AI-решения для автоматизации

Май 2024 г.

kept

Основа AI-решения - применение технологий RAG и LLM

01

Высокоэффективные технологии RAG и локальные языковые модели

В 2023 году **появились две технологии, которые позволяют автоматизировать бизнес-процессы и максимально снизить уровень человеческого труда** на рутинные задачи в различных подразделениях компании

RAG (Retrieval Augmented Generation) - архитектура, ограничивающая возможности AI-галлюцинаций и позволяющая передавать контекст

Возможность локального использования больших языковых моделей (mistral, llama3, alpha, wizard, vikuna и т.д) как при помощи решения **ollama**, так и в связке с **langchain** или **llamaindex**

RAG

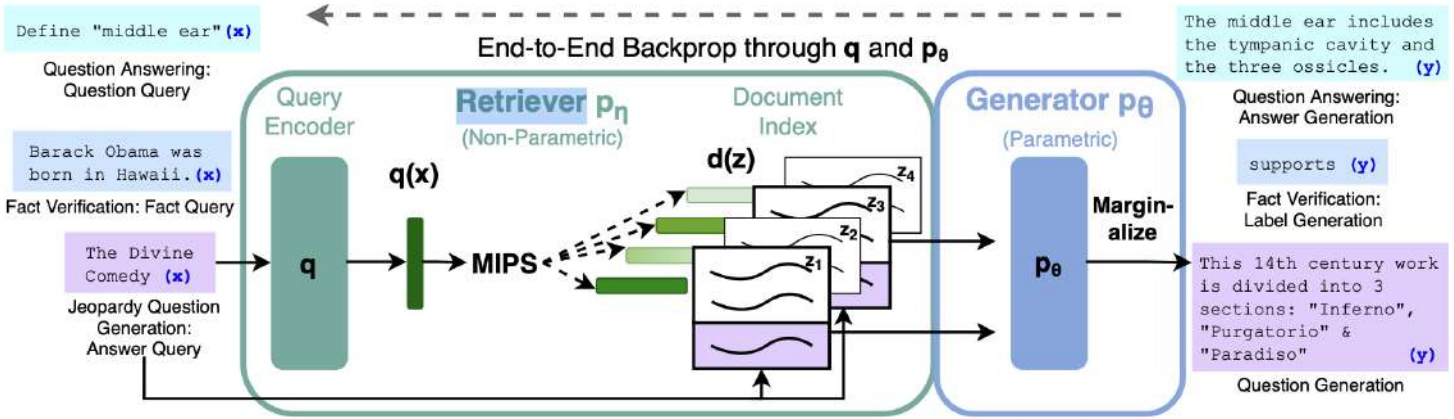
Retrieval Augmented Generation представляет собой способ избавить большие языковые модели (LLM) от галлюцинаций и недостоверных фактов

Функциональность RAG:

1. **Использует LLM для извлечения информации из цепочек связанных документов** путем интеллектуального анализа, а не разметки страниц

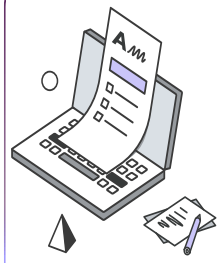
2. **Задает контекст в виде фрагментов текста**, на базе которых LLM должна скомпоновать ответ

3. **Позволяет безопасно использовать** локальные **LLM модели**



LLama.cpp

LLaMA (Large Language Model Meta AI) — большая языковая модель (LLM)



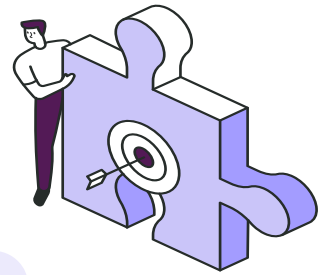
Позволяет использовать локально(не в облаке) генеративные языковые модели уровня GPT 3.5 и выше

- Позволяет **использовать различные варианты "до-обученных"** моделей и интеграции
- **Не требует облачных интеграций и передачи в публичное облако** критических корпоративных данных
- Существуют **совместимые модели, обученные на русском языке** и способные хорошо понимать и формулировать результаты на русском языке

Llama опережает другие модели такие как Mistral, Gemma, Google — это превосходство достигается по результатам девяти тестов: MATH, MMLU, AGIEval, ARC и др.

Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH	AGI Eval
MPT	7B	20.5	57.4	41.0	57.5	4.9	26.8	31.0	23.5
	30B	28.9	64.9	50.0	64.7	9.1	46.9	38.0	33.8
Falcon	7B	5.6	56.1	42.8	36.0	4.6	26.2	28.0	21.2
	40B	15.2	69.2	56.7	65.7	12.6	55.4	37.1	37.0
LLAMA 1	7B	14.1	60.8	46.2	58.5	6.95	35.1	30.3	23.9
	13B	18.9	66.1	52.6	62.3	10.9	46.9	37.0	33.9
	33B	26.0	70.0	58.4	67.6	21.4	57.8	39.8	41.7
	65B	30.7	70.7	60.5	68.6	30.8	63.4	43.5	47.6
LLAMA 2	7B	16.8	63.9	48.9	61.3	14.6	45.3	32.6	29.3
	13B	24.5	66.9	55.4	65.8	28.7	54.8	39.4	39.1
	34B	27.8	69.9	58.7	68.0	24.2	62.6	44.1	43.4
	70B	**37.5**	**71.9**	**63.6**	**69.4**	**35.2**	**68.9**	**51.2**	**54.2**

Три шага к автоматизации



01

Обучаем традиционный **ML-алгоритм связям выполняемых операций с действиями в системах** (проводки, записи, справочники и т.п) на базе экспорта данных из Legacy-систем

02

Обучаем локальную LLM определять типы, виды операций, действий по документам и регламентам/контекстной информации

03

Соединяем три компонента в RAG-подходе:
1. Определяем типы операций
2. Определяем действия, которые необходимо выполнить (выполнить расчеты, сверку, сформировать проводки)
3. Подключаем систему роботизации или скрипты

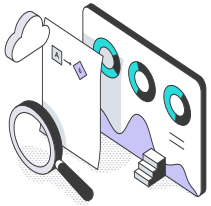
Сложности и риски использования AI

Галлюцинации языковых моделей

Если современную LLM-модель до обучить на массиве новых документов, **то она сможет отвечать на вопросы по ним**. Однако, в случаях, **когда у модели не хватает информации, она переходит в режим "галлюцинирования"**, когда она генерирует правдоподобный текст или данные, не основываясь на фактах

Сложный контекст

Нельзя делать выводы о хозяйственной операции на базе единственного поступившего документа, **всегда нужно анализировать связанные с ним документы (иногда другого типа), а так же "изучить" правила, инструкции и сложившиеся практики**



Безопасность

Большинство LLM-моделей работают в облаке с неизвестным количеством людей, имеющих доступ к данным модели. Это крайне нежелательно для работы с документацией, содержащей персональные данные и коммерческую тайну

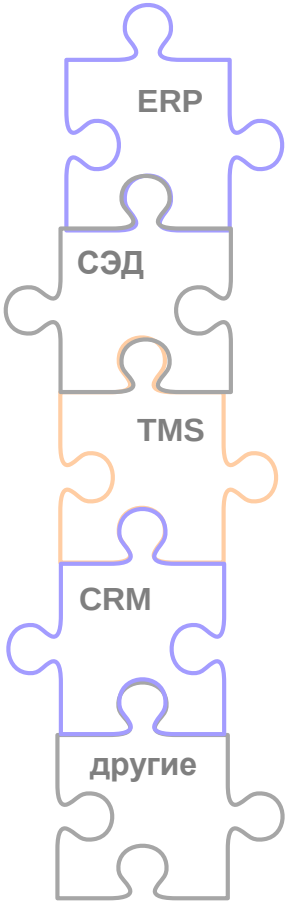
Практические кейсы и результаты

02

Практическая схема реализации кейсов

Кейсы	Входящая информация	LLM	Результат
1 Нормализация НСИ	Справочники, классификаторы, стандарты, ГОСТы и др.	LLM использует контекст и готовит ответ на основе этих знаний	**Устранение** дубликатов **Проверка на соответствие** ГОСТам
2 Due Date	Заявки на платеж, карточки договоров и договорные документы и др.		**Определение целевых дат оплат** по договорным документам
3 Контроль и аудит	Первичные бухгалтерские документы, стандарты ПБУ, МСФО и др.	LLM	**Распознавание** первичных документов и **заполнение** Excel таблиц данными
4 Формирование авансовых отчетов	Чеки, счета, выписки по операциям, билеты и др.		**Распознавание** чеков, **формирование** проводок и авансового отчета
5 Переоценка финансовых вложений	Финансовая отчетность, письма и др.	Пользователь дает задание, которое формирует выборку по embedding-векторам и формирует контекст	**Выполнение** расчетов в Excel на основе распознанной и полученной информации
6 Проверка документов на соблюдение ЛФП*	Карточки договоров, документы авторизации (письма разных форматов)		**Контроль и проверка** документов авторизации на соблюдение ЛФП
7 Формирование, заполнение карточки договора	Договорные документы и их карточки в системах и др.	RAG	**Формирование, заполнение, корректировка** карточки договора
8 Запросы по анализу налоговой практики	Документация (декларации, статьи НК РФ, письма ФНС и др)		**Анализ** налоговой практики к судебным решениям и разъяснениям

Системы



Входящая информация – неструктурированная и структурированная информация различного типа и формата в pdf, png, jpg, txt и др.

ЛФП – лимиты финансовых полномочий

Демо практической реализации кейсов

Кейсы	Ссылка на демонстрацию кейсов
1 Нормализация НСИ	Дедупликация записей в справочнике, кластеризация данных, приведение к единому стандарту
2 Due Date	Анализ договора и сопутствующих документов для определения даты платежа
3 Контроль и аудит	Поиск данных в массивах первичных документов
4 Формирование авансовых отчетов	Обнаружение печати и подписи на финансовых документах
5 Формирование заданий на платеж для сложных условий оплаты	Анализ договора и определение условий для выплаты аванса
6 Проверка документов на соблюдение ЛФП	Анализ авторизационных писем перед подписанием расходных договоров за рамками лимитов
7 Формирование, заполнение карточки договора	Поиск данных в договоре для заполнения сводки или метаданных документа
8 Запросы по анализу налоговой практики	Анализ судебной практики и разъяснений по законодательству

Кейсы в разработке

kept

1 ТОиР

Заполнение технических карт

2 Закупки

Сверка заявки на закупку с ГОСТ, с справочниками, с остатками на складах

3 Стандарты бухгалтерского учета

Формирование автоматических запросов и **генерация** ответов к МСФО, РСБУ

4 Строительство

Чтение, распознавание ПСД, **сравнение** актов КС и **перепроверка, выверка** с бюджетами

5 Бюджетирование

Формирование автоматических запросов и **генерация** ответов к модели бюджетирования и УУ

Функционально технические требования для проектов AI

03

Первый шаг - Экспериментальный проект

Экспериментальный проект состоит из 3-х этапов:

- **Загрузка и формирование embeddings** по документации и др.
- **Формирование запросов в LLM в рамках RAG** для выдачи структурированных данных из документации в JSON-формате
- **Обучение второй ML-модели**, определяющей операции по информации от RAG

01

Обучение модели на исторических данных

- **Загрузка** исторических данных **и проверка** совпадения действий (проводок) с реальными
- **Тестирование** обработки новых документов

02

Обработка данных

- **Подключение скриптов и средств роботизации** для исполнения инструкций по проводкам (и прочим действиям) в реальной учетной системе

03

Подключение скриптов и средств роботизации

Этапы проекта

kept

1 Первичное обследование и данные для обучения

На этом этапе оформляется:

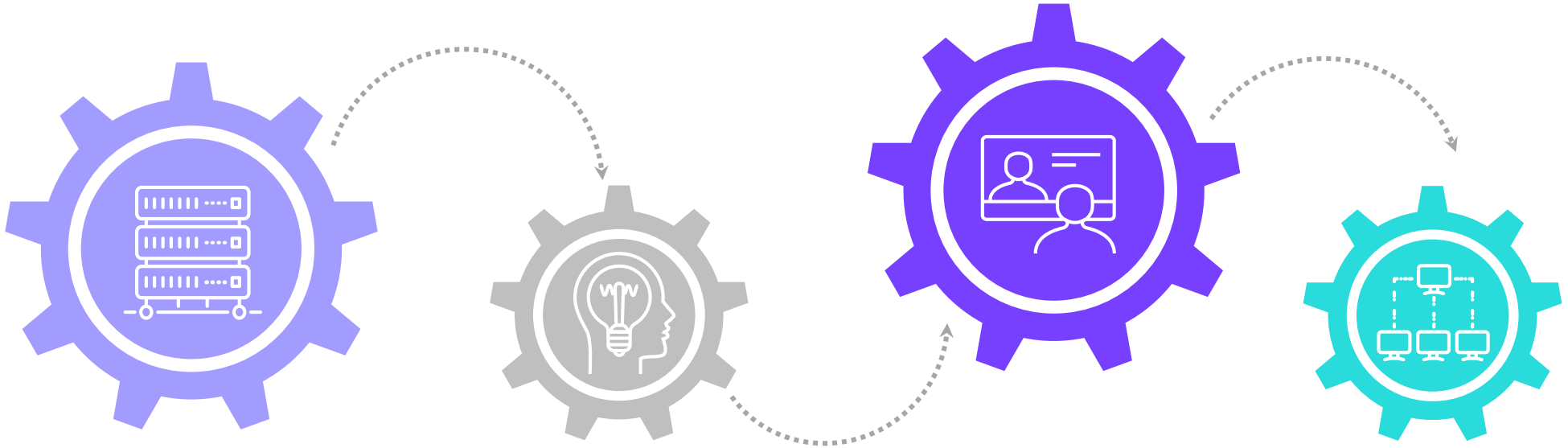
1. Документ с целеполаганием
2. Описываются входные структуры данных с указанием источников
3. Клиент выгружает данные из имеющихся систем и предоставляет доступ к ним
4. Формируется техническое задание на систему

2 Обучение моделей и тестирование на данных

На этом этапе происходит разработка решения:

1. Разворачивается инфраструктура
2. Создается система связанных моделей RAG + ML
3. Проводится серия экспериментов с данными и дополнением контекста

Инфраструктурные требования



Требования к серверу

- 12 vCPU/96 GB RAM/nVidia A40 или A100
- 48Gb памяти доступной для CUDA-оборудования.
- Дисковое пространство > 1Tb (для моделей и тестовых массивов документации)

Требования к списку open-source технологий

Для реализации проекта необходимо использовать только такие решения с открытым кодом, которые можно использовать для последующего включения продукта в реестр Российского ПО

Требования к доступам

- Требуется доступ по RDP к среде разработки (VS Code + python + pip + poetry + библиотеки, git)
- SSH доступ к серверу с CUDA для работы с : ollama, системными библиотеками для CUDA, Docker и др.

Требования к интеграционному ландшафту

Для разработки прототипов и MVP нужна возможность ручной выгрузки внутренней документации на доступное из рабочей среды хранилище. Минимум по дисковому пространству - 2Тб.

Контакты

Роман Малюга

Партнер,
Руководитель
Группы «Цифровые финансы»

T: +7 (968) 691 10 57
E: rmalyuga@kept.ru

www.kept.ru