Оффлайн оценка качества поиска по технической документации

Шелепугин И.М.

ОЕЯ

09.10.2026

План

1. Изменения в системе

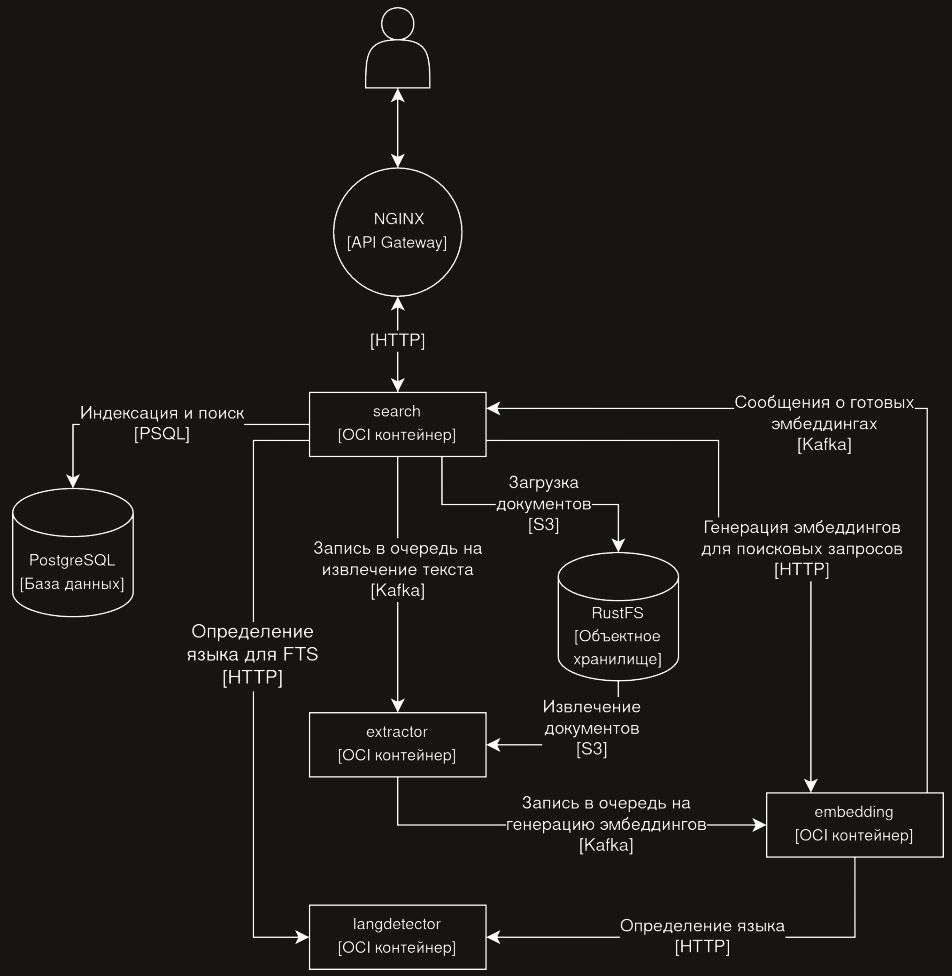
2. Метрики качества поиска

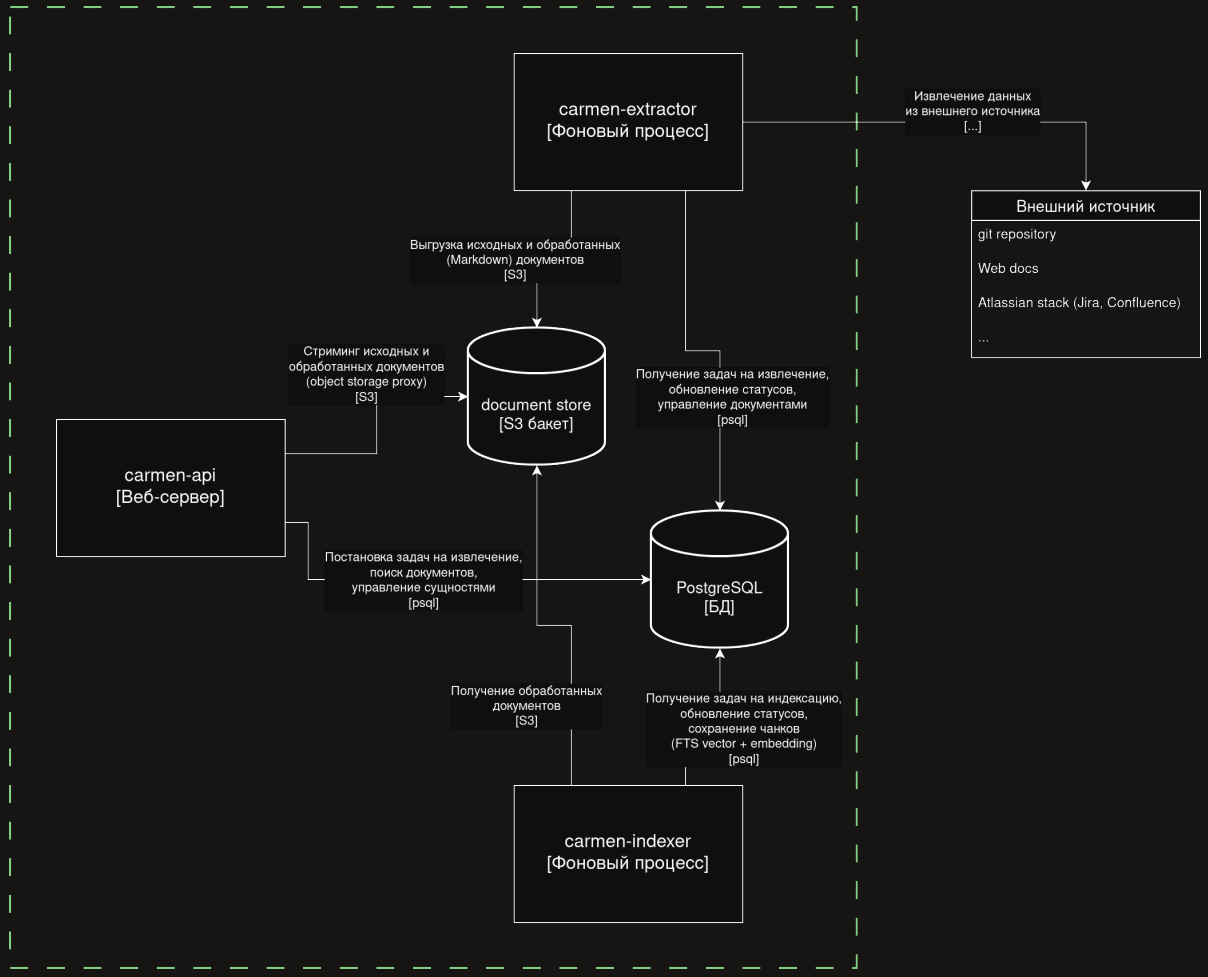
3. Измерение метрик

Архитектура

Минусы предыдущей архитектуры:

1. Очень много времени уходит на передачу сообщений
2. Нет источника правды для состояния коллекции/документов
3. Избыточное дробление на сервисы





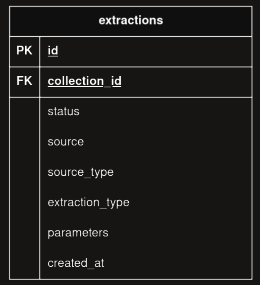
Новая архитектура — попытка решить описанные проблемы.

1. БД как единственный источник правды
2. Взаимодействие сервисов посредством данных

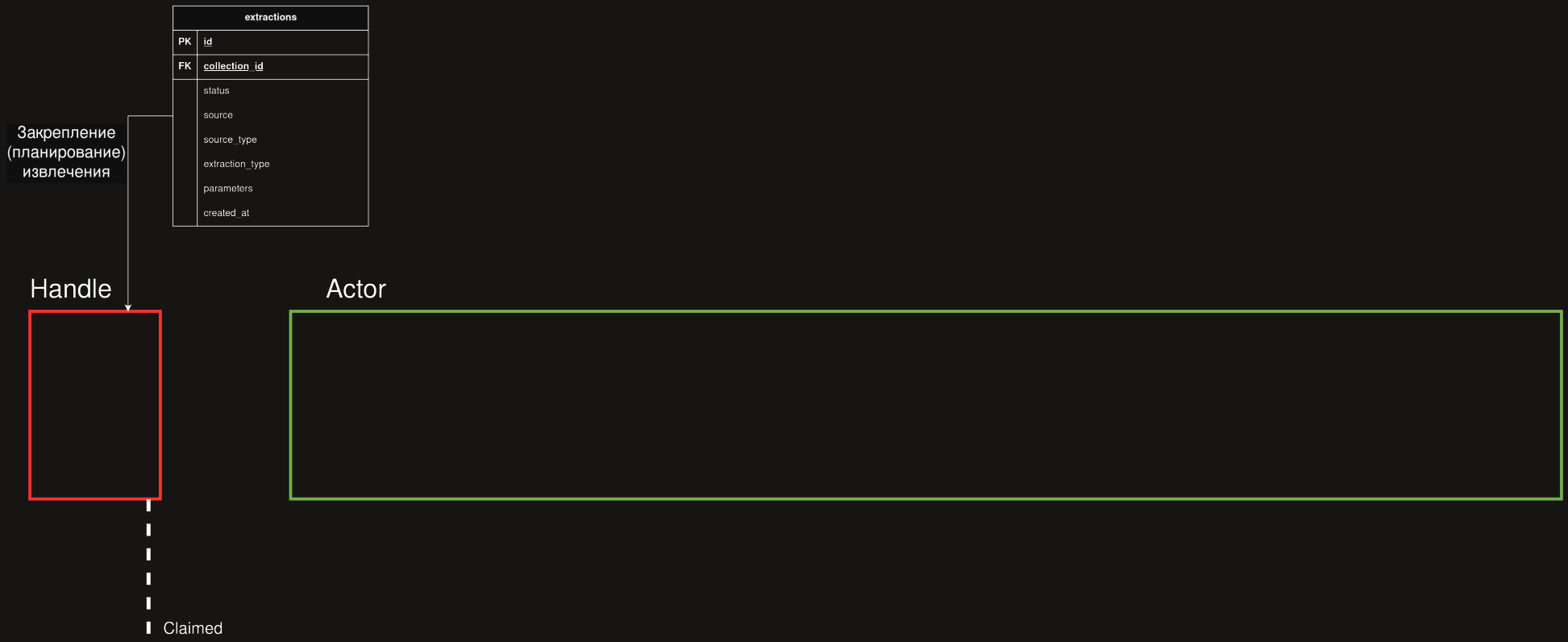
Extractor

Подсистема для извлечения документации из внешних источников.
Ранее поддерживалась только прямая загрузка документов.

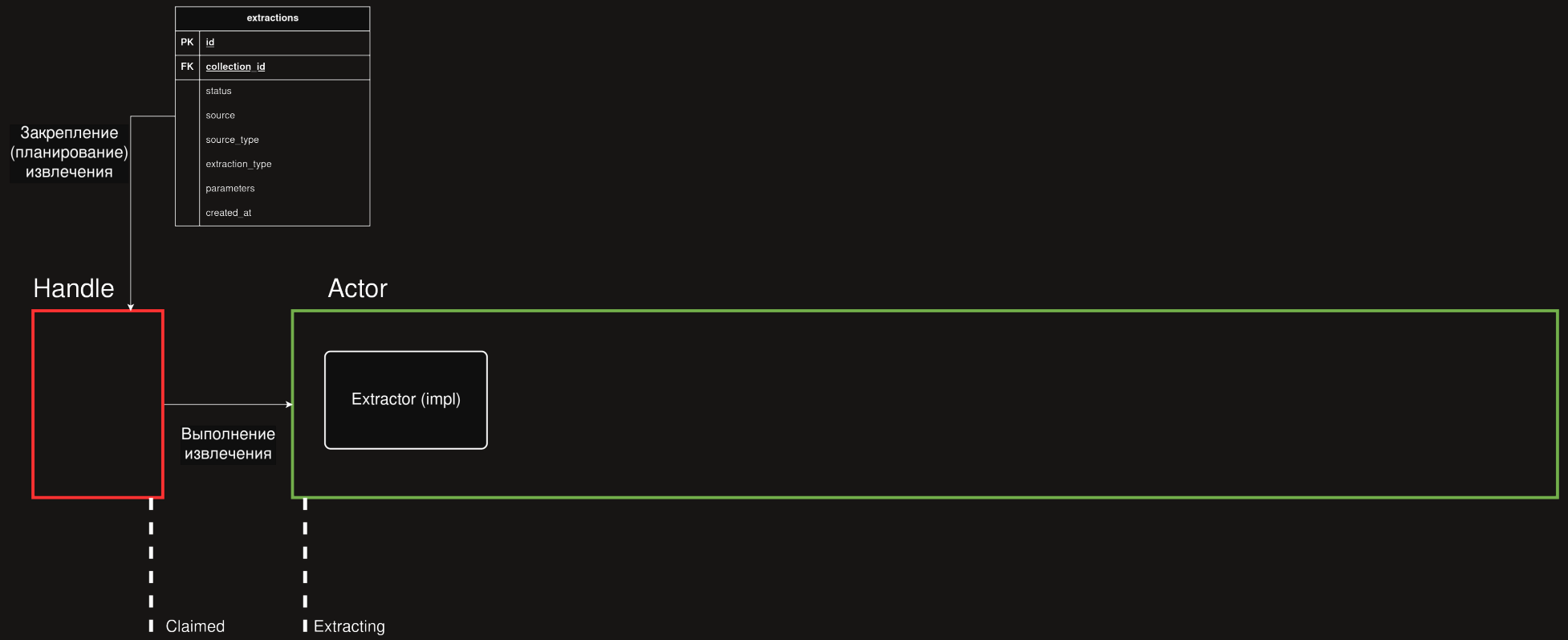
- Event sourcing, все извлечения хранятся в таблице `extractions`
  - Есть источник правды для состояния коллекции
- Без состояния, простое горизонтальное масштабирование
- Поддерживает отмену задач
- Идемпотентность



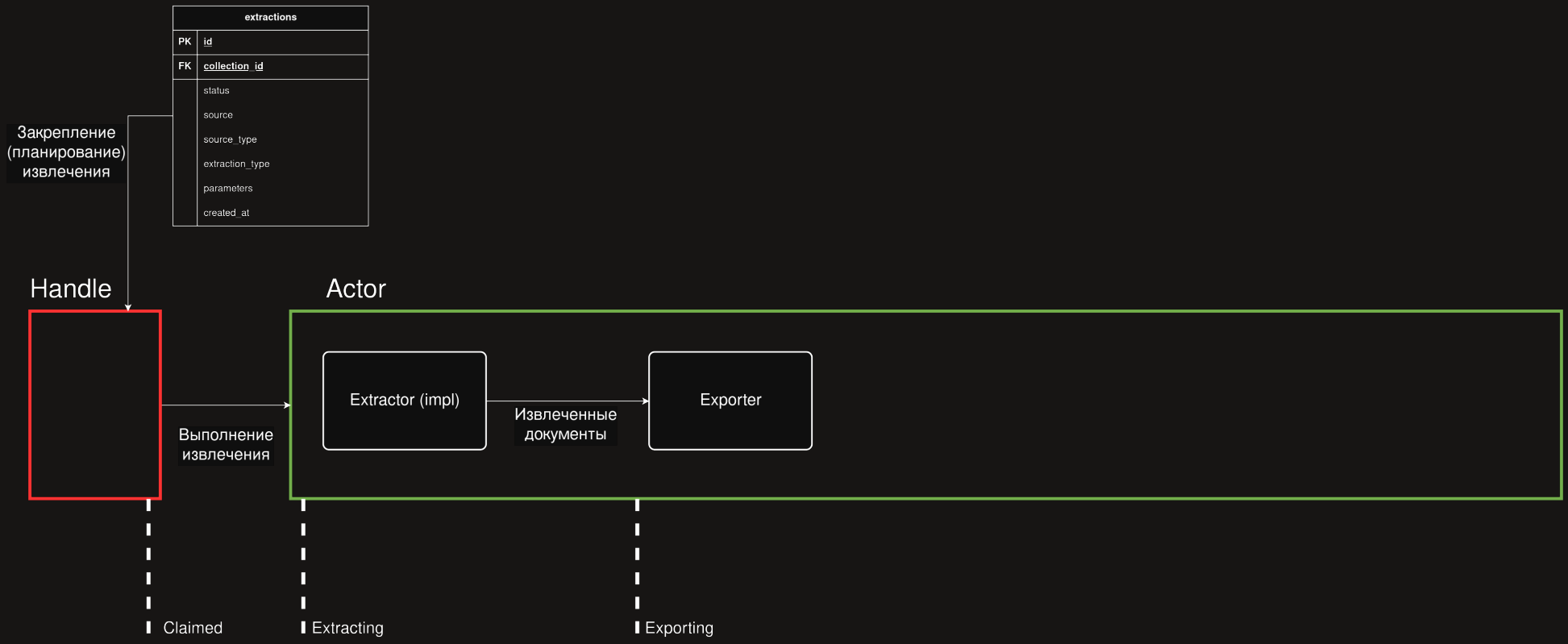
1. Раз в N секунд проверяет таблицу `extractions`



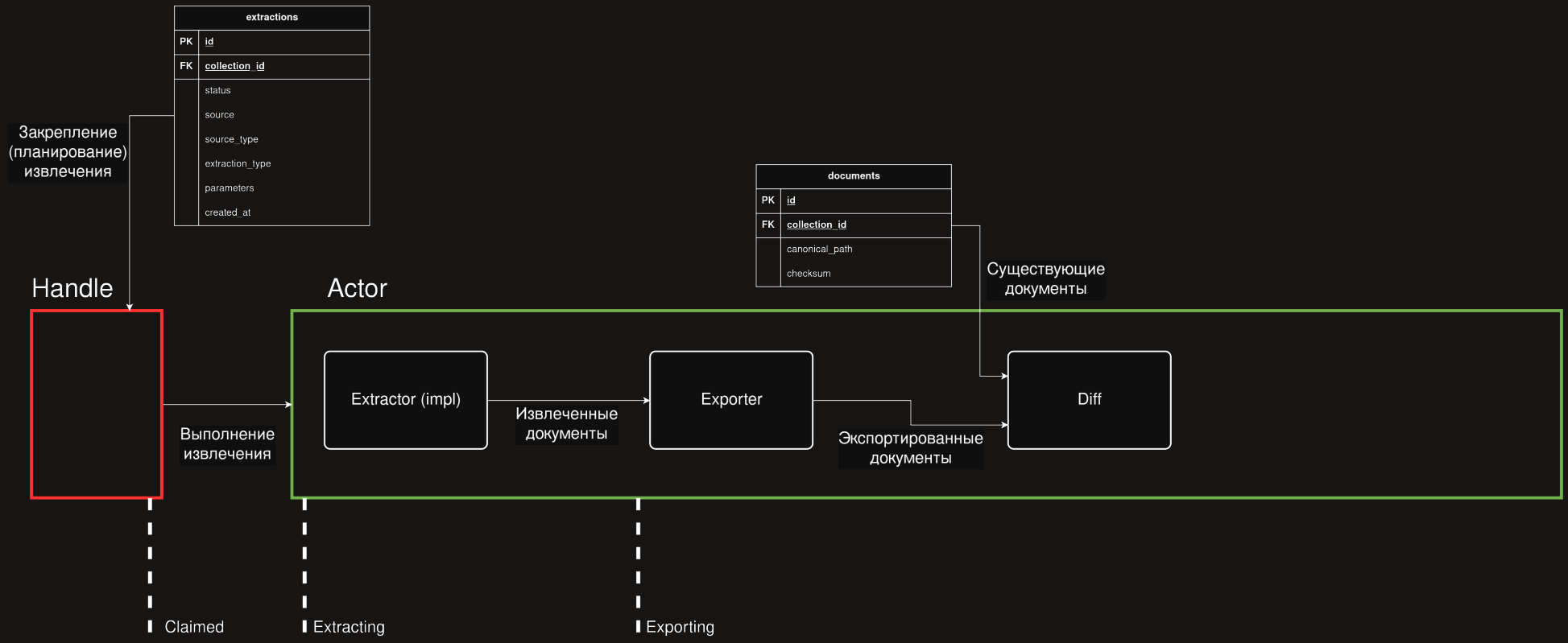
2. Если есть извлечение (задача) с `status = pending`, забирает её



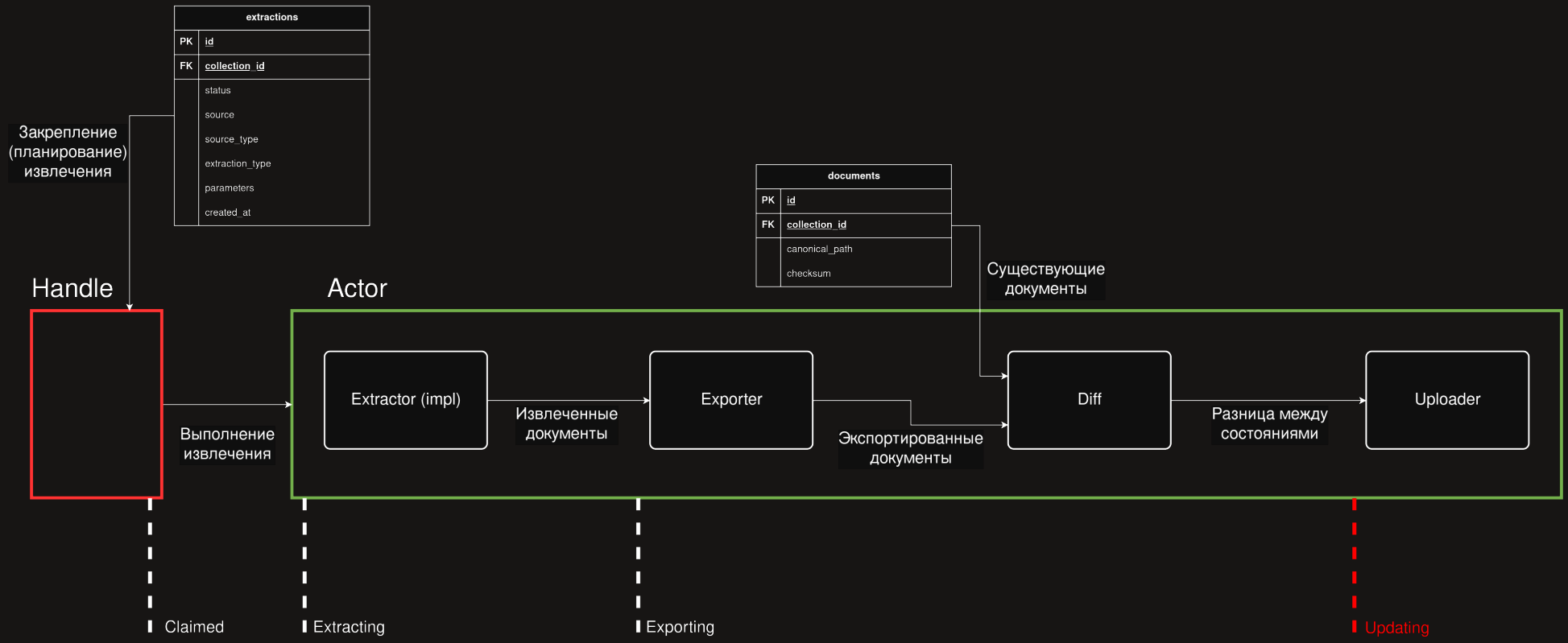
3. Выбирает экстрактор для указанного источника и извлекает документы



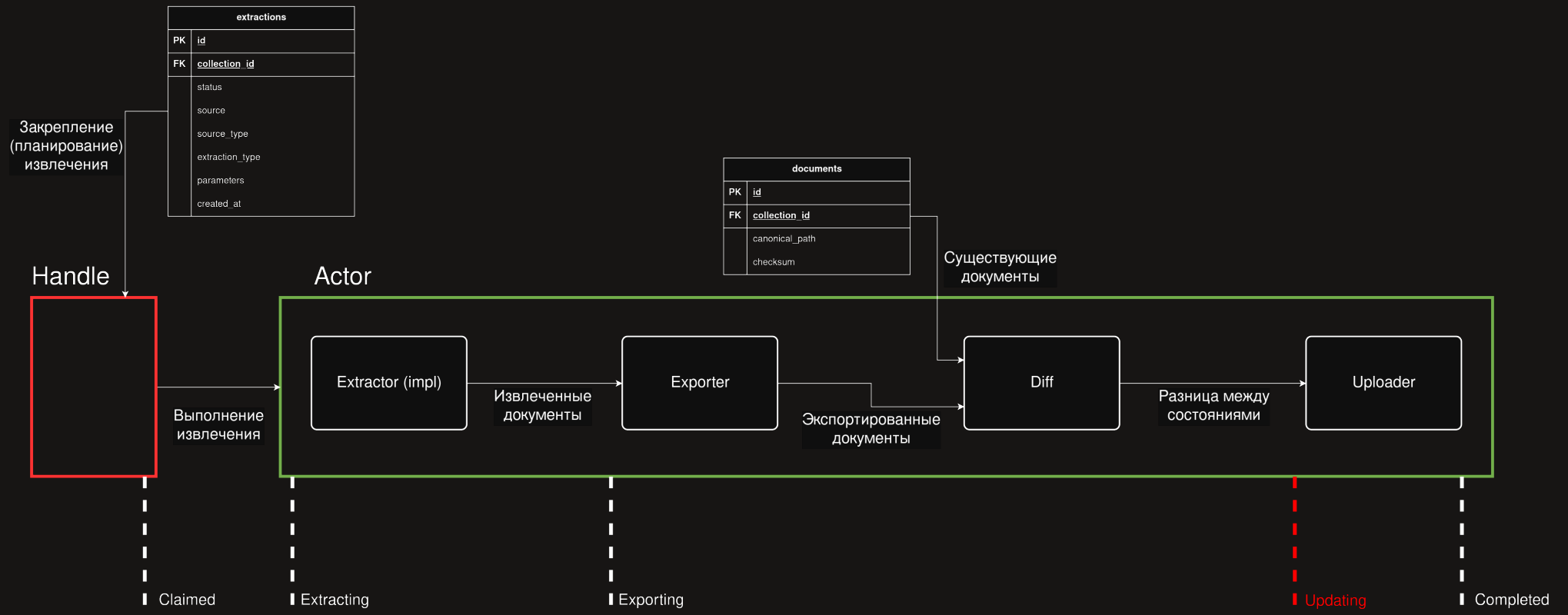
4. Экспортирует документы в Markdown



5. Вычисляет разницу между состояниями коллекции



6. Загружает документы в S3, обновляет данные в БД


extractions
PK id
FK collection_id
status
source
source_type
extraction_type
parameters
created_at
documents
PK id
FK collection_id
canonical_path
checksum
Закрепление (планирование) извлечения
Handle
Actor
Существующие документы
Extractor (impl)
Exporter
Diff
Uploader
Выполнение извлечения
Извлеченные документы
Экспортированные документы
Разница между состояниями
Claimed
Extracting
Exporting
Updating
Completed

Extractor: Отмена задач

Задачи могут быть отменены до этапа 5 (`status = exporting`) включительно

- Отмена со стороны пользователя: `pg_notify`
- Завершение процесса: `SIGINT` / `SIGTERM`

Отмена на этапе 6 (`status = uploading`) невозможна

- `pg_notify` игнорируется
- Остановка после сигнала откладывается, пока извлечение не завершится

Extractor: Идемпотентность

Документ идентифицируется **каноническим путем** и **контрольной суммой**.



Extractor: Идемпотентность

Документ идентифицируется **каноническим путем** и **контрольной суммой**.

Канонические пути разбивают множество документов на:

- Старые
- Общие
- Новые



Extractor: Идемпотентность

Документ идентифицируется **каноническим путем** и **контрольной суммой**.

Контрольные суммы разбивают множество **общих** документов на:

- Обновленные — к.с. различаются
- Без изменений — к.с. совпадают



Extractor: Идемпотентность

Два типа извлечения: слияние (merge) и перезапись (overwrite).

	Слияние	**Перезапись**
Старые	Сохраняются	Удаляются
Без изменений	Сохраняются	
Обновленные	Обновляются	
Новые	Добавляются	

План

1. Изменения в системе

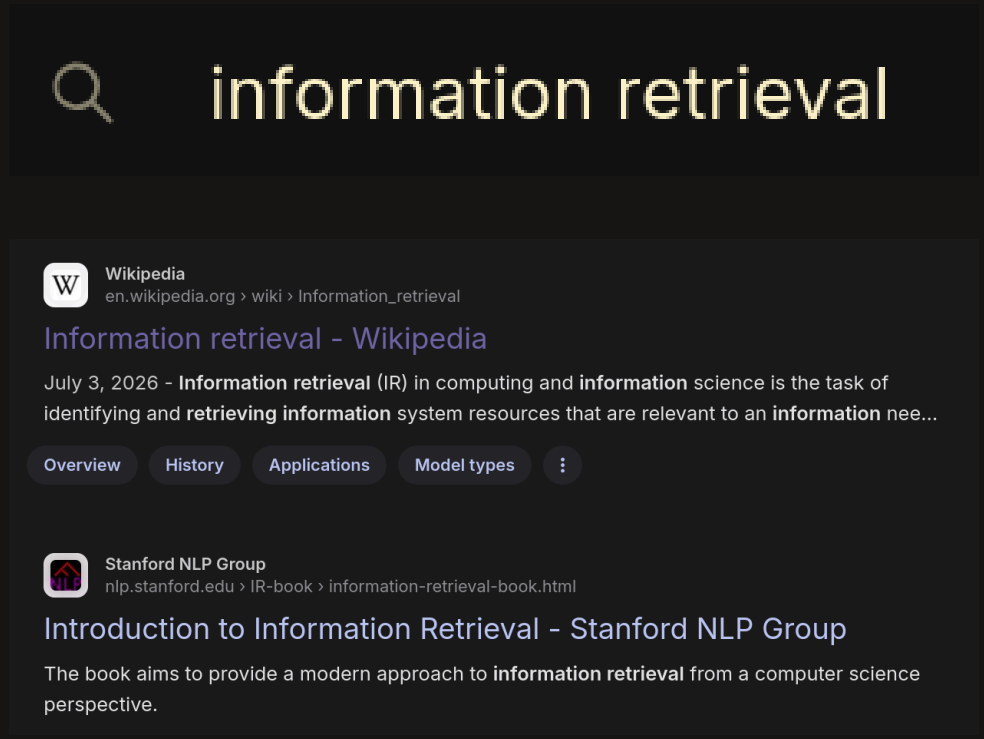
2. Метрики качества поиска

3. Измерение метрик

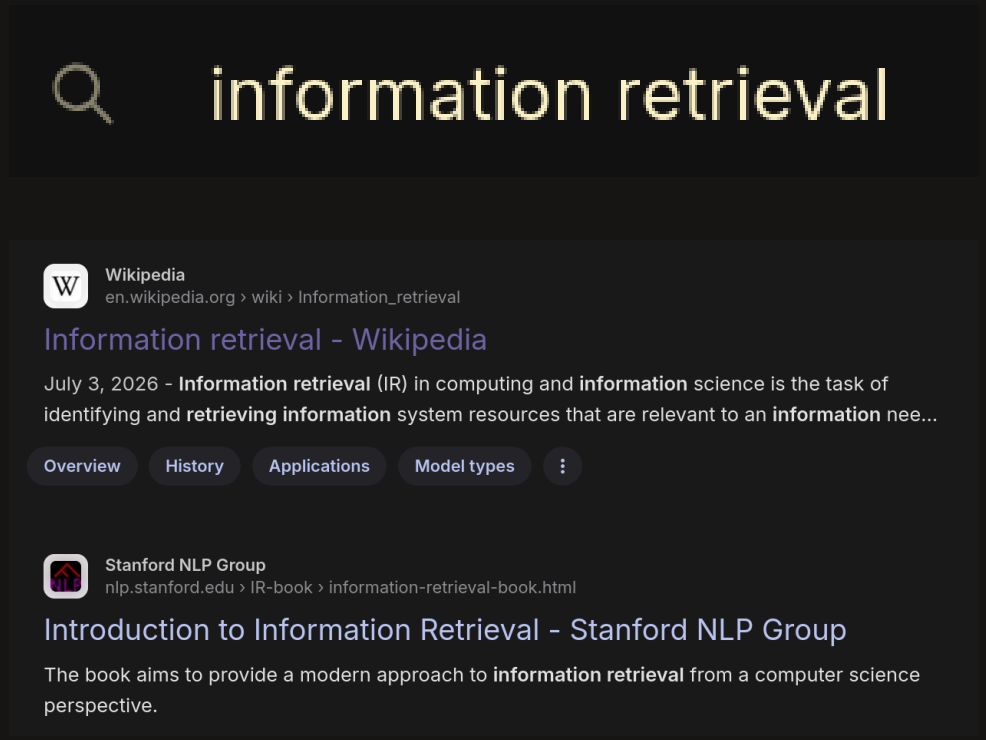
Качество поиска



Качество поиска



Качество поиска



Качество поиска можно определить как общую релевантность результатов по отношению к запросу

О важности качества поиска

Выделяют 3 характеристики:

1. Качество
2. Свежесть
3. Объем



Классификация поисковых систем.
Baeza-Yates, Ribeiro-Neto
“Modern Information Retrieval”

О важности качества поиска

Выделяют 3 характеристики:

1. Качество
2. Свежесть
3. Объем

Carmen — это **вертикальная поисковая система**:

- Одна предметная область
- Коллекции связанных документов



Классификация поисковых систем.
Baeza-Yates, Ribeiro-Neto
“Modern Information Retrieval”

О важности качества поиска

Выделяют 3 характеристики:

1. Качество
2. Свежесть
3. Объем

Carmen — это **вертикальная поисковая система**:

- Одна предметная область
- Коллекции связанных документов

В данном случае качество — наиболее важная характеристика.



Классификация поисковых систем.
Baeza-Yates, Ribeiro-Neto
“Modern Information Retrieval”

Оценка качества поиска

Оценка качества поиска

Оффлайн

- Без реальных пользователей
- Измерение на основе экспертных оценок

Оценка качества поиска

Оффлайн

- Без реальных пользователей
- Измерение на основе экспертных оценок

Онлайн

- Реальные пользователи
- Измерение на основе поведения пользователей (анализ логов)

Оценка качества поиска

Оффлайн

- Без реальных пользователей
- Измерение на основе экспертных оценок

Онлайн

- Реальные пользователи
- Измерение на основе поведения пользователей (анализ логов)

На данном этапе более применимы оффлайн оценки:

- Быстрые измерения при изменении системы
- На стадии разработки нет реальных пользователей

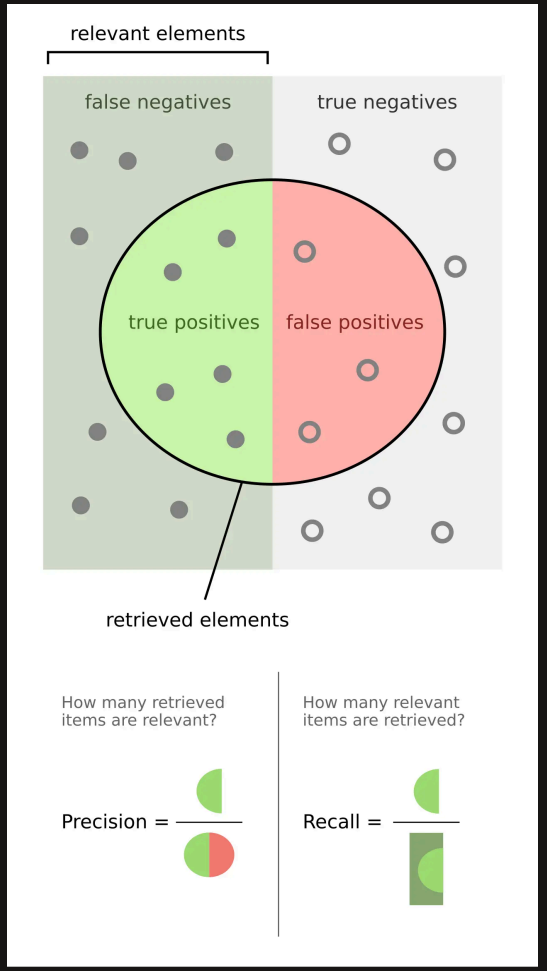
Точность и полнота

Точность — сколько релевантных документов среди возвращенных

$$P = \frac{|\text{возвращенные} \cap \text{релевантные}|}{|\text{возвращенные}|}$$

Полнота — сколько релевантных документов извлечено среди всех релевантных

$$R = \frac{|\text{возвращенные} \cap \text{релевантные}|}{|\text{релевантные}|}$$



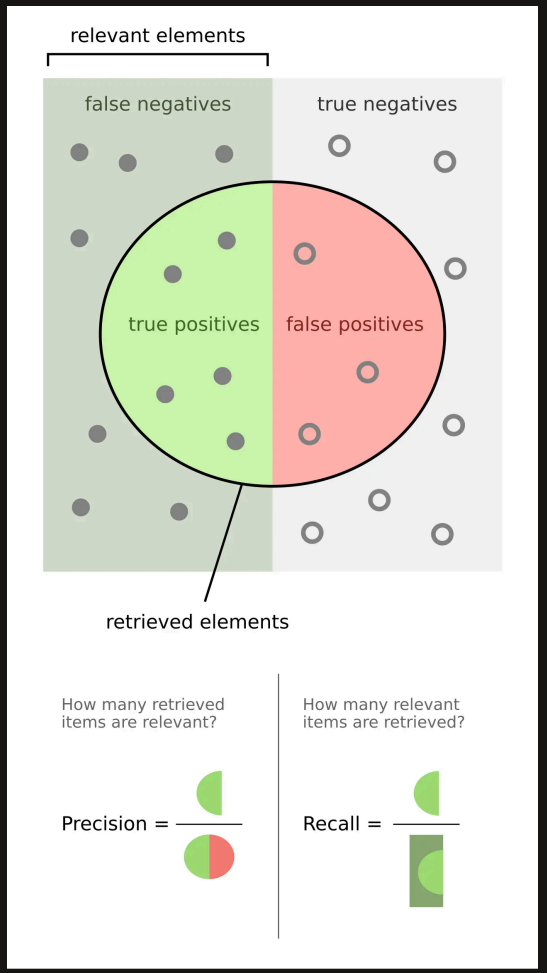
Точность и полнота (Википедия)

Точность и полнота

В терминах теории вероятностей:

$$P = \mathbb{P}(\text{релевантен} | \text{возвращен})$$

$$R = \mathbb{P}(\text{возвращен} | \text{релевантен})$$



Точность и полнота (Википедия)

Точность и полнота

В терминах логических ошибок:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

		Документ релевантный	
		Да	Нет
Документ возвращен	Да	TP	FP
	Нет	FN	TN

F-мера

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

Наиболее распространена мера F_1 ($\alpha = 0.5$) — среднее гармоническое

$$F_1 = \frac{1}{\frac{1}{2} \cdot \frac{1}{P} + \frac{1}{2} \cdot \frac{1}{R}} = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R}$$

Замечание: Релевантность документа

Релевантность документа не обязательно является бинарной шкалой (0/1, true/false).

- Часто используется шкала 0–4
- Алгоритмы поиска возвращают $r \in [0, 1]$

Перевод в бинарную шкалу с помощью квантилей:

$$r_{\{0,1\}} = \begin{cases} 1, & r \geq \text{threshold} \\ 0, & \text{иначе} \end{cases}$$

Точность / полнота при ранге k

Те же метрики, но для первых k результатов

$$P@k = \frac{|\text{релевантные на ранге k}|}{k}$$

$$R@k = \frac{|\text{релевантные на ранге k}|}{|\text{релевантные}|}$$

Обратный ранг

$$\mathrm{RR} = \frac{1}{\text{Ранг 1-го релевантного документа}}$$

Средняя точность

$$\mathrm{AP} = \frac{\sum_{k \in H} P@k}{|\text{релевантные}|}$$

- H — ранги релевантных документов

“Средняя точность при пройденных релевантных документах”

Discounted cumulative gain (DCG)

Cumulative gain:

$$\mathrm{CG} = \sum_{k=1}^{N} (2^{R_k} - 1)$$

- N — кол-во результатов
- R_k — шкала релевантности

Discount:

$$D(k) = \frac{1}{\log(1+k)}$$

Discounted cumulative gain (DCG)

$$\mathrm{DCG} = \sum_{k=1}^{N} \frac{2^{R_k} - 1}{\log(k+1)}$$

Релевантные документы, находящиеся ниже в списке результатов, штрафуются. Результаты, находящиеся выше в списке, считаются более полезными.

Normalized DCG

Проблема: значение DCG не нормализовано.

$\mathrm{DCG}_{\mathrm{ideal}}$ — DCG от топ N документов по релевантности.

- Иными словами, DCG наилучшего набора результатов, который могла вернуть поисковая система

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{DCG}_{\mathrm{ideal}}}$$

Резюме

- Не учитывают ранги: точность, полнота, F-мера
- Учитывают ранги: точность / полнота при ранге k, обратный ранг, средняя точность
- Учитывает поведение пользователей (как модель): DCG

Существует множество других метрик.

План

1. Изменения в системе

2. Метрики качества поиска

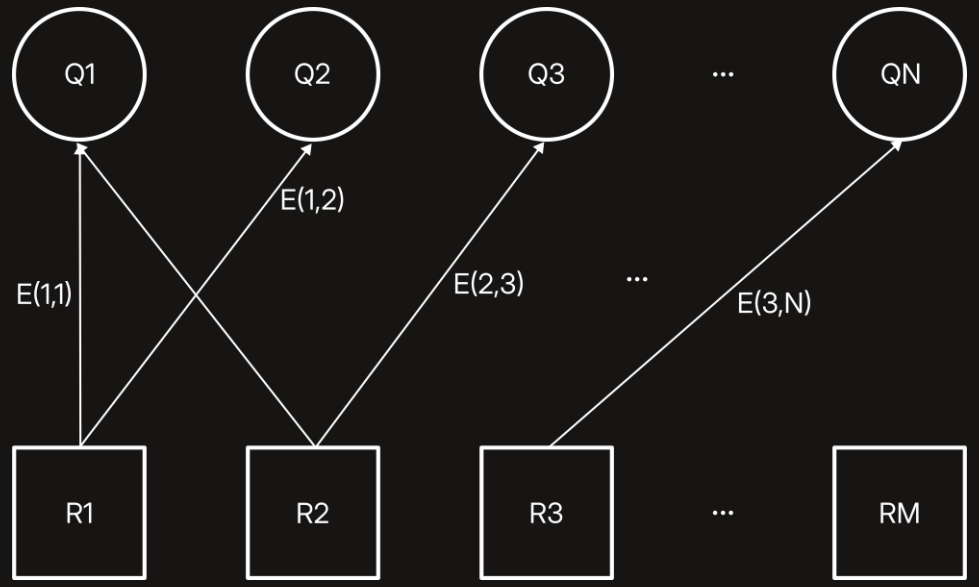
3. Измерение метрик

Тестовые коллекции

Состоят из запросов, результатов поиска и оценок релевантности.

Можно представить в виде двудольного графа:

- R_i — результат
- Q_j — запрос
- $E(i, j)$ — релевантность i-ого результата j-ому запросу
  - Можно использовать любую шкалу релевантности



Методология измерений

1. Составить тестовую коллекцию
2. Измерить метрики (P, R, F-1, RR, AP, DCG, ...)
   - Выполнить поисковый запрос с Q_j
   - Запросить релевантность полученных результатов: $E(i, j), i \in \mathrm{Res}$
   - Вычислить нужные метрики по формулам

Недостатки

- $M \cdot N$ оценок, которые необходимо добавить вручную
  - Можно уменьшить, если считать отсутствие связи за 0
- Система извлекает документы, а ищет по чанкам
  - Декомпозиция поиска в отдельную систему
  - Синхронизация результатов поиска

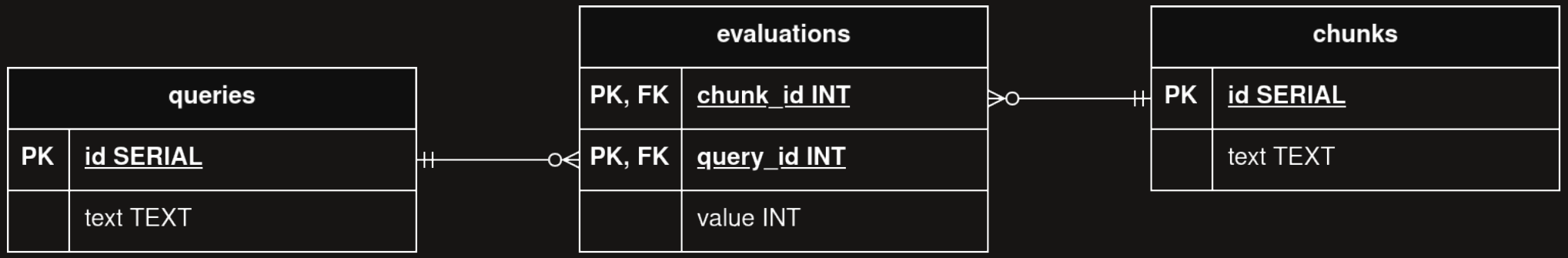
Реализация

`wing` — CLI для измерения оффлайн метрик качества поиска.

- CRUD для запросов, результатов, оценок
- Синхронизация с Carmen
  - Импорт существующей коллекции в тестовую
  - Создание коллекции из тестовой
- Вычисление метрик

Реализация

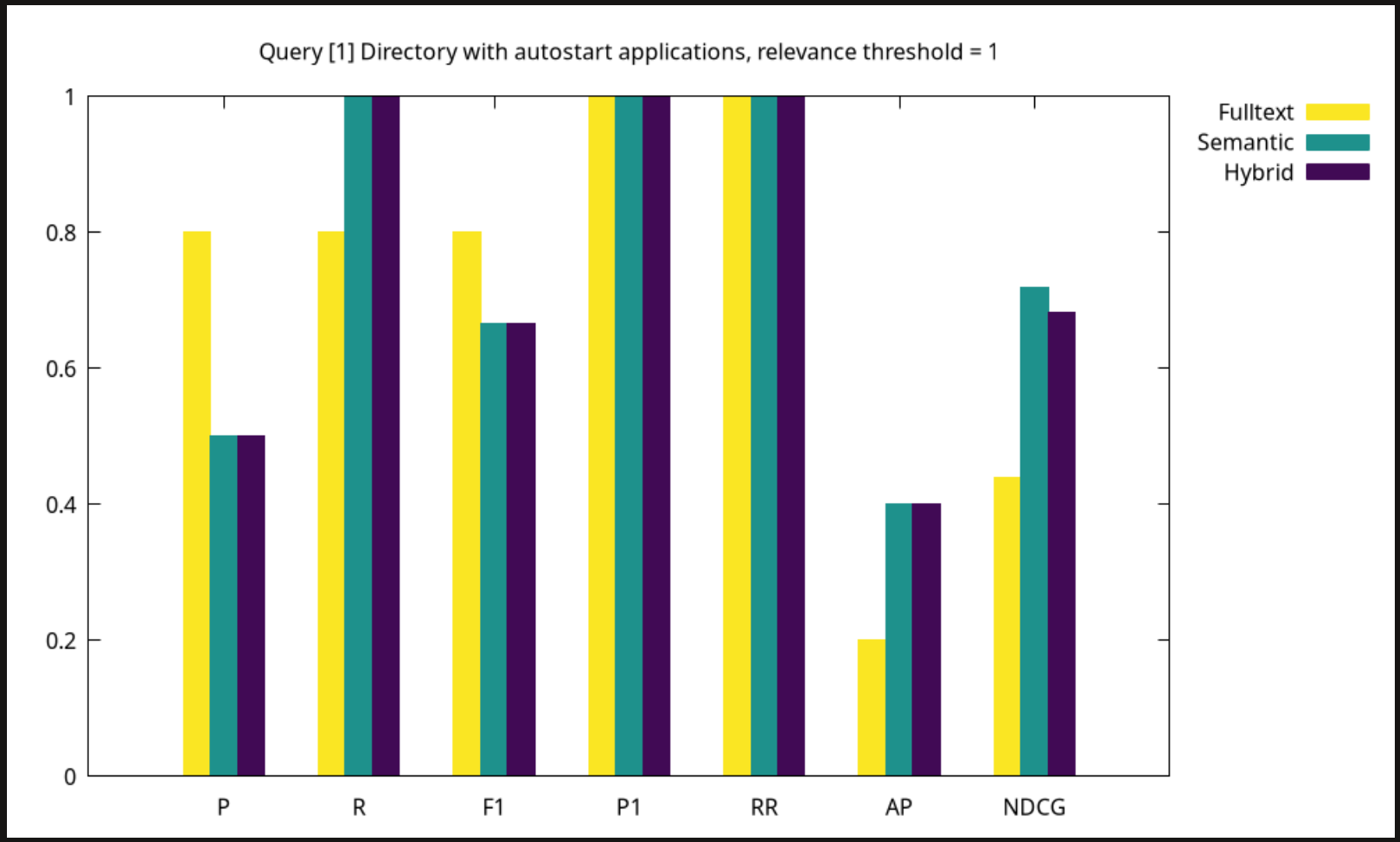
Представление тестовой коллекции

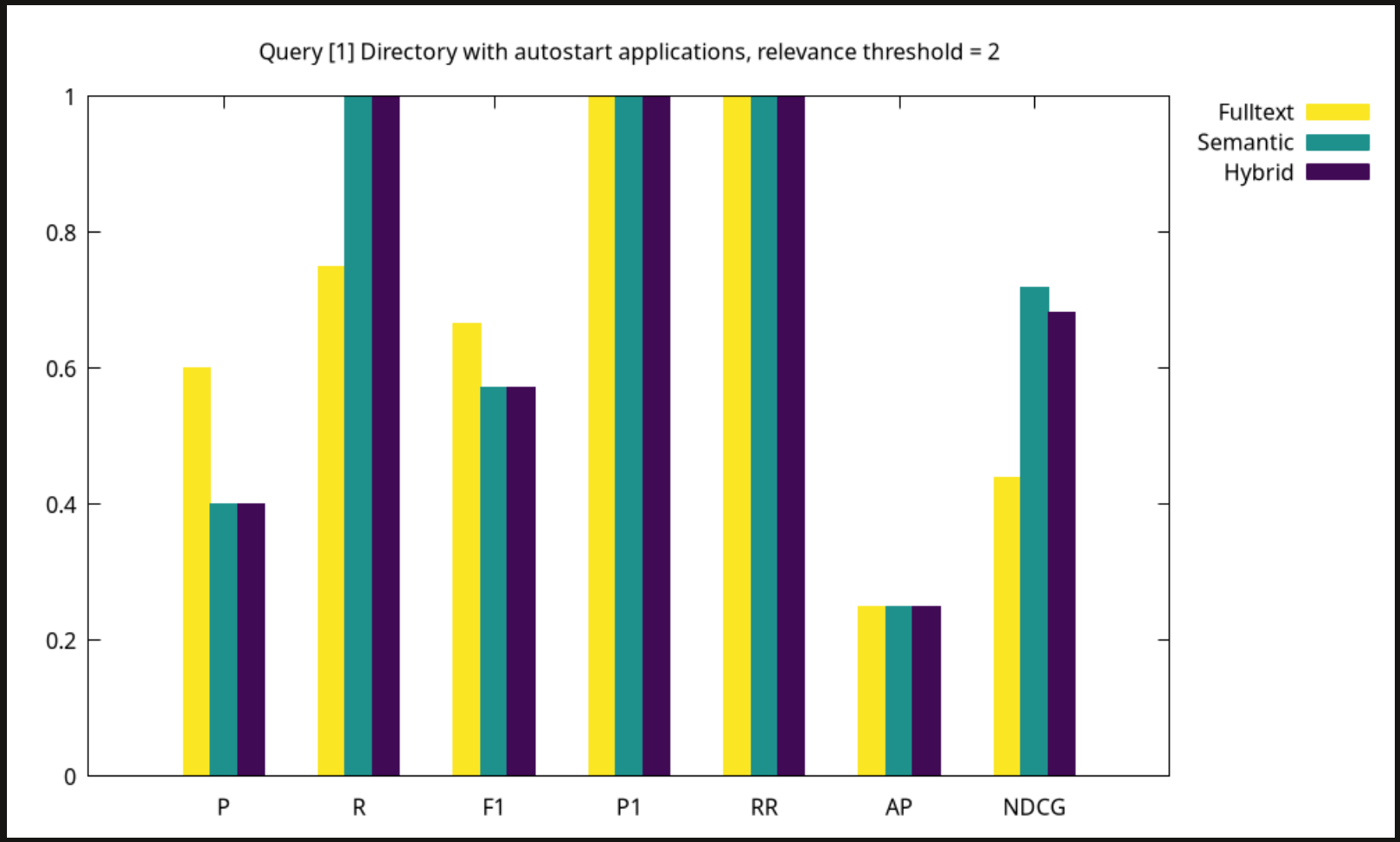


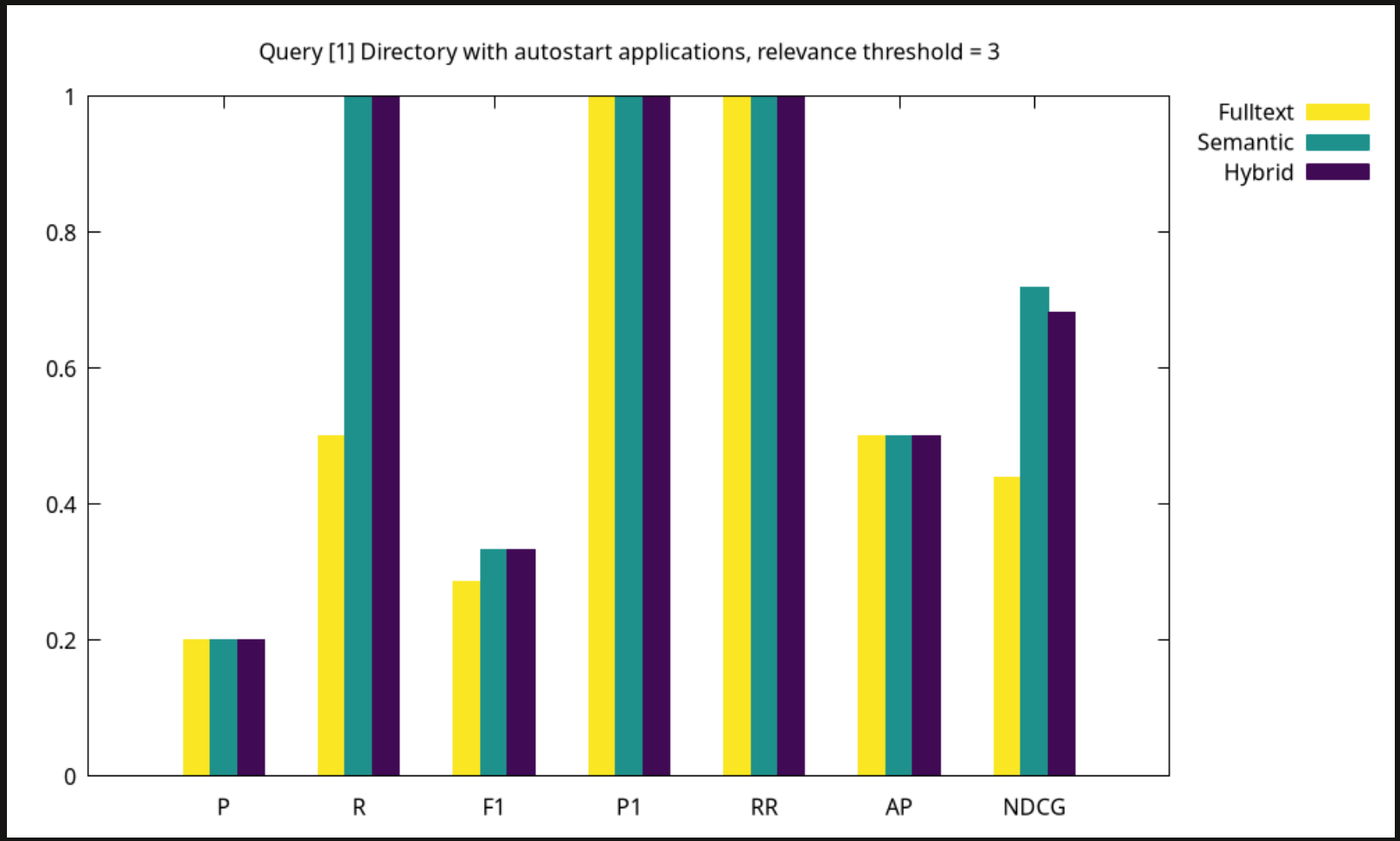
Тестовая коллекция

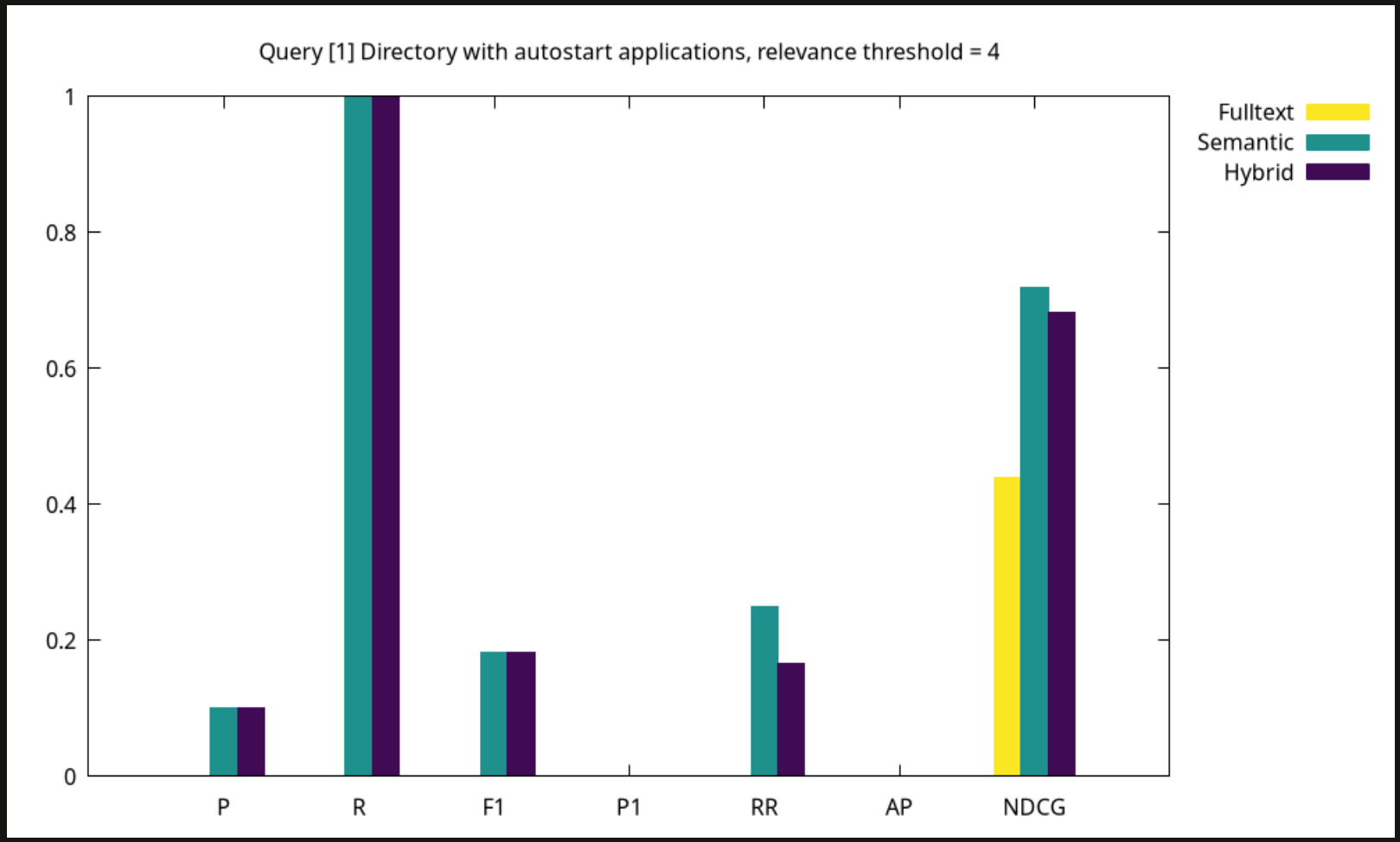
В качестве тестовой коллекции рассматривался набор спецификаций Freedesktop (https://www.freedesktop.org/wiki/Specifications/)

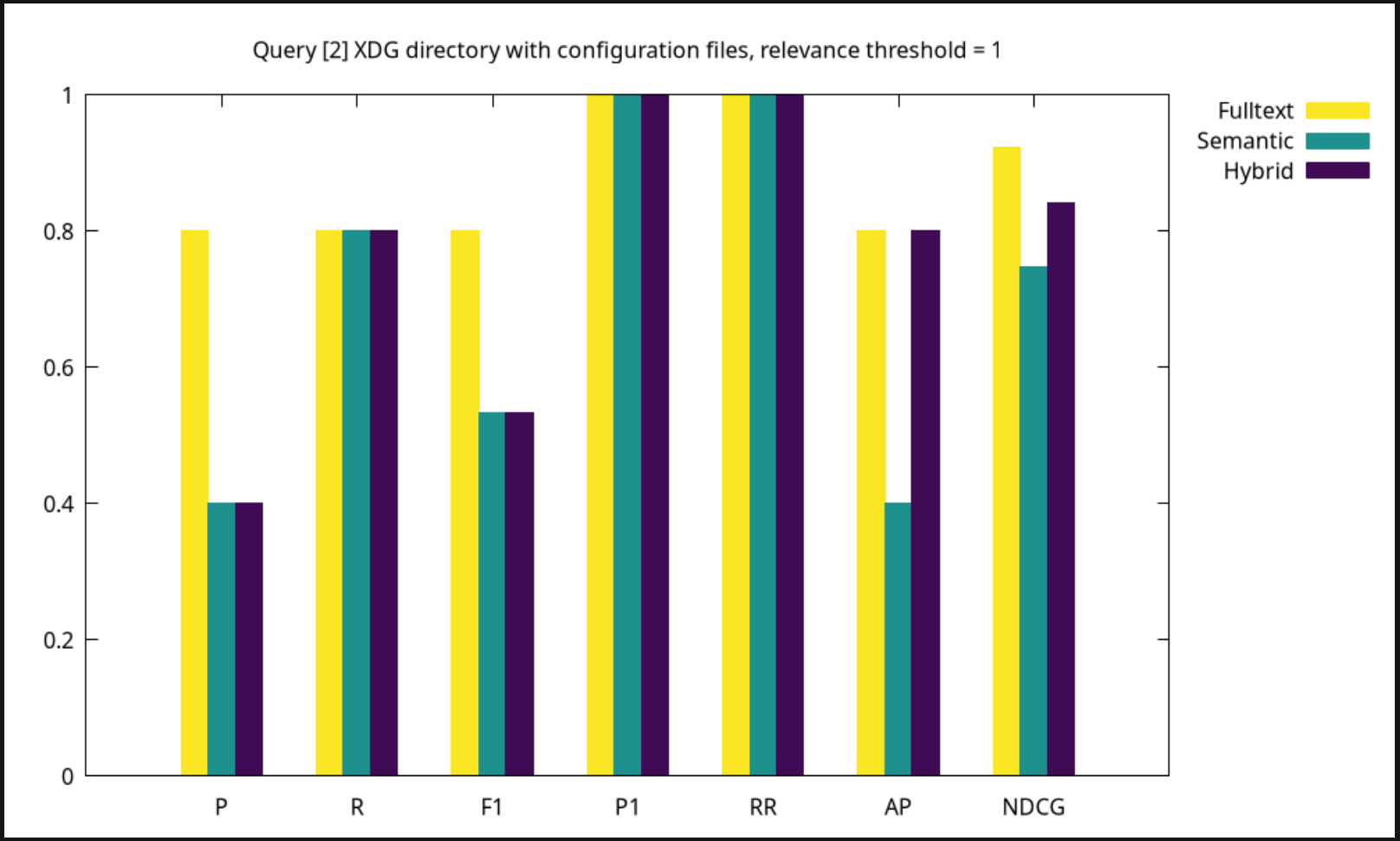
- Autostart
- StatusNotifierItem
- File URI

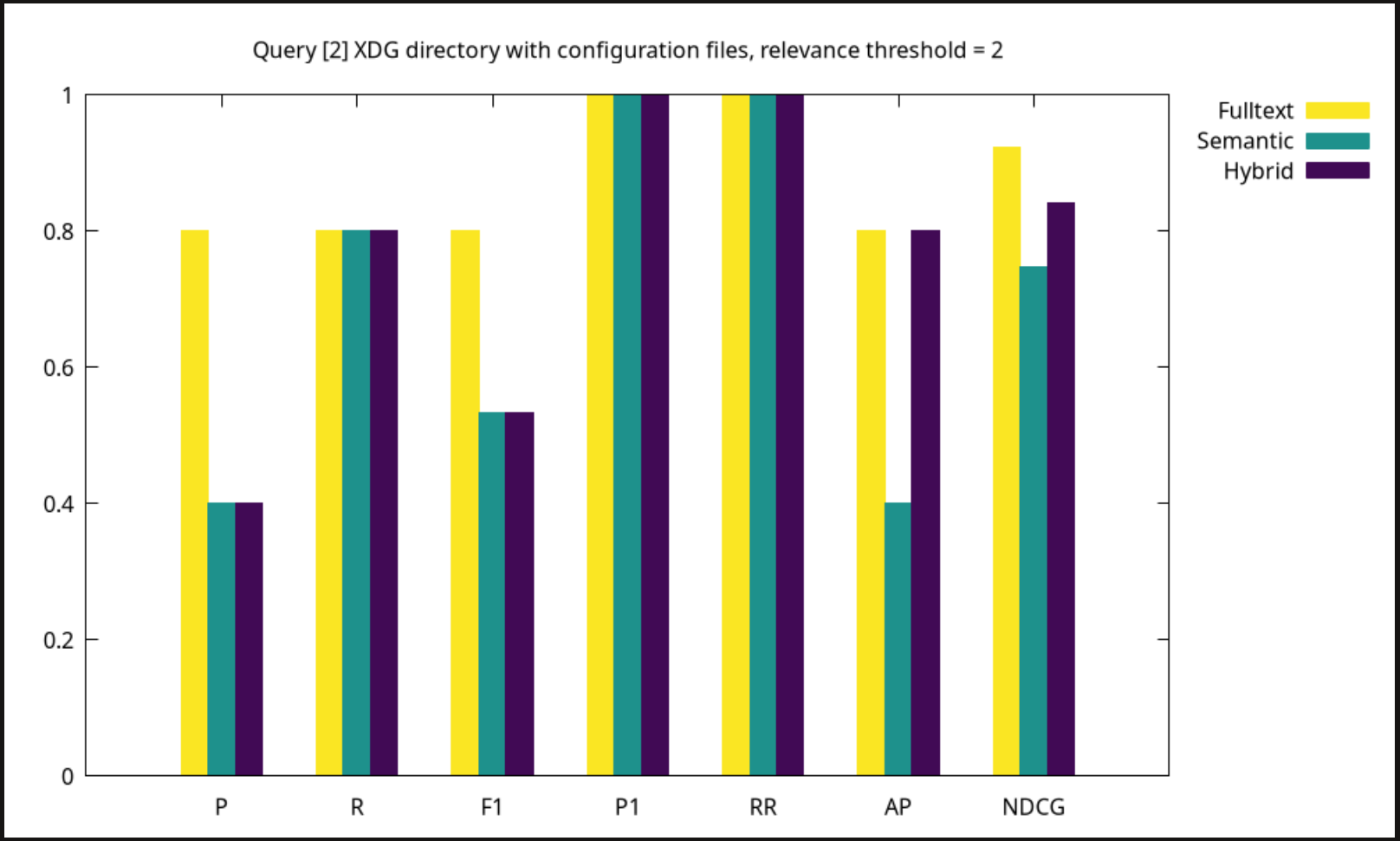

Query [1] Directory with autostart applications, relevance threshold = 1
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

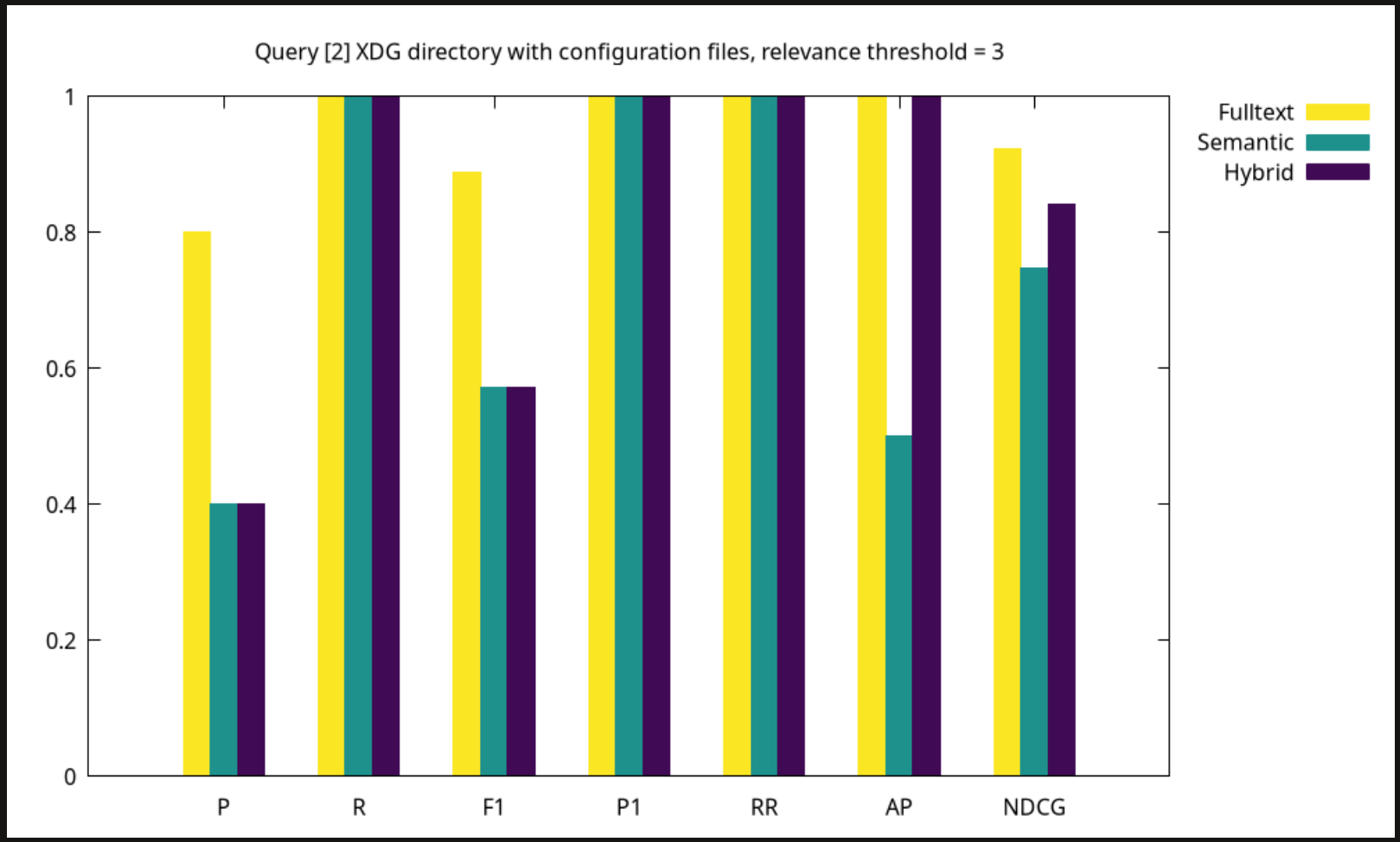

Query [1] Directory with autostart applications, relevance threshold = 2
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

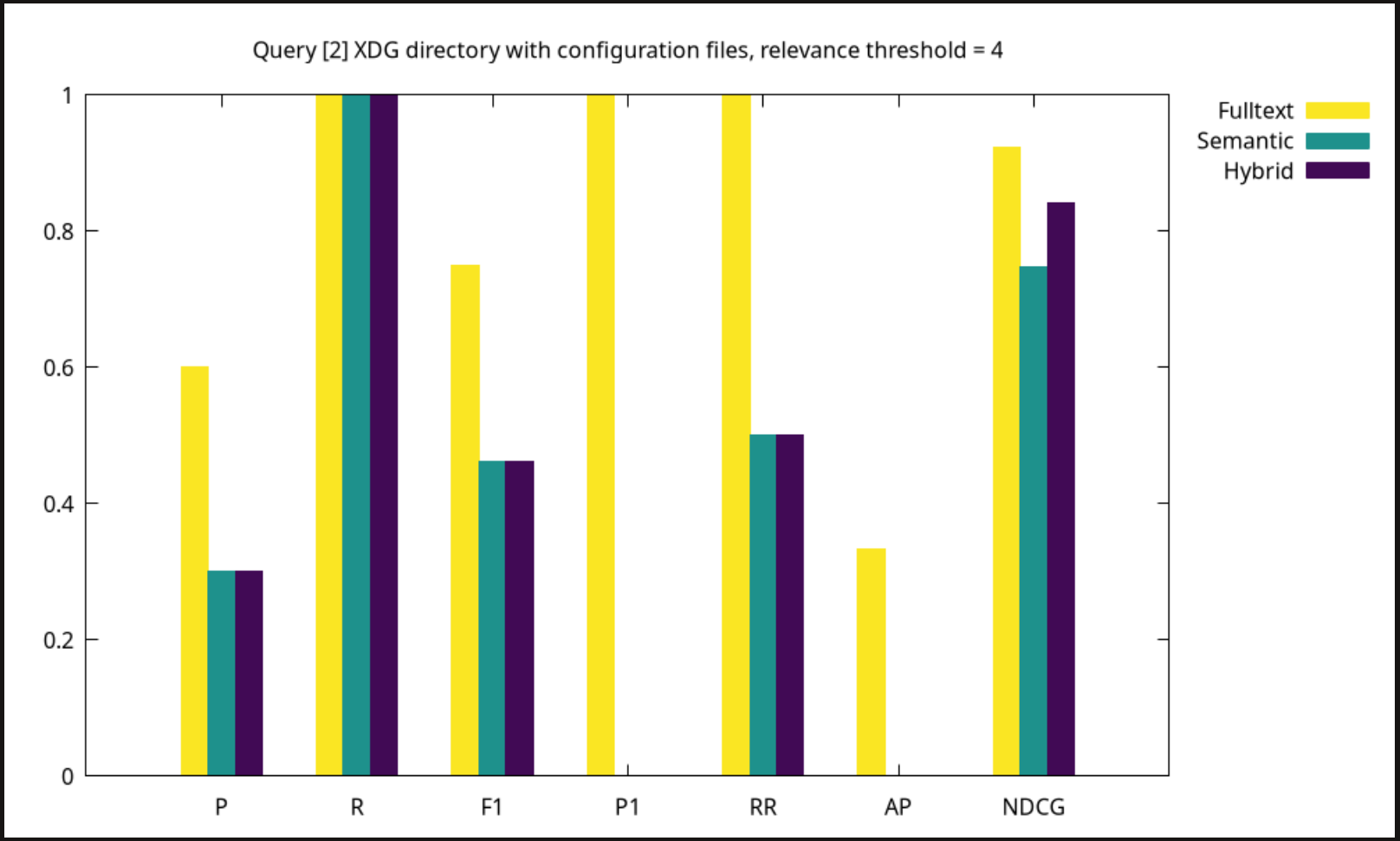

Query [1] Directory with autostart applications, relevance threshold = 3
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

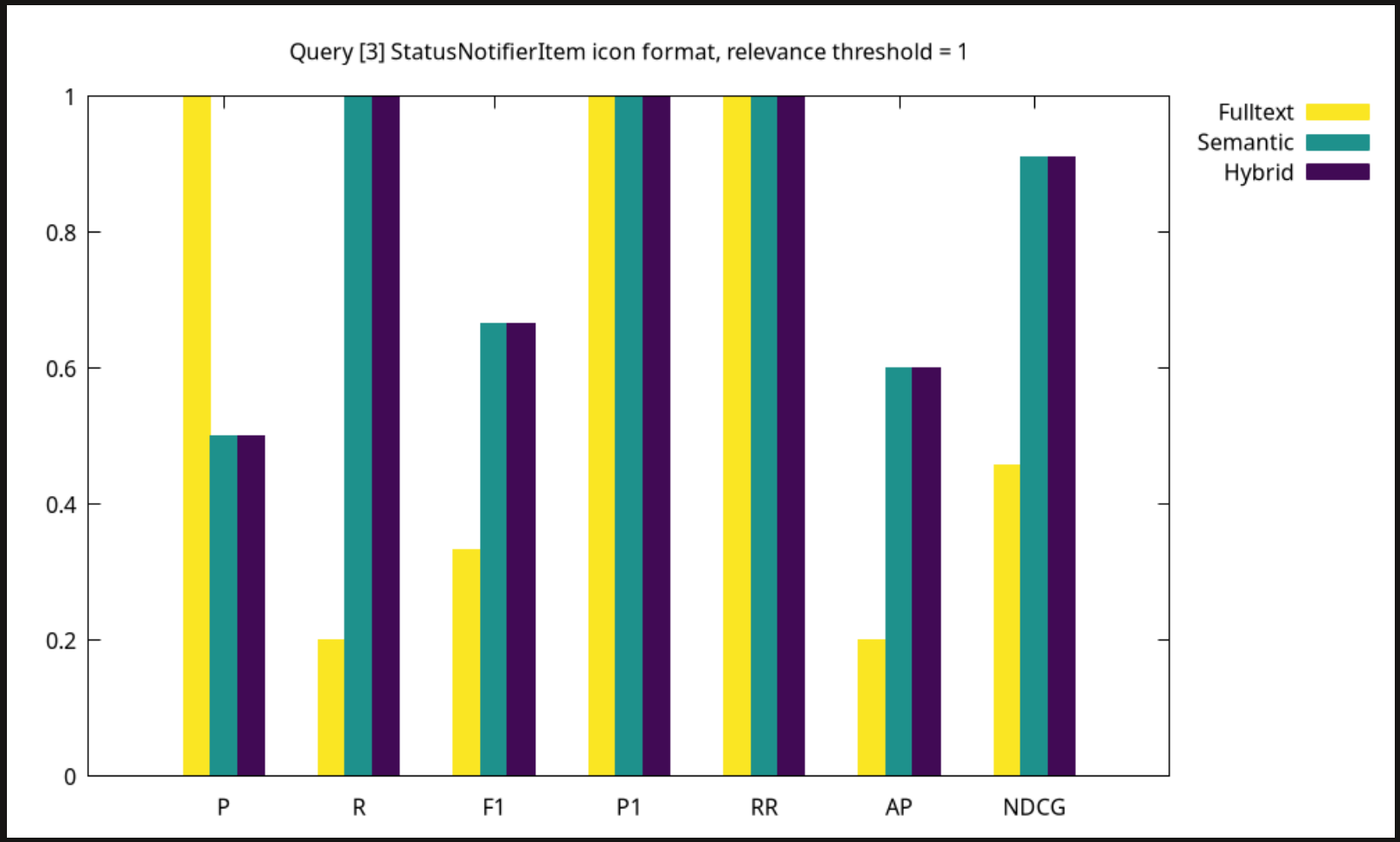

Query [1] Directory with autostart applications, relevance threshold = 4
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

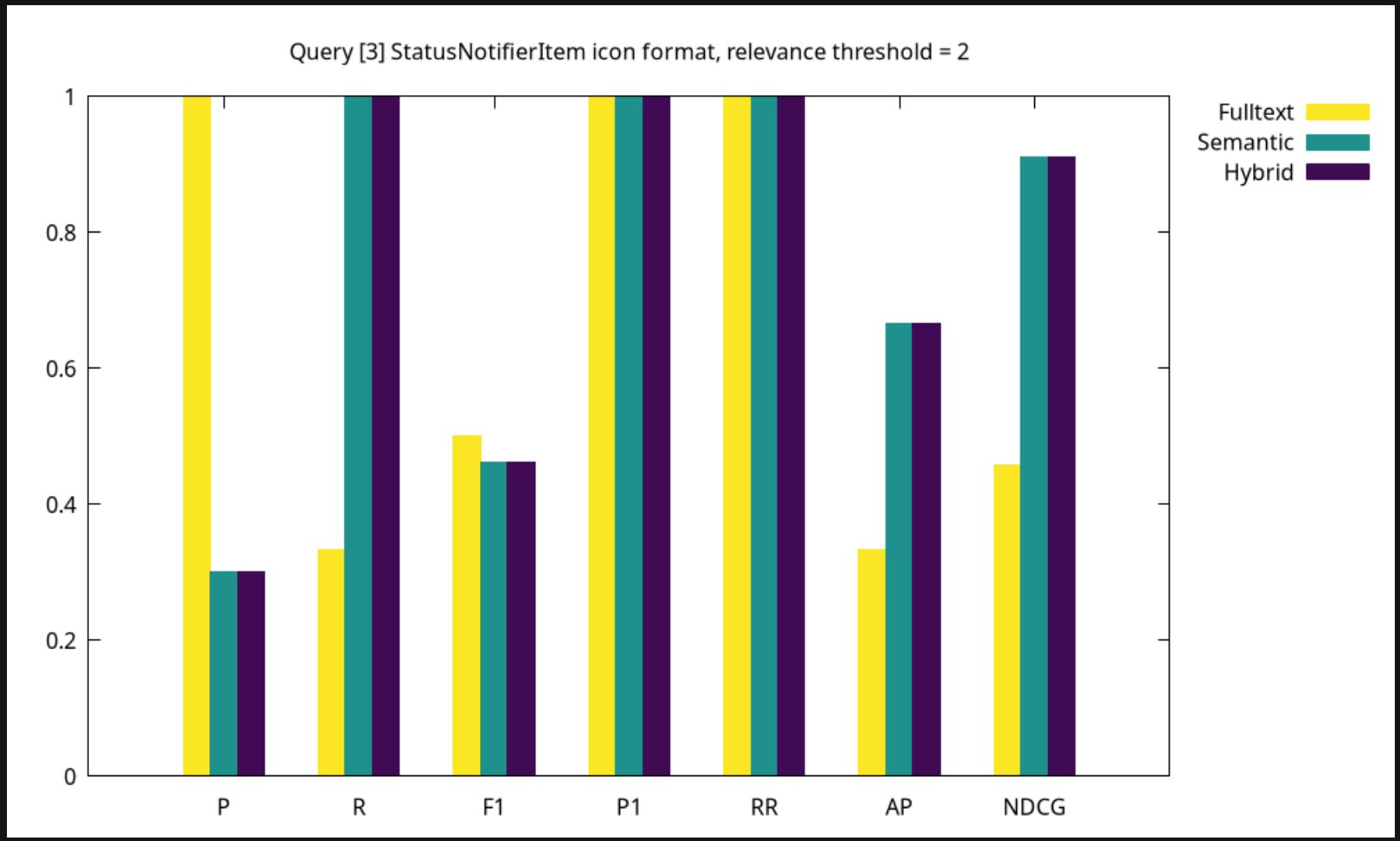

Query [2] XDG directory with configuration files, relevance threshold = 1
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

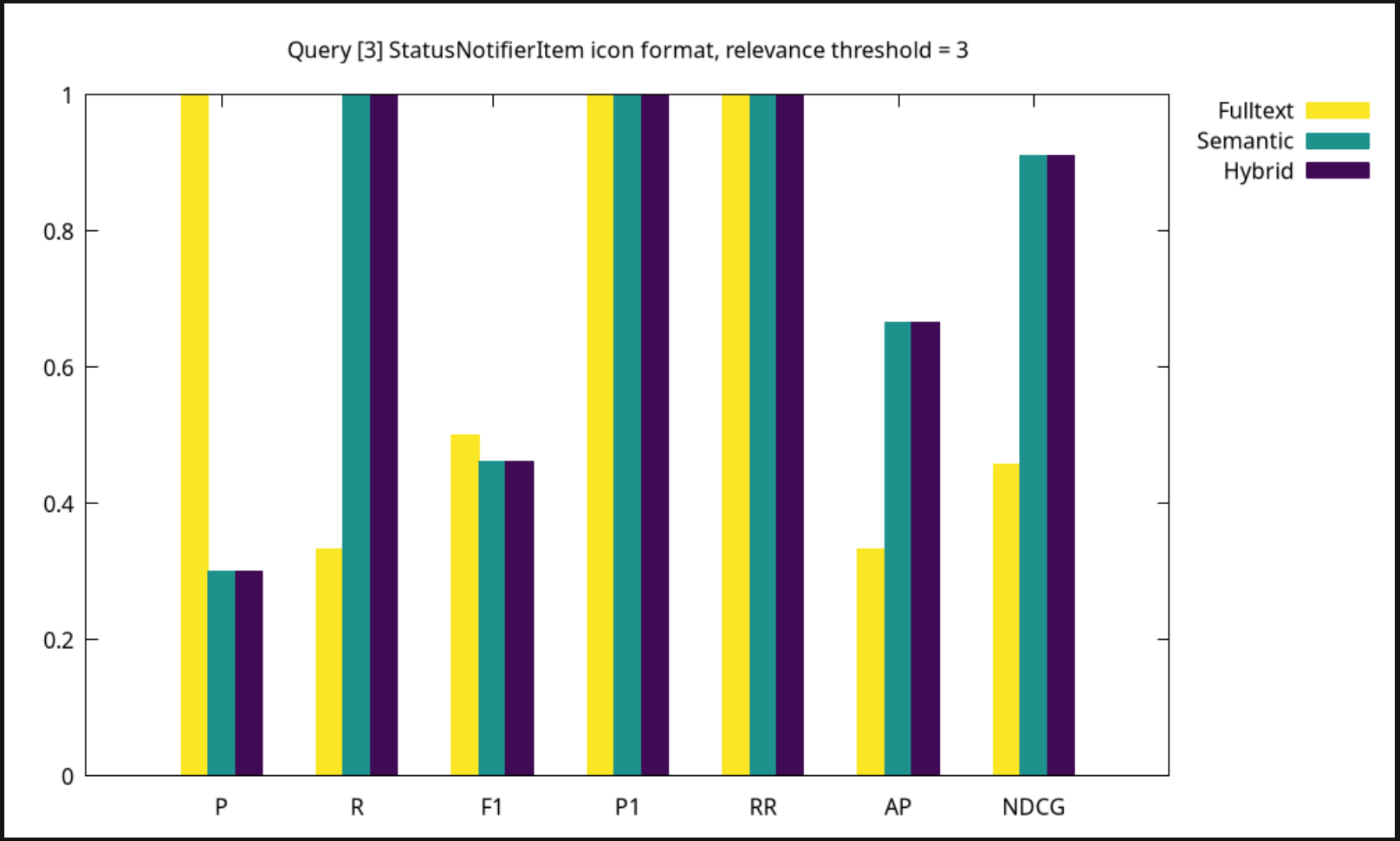

Query [2] XDG directory with configuration files, relevance threshold = 2
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

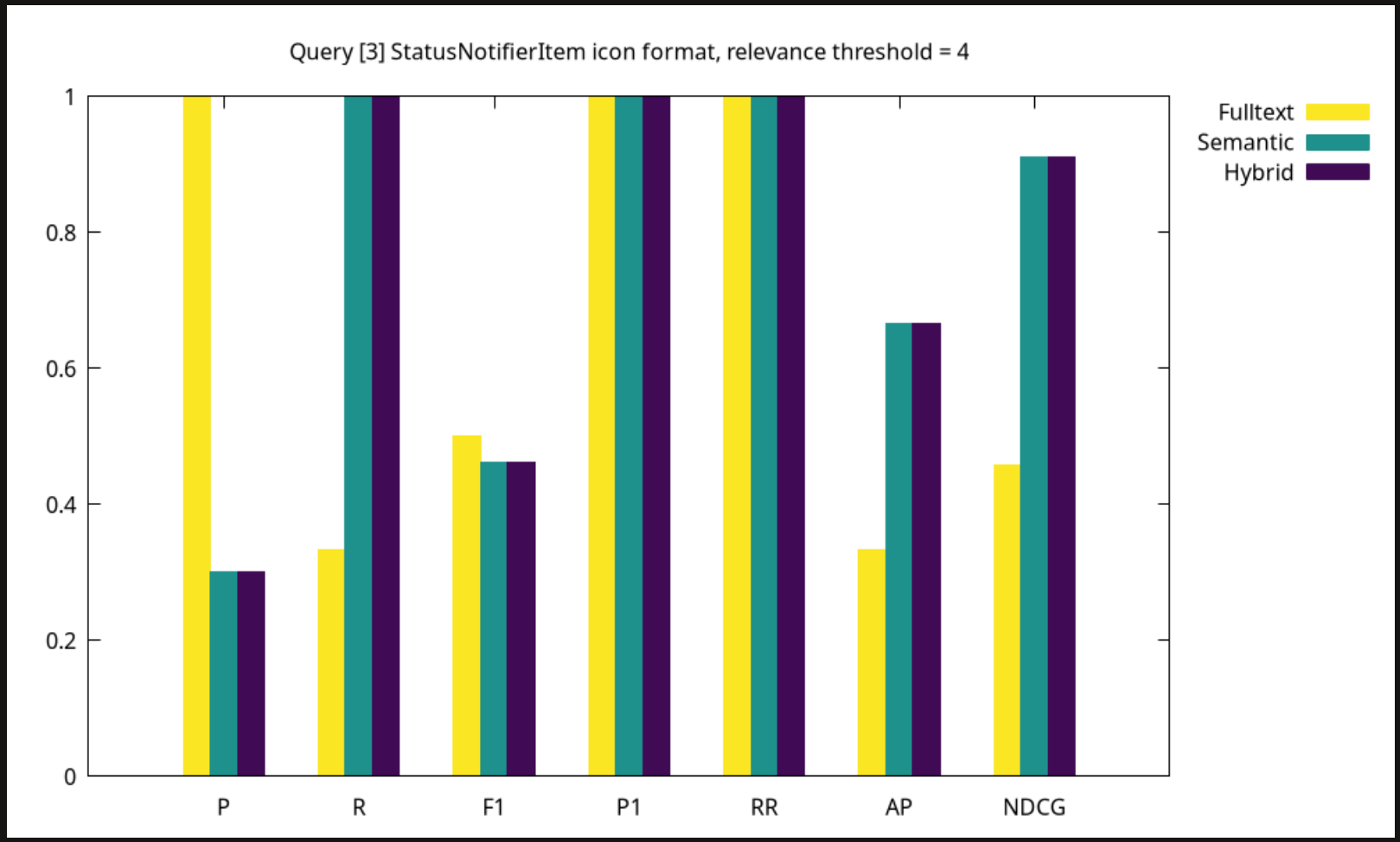

Query [2] XDG directory with configuration files, relevance threshold = 3
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid


Query [2] XDG directory with configuration files, relevance threshold = 4
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid


Query [3] StatusNotifierItem icon format, relevance threshold = 1
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid


Query [3] StatusNotifierItem icon format, relevance threshold = 2
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid


Query [3] StatusNotifierItem icon format, relevance threshold = 3
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid


Query [3] StatusNotifierItem icon format, relevance threshold = 4
1
0.8
0.6
0.4
0.2
0
P
R
F1
P1
RR
AP
NDCG
Fulltext
Semantic
Hybrid

Интерпретация результатов

- Полнотекстовый поиск выигрывает в точности (10 / 12) и F1 (9 / 12), но проигрывает в полноте (8 / 12)
- Полнотекстовый поиск менее устойчив с точки зрения подбора слов (запросы 1 и 2)
- Показатели семантического и гибридного поиска схожи:
  - При прочих равных может выдавать более релевантные документы раньше (запрос 1). Это позволяет использовать его для финального переранжирования (реранкер), что увеличивает DCG.

Исходный код





- https://git.shelepugin.ru/carmen/carmen/src/tag/2026-10-09-etu-nlp
- https://git.shelepugin.ru/carmen/wing/src/tag/2026-10-09-etu-nlp