LLM4Rec в маркетплейсе: от encoder-decoder к small language model

Персонализированные рекомендаций с использованием LLM

Карина Романова

Team Lead CoreLLM: user behaviour

Вместе с командой CoreLLM занимаюсь персонализацией посредством адаптации LLM

Читаю лекции в МФТИ по LLM, мультимодальности и другим темам

ex-SberDevices, Innotech

Лина Вельтман

Senior Data Scientist

Работаю с LLM в задачах анализа поведения пользователей

Опыт в прикладных задачах NLP и Computer Vision

ex Samsung Research Center

О чем поговорим?

Задача рекомендаций

Переход от P5 к LLM

Подход LLM — почему?

Подробнее о данных

Оптимизация модели

Результаты исследования

Задача рекомендаций

Масштабы задачи и вызовы персонализации

Масштаб

Работа с более чем 55 миллионами активных пользователей за неделю

Холодный старт

Рекомендации для новых пользователей без истории

Контекст

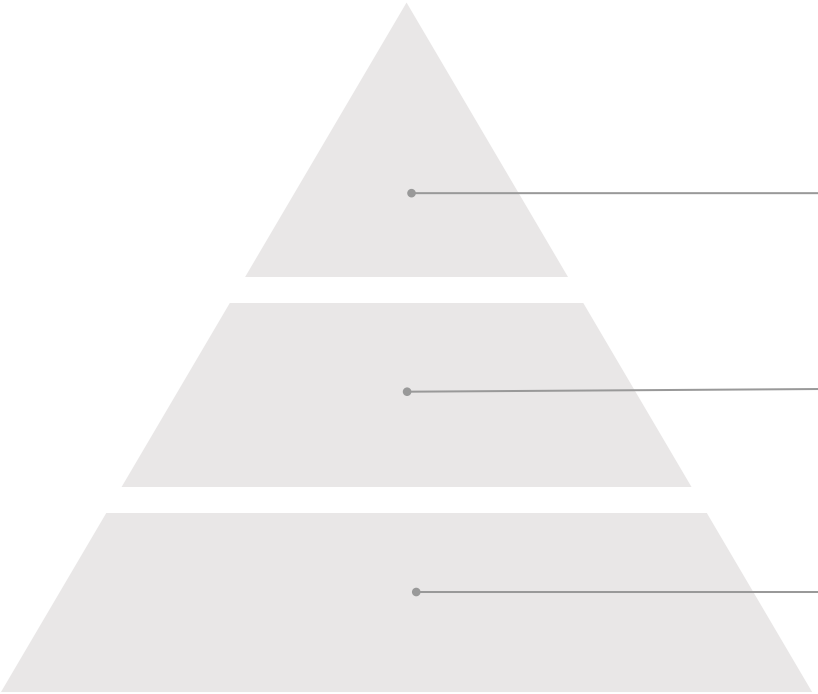
Учет ситуационных факторов при формировании рекомендаций

Динамика

Изменение предпочтений пользователей со временем

От P5 к LLM

Обзор архитектуры P5



Predict
Генерация персонализированных рекомендаций

Personalized Prompt
Формирование запросов с учетом пользователя

Pretrain
Предобучение на больших объемах данных

https://arxiv.org/pdf/2203.13366

Переход к decoder-only архитектуре

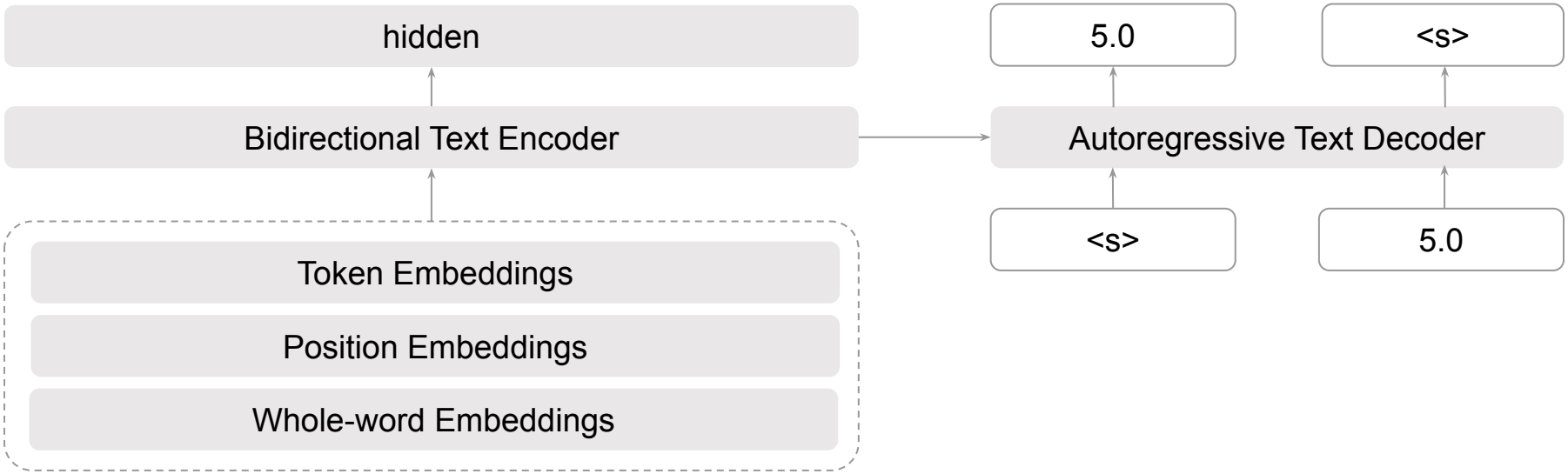
Архитектура	Среднее время инференса (сек/юзер)	Общее время на инференс (часов)	Количество GPU (A800 80GB)
P5	0,1227	170,42	2
Decoder-only LLM (5M users, 1 GPU)	0,023	32	1
Decoder-only LLM (5M users, 4 GPUs)	0,0058	8	4

Переход с T5 → LLaMA / Qwen (устаревшая архитектура, слабая поддержка)

Decoder-only LLM быстрее в 5–20 раз и проще масштабируется

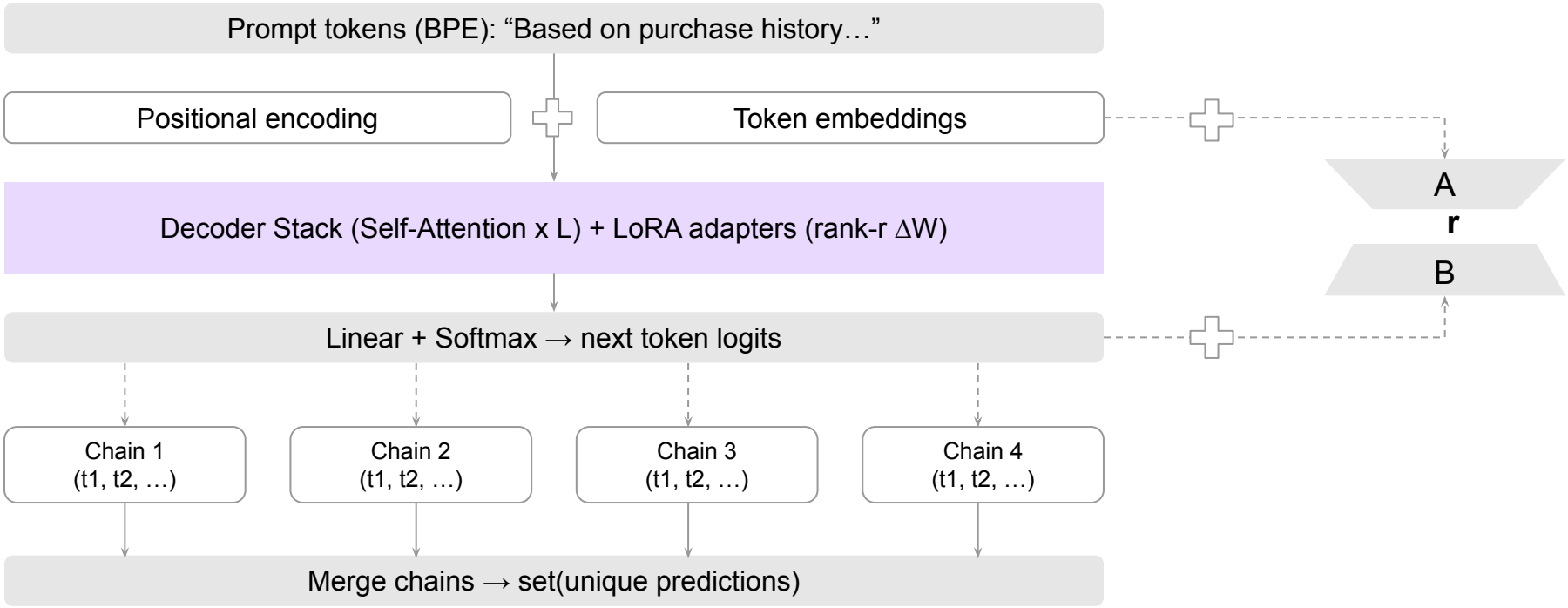
** Meta — запрещенная на территории РФ организация*

Переход к decoder-only архитектуре



Отказ от whole word embeddings (wwe): замедлял инференс, работает сразу над id, дополнительное связывание не нужно

Переход к decoder-only архитектуре



Подход LLM — почему?

Единая задача – универсальная модель

Обучаем LLM только на задаче генерации корзины

Промпты и таргет позволяют получать рекомендации под разные сценарии без переобучения

Гибкая настройка без новой модели

Поменяли таргет → получили похожие товары

Изменили промпт → адаптировались под cold start или категорию

Семантическое поведение

Модель учитывает не только ID,
но и паттерны:

контекст + история + интересы →
персонализированный ответ

Простое масштабирование

Один стек под все: смена архитектуры (T5 → LLaMA) и токенизатора без переписывания пайплайна

** Meta — запрещенная на территории РФ организация*

Подробнее о данных

Работа с пользовательскими данными

История покупок

Данные о покупках за последние 6 месяцев

Атрибуты пользователя

Пол, возраст, гео, покупательская способность

Вишлисты

Товары, добавленные в избранное

Корзины

Товары в корзине пользователя

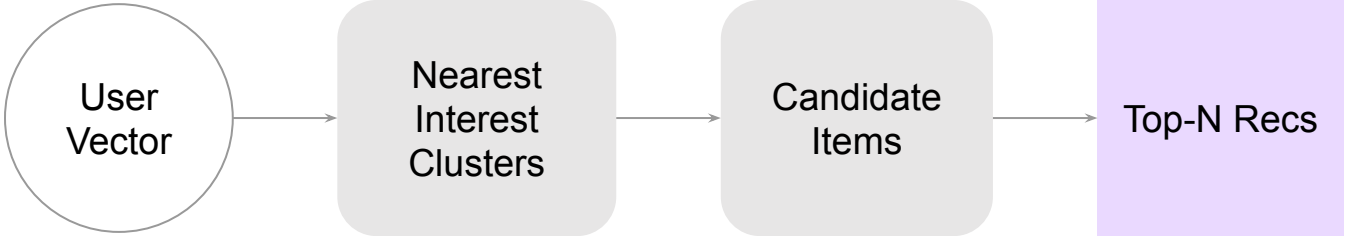
Работа с пользовательскими данными

1

Выгрузка сырых данных (SQL, Hadoop, HDFS)

2

Расчет кластеров интересов



Работа с пользовательскими данными

1

Выгрузка сырых данных (SQL, Hadoop, HDFS)

2

Расчет кластеров интересов

3

Выгрузка вишлистов, корзин

4

Выгрузка похожих товаров

5

Выгрузка похожих пользователей

6

Объединение данных

7

Предобработка и сплит (train/val/test)

Подбор фич

Атрибуты пользователей

Демографические данные и история покупок показали наибольшую значимость

Атрибуты товаров

Ценовой сегмент существенно влияет на точность рекомендаций

Данные

Доработка структуры данных увеличила Hit Rate на 12%

Промпты

Улучшение формулировок промптов повысило релевантность на 15%

Работа с пользовательскими данными

Состоит из двух частей. Работы с промптами,
а именно формированием input и target.

Пример:

инпут:
социально-демографические признаки,

история покупок

таргет:
корзина, вишлист, товары от похожих пользователей, похожие товары на корзину, похожие товары на вишлист, товары из одинаковых кластеров интересов по купленным товарам

Эксперименты с промптами. Управляющие слова

- Управляющие слова в нашем промпте — это описания признаков пользователя (geo, age, gender, purchasing power), которые остаются неизменными, меняются только числовые значения при них
- Пришли от 8 шаблонов к 1
- Русский язык ⟶ Английский язык

Input:

Промпт: "Based on the purchase history: [8363834, 482976, 4710760, …]. What next products will be bought by the user: 123, geo: 567, age: 48, gender: 1, purchasing power: 0.5?"

Эксперименты с промптами. Эксперименты с таргетом

Target: [483957, 73830364, 237494962, 8394372, …]

- Товары от похожих пользователей (50% товаров из вишлиста и корзины от похожих юзеров, остальное — вишлист/корзина текущего юзера)
- Похожие товары (25% похожих на вишлист, 25% — на корзину, остальное — вишлист / корзина)
- Комбинированный таргет (товары от похожих пользователей + похожие товары + товары из кластеров интересов + собственные товары)

Эксперименты с промптами. Ключевые результаты

- Рост Precision & Recall при использовании похожих товаров
  и данных от похожих пользователей
- Лучшие метрики у стратегии с похожими товарами (~17.5%)
- Оптимальное распределение таргета улучшает метрики
  без увеличения объема данных

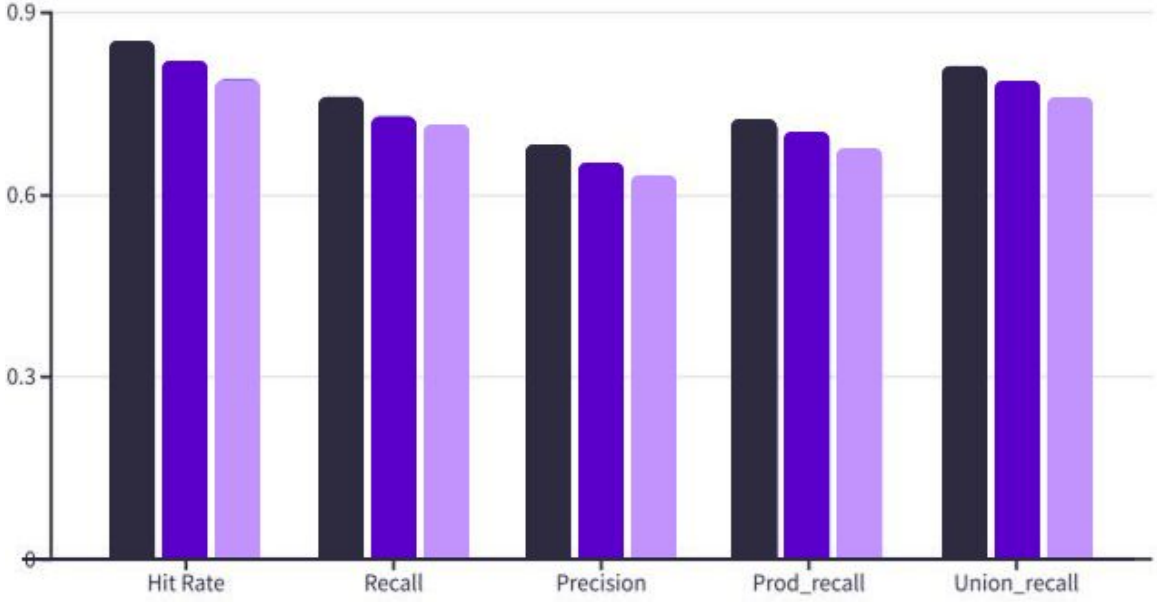
Эксперименты с промптами

Модель	ΔPrecision	ΔRecall	ΔRevenue
Baseline (125K)	−99.92%	−1.75%	−99.87%
Baseline (1M)	−93.61%	−1.62%	−94.24%
Похожие пользователи (1M)	+43.81%	+0.73%	+39.81%
Похожие товары (125K)	**+54.26%**	**+0.90%**	**+42.72%**
Комбинированный таргет (125K)	+28.23%	+0.42%	+22.33%

Δ = относительная разница к **Baseline (5M)**

Оптимизация таргета за счет использования данных о похожих пользователях и товарах позволяет значительно улучшить метрики рекомендаций. Наилучшие результаты показала стратегия с **похожими товарами**

Выбор количества данных для обучения



125k пользователей

1 млн пользователей

5 млн пользователей

Оптимизация модели

Влияние выбора токенизатора

Стандартная токенизация

1 – 2 – 3 – 4 – 5 – 6

- Каждая цифра как отдельный токен
- Длинные последовательности, медленный инференс

Оптимизированная токенизация

1 2 3 – 4 5 6

- Плотная группировка
- Короче последовательности, быстрее обработка

Тип токенайзер	часть текста	средняя длина	сокращение длины
individual_digits=True	input	658	0 %
individual_digits=True	output	2614	0 %
individual_digits=False	input	310	53%
individual_digits=False	output	1160	56%
обученный на наших данных с individual_digits=False	input	284	57%
обученный на наших данных с individual_digits=False	output	1042	60%

Влияние выбора токенизатора. Результаты

Сокращение длины последовательности на 40%

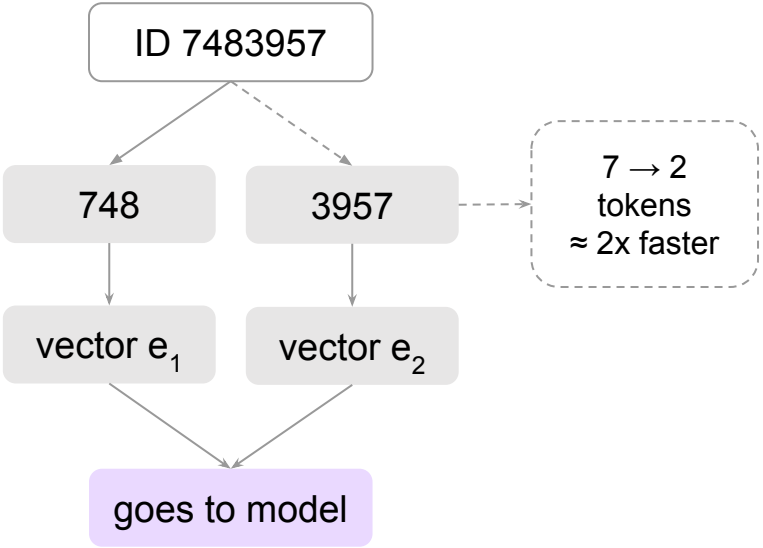
Ускорение инференса в 1.5-2 раза

model	sec / user	токенов на id товара
qwen-2.5-0.5b	0.2	9
llama-3.2-1b*	0.11	3

** Meta — запрещенная на территории РФ организация*

Почему это работает?

- ID ≠ магический объект — для модели это просто строка
- BPE режет строку на частые кусочки: 7483957 → 748 + 3957 (2 токена вместо 7 цифр)
- Каждый кусочек → индекс → вектор:embedding-слой учится, когда такие кусочки встречаются рядом
- Меньше токенов → быстрее: 7 → 2 токена ≈ в 2 раза меньше вычислений в self-attention
- Обобщение «по префиксу»: ID, начинающиеся на 748…, делят общий вектор «748» → модель переносит знания



Настройка параметров генерации

Промпт:

"Based on the purchase history: [8573934, 482975, 472075, …]. What next products will be bought by the user: 123, geo: 567, age: 48, gender: 1, purchasing power: 0.5?"

1 цепочка:

7483957,
73830364,
237494962,
8394372, …

2 цепочка:

947295,
8394372,
73830364,
8394372, …

3 цепочка:

8394372,
73830364,
947295,
8394372, …

4 цепочка:

73830364,
947295,
237494962,
8394372, …

Все независимые цепочки мержим и оставляем только уникальные

4 — количество независимых генераций

Ускорение до 28 раз (на 15%)

Настройка параметров генерации

0.7 Temperature

Оптимальное значение для баланса разнообразия и точности

50 Top-k

Количество наиболее вероятных токенов для выбора

310 токенов — максимальная длина последовательности

Сокращение времени генерации

0.9 Top-p

Порог вероятности для отбора токенов
Дает баланс детерминированности и случайности

Настройка параметров генерации: результаты экспериментов

N — количество независимых последовательностей рекомендаций	Δ (10 vs 1)	Δ (20 vs 1)
Hit rate	+189.3%	+201.4%
Precision	+170.0%	+180.0%
Recall	+179.6%	+232.5%
Revenue	+231.0%	+231.8%
Top bias	0.0%	−66.7%
Recall by category	+43.3%	+45.6%
Ускорение генерации	11%	15%
Среднее кол-во уникальных рекомендаций для юзера	0.0%	-0.43%

Итоговый сетап

Модель: Facebook OPT-69M (OPT-125M с 4 скрытыми слоями вместо 12)

Инференс: TRT-LLM, FP16

Производительность: Batch ~4096, ускоренная генерация

Архитектура вывода: 4 независимые цепочки рекомендаций

Токенайзер: Оригинальный, без переобучения

** Meta — запрещенная на территории РФ организация*

Результаты оптимизаций

Количество юзеров	Количество A800 80GB	Скорость, сек/юзер	Времени на инференс, часы
1 млн	1	0.023	6.4
1 млн	2	0.0115	3.2
1 млн	4	0.0058	1.6
5 млн	1	0.023	32
5 млн	2	0.0115	16
5 млн	4	0.0058	8
10 млн	1	0.023	128
10 млн	2	0.0115	64
10 млн	4	0.0058	32
50 млн	1	0.023	320
50 млн	2	0.0115	160
50 млн	4	0.0058	80

Мы достигли высокой скорости обработки данных. Время инференса составляет всего 0.023 секунды на пользователя

Результаты исследования

Результаты исследования

	Прирост SLM (facebook/opt-125m) обучена на 125k юзеров **(x раз)** относительно **P5 (baseline)** 3_500_000 юзеров
HitRate top (200)	40.70×
Revenue top (200)	36.34×
Top bias (top 200)	0.64× ↓
Recall by Category (top 200)	2.53×
Precision (top 200)	9.13×
Recall (top 200)	50.61×
MAP	36.63×
NDCG	28.46×
Novelty	4.93×
Serendipity*1000	33.35×

** Meta — запрещенная на территории РФ организация*

Выводы

- LLM решают cold start и адаптацию к изменяющимся интересам
- История покупок + демография = ключевые признаки
- Похожие товары/пользователи улучшают Precision, Recall, Revenue
- Decoder-only модели ускорили обучение и упростили поддержку
- Токенизация + генерация = 28× ускорение, 0.023 сек/пользователя @ batch=4096
- Модель масштабируема и готова к росту

Потенциал модели и дальнейшие шаги

Масштабирование моделей

Переход на модели большего размера: 0.5b, 1b, 1.5b

Доработка данных

Добавление сезонности, временных фич, мультимодальности, RL

Оптимизация инференса

Квантизация, кастомный токенизатор, новые фреймворки

Спасибо за внимание и до встречи!

Telegram-канал
что-то на DL-ском

Telegram-канал
WB Space

Блог WB Tech
на Habr