Тема №6

DeepSeek Chat представляет собой крупноязыковую модель (Large Language Model, LLM), разработанную компанией DeepSeek и относящуюся к семейству генеративных искусственных интеллектов трансформерной архитектуры. Данная система основана на нейросетевых механизмах обработки естественного языка (Natural Language Processing, NLP) с применением внимания (attention mechanisms) и глубинного обучения (deep learning).

Модель использует массивные предобученные параметры (сотни миллиардов весовых коэффициентов), что позволяет ей анализировать, синтезировать и генерировать текстовые данные с высокой степенью когерентности и контекстуальной релевантности. Важной отличительной чертой DeepSeek Chat является поддержка расширенного контекстного окна (128K токенов), что существенно превосходит многие аналоги (например, GPT-4 Turbo имеет 128K лишь в некоторых режимах). Это позволяет модели эффективно обрабатывать длинные документы, сохраняя смысловую связность.

С точки зрения педагогической практики, DeepSeek Chat может быть охарактеризован как интеллектуальный ассистент, способный выполнять следующие дидактические функции:

- Генерация учебных материалов (конспекты, методические рекомендации, сценарии уроков) на основе заданных параметров (возрастная группа, уровень сложности, предметная область).
- Адаптивное объяснение сложных концепций с возможностью кастомизации стиля изложения (научный, популярный, схематизированный).
- Автоматизированная проверка и анализ текстов (эссе, рефератов, программного кода) с предоставлением обратной связи.
- Создание интерактивных диалоговых симуляций для отработки педагогических сценариев (например, объяснение темы "алгоритмы" ученику с визуальным мышлением).

Применение DeepSeek Chat в образовании требует учета следующих факторов:

- Промпт-инжиниринг (Prompt Engineering) – наука о формулировке запросов к ИИ. Для эффективного взаимодействия необходимо использовать контекстные уточнения ("Как учитель информатики, объясни рекурсию на примере Fibonacci для 10 класса"), структурирование вывода ("Представь ответ в виде таблицы") и итеративное уточнение ("Дай более простой вариант").

- Критическая верификация выходных данных. Несмотря на высокую точность, LLM могут генерировать галлюцинации (hallucinations) – ложные утверждения, представленные как факты. Преподаватель должен осуществлять экспертный контроль контента.
- Этические и нормативные ограничения. Использование ИИ-сгенерированных материалов должно соответствовать принципам академической честности (например, запрет на полное замещение авторских методических разработок).

Актуальная версия DeepSeek Chat доступна по адресу:

<https://chat.deepseek.com/>

Для углубленного изучения архитектуры модели рекомендуется ознакомиться с технической документацией на <https://deepseek.com>.

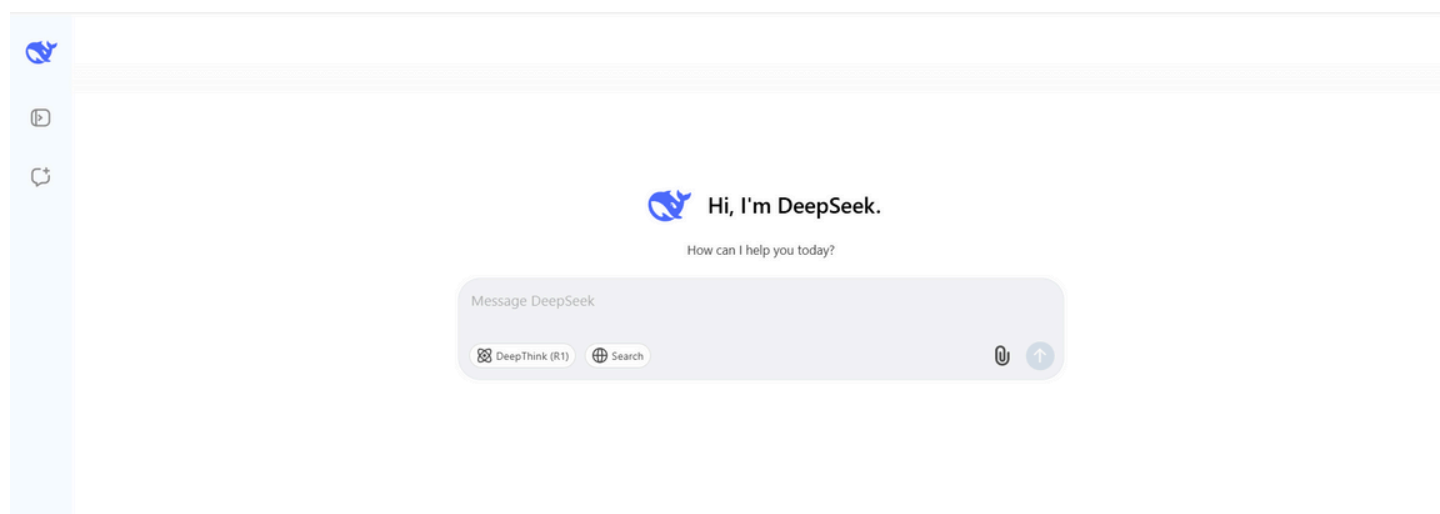


Рисунок 1. Стартовая страница чата DeepSeek

Использование DeepSeek Chat для создания интерактивных лекций по информатике

Для разработки лекций в DeepSeek необходимо сформулировать корректный запрос, максимально детально описывающий получение необходимой информации. Примером такого запроса может являться следующее сообщение:

“Разработай подробный план лекции по теме «Алгоритмы сортировки» для 10 класса. Включи разделы: объяснение теории, наглядные примеры (с кодом на Python), сравнение эффективности алгоритмов, практические задания. Оформи в виде маркированного списка с подразделами.”

Итогом данного запроса является четкая структурированная лекция, которую эффективно можно использовать для теоретической части урока. Пример представлен на рисунке 2.

2. Основные алгоритмы сортировки

• 2.1. Сортировка пузырьком (Bubble Sort)

- Принцип работы (последовательное сравнение соседних элементов)
- Наглядный пример с анимацией или пошаговой визуализацией
- Код на Python:

python

 Copy  Download

```
def bubble_sort(arr):  
    n = len(arr)  
    for i in range(n):  
        for j in range(0, n-i-1):  
            if arr[j] > arr[j+1]:  
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

- Временная сложность: $O(n^2)$

• 2.2. Сортировка выбором (Selection Sort)

- Принцип работы (поиск минимального элемента и обмен)
- Пример пошаговой работы
- Код на Python:



Рисунок 2. Пример ответа на запрос по генерации лекционного материала

Улучшить лекцию возможно с помощью дополнения данного материала вопросами, тестами. Сделать это можно новым запросом, пример промта может быть таким:

"Придумай 5 интерактивных вопросов по теме «Сортировка алгоритмов», которые можно задать во время лекции. Включи:

- Вопрос с подвохом для обсуждения,
- Практическую задачу на сортировку массивов,
- Вопрос на сравнение разных алгоритмов".

Результат такого запроса представлен на рисунке 3.



Вот 5 интерактивных вопросов по теме «**Сортировка алгоритмов**», которые можно задать во время лекции:

1. Вопрос с подвохом для обсуждения

«Почему бы не всегда использовать быструю сортировку (QuickSort), если она в среднем работает за $O(n \log n)$ и считается одной из самых быстрых?»

(Подвох: QuickSort может деградировать до $O(n^2)$ в худшем случае, требует дополнительной памяти для рекурсии, а в некоторых случаях (например, почти отсортированных данных) другие алгоритмы, такие как TimSort или InsertionSort, работают эффективнее.)

2. Практическая задача на сортировку массивов

«Дан массив чисел: [3, 1, 4, 1, 5, 9, 2, 6, 5]. Покажите шаги сортировки этого массива с помощью сортировки слиянием (MergeSort). Сколько сравнений потребуется?»

(Цель: проверить понимание принципа «разделяй и властвуй» и устойчивости алгоритма.)

3. Вопрос на сравнение алгоритмов

«В каких ситуациях сортировка вставками (InsertionSort) будет работать быстрее, чем сортировка выбором (SelectionSort), и почему?»

(Ожидаемый ответ: InsertionSort эффективнее на почти отсортированных данных, так как делает меньше сравнений и сдвигов, тогда как SelectionSort всегда работает за $O(n^2)$, независимо от исходного порядка.)



Рисунок 3. Ответ на запрос по дополнительным заданиям

Большим преимуществом больших языковых моделей в образовании, можно считать возможность объяснять сложные концепции простым языком. Примером может являться следующий запрос:

"Объясни принцип работы стека вызовов при рекурсии на аналогии из реальной жизни. Подготовь текст для слайда презентации."



Слайд: Принцип работы стека вызовов в рекурсии

Аналогия из жизни:

Представьте, что вы читаете книгу, и вдруг наткнётесь на неизвестное слово. Вы откладываете текущую страницу (чтобы вернуться к ней позже), берёте словарь и ищете значение. Но в словаре тоже встречается непонятный термин!

Тогда вы:

- 1 Откладываете словарь (но помните, где остановились).
- 2 Берёте энциклопедию и ищете термин.
- 3 Найдя ответ, возвращаетесь к словарю, а затем — к книге.

Как это связано со стеком вызовов?

- Каждый новый вызов рекурсии — как новая «отложенная книга» (кадр стека).
- Результат вычисляется в обратном порядке (последний вызов завершается первым).

Итог:

Стек вызовов — это «стопка отложенных задач», где рекурсия раскручивается, пока не достигнет базового случая.

Пример для кода:

```
python
def factorial(n):
    if n == 1: # базовый случай
        return 1
    return n * factorial(n - 1) # рекурсивный вызов
```

(Можно добавить схематичную картинку стека с вызовами `factorial(3)` → `factorial(2)` → `factorial(1)`)

Рисунок 4. Ответ на запрос объяснение сложной концепции.

Для решения сложных задач в информатике, существует возможность генерации кода. Для формирования таких материалов, запрос может быть следующим:

"Напиши код на Python для бинарного поиска с подробными комментариями для каждого шага. Добавь пример вызова функции с тестовыми данными."

Ответом на данный запрос, будет являться следующее сообщение:

```
python
def binary_search(arr, target):
    left, right = 0, len(arr) - 1 # Границы поиска
    while left <= right:
        mid = (left + right) // 2 # Средний индекс
        if arr[mid] == target:
            return mid # Элемент найден
        elif arr[mid] < target:
            left = mid + 1 # Ищем в правой половине
        else:
            right = mid - 1 # Ищем в левой половине
    return -1 # Элемент отсутствует

# Пример использования:
sorted_list = [1, 3, 5, 7, 9]
print(binary_search(sorted_list, 5)) # Вывод: 2 (индекс элемента)
```

Рисунок 5. Ответ на запрос по генерации кода.

Еще одной удивительной возможностью DeepSeek - это разработка кейсов для групповой работы. Разработать такую задачу возможно с помощью следующего промта:

"Составь сценарий группового задания по теме «Безопасность в интернете». Условия:

- 3 команды по 4 человека,
- Время выполнения — 20 минут,
- Этапы: анализ угроз, разработка правил, презентация решений."



Сценарий группового задания «Безопасность в интернете»

Цель:

Разработать и представить правила безопасного поведения в интернете, проанализировав основные угрозы.

Формат:

3 команды по 4 человека. Время выполнения — 20 минут.

Этапы задания:

1. Анализ угроз (5 минут)

Каждая команда получает список возможных угроз в интернете (например: фишинг, вирусы, кибербуллинг, мошенничество, утечка данных).

Задача:

- Выбрать 3 наиболее опасные угрозы.
- Кратко описать, как они работают и чем опасны.

2. Разработка правил (10 минут)

На основе выбранных угроз команды составляют 5 **ключевых правил безопасности** для пользователей.

Примеры:

- «Не переходите по подозрительным ссылкам».
- «Используйте двухфакторную аутентификацию».
- «Не публикуйте личную информацию в соцсетях».

Рисунок 6 . Пример ответа на запрос по подготовке группового задания

Помимо этого DeepSeek позволяет загружать собственные конспекты лекций по любой тематике, а затем может выделять из них самое важное или объяснять сложные концепции простым языком. Пример такой работы представлен на рисунке №7.



Globalnaya-kompyuternay...
PDF 160.26KB

"Проанализируй текст на сложность. Упрости объяснение для школьников 8 класса. Выдели ключевые термины для глоссария."



Упрощенное объяснение для школьников 8 класса:

Что такое Интернет?

Интернет — это огромная сеть, которая соединяет компьютеры по всему миру. Она позволяет людям обмениваться информацией, учиться, работать и общаться, даже если они находятся в разных странах.

Как устроен Интернет?

1. Компьютеры и серверы:

- Обычные компьютеры (рабочие станции) используются людьми для работы в сети.
- Серверы — это мощные компьютеры, которые хранят информацию (например, сайты) и предоставляют её по запросу.

2. Провайдеры:

Чтобы подключиться к Интернету, нужен провайдер — компания, которая обеспечивает доступ



Message DeepSeek



DeepThink (R1)



Search



AI-generated content. For reference only.

Рисунок 7. Анализ и облегчение учебного материала для 8 класса

DeepSeek трансформирует традиционную лекцию в динамичный интерактивный формат, где:

- Теория дополняется мгновенными примерами,
- Студенты получают персонализированные объяснения,
- Преподаватель экономит время на рутинных задачах.

Для максимальной эффективности важно комбинировать ИИ-генерацию с педагогическим контролем.

Практические задания

Задание № 1. Разработка структуры урока с использованием Deep Seek

1. Откройте DeepSeek Chat и введите запрос:
«Составь подробный план урока по теме "[Ваша тема, напр., "Циклы в Python"]" для [класс/курс]. Включи: цели, этапы (актуализация знаний, объяснение нового, закрепление), время на каждый этап, виды деятельности (фронтальная/групповая работа), домашнее задание.»
2. Проанализируйте предложенный план:
 - a. Соответствует ли он ФГОС?
 - b. Достаточно ли в нем интерактива?
3. Модифицируйте план, добавив 2 собственных элемента (например, блиц-опрос или работу с ошибками).

Задание №2. Создание дифференцированных заданий

1. Запросите у DeepSeek:
«Придумай 3 варианта задачи на тему "[тема, напр., "Функции в Python"]" для:
 - a. начинающих (базовый синтаксис),
 - b. среднего уровня (использование аргументов),
 - c. продвинутых (рекурсия).»
2. Добавьте к каждому варианту:
 - a. Критерии оценки,
 - b. Пример решения (попросите ИИ его сгенерировать).

Задание №3. Генерация вопросов для лекции

1. Введите запрос:
«Создай 5 вопросов по теме "[тема]" для проверки понимания во время лекции. Включи:
 - a. 2 вопроса с подвохом,
 - b. 1 открытый вопрос,
 - c. 2 тестовых вопроса с вариантами ответов.»
2. Проведите мини-опрос в группе, используя эти вопросы.

Задание № 4. Подготовка аналогий для объяснения сложных тем

1. Задайте DeepSeek:
«Объясни тему "[напр., "Рекурсия"]" через аналогию из повседневной жизни. Подготовь текст для слайда презентации.»
2. Создайте слайд в PowerPoint на основе этой аналогии + добавьте реальный пример кода.

Задание № 5. Проверка и комментирование работ обучающихся

1. Загрузите в DeepSeek текст студенческого реферата/эссе (или используйте пример).

2. Запросите:
«Проанализируй текст на:
а. логические ошибки,
б. грамматику,
с. соответствие теме.
Дай рекомендации по улучшению.»
3. На основе анализа составьте развернутый комментарий для обучающегося.

Задание №6. Генерация образовательного теста

1. Попросите DeepSeek:
«Сгенерируй 10 тестовых вопросов по теме "[тема]" в формате Яндекс формы. Включи:
а. 5 вопросов с выбором ответа,
б. 3 вопроса на соответствие,
с. 2 открытых вопроса.»
2. Импортируйте вопросы в Яндекс.Формы и протестируйте на коллегих.

Задание №7. Разработка кейса для групповой работы

1. Введите:
«Придумай кейс по теме "[напр., "Этика в интернете"]" для работы в группах.
Описание:
а. проблема,
б. роли участников,
с. этапы решения,
д. критерии оценки.»
2. Проведите пробное обсуждение кейса в мини-группах (3-4 человека).

Задание № 8. Анализ эффективности инструментов искусственного интеллекта

1. Напишите эссе (1-2 страницы) на тему:
«Как DeepSeek может дополнить работу учителя? Какие задачи нельзя делегировать ИИ?»
2. Используйте примеры из выполненных заданий.

