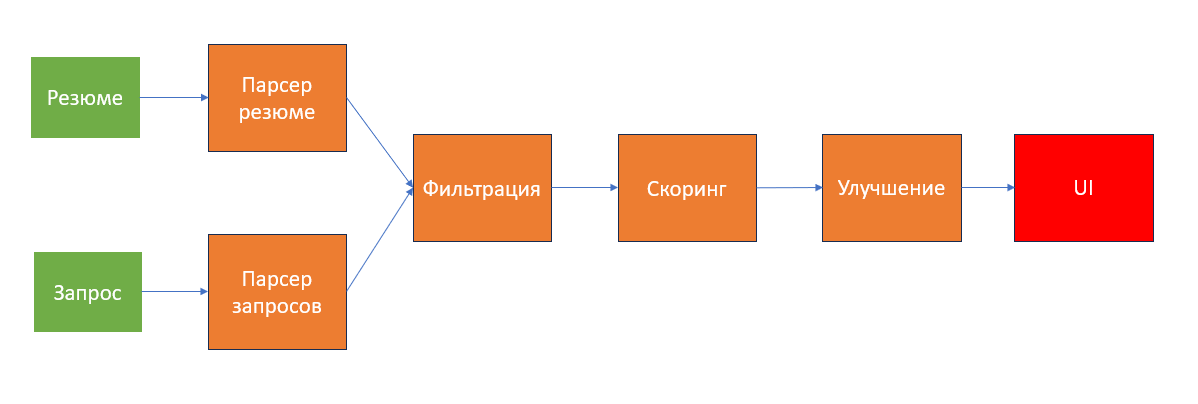
Интеллектуальный поиск резюме

Компания Юзтех располагает базой в сотни тысяч кандидатов, и рекрутёрам сориентироваться в этой базе достаточно сложно. Поэтому перед DS-командой Юзтех была поставлена задача разработать систему интеллектуального поиска. Вот так можно сформулировать основные требования:

* на вход система должна принимать запрос на специалиста в произвольной текстовой форме
* система должна выдавать упорядоченный короткий список кандидатов (вплоть до 1000 человек), удовлетворяющих запросу, каждому кандидату сопоставляется процент соответствия запросу
* для процента соответствия запросу должна предоставляться интерпретация, т.е. он должен разбиваться на составляющие --- навыки, специальность, образование, компания, локация и др.
* должна быть представлена возможность улучшения резюме (чего резюме не хватает, для того чтобы стать идеальным)

Схема решения

Наше решение состоит из компонент, указанных на схеме:



Основная логика может быть разбита на 5 блоков:

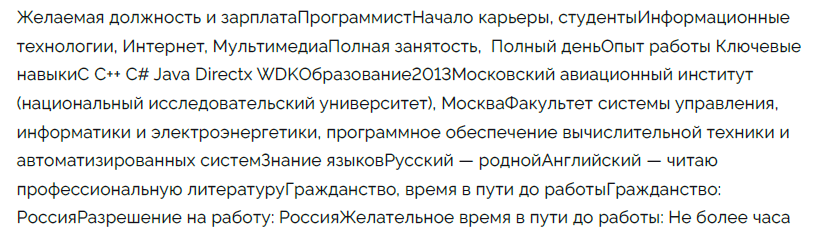
* парсер резюме
* парсер запросов
* фильтрация
* скоринг
* улучшение

Коротко обсудим далее каждый из этих 5 блоков.

Парсер резюме

Задача парсера состоит в выделении нескольких десятков признаков из резюме, эти признаки используются как в системе моделей, так и выводятся на UI. Для простых признаков вроде занятости мы используем регулярные выражения, для более сложных признаков вроде компаний, в которых работал кандидат, мы используем NER-модели, для уточнения признаков мы используем LLM (например, в тех случаях, когда мы не смогли определить пол кандидата обычными методами). Парсинг резюме запускается только для новых резюме и происходит ночью. Кроме того, в конце парсинга создаются и обновляются определённые структуры, для того чтобы последующий поиск был более эффективным, чтобы можно было быстро найти кандидатов, владеющих pega, или работавших на Мосбиржу.

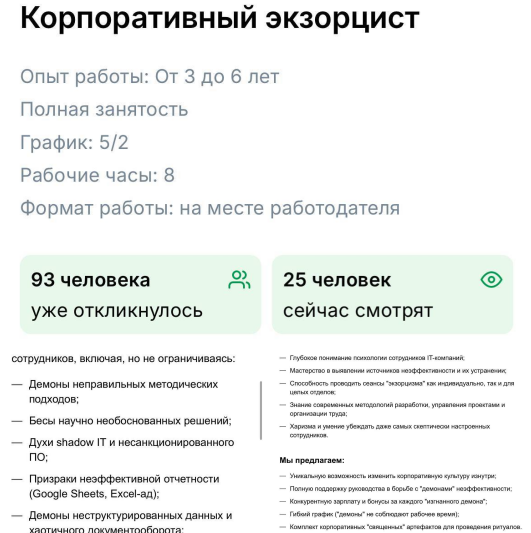
Отметим, что не все резюме легко распарсить даже человеку:



Парсер запросов

Задача данного парсера состоит в выделении ключевых требований из запроса. Основная ставка делается на выделение ключевых навыков, однако также мы выделяем специальность кандидата, грейд кандидата, компанию, в которую ищут кандидата, а также ряд других факторов. В парсере активно используется LLM. Парсер способен эффективно работать с запросами разной длины от одного слова, до pdf-файла из нескольких страниц. Таким образом, запрос на поиск похожего специалиста с резюме из 6 страниц сжимается до 3 строчек.

Запрос может быть в любой форме, однако не на все запросы можно будет найти кандидатов:



Фильтрация

База Юзтех содержит несколько сотен тысяч кандидатов. Однако большинство из них нет никакого смысла скорить на конкретную вакансию. Если ищут BE-разработчика, то нет смысла рассматривать тестировщиков. Поэтому мы разработали классификационные модели, которые определяют специализацию кандидата (пара десятков специализаций), а также грейд кандидата. Далее мы фильтруем кандидатов, оставляя только тех, чья специализация релевантна специализации запроса, и чей грейд не сильно отличается от грейда запроса. Также на этапе фильтрации с помощью LLM определяется ведущий навык. Если количество кандидатов удовлетворяющих всем фильтрам слишком мало, то фильтрация по специализации и грейду отключается, и мы ищем по всем кандидатам, обладающих ведущим навыком. Специализации и грейды кандидатов определяются в конце парсинга резюме, ночью, когда готовится база, поэтому фильтрация происходит очень быстро.

Скоринг

На данном этапе мы считаем процент соответствия кандидата вакансии, состоящий из нескольких компонент:

* скор по ключевым словам
* скор по образованию
* скор по компании
* скор по грейду
* скор по специализации
* ….

Это самый долгий этап поиска (весь поиск в среднем занимает минуту). Мы стараемся, с одной стороны, сделать так, чтобы скоры считались максимально быстро, а с другой стороны, чтобы они были максимально интерпретируемыми, как и финальный процент соответствия вакансии.

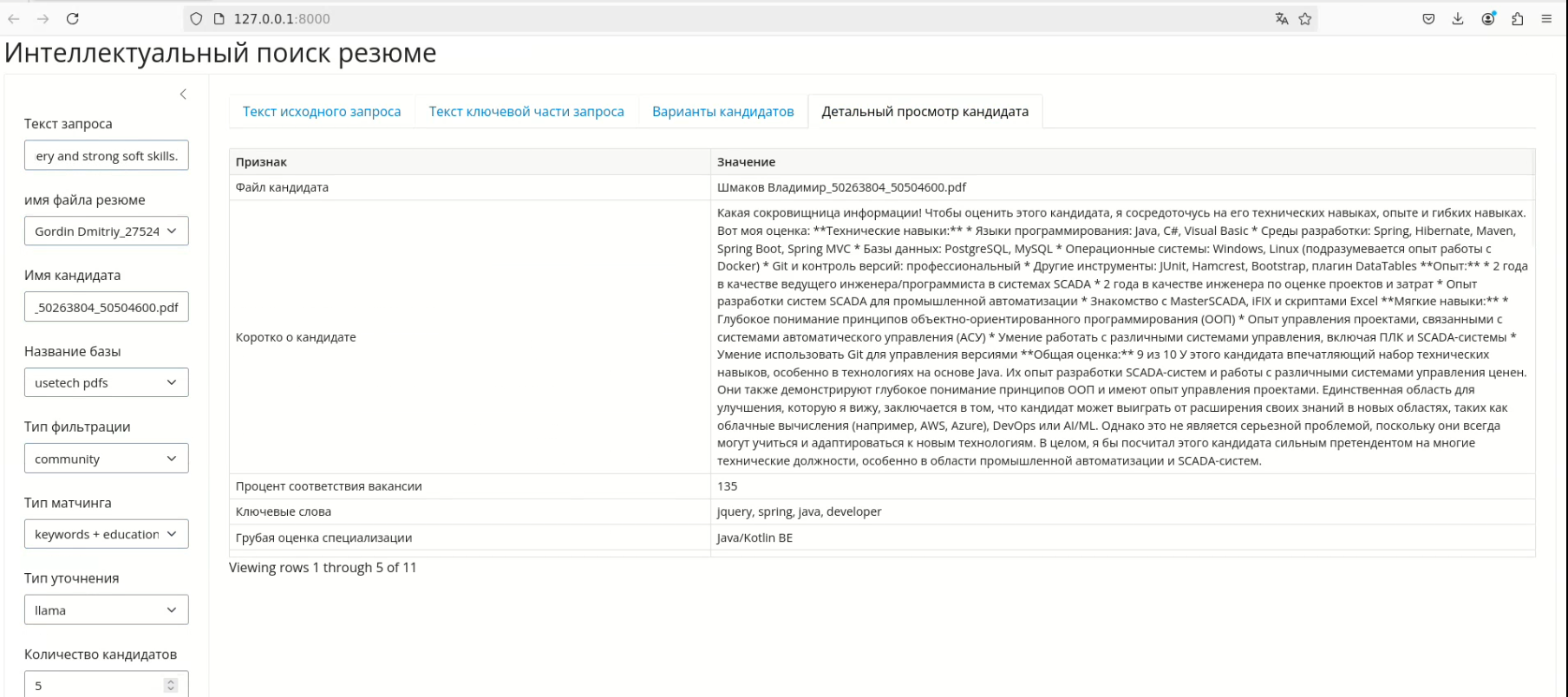
Отметим, что в первой версии прототипа скор мог быть выше 100%.  


Улучшение

Данный блок вызывается только для избранных резюме, которые бы рекрутёр хотел бы улучшить. Мы пытаемся ответить на вопрос, что надо сделать с резюме, чтобы оно лучше соответствовало требованиям запроса. Мы находим ключевые навыки, которые есть смысл добавить, активно используем LLM, смотрим на другие факторы. Дальше изменённая информация вставляется в распарсенное резюме и скор резюме пересчитывается. Кроме того, мы выводим и мнение LLM в свободной форме, содержащее как общие рекомендации (вроде расписывать достижения в опыте работы), так и рекомендации специфичные для резюме и запроса (подчеркните опыт кандидата с TestIt и Kaiten в мобильном тестировании). Кроме того, мы можем по двум резюме и запросу определить, чего нужно добавить из одного резюме в другое резюме, чтобы оно соответствовало запросу лучше.

Эволюция

Наш первый прототип был сделан на shiny, процент соответствия был произвольным числом, мы не следили за форматированием, поиск считался 3 минуты.



Теперь наш UI --- это сваггер.



А наш микросервис стал частью большого приложения, поиск выполняется за минуту, проблем с форматированием нет, процент соответствия не может быть больше 100%.

