# Проблема полноты поиска в B2B-каталогах DIY-товаров: ограничения семантических эмбеддингов и сущностно-ориентированный подход

Краснов Федор

24 февраля 2026 г.

**Аннотация**

В статье рассматривается задача обеспечения высокой полноты поиска в B2B-каталогах DIY-товаров (инструменты и материалы для строительства и ремонта). На практике B2B-поиск предъявляет существенно более строгие требования к полноте, чем B2C-рекомендательные сценарии. Анализируется эволюция архитектуры поиска: от чистого dense retrieval на базе трансформерных эмбеддингов (ModernBERT, triplet loss) к гибридной системе, основанной на детекции сущностей (бренд, модель, технические характеристики) и структурированном сопоставлении (entity matching).

Показано, что embedding-based retrieval не гарантирует полноту из-за сглаживания числовых токенов, семантической компрессии идентификаторов и особенностей ANN-поиска. Предложенная архитектура anchor-based retrieval обеспечивает значительный рост Recall@10 (с 0.65 до 0.97 на реальных данных) при сохранении приемлемой латентности и высокой объяснимости.

Экспериментальная часть выполнена на промышленном каталоге более 10 млн позиций и выборке из 5 тыс. реальных B2B-запросов. Работа ориентирована на практическое применение и может быть использована при проектировании поисковых систем в сегменте B2B e-commerce.

Ключевые слова: B2B-поиск, DIY-товары, полнота поиска, recall, dense retrieval, entity matching, гибридные архитектуры.

## Введение

В B2B-сценариях поиск товаров является частью операционного процесса: закупки, выставление счетов, тендерные процедуры, формирование спецификаций. В отличие от B2C, где допустима семантическая приблизительность (“похожие товары”) [1, 2, 3, 4, 5], в B2B-контексте критична точность идентификации [6, 7, 8]. Ошибка в выдаче приводит не только к снижению пользовательского удовлетворения, но и к прямым операционным издержкам.

Формально, пусть $Q$ — множество пользовательских запросов, $D$ — каталог товаров, $R(q) \subset D$ — множество релевантных документов для запроса $q$. Полнота поиска определяется как [9]

$$Recall@K(q) = \frac{|TopK(q) \cap R(q)|}{|R(q)|}.$$

В B2B-контексте множество релевантных документов для значительной доли запросов вырождается [10]:

$$|R(q)| = 1.$$

Обозначим единственный релевантный документ как $d^*$. Тогда формула полноты упрощается:

$$Recall@K(q) = \begin{cases} 1, & \text{если } d^* \in TopK(q), \\ 0, & \text{иначе.} \end{cases}$$

Иными словами, при $|R(q)| = 1$ метрика Recall@K становится эквивалентной индикаторной функции:

$$Recall@K(q) = \mathbf{1}\{d^* \in TopK(q)\}.$$

Таким образом, полнота в B2B-задаче приобретает бинарный характер: система либо извлекла корректную сущность, либо произошла потеря релевантного объекта. В отличие от B2C-сценариев, где допускается частичная релевантность и множественность релевантных результатов, здесь отклонение даже на одну позицию означает фактическую ошибку поиска.

Рассмотрим вероятностную интерпретацию [11, 12, 13]. Пусть случайная величина $X_q = \mathbf{1}\{d^* \in TopK(q)\}$ описывает успешность извлечения для запроса $q$. Тогда математическое ожидание

$$\mathbb{E}[X_q] = \mathbb{P}(d^* \in TopK(q))$$

интерпретируется как вероятность успешного извлечения корректной товарной позиции. Средний Recall@K по выборке запросов является эмпирической оценкой этой вероятности:

$$Recall@K = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}\{d^* \in TopK(q)\}.$$

Следовательно, задача повышения полноты в B2B-каталогах сводится к максимизации вероятности точного извлечения конкретной сущности. Даже незначительное снижение данной вероятности приводит к прямым бизнес-рискам.

Подобная постановка задачи накладывает существенные ограничения на применимость методов, основанных исключительно на семантической близости. В dense retrieval релевантность моделируется через непрерывную функцию сходства в векторном пространстве:

$$Score(q, d) = \cos(\mathbf{e}_q, \mathbf{e}_d),$$

где $\mathbf{e}_q, \mathbf{e}_d$ — эмбеддинги запроса и документа. Оптимизация такой функции направлена на минимизацию среднего расстояния между семантически близкими объектами, однако не гарантирует сохранение дискретных идентификаторов (модель, артикул, точное числовое значение).

В условиях, когда целевая метрика эквивалентна индикаторной функции, даже малое искажение расстояний в embedding-пространстве может привести к выпадению $d^*$ из множества $TopK(q)$. Таким образом, непрерывная природа dense retrieval вступает в противоречие с дискретной структурой целевой задачи.

Особенно ярко данное противоречие проявляется в сегменте DIY-товаров (инструменты и материалы для строительства и ремонта). DIY-каталоги характеризуются следующими особенностями:

- Высокой долей числовых характеристик (мощность, размеры, напряжение, обороты).
- Наличием уникальных идентификаторов (модель, артикул, модификация).
- Транслитерацией и смешением алфавитов (латиница/кириллица).

- Выраженным long tail — значительным количеством редких SKU.

Запросы часто имеют структурированный характер и одновременно содержат тип товара, модель и технические характеристики, например:

"Перфоратор HR 3200 C 850 Вт SDS+"

В подобных случаях задача поиска сводится не к определению семантической близости, а к точному сопоставлению набора сущностей и параметров. Это обстоятельство служит основанием для выдвижения гипотезы о принципиальных ограничениях чистого dense retrieval в задачах обеспечения полноты B2B-поиска и мотивирует переход к сущностно-ориентированным архитектурам.

## Конфликт непрерывной функции сходства и дискретной целевой функции

Целевая функция в B2B-поиске при $|R(q)| = 1$ имеет дискретный характер и может быть записана как

$$L_{\text{task}}(q) = 1 - \mathbf{1}\{d^* \in TopK(q)\}.$$

Данная функция является разрывной по отношению к параметрам модели: сколь угодно малое изменение скоринговой функции $Score(q, d)$ может изменить порядок документов и привести к скачкообразному изменению $L_{\text{task}}$.

В то же время dense retrieval обучается посредством непрерывной surrogate-функции потерь, например triplet loss [14]:

$$L_{\text{triplet}} = \max\big(0, d(\mathbf{e}_q, \mathbf{e}_{d^+}) - d(\mathbf{e}_q, \mathbf{e}_{d^-}) + m\big),$$

где $d(\cdot, \cdot)$ — непрерывная метрика в embedding-пространстве [15]. Оптимизация $L_{\text{triplet}}$ минимизирует среднее расстояние между позитивными и негативными парами, однако не минимизирует напрямую $L_{\text{task}}$.

Таким образом возникает *loss-mismatch*:

$$\min L_{\text{triplet}} \not\Rightarrow \min L_{\text{task}}.$$

Даже если выполняется

$$d(\mathbf{e}_q, \mathbf{e}_{d^*}) < d(\mathbf{e}_q, \mathbf{e}_d),$$

для большинства $d$, это не гарантирует, что $d^*$ войдёт в $TopK(q)$ при наличии большого числа близких по расстоянию кандидатов. В условиях высокой плотности embedding-пространства малые флуктуации расстояний приводят к выпадению релевантного документа из топа, что критично при бинарной природе целевой функции.

## Влияние Approximate Nearest Neighbors

На практике поиск осуществляется не по точной метрике, а с использованием Approximate Nearest Neighbors (ANN) [16, 17]. Пусть $\mathcal{N}_K(q)$ — множество истинных $K$ ближайших соседей, а $\widehat{\mathcal{N}}_K(q)$ — результат ANN-поиска. Тогда существует вероятность ошибки:

$$\mathbb{P}(\widehat{\mathcal{N}}_K(q) \neq \mathcal{N}_K(q)) > 0.$$

Обозначим $p_{ann} = \mathbb{P}(d^* \in \widehat{\mathcal{N}}_K(q) \mid d^* \in \mathcal{N}_K(q))$ — вероятность корректного возврата истинного соседа алгоритмом ANN.

Тогда итоговая вероятность извлечения релевантного документа ограничена сверху:

$$\mathbb{P}(d^* \in \widehat{\mathcal{N}}_K(q)) \leq \mathbb{P}(d^* \in \mathcal{N}_K(q)) \cdot p_{ann}.$$

Следовательно,

$$Recall^{ANN}@K \leq Recall^{Exact}@K \cdot p_{ann}.$$

Даже при идеальной embedding-модели ($Recall^{Exact}@K \approx 1$) ограничение $p_{ann} < 1$ индуцирует верхнюю границу на достижимую полноту. В задачах с $|R(q)| = 1$ это означает фундаментальное ограничение вероятности успеха.

## Необходимость структурированной постановки

Рассмотренные противоречия указывают на то, что задача поиска в B2B-каталогах носит не только метрический, но и структурированный характер. Пусть запрос представляется как набор сущностей:

$$q \to y = (y_{\text{type}}, y_{\text{brand}}, y_{\text{model}}, y_{\text{tx}_1}, \ldots, y_{\text{tx}_m}),$$

где каждая компонента принадлежит конечному дискретному множеству.

Тогда задача поиска может быть интерпретирована как задача структурированного предсказания:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} F(q, y),$$

где $F$ — скоринговая функция, учитывающая совместимость между запросом и структурой товара.

В отличие от непрерывного embedding-пространства, пространство $\mathcal{Y}$ дискретно и факторизуемо, что позволяет:

- декомпозировать задачу по сущностям,
- использовать детерминированные правила сопоставления,
- контролировать полноту на уровне отдельных компонент.

Таким образом, переход от чистого dense retrieval к структурированной модели поиска является не эвристическим, а теоретически мотивированным шагом, вытекающим из бинарной природы целевой функции и ограничений ANN.

Dense retrieval оптимизирует семантическую близость. Однако в B2B требуется точное совпадение сущностей. Семантическая близость не эквивалентна идентичности.

Основная гипотеза работы:

> Embedding-based retrieval не обеспечивает достаточную полноту в B2B DIY-каталогах из-за потери информации о сущностях; entity-based matching обеспечивает более высокую полноту.

Далее рассматривается реализация сущностно-ориентированной архитектуры и её интеграция с embedding-моделями в гибридной системе поиска.

## Методика исследования

Пусть задан каталог товаров

$$\mathcal{D} = \{d_1, \ldots, d_N\},$$

и множество пользовательских запросов

$$\mathcal{Q} = \{q_1, \ldots, q_M\}.$$

Для каждого запроса $q \in \mathcal{Q}$ определено множество релевантных документов

$$R(q) \subseteq \mathcal{D}.$$

В рассматриваемом B2B-сценарии преимущественно выполняется условие

$$|R(q)| = 1,$$

что соответствует поиску конкретной модели, артикула либо товара с заданной спецификацией.

Требуется построить функцию ранжирования

$$f : \mathcal{Q} \times \mathcal{D} \to \mathbb{R},$$

максимизирующую полноту извлечения (Recall@K) при ограничениях на вычислительные ресурсы и латентность.

В качестве базовой модели использована трансформерная архитектура ModernBERT-base [18, 19] (149M параметров), отображающая текст в векторное пространство размерности 768:

$$\phi : \mathcal{T} \to \mathbb{R}^{768}.$$

Сходство запроса $q$ и документа $d$ определяется косинусной мерой:

$$s(q, d) = \cos(\phi(q), \phi(d)).$$

Ранжирование осуществляется по убыванию значения $s(q, d)$.

Дообучение модели выполнялось на парах «пользовательский запрос – название товара» с использованием триплетной функции потерь:

$$L(a, p, n) = \max\big(0,\ d(a, p) - d(a, n) + m\big),$$

где:

- $a$ — embedding запроса (anchor),
- $p$ — релевантный товар (positive),
- $n$ — нерелевантный товар (negative),
- $d(\cdot, \cdot)$ — косинусная дистанция,
- $m = 0.3$ — margin.

Минимизируется эмпирический риск:

$$\mathcal{L} = \mathbb{E}_{(a,p,n)\sim\mathcal{S}} L(a,p,n).$$

Негативные примеры формировались с использованием стратегии hard-negative mining внутри батча.

Без аугментации наблюдалось быстрое насыщение функции потерь, что указывало на преобладание лёгких негативных примеров.

Для увеличения сложности обучающей выборки применялись следующие преобразования запросов:

- симуляция опечаток;
- транслитерация (латиница $\leftrightarrow$ кириллица);
- перестановка токенов;
- удаление токенов;
- замена синонимами и техническими эквивалентами.

Формально аугментация реализует отображение

$$A : q \mapsto \tilde{q},$$

где распределение $\tilde{q}$ аппроксимирует эмпирическое распределение искажённых пользовательских формулировок.

Для масштабируемого поиска использовался алгоритм Approximate Nearest Neighbors (ANN) на основе графа малых миров (HNSW) [17, 20] в библиотеке uSearch [21].

Пусть

$$\mathcal{I}_{ANN}(q,K)$$

— множество документов, возвращаемых ANN-алгоритмом.

Тогда верхняя граница полноты ограничена вероятностью попадания истинного ближайшего соседа в результирующее множество:

$$Recall@K \leq P\big(d^* \in \mathcal{I}_{ANN}(q,K)\big),$$

где $d^*$ — истинный ближайший сосед в полном пространстве.

При размере каталога порядка $10^7$ товаров среднее время ответа составляло около $50$ мс.

## Entity-based retrieval

С учётом выявленного конфликта между непрерывностью функции сходства в embedding-пространстве и дискретной природой релевантности в B2B-поиске дополнительно реализован сущностно-ориентированный слой извлечения.

Ключевое предположение данного слоя состоит в том, что значительная доля B2B-запросов содержит явно структурированные компоненты (бренд, модель, артикул, технические характеристики), а их точное совпадение является необходимым условием релевантности.

В отличие от dense-подхода, аппроксимирующего близость в непрерывном векторном пространстве, сущностный слой функционирует в дискретном символическом пространстве и накладывает структурные ограничения до этапа семантического ранжирования.

**Извлечение брендов**

Пусть $B$ — словарь брендов, $|B| > 10^4$, сформированный на основе нормализованных метаданных каталога и дополненный распространёнными вариантами написания и транслитерации.

Функция детерминированного совпадения задаётся как:

$$BrandMatch(q, d) = \begin{cases} 1, & \text{если } brand(q) = brand(d), \\ 0, & \text{иначе.} \end{cases}$$

Извлечение бренда реализуется посредством словарного сопоставления с предварительной нормализацией (приведение к единому регистру, транслитерация, удаление юридических суффиксов).

Данный компонент обеспечивает категориальную согласованность и устраняет кросс-брендовую семантическую близость, часто возникающую в embedding-пространстве вследствие схожести текстовых описаний.

**Извлечение моделей и артикулов**

Модели и артикулы описываются регулярным языком $\mathcal{L}_{model}$, охватывающим буквенно-цифровые шаблоны, дефисные конструкции и смешанные символьные последовательности, характерные для промышленного оборудования.

Функция совпадения определяется следующим образом:

$$ModelMatch(q, d) = \mathbf{1}\{model(q) = model(d)\}.$$

Данный компонент реализует жёсткое структурное ограничение. В отличие от embedding-подхода, который склонен сглаживать числовые токены и частично игнорировать символьные различия, точное сопоставление идентификаторов гарантирует сохранение дискретных различий (например, 850W vs 800W, HR3200C vs HR3200).

Формально данный модуль снижает вероятность ложноположительных совпадений, обусловленных локальной геометрической близостью векторных представлений.

**Обработка технических характеристик**

Пусть $TX_q,\ TX_d$ — множества нормализованных технических атрибутов запроса и документа соответственно. Нормализация атрибутов включает:

- унификацию единиц измерения (например, W, kW → единое представление),
- канонизацию числовых значений,
- отображение синонимов названий характеристик.

Частичная структурная согласованность определяется как:

$$Score_{tx} = |TX_q \cap TX_d|.$$

В отличие от бинарных компонентов совпадения бренда и модели, данный модуль допускает градуированную оценку совместимости при сохранении интерпретируемости и символической природы сопоставления.

## Гибридная архитектура

Итоговая функция скоринга определяется как взвешенная линейная комбинация структурных и семантических компонентов:

$$Score(q, d) = 3 \cdot ModelMatch + 2 \cdot BrandMatch + 1 \cdot TypeMatch + \alpha \cdot Score_{tx} + \beta \cdot s(q, d),$$

где $s(q, d)$ — косинусное сходство dense-эмбеддингов запроса и документа.
Система весов отражает структурную иерархию идентификаторов в B2B-поиске:

- совпадение модели имеет наивысший приоритет (точная идентичность),
- совпадение бренда обеспечивает категориальную согласованность,
- совпадение типа контролирует соответствие товарной категории,
- пересечение технических характеристик уточняет совместимость,
- dense-компонент выступает как вторичный сигнал для ранжирования.

Принципиально важно, что dense-компонент применяется преимущественно к отфильтрованному подмножеству

$$\mathcal{D}'(q) \subset \mathcal{D},$$

полученному после сущностной фильтрации.
Тем самым процесс извлечения реализует двухэтапную архитектуру:

1. структурная фильтрация (entity-based pruning);
2. dense reranking внутри множества $\mathcal{D}'(q)$.

Данная декомпозиция разделяет фазу дискретной идентификации и фазу непрерывного семантического ранжирования, что приводит к согласованию оптимизируемой функции с бинарной природой recall в B2B-поиске.

Предложенная методика позволяет эмпирически проверить гипотезу о том, что одних лишь непрерывных семантических эмбеддингов недостаточно для обеспечения требуемой полноты поиска в масштабных B2B-каталогах DIY-товаров. Основные причины заключаются в:

- сглаживании дискретных идентификаторов в embedding-пространстве,
- нечувствительности к малым числовым различиям,
- аппроксимационных ошибках ANN-индексации,
- геометрической близости семантически похожих, но структурно различных товаров.

Введение явного сущностно-ориентированного слоя восстанавливает структурную детерминированность поиска при сохранении гибкости ранжирования, обеспечиваемой dense-моделями.

# Эксперимент

Цель эксперимента — количественная проверка гипотезы о недостаточной полноте dense retrieval в B2B-каталогах DIY-товаров и оценка эффекта сущностно-ориентированной и гибридной архитектуры.

Эксперимент проводился на промышленном каталоге строительных и технических товаров.

- Размер каталога: $10.4$ млн SKU.
- Обучающая выборка: $100\,000$ пар «запрос – товар».
- Тестовая выборка: $5\,000$ реальных B2B-запросов.

Тестовый набор формировался из производственного лога и включал запросы следующих типов:

1. точный поиск модели или артикула;
2. запросы с указанием технических характеристик;
3. обобщённые категорийные запросы;
4. шумовые и неполные формулировки.

Распределение запросов отражает реальное поведение B2B-аудитории, где доминируют сущностно-определённые формулировки.

Оценка качества производилась по следующим метрикам:

- $Recall@10$ — полнота извлечения;
- $Precision@10$ — точность в топ-10;
- $MRR$ — средняя обратная позиция первого релевантного результата;
- Latency — среднее время ответа;
- Explainability score — экспертная оценка интерпретируемости (шкала 1–5).

Метрика Explainability score отражает степень прозрачности логики ранжирования и воспроизводимость причин попадания документа в выдачу.

Были реализованы и сопоставлены четыре архитектуры:

1. Dense retrieval (ModernBERT + uSearch HNSW);
2. BM25;
3. Entity-based retrieval;
4. Hybrid (entity pruning + dense reranking).

Таблица 1: Сравнение архитектур поиска

| Подход | Recall@10 | Precision@10 | Latency | Explainability |
|---|---|---|---|---|
| Dense | 0.65 | 0.70 | 50 ms | 2 |
| BM25 | 0.80 | 0.75 | 30 ms | 3 |
| Entity | 0.95 | 0.85 | 40 ms | 5 |
| Hybrid | 0.97 | 0.88 | 60 ms | 5 |

## Результаты эксперимента

Dense retrieval продемонстрировал ограниченную полноту ($Recall@10 = 0.65$), как показывает Таблица 1, что подтверждает гипотезу о сглаживании дискретных идентификаторов в embedding-пространстве.

BM25 существенно превзошёл dense-подход по полноте (+23% относительно dense), что свидетельствует о важности точного токенного совпадения в B2B-сценариях.

Entity-based retrieval обеспечил значительный прирост полноты (+46% относительно dense), достигнув $Recall@10 = 0.95$. Наиболее существенный прирост наблюдался на запросах, содержащих:

- точное указание модели;
- числовые технические характеристики;
- артикулярные идентификаторы.

Hybrid-архитектура обеспечила максимальное качество ($Recall@10 = 0.97$) при умеренном увеличении латентности (до 60 мс). Таким образом, добавление dense reranking после структурной фильтрации позволяет сохранить высокую полноту и улучшить ранжирование внутри сущностно-согласованного множества.

## Анализ ошибок

### Dense retrieval

Основные ошибки:

- подмена близких моделей (например, различие в одной цифре);
- сглаживание числовых характеристик (2800 Вт vs 3000 Вт);
- игнорирование артикулов как “шумовых” токенов.

Эти ошибки непосредственно иллюстрируют loss-mismatch между непрерывной функцией сходства и дискретной целевой функцией релевантности.

### Entity-based retrieval

Ошибки носили преимущественно технический характер:

- некорректная нормализация единиц измерения;
- неполное извлечение модели из запроса.

**Hybrid**

Ошибки ограничивались редкими long-tail сценариями:

- редкие бренды вне словаря;
- нестандартные сокращения;
- многокомпонентные составные запросы.

### Выводы эксперимента

Полученные результаты подтверждают исходную гипотезу: непрерывные семантические эмбеддинги в изоляции не обеспечивают требуемой полноты поиска в B2B-каталогах DIY-товаров.

Сущностно-ориентированная фильтрация устраняет систематические ошибки dense retrieval, а гибридная архитектура позволяет совместить дискретную строгость и семантическую гибкость, достигая максимального качества при промышленно допустимой латентности.

## Дискуссия

### Почему эмбеддинги теряют полноту

Embedding-модель оптимизирует непрерывную функцию сходства в евклидовом или сферическом пространстве, минимизируя среднюю дистанцию между релевантными парами. Формально минимизируется эмпирический риск вида

$$\mathbb{E}_{(q,d^+)}\, \ell\big(s(q, d^+)\big),$$

где $s(q, d)$ — непрерывная функция сходства.

Однако целевая функция релевантности в B2B-сценарии носит дискретный характер:

$$Rel(q, d) \in \{0, 1\},$$

причём в большинстве случаев $|R(q)| = 1$. Следовательно, задача сводится к точному извлечению единственного элемента множества.

Оптимизация similarity приводит к инвариантности по числовым значениям и идентификаторам, если их вклад в среднюю функцию потерь статистически невелик. Векторное пространство стремится сгладить локальные различия, минимизируя глобальную ошибку. В результате:

- близкие числовые характеристики оказываются почти неразличимыми;
- артикула и модели интерпретируются как шумовые токены;
- возникает конфликт между непрерывной аппроксимацией и дискретной целью.

Таким образом, потеря полноты является не случайной ошибкой, а следствием геометрических свойств embedding-пространства и выбранной функции оптимизации.

## Почему entity matching обеспечивает высокую полноту

Сущностно-ориентированный подход основан на дискретных проверках совпадения идентификаторов и атрибутов. Его свойства принципиально отличаются от dense retrieval:

- **Детерминированность:** совпадение модели или артикула определяется однозначно.
- **Прямая проверка идентификаторов:** отсутствует аппроксимация через непрерывную метрику.
- **Структурная интерпретируемость:** каждый фактор ранжирования прозрачен и объясним.

Формально, если $ModelMatch(q, d) = 1$, то вероятность релевантности стремится к единице:

$$P(Rel(q, d) = 1 \mid ModelMatch(q, d) = 1) \approx 1,$$

что делает данный компонент мощным якорем (anchor) в архитектуре retrieval.

Именно детерминированная природа проверки сущностей объясняет резкий рост $Recall@10$ по сравнению с чистым embedding-подходом.

## Инженерные ограничения

Разработка архитектуры проводилась с учётом производственных ограничений:

- Latency $< 100$ мс при каталоге $> 10^7$ SKU;
- ограничение оперативной памяти индекса;
- горизонтальная масштабируемость.

Dense retrieval требует ANN-структур и увеличивает потребление памяти. Entity-based фильтрация, напротив, позволяет сократить пространство поиска до структурно согласованного подмножества:

$$\mathcal{D}'(q) \subset \mathcal{D}, \quad |\mathcal{D}'(q)| \ll |\mathcal{D}|.$$

Гибридная архитектура обеспечивает баланс между качеством и вычислительной эффективностью.

На основании проведённого исследования можно сформулировать следующие рекомендации для B2B-каталогов технических товаров:

1. Обеспечивать полноту через entity matching до применения семантических моделей.
2. Использовать dense retrieval преимущественно для reranking внутри структурно отфильтрованного множества.
3. Контролировать метрики полноты отдельно от точности, поскольку снижение recall в B2B-задачах приводит к прямым бизнес-потерям.
4. Явно учитывать различие между задачами семантического поиска и задачами точного сущностного извлечения.

## Заключение

В работе показано, что embedding-based retrieval недостаточен для B2B DIY-каталогов при приоритете полноты извлечения. Экспериментально продемонстрировано увеличение $Recall@10$ с $0.65$ до $0.97$ при переходе к гибридной архитектуре с сущностным якорением.

Полученные результаты подтверждают гипотезу о фундаментальном конфликте между непрерывной оптимизацией similarity и дискретной природой релевантности в задачах точного товарного поиска.

Ключевой вывод состоит в следующем: в B2B-задачах retrieval архитектура должна строиться вокруг сущностей (модели, артикулы, технические характеристики), а семантическая близость должна играть вспомогательную роль.

В качестве направлений развития рассматриваются:

- графовые модели сущностей и их связей;
- использование LLM для генерации и нормализации синонимов;
- learning-to-rank поверх сущностно отфильтрованных кандидатов;
- формализация задачи как structured prediction с дискретными ограничениями.

## Список литературы


[1] Краснов Ф. В. Embedding-based retrieval: measures of threshold recall and precision to evaluate product search //Бизнес-информатика. – 2024. – Т. 18. – №. 2. – С. 22-34.

[2] Krasnov F., Kurushin F., Mogilevich E. Custom shared encoder for enhanced recall in e-commerce product search task //Second International Conference on Computing, Machine Learning, and Data Science (CMLDS 2025). – SPIE, 2025. – Т. 13730. – С. 84-91.

[3] Краснов Ф. В. Повышение полноты и точности поиска товаров на торговых интернет-площадках //ПРИКЛАДНАЯ ИНФОРМАТИКА Учредители: Московский университет"Синергия". – 2024. – Т. 19. – №. 2. – С. 118-136.

[4] Gan Y. et al. Binary embedding-based retrieval at Tencent //Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. – 2023. – С. 4056-4067.

[5] Li S. et al. Embedding-based product retrieval in taobao search //Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. – 2021. – С. 3181-3189.

[6] Weller O. et al. On the theoretical limitations of embedding-based retrieval //arXiv preprint arXiv:2508.21038. – 2025.

[7] Lin J. et al. Enhancing relevance of embedding-based retrieval at walmart //Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. – 2024. – С. 4694-4701.

[8] Ren Z. et al. Information Discovery in E-commerce //Foundations and TrendsŴ in Accounting. – 2024. – Т. 18. – №. 4-5. – С. 417-690.

[9] Schütze H., Manning C. D., Raghavan P. Introduction to information retrieval. – Cambridge : Cambridge University Press, 2008. – T. 39. – C. 234-265.

[10] Azzopardi L., De Rijke M., Balog K. Building simulated queries for known-item topics: an analysis using six european languages //Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. – 2007. – C. 455-462.

[11] Park L. A. F. Confidence intervals for information retrieval evaluation //ADCS 2010. – 2010. – C. 97.

[12] Hull D. Using statistical testing in the evaluation of retrieval experiments //Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. – 1993. – C. 329-338.

[13] Robertson S. E. The probability ranking principle in IR //Journal of documentation. – 1977. – T. 33. – №. 4. – C. 294-304.

[14] Schroff F., Kalenichenko D., Philbin J. Facenet: A unified embedding for face recognition and clustering //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 815-823.

[15] Karpukhin V. et al. Dense passage retrieval for open-domain question answering //Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). – 2020. – C. 6769-6781.

[16] Malkov Y. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs //IEEE transactions on pattern analysis and machine intelligence. – 2018. – T. 42. – №. 4. – C. 824-836.

[17] Aumüller M., Bernhardsson E., Faithfull A. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms //Information Systems. – 2020. – T. 87. – C. 101374.

[18] Devlin J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL, 2019.

[19] Warner B. et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference //Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). – 2025. – C. 2526-2547.

[20] Johnson J. et al. Billion-scale similarity search with GPUs. arXiv, 2019.

[21] Варданян А. USearch by Unum Cloud : программное обеспечение. Версия 2.24.0. 2023. URL: https://github.com/unum-cloud/usearch (дата обращения: 21.02.2026). DOI: 10.5281/zenodo.7949416.