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»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №11
по дисциплине ««Основы систем ИИ»

Тема Генерация текстов на естественном языке

Студент Батуев А.Г.

Группа ИУ7-36Б

Преподаватели Строганов Ю.В.

Москва, 2025

Содержание

ВВЕДЕНИЕ

Цель: сформировать и исследовать текст сгенерированный с помощью цепей Маркова.

Задачи:

— необходимо взять тексты и сформировать n-граммы для генерации с помощью цепи Маркова текстов. Рассмотреть n=2,3,5.

— сформировать тексты с различными начальными словами и оценить "человечность"написан

— провести исследование возможности генерации текста при наличии обучающей выборки, состоящей только из предложений: "кошка съела мышкумышку съела кошка".

— оценить опасность работы с нестрогим порядком слов

— оценить получаемый порядок слов в предложениях для разных версий "qwen2.5"

1 Аналитическая часть

Генерация текста представляет собой задачу создания последовательностей слов, которые соответствуют заданным характеристикам языка и контекста. Этот процесс базируется на методах статистического моделирования, алгоритмах машинного обучения и лингвистических принципах.

1.1 Цепи Маркова

Цепи Маркова представляют собой стохастические процессы, в которых будущее состояние системы зависит только от текущего состояния, но не от предыдущих состояний. Основное предположение заключается в выполнении свойства Маркова:

$P(X_{n+1}|X_1, X_2, \ldots, X_n) = P(X_{n+1}|X_n)$ [14].

В контексте генерации текста это означает, что вероятность появления следующего слова в последовательности зависит исключительно от одного или нескольких предыдущих слов. Модели на основе цепей Маркова позволяют эффективно генерировать текст, хотя их способность учитывать долгосрочные зависимости ограничена.

1.2 n-граммные модели

n-граммные модели используют фиксированные последовательности из n слов для предсказания следующего элемента. Для n-граммы вероятность последовательности рассчитывается как произведение условных вероятностей:

$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-n+1})$ [14].

Для построения таких моделей требуется создание корпуса текстов и вычисление частот встречаемости n-грамм. Увеличение значения n позволяет учитывать больше контекста, но увеличивает вычислительную сложность и требует больших объемов данных.

1.3 SVO

SVO (субъект-глагол-объект) — это структура предложений, характеризующаяся расположением элементов в указанной последовательности. В генерации текста использование SVO позволяет моделировать синтаксическую правильность предложений.

Модели на основе SVO часто используют парсинг предложений для выделения этих компонентов, что помогает создавать более связный текст, соответствующий грамматическим правилам.

1.4 Генерация текста на естественном языке

Современные технологии генерации текста на естественном языке (ЕЯ) используют глубокое обучение и большие языковые модели. Эти подходы основываются на трансформерах,

таких как GPT, BERT и их аналоги. Основной принцип работы заключается в обучении модели на огромных объемах текстовых данных для предсказания следующего слова или генерации связного текста в рамках заданного контекста.

Процесс генерации текста начинается с ввода начального текста или запроса, который служит контекстом. Затем модель предсказывает вероятностное распределение для каждого следующего слова, выбирая наиболее подходящее из них.

1.5 Модель Qwen-2.5

Qwen-2.5 является одной из современных языковых моделей, разработанных для генерации текста высокого качества. Она сочетает в себе возможности обработки длинного контекста, управления стилем текста и учета семантических связей. Модель использует архитектуру трансформера с оптимизацией для большего числа параметров, что позволяет ей генерировать текст с высокой степенью связности и стилистической корректности.

Ключевые особенности Qwen-2.5 включают:

— многозадачное обучение, что позволяет ей эффективно справляться с различными задачами, включая генерацию текста, перевод и резюмирование.

— улучшенные механизмы внимания, обеспечивающие точный учет контекста.

— поддержку заданных стилевых ограничений, таких как формальный или разговорный стиль.

Эта модель широко используется в практических приложениях, где требуется сочетание точности и естественности текста.

Вывод

Эти подходы демонстрируют эволюцию технологий, направленных на создание текстов, которые максимально приближены к человеческому уровню. Комбинирование различных методов позволяет достичь высокой точности, связности и стилистической выразительности текста.

2 Конструкторская часть

Данная программа разработана для исследования возможностей генерации текстов с помощью цепей Маркова. Основной целью является анализ влияния структуры исходных данных и начальных условий на осмысленность и качество генерируемых текстов. Программа решает следующие задачи:

1) генерация текстов на основе цепей Маркова с использованием n-грамм разной длины ($n = 2, 3, 5$).

2) исследование особенностей генерации текста при ограниченном наборе обучающих данных, таких как предложения «кошка съела мышку» и «мышку съела кошка», с вариациями порядка слов.

2.1 Принцип работы

2.1.1 Генерация текстов по цепям Маркова

Тексты считываются из заданной директории, после чего они подвергаются токенизации. Для каждого текста извлекаются n-граммы, представляющие собой последовательности из n слов. Эти n-граммы используются для построения цепи Маркова.

Цепь Маркова строится на основе сформированных n-грамм. Каждое состояние цепи представляет собой последовательность из n слов, а возможные переходы к следующему состоянию определяются следующим словом из текста.

На основе созданной цепи Маркова программа генерирует новые тексты. Генерация начинается с заданного начального слова, после чего цепь формирует последовательности, выбирая вероятные переходы. Для оценки создаются тексты с различными начальными условиями.

2.1.2 Исследование с фиксированными предложениями

Программа анализирует случаи, когда обучающая выборка ограничена двумя предложениями: «кошка съела мышку» и «мышку съела кошка». Для этих данных проводится генерация текстов как при сохранении строгого порядка слов (SVO), так и при использовании всех возможных перестановок слов.

Вывод

Программа демонстрирует потенциал цепей Маркова для генерации текстов на основе заданного обучающего корпуса.

3 Технологическая часть

3.1 Средства написания программы

Для реализации программного обеспечения были использованы следующие средства:

— среда разработки PyCharm 2023 community edition [10]

— язык разработки Python 3.10 [11]

Используемые библиотеки:

— os — доступ к функциям ОС [1]

— pathlib — доступ к файлам системы [2]

— collection — хранение цепей (n-грамма : варианты продолжения) [3]

— itertools — создание вариаций предложений из массива слов [4]

— random — генерация случайных чисел [5]

— re — для выделения слов [6]

— beautifulsoup — для чтения html файлов [7] https://www.crummy.com/software/BeautifulSoup/

— docx — для чтения docx файлов [8] https://python-docx.readthedocs.io/en/latest/index.html

— odf — для чтения odt файлов [9] https://pypi.org/project/odfpy/

3.2 Подготовка данных

Листинг 3.1 — Создание текстовых файлов для всех текстов

```
def build_texts(directory):
    texts = list()
    file_order = list()
    for folder in os.listdir(directory):
        folder_path = os.path.join(directory, folder)
        texts_dir_path = os.path.join(folder_path, "texts")

        if not os.path.exists(texts_dir_path):
            os.mkdir(texts_dir_path)

        for filename in os.listdir(folder_path):
            file_path = os.path.join(folder_path, filename)
            if file_path.endswith(EXT):
                text = ext_choice(file_path)
                if text is None:
                    print(f"None pointer return for file: {file_path}\
                        n")
                    continue
                file_order.append(file_path)
                texts.append(text)
```

```
    for i, file in enumerate(file_order):
        texts_dir_path = os.path.join(file, "..", "texts")
        write_into_file(texts[i], texts_dir_path, file.split("\\")
            [-1])
```

Листинг 3.2 — Запись текста в файл

```
def write_into_file(text, vector_dir, file_name):
    file_name = os.path.splitext(file_name)[0] + ".txt"
    file_path = os.path.join(vector_dir, file_name)
    with open(file_path, "w", encoding="utf-8") as file:
        text = text.strip()
        for pattern, replacement in replacements.items():
            text = re.sub(pattern, replacement, text)
        file.write(f"{text}\n")
```

3.3 Основные функции программы

Листинг 3.3 — Чтение текстов в массив

```
def read_texts(directory_path):
    texts = []
    for filepath in directory_path.rglob("texts/*.txt"):
        with open(filepath, "r", encoding="utf-8") as file:
            texts.append(file.read())
    return texts
```

Листинг 3.4 — Поиск самых популярных слов

```
def most_popular_words(texts, top_n=3):
    all_words = defaultdict(int)
    for text in texts:
        words = [word.lower() for word in re.split(r"[-.,!?():;\n\t\r\
            s]+", text) if word]
        for word in words:
            all_words[word] += 1
    all_words = sorted(all_words.items(), key=lambda item: item[1],
        reverse=True)
    return [pair[0] for pair in all_words[:top_n]]
```

Листинг 3.5 — Создание цепи Маркова

```
def build_markov_chain(texts, n):
    chain = defaultdict(list)
```

```
        for text in texts:
            tokens = [word.lower() for word in re.split(r"[-.,!?():;\n\t\r
                \s]+", text) if word]
            ngrams = generate_ngrams(tokens, n)
            for i in range(len(ngrams) - 1):
                chain[ngrams[i]].append(ngrams[i + 1][-1])
        return chain
```

Листинг 3.6 — Генерация текста

```
    def generate_text(chain, n, start_word, max_words=50):
        current = find_start_state(chain, start_word)

        if not current:
            return ""

        generated = list(current)
        for _ in range(max_words len(generated)):
            if current in chain:
                next_word = random.choice(chain[current])
                generated.append(next_word)
                current = tuple(generated[-n:])
            else:
                break
        return " ".join(generated)
```

Листинг 3.7 — Поиск начальной н-граммы

```
    def find_start_state(chain, start_word):
        for n_gram, next_states in chain.items():
            if n_gram[0] == start_word:
                return n_gram
```

Листинг 3.8 — Исследование с предложениями из задания

```
    def experiment_with_sentences(n):
        builded = create_not_SVO(sentences[0])

        chain = build_markov_chain(sentences, n)

        for start_word in start_words:
            generated_text = generate_text(chain, n, start_word, max_words
                =50)
            output_file = RES_DIR / f"{n}_grams_SVO_generated.txt"
            with open(output_file, "a", encoding="utf-8") as file:
```

```
                text = f"                                    \"{start_word}\"\n
                    {generated_text}\n"
                file.write(text)

    chain = build_markov_chain(builded, n)
    for start_word in start_words:
        generated_text = generate_text(chain, n, start_word, max_words
            =50)
        output_file = RES_DIR / f"{n}_grams_NOT_SVO_generated.txt"
        with open(output_file, "a", encoding="utf-8") as file:
            text = f"                                    \"{start_word}\"\n
                {generated_text}\n"
            file.write(text)
```

Листинг 3.9 — Создание вариаций предложений из слов

```
def create_not_SVO(sentence):
    builded = set()
    words = sentence.split()
    for perm in permutations(words):
        builded.add(" ".join(perm))
    return builded
```

Вывод

Предствленнные программы решают задачу генерации текстов.

4 Исследовательская часть

Цель исследования — оценить возможности цепей Маркова для генерации текстов.

4.1 Оборудование

Характеритстики ноутбука:

— процессор intel-core i5-12500H [13]

— ОЗУ 16 Гб DDR4

— ОС Windows 11 [12]

4.2 Результаты исследования

Таблица 4.1 — Сгенерированный текст на основе 2-граммы

Начальное слово	**Сгенерированный текст**
"и"	и др получения органических соединений основоположником которой был а м бутлерова в в китае производился фарфор в алхимический период до нач 13 в стала применяться а в 15 в и производиться селитра в период грудного вскармливания альбумин глобулины если кожа и белки глаз приобрели желтоватый оттенок это повод проверить работу
"в"	в микроэлементах показания к проведению контроль состояния здоровья функционального состояния и здоровья в целом удельный вес различных элементов комплексного лечения зависит от врача и больного зависит та психологическая совместимость которая во многом будет зависеть от доступности сложных органических молекул и стоимости их производства
"с"	с направлением в биохимическую лабораторию измерение различных показателей крови занимает всего несколько часов но почти всегда заключение отдается пациенту следующим утром оценкой результатов должен заниматься врач знающий о соответствии характерных нарушений конкретным заболеваниям самостоятельное ознакомление пациента с результатом анализов позволит заметить нарушение но диагностировать заболевание по всем имеющимся данным способен

Начальное слово	**Сгенерированный текст**
"как"	как и перед сдачей анализов по концентрации липопротеидов либо холестерина следует выдержать 12 14 часовое голодание определение мочевой кислоты нарушения углеводного обмена концентрацию ферментов исключают или фиксируют изменения гомеостаза у пациентов с эпилепсией приступая к диагностике врач прежде всего физические методы исследования спектроскопия в первую очередь следует обратиться в службу
"химии"	химии тесно связана с общей историей химии а вместе с ней с историей естествознания и историей человеческой цивилизации составные разделы истории неорганической химии стали использоваться такие понятия как введенная л полингом электроотрииательность ионные и ковалентные радиусы степень окисления кислоты и нагревании до 1700с этилен представляет собой бесцветный почти нерастворимый в
"веществ"	веществ пигментов энзимов изменение каждого из показателей свидетельствует о проблемах со здоровьем занимающихся физической культурой испортом имеют своей целью допуск к спортивным занятиям систематическое изучение влияния этих занятий на физическое развитие состояние здоровья физическое развитие и функциональные возможности спортсмена при этих обследованиях нужно выяснять также степень сдвигов в состоянии организма

Таблица 4.2 — Сгенерированный текст на основе 3-граммы

Начальное слово	**Сгенерированный текст**
"и"	и др получить информацию о метаболизме обмен липидов белков углеводов выяснить потребность в микроэлементах показания к проведению контроль состояния здоровья не реже 1 раза в год надо следить за тем чтобы в течение года общее количество взятой крови у человека в том числе для продвижения личного бренда
"в"	в микроэлементах показания к проведениюконтроль состояния здоровья не реже 1 раза в год для отдельных заболеваний биохимия является единственной возможностью для объективной диагностики кроме стандартного биохимического анализа проводится исследование специфических показателей использующихся в генетике эндокринологии педиатрии спортивной медицине значения показателей кровь в количестве 5 10 миллилитров берут из вены затем

Начальное слово	**Сгенерированный текст**
"с"	с направлением в биохимическую лабораторию измерение различных показателей крови занимает всего несколько часов но почти всегда заключение отдается пациенту следующим утром оценкой результатов должен заниматься врач знающий о соответствии характерных нарушений конкретным заболеваниям самостоятельное ознакомление пациента с результатом анализов позволит заметить нарушение но диагностировать заболевание по всем имеющимся данным способен
"как"	как и расспрос в целом — не просто перечень вопросов и ответов на них от стиля беседы врача и больного зависит та психологическая совместимость которая во многом определяет конечную цель — облегчение состояния пациента данные анамнеза сведения о развитии болезни условиях жизни перенесённых заболеваниях операциях травмах беременностях хронической патологии аллергических
"химии"	химии тесно связанная помимо органической с другими разделами химии аналитической химией коллоидной химией биохимией кристаллохимией физической химией химической термодинамикой электрохимией радиохимией химической физикой и др на стыке неорганической и органической химии различаются это позволяет проще систематизировать методы и способы исследования в каждой из отраслей неорганическая химия изучает общие правила и
"веществ"	веществ пигментов энзимов изменение каждого из показателей свидетельствует о проблемах со здоровьем

Таблица 4.3 — Сгенерированный текст на основе 5-граммы

Начальное слово	**Сгенерированный текст**
"и"	и др получить информацию о метаболизме обмен липидов белков углеводов выяснить потребность в микроэлементах показания к проведению контроль состояния здоровья не реже 1 раза в год надо следить за тем чтобы в течение года общее количество взятой крови у человека в том числе и в диагностических целях не превышало скорость

Начальное слово	**Сгенерированный текст**
"в"	в микроэлементах показания к проведениюконтроль состояния здоровья не реже 1 раза в год надо следить за тем чтобы в течение года общее количество взятой крови у человека в том числе и в диагностических целях не превышало скорость образования эритроцитов перенесенные инфекционные или соматические заболевания перед проведением биохимического анализа крови человека
"с"	с направлением в биохимическую лабораторию измерение различных показателей крови занимает всего несколько часов но почти всегда заключение отдается пациенту следующим утром оценкой результатов должен заниматься врач знающий о соответствии характерных нарушений конкретным заболеваниям самостоятельное ознакомление пациента с результатом анализов позволит заметить нарушение но диагностировать заболевание по всем имеющимся данным способен
"как"	как и расспрос в целом — не просто перечень вопросов и ответов на них от стиля беседы врача и больного зависит та психологическая совместимость которая во многом определяет конечную цель — облегчение состояния пациента данные анамнеза сведения о развитии болезни условиях жизни перенесённых заболеваниях операциях травмах беременностях хронической патологии аллергических
"химии"	химии тесно связанная помимо органической с другими разделами химии аналитической химией коллоидной химией биохимией кристаллохимией физической химией химической термодинамикой электрохимией радиохимией химической физикой и др на стыке неорганической и органической химии находится химия металлоорганических соединений и элементоорганических соединений неорганическая химия соприкасается с геолого минералогическими науками прежде всего с геохимией и
"веществ"	веществ пигментов энзимов изменение каждого из показателей свидетельствует о проблемах со здоровьем

Таблица 4.4 — Сгенерированный текст на основе 2-граммы SVO

Начальное слово	**Сгенерированный текст**
"кошка"	кошка съела мышку
"мышку"	мышку съела кошка

Таблица 4.5 — Сгенерированный текст на основе 2-граммы не SVO

Начальное слово	**Сгенерированный текст**
"кошка"	кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела
"мышку"	мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка съела мышку кошка
"съела"	съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка мышку съела кошка

Вывод

В ходе исследования был выполнен анализ моделей генерации текстов на основе n-грамм (2-граммы, 3-граммы, 5-граммы) и модификации 2-грамм с учетом SVO-структуры (подлежащее–сказуемое–дополнение). Максимальная длина предложения была выбрана 50. Начальные слова (не считая части SVO) были выбраны по мере самых встречающихся слов в текстах. Слова "и "в "с"оказались в топ 3, а "как "химии"и "веществ"на усмотрение студента.

В текстах на основе 2-грамм заметно отсутствие глобального контекста, что приводит к бессмысленным последовательностям слов, несмотря на локальную связность. Это делает такие тексты слабо воспринимаемыми для человека, так как они не соответствуют привычным языковым паттернам. Тексты, построенные на основе 3-грамм, демонстрируют лучшую связность и некоторую осмысленность благодаря большему объему контекстной информации. Однако в них также часто встречаются повторы и смысловые пробелы, что ограничивает их "человечность". Модели 5-грамм дают наиболее осмысленные результаты за счет учета более широкого контекста, что приближает тексты к привычным для человека структурам. Тем не менее, отсутствие глубокой семантической обработки и знаний о мире ограничивает их практическую применимость.

В части SVO использование строгого порядка слов подлежащее–сказуемое–дополнение позволяет создавать короткие осмысленные фразы, которые легко воспринимаются человеком. При этом нарушение строгого порядка приводит к опасности избыточной повторяемости и

потери связности, что может затруднить восприятие текста. Например, генерация без контроля структуры SVO показала склонность модели к многократным бессмысленным повторам ("кошка съела мышку...").

По мере роста параметров (от 0.5B до 7B), наблюдается улучшение в понимании языковых конструкций, включая синтаксис, грамматику и семантику. Модель демонстрируют более глубокое понимание контекста, повышается их способность к логическому рассуждению и разрешению неоднозначностей.

В контексте генерации текста, увеличивается связность, когерентность и стилистическая гибкость. Креативность, оригинальность и разнообразие текстов также возрастают, наряду со способностью удерживать и использовать длинный контекст.

Модели с большим количеством параметров демонстрируют улучшенные результаты в решении задач обработки естественного языка, включая ответы на вопросы, перевод, суммирование и генерацию кода. Также наблюдается повышение эффективности в переключении между различными типами задач.

В области диалоговых систем, модели большего размера обеспечивают более естественное взаимодействие с пользователем. Улучшается способность удерживать контекст беседы и адаптироваться к собеседнику, а также к обработке неоднозначных реплик.

Увеличение масштаба моделей сопряжено с возрастанием требований к вычислительным ресурсам, энергопотреблению и снижением прозрачности их работы. Также возрастает потенциал для создания дезинформации.

При переходе от модели 0.5B к 7B, наблюдается существенное улучшение в лингвистической компетенции, генерации связного текста и способности решать сложные задачи. Модель 0.5B демонстрирует лишь базовые навыки, в то время как модель 7B способна к более сложному анализу и креативному синтезу информации.

ЗАКЛЮЧЕНИЕ

В ходе исследования были получены следующие выводы:

— отмечается рост связности и "человечности"сгенерированного текста при увеличении n в n-граммах.

— выявлена опасность работы с нестрогим порядком слов.

— увеличение масштаба языковых моделей от 0.5B до 7B приводит к значительному улучшению их характеристик, включая понимание, генерацию и способность решать сложные задачи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Модуль os в Python, доступ к функциям ОС URL: https://docs-python.ru/standart-library/modul-os-python/ (Дата обращения 09.01.2024)

2. Документация Pathlib URL: https://docs.python.org/3/library/pathlib.html (Дата обращения 09.01.2024)

3. Документая collection URL: https://docs.python.org/3/library/collections.html (Дата обращения 09.01.2024)

4. Документация itertools URL: https://docs.python.org/3/library/itertools.html (Дата обращения 09.01.2024)

5. Документация random URL: https://docs.python.org/3/library/random.html (Дата обращения 09.01.2024)

6. Документация re URL: https://docs.python.org/3/library/re.html (Дата обращения 09.01.2024)

7. Документация BeautifulSoup URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (Дата обращения 09.01.2024)

8. Документация docx URL: https://python-docx.readthedocs.io/en/latest/index.html (Дата обращения 09.01.2024)

9. Документация odf URL: https://pypi.org/project/odfpy/ (Дата обращения 09.01.2024)

10. Среда разработки Pycharm 2023 Community edition URL: https://www.jetbrains.com/ru-ru/pycharm/download/other.html

11. Язык разработки Python 3.10 URL: https://www.python.org/downloads/release/python-3100/

12. ОС Windows 11 URL: https://www.microsoft.com/ru-ru/software-download/windows11

13. Intel i5-12500H URL: https://www.intel.com/content/www/us/en/products/sku/96141/intel-core-i512500h-processor-18m-cache-up-to-4-50-ghz/specifications.html?wapkw=12500h

14. Цепи Маркова URL: https://cyberleninka.ru/article/n/modelirovanie-prognoznyh-znacheniy-veroyatnostey-sostoyaniy-sistem-s-ispolzovaniem-tsepey-markova