НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2025 Том 25 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
Jule–August 2025 Vol. 25 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2025-25-4-710-717
УДК 004.89

K-sparse энкодер для эффективного информационного поиска

Вячеслав Юрьевич Добрынин✉

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
vidobrynin@itmo.ru✉, https://orcid.org/0009-0004-3056-8403

Аннотация

Введение. Современные промышленные поисковые системы, как правило, используют двухстадийный конвейер — быстрый отбор кандидатов и последующее ранжирование, что неизбежно ведет к потере части релевантных документов из-за простых алгоритмов на первой стадии. В работе предлагается одностадийный подход, сочетающий преимущества плотных моделей семантического поиска и эффективности инвертированных индексов. Ключевым компонентом решения является K-sparse энкодер, применяемый для преобразования плотных векторов в разреженные, совместимые с инвертированными индексами библиотеки Lucene. **Метод.** В отличие от ранее исследованного идентифицируемого вариационного автоэнкодера, предлагаемая модель основана на автоэнкодере с функцией активации TopK, которая явно фиксирует число ненулевых координат на этапе обучения. Такая функция активации делает процесс получения разреженного вектора дифференцируемым, устраняет необходимость постобработки и упрощает функцию потерь до суммы ошибки восстановления и компоненты, сохраняющей относительные расстояния между плотными и разреженными представлениями. Обучение выполнялось на подмножестве из 300 тыс. документов набора данных MS MARCO с использованием PyTorch и GPU NVIDIA L4. **Основные результаты.** Предложенная модель достигает 96,6 % качества исходной плотной модели по метрике NDCG@10 (0,57 против 0,59) на наборе данных SciFact при 80 % разреженности векторов. Дополнительно показано, что дальнейшее увеличение разреженности снижает объем индекса и ускоряет время поиска, сохраняя приемлемое качество поиска. По используемой памяти решение превосходит графовый алгоритм Hierarchical Navigable Small World, а по скорости приближается к нему при высоких уровнях разреженности. **Обсуждение.** Работа подтверждает применимость предложенного подхода для поиска неструктурированных данных. Прямое управление степенью разреженности дает возможность балансировать между качеством, задержкой поиска и требованиями к памяти. Благодаря использованию инвертированного индекса на базе библиотеки Lucene, предлагаемое решение может быть эффективно применено в промышленных поисковых системах. В качестве направлений дальнейших исследований рассматриваются интерпретируемость извлекаемых признаков и повышение качества поиска при значительной разреженности представлений.

Ключевые слова

информационный поиск, разреженные векторные представления, K-sparse автоэнкодер, функция активации TopK, инвертированный индекс, одностадийная архитектура

Ссылка для цитирования: Добрынин В.Ю. K-sparse энкодер для эффективного информационного поиска // Научно-технический вестник информационных технологий, механики и оптики. 2025. Т. 25, № 4. С. 710–717. doi: 10.17586/2226-1494-2025-25-4-710-717

K-sparse encoder for efficient information retrieval

Viacheslav Yu. Dobrynin✉

ITMO University, Saint Petersburg, 197101, Russian Federation
vidobrynin@itmo.ru✉, https://orcid.org/0009-0004-3056-8403

Abstract

Modern industrial search engines typically employ a two-stage pipeline: fast candidate retrieval followed by reranking. This approach inevitably leads to the loss of some relevant documents due to the simplicity of algorithms used in the first stage. This work proposes a single-stage approach that combines the advantages of dense semantic search models with the efficiency of inverted indices. The key component of the solution is a K-sparse encoder used to convert dense

vectors into sparse ones compatible with inverted indices of the Lucene library. In contrast to the previously studied identifiable variational autoencoder, the proposed model is based on an autoencoder with a TopK activation function which explicitly enforces a fixed number of non-zero coordinates during training. This activation function makes the sparse vector generation process differentiable, eliminates the need for post-processing, and simplifies the loss function to a sum of reconstruction error and a component preserving relative distances between dense and sparse representations. The model was trained on a 300,000-document subset of the MS MARCO dataset using PyTorch and an NVIDIA L4 GPU. The proposed model achieves 96.6 % of the quality of the original dense model in terms of the NDCG@10 metric (0.57 vs. 0.59) on the SciFact dataset with 80 % sparsity. It is also shown that further increasing sparsity reduces index size and improves retrieval speed while maintaining acceptable search quality. In terms of memory usage, the approach outperforms the Hierarchical Navigable Small World (HNSW) graph-based algorithm, and at high sparsity levels, its speed approaches that of HNSW. The results confirm the applicability of the proposed approach to unstructured data retrieval. Direct control over sparsity enables balancing between search quality, latency, and memory requirements. Thanks to the use of an inverted index based on the Lucene library, the proposed solution is well suited for industrial-scale search systems. Future research directions include interpretability of the extracted features and improving retrieval quality under high sparsity conditions.

Keywords
information retrieval, sparse vector representations, autoencoder, TopK activation function, inverted index, single-stage architecture

For citation: Dobrynin V.Yu. K-sparse encoder for efficient information retrieval. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2025, vol. 25, no. 4, pp. 710–717 (in Russian). doi: 10.17586/2226-1494-2025-25-4-710-717

Введение

Архитектура современных промышленных поисковых систем состоит из двух стадий [1, 2]: быстрое извлечение документов-кандидатов и ранжирование полученного списка. Хотя данный подход сочетает скорость и качество каждого из этапов, первый этап фундаментально ограничивает его, поскольку теряет релевантные документы из-за своей простоты [3, 4]. Решить данную проблему позволяет переход к одностадийной архитектуре. Впервые подобный подход был продемонстрирован в моделях Standalone Neural Ranking Model [5], Sparse Lexical and Expansion Model [6] и других. Также следует отметить, что подобные решения были представлены в работах [7, 8].

В настоящей работе продолжены исследования алгоритма, предложенного в [7]. Основная идея заключается в использовании энкодера, который конвертирует плотные векторные представления в разреженные. Эти разреженные представления можно использовать с инвертированным индексом, обеспечивающим высокую эффективность поиска. Требуемый энкодер получается в ходе обучения автоэнкодера, состоящего из энкодера и декодера.

В работе [7] была выдвинута гипотеза о том, что для разреженного пространства необходимы ограничения согласованности, независимости, разреженности, а также сохранения относительных расстояний между плотными и разреженными представлениями. Однако эксперименты показали, что ключевую роль для качества поиска играла функция потерь, отвечающая за сохранение относительных расстояний.

Ввиду выявленных ограничений предлагается усовершенствованная архитектура модели, позволяющая учесть недостатки ранее предложенного подхода.

Ограничения предложенной модели

В основе архитектуры лежит модель, предложенная в работе [7], в которой использовался идентифицируемый вариационный автоэнкодер (Identifiable Variational Autoencoder, iVAE) [9], предназначенный для ввода ограничения независимости на получаемое скрытое пространство. Однако вариационная природа данной модели имеет существенное ограничение — энкодер генерирует параметры нормального распределения, из которого сэмплируются конкретные значения разреженного вектора. Данная особенность приводит к тому, что в результирующем векторе отсутствуют значения, равные нулю, а присутствуют лишь близкие к нему. Это вынуждает использовать механизм зануления значений, близких к нулю. Так, в работе [7] использовался определенный перцентиль наибольших значений по модулю. Однако такое зануление приводит к потере информации, поскольку во время обучения автоэнкодер не учитывает этот процесс, что создает несоответствие между обучением и использованием модели. Включить такое зануление в процесс обучения невозможно, поскольку без дополнительных модификаций эта операция не является дифференцируемой [10].

Для обучения модели использовалась следующая общая функция потерь:

$$L = \alpha_1 L_{ELBO} + \alpha_2 L_{dist} + \alpha_3 L_{FLOPS}, \quad (1)$$

где L_{ELBO} — вариационная нижняя граница (Evidence Lower Bound, ELBO), необходимая как для обеспечения независимости координат скрытого пространства, так и для восстановления входных данных автоэнкодера; L_{dist} — функция потерь (distance loss, dist), сохраняющая относительные расстояния между плотными и разреженными векторами; L_{FLOPS} — функция потерь (Floating-Point Operations, FLOPS), необходимая для получения разреженного вектора; α_1, α_2 и α_3 — коэффициенты, регулирующие вклад каждой из компонент. Подробное описание функции потерь представлено в работе [7].

Подбор значений коэффициентов функции потерь (1) осуществлялся эмпирически, и влияние их выбора показано на рис. 1.

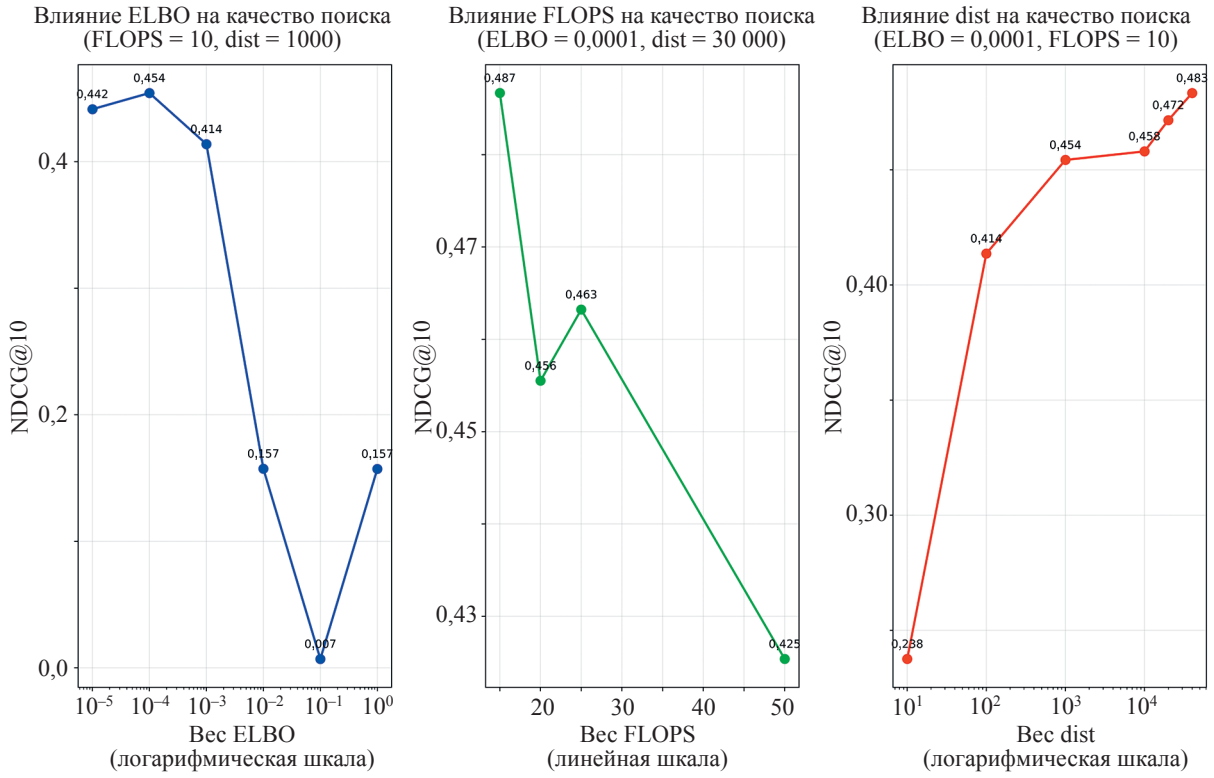

Рис. 1. Влияние компонентов функции потерь на качество поиска по метрике Normalized Discounted Cumulative Gain (NDCG)

Fig. 1. Impact of loss function components on search quality measured by Normalized Discounted Cumulative Gain (NDCG)

Таким образом, проведенные исследования показали, что качество поиска существенно растет при увеличении вклада функции потерь, отвечающей за сохранение относительных расстояний между плотными и разреженными векторами. В то же время наблюдается снижение качества при увеличении вклада нижней вариационной границы, используемой в iVAE для обучения автоэнкодера.

Предлагаемые изменения

Вместо энкодера iVAE в текущей работе предлагается использовать K-sparse энкодер [11] с применением специальной функции активации TopK (рис. 2). Функция TopK позволяет явно ограничить количество ненулевых компонент в скрытом пространстве, при этом все еще поддерживает использование градиентного спуска для оптимизации, что является критически важным свойством.

В работе [12] исследователи из компании OpenAI используют K-sparse энкодер для интерпретации нейронов больших языковых моделей.

В настоящей работе предлагается адаптировать данный энкодер для задачи извлечения интерпретируемых компонент из плотного вектора **x** с целью их дальнейшего использования с инвертированным индексом в системах информационного поиска.

Таким образом, вместо iVAE будет обучаться обычный автоэнкодер с функцией потерь Mean Square Error. Это позволит существенно упростить модель и пол-

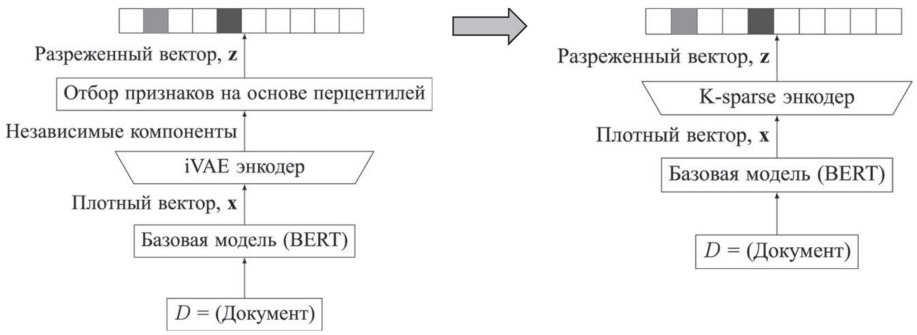

Рис. 2. Упрощение процесса разреживания плотного вектора модели Bidirectional Encoder Representations from Transformers (BERT)

Fig. 2. Simplification of the dense vector sparsification process for the Bidirectional Encoder Representations from Transformers (BERT) model

ностью устранить необходимость в блоке зануления компонент, близких к нулю, так как функция TopK уже обеспечивает требуемую разреженность.

Важным компонентом новой архитектуры модели является функция потерь, предложенная в работе [7], которая отвечает за сохранение относительных расстояний между плотными и разреженными векторами:

$$L_{dist} = \frac{1}{N}\left|\left|\text{diag}(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{X}^T - \text{diag}(\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{Z}^T\right|\right|_F, \quad (2)$$

где N — размер пакета (батча); $\mathbf{X}$ и $\mathbf{Z}$ — пакеты плотных и разреженных векторных представлений; diag($\mathbf{M}$) — диагональная матрица из диагональных элементов матрицы $\mathbf{M}$; $\left|\left|.\right|\right|_F$ — норма Фробениуса.

Это особенно критично, поскольку именно порядок ранжирования документов является ключевым свойством в задаче информационного поиска, для которой будут использоваться полученные разреженные представления.

Результирующая функция потерь имеет следующий вид:

$$L = \alpha_1 L_{MSE} + \alpha_2 L_{dist}. \quad (3)$$

Применение функции потерь (3) в процессе обучения представлено на рис. 3.

Предложенные изменения позволяют обучать энкодер, который изначально формирует представление с фиксированным числом ненулевых компонент, устраняя необходимость в последующем занулении.

Независимость координат разреженного пространства

В данной работе явно не накладывается ограничение на независимость в скрытом пространстве. Однако это не означает, что такое ограничение не является полезным. Компоненты плотных векторных представлений могут одновременно отражать несколько концептов — аналогично нейронам в больших языковых моделях, активирующимся в различных контекстах. Тем не менее, как показано в работе исследователей из компании Anthropic [13], использование разреженных автоэнкодеров с избыточной размерностью скрытого пространства (overcomplete) позволяет «распутать» нейроны или компоненты плотного вектора до более однозначных интерпретаций. Это свойство является желательным при построении поиска на основе инвертированного индекса.

Экспериментальное исследование

Модель обучалась с использованием языка программирования Python и фреймворка машинного обучения PyTorch [14]. Обучение выполнялось на графическом ускорителе NVIDIA L4 с 23 ГБ видеопамяти.

В качестве набора данных для обучения применено подмножество из 300 тыс. документов коллекции MS MARCO [15]. Для оперативной проверки качества в процессе разработки использовался сравнительно небольшой набор данных SciFact [16].

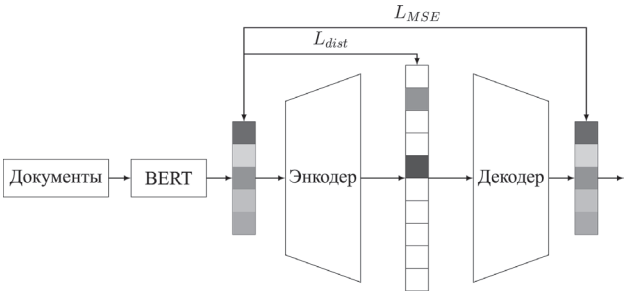

Рис. 3. Схема обучения K-sparse автоэнкодера

Fig. 3. Training scheme of the K-sparse autoencoder

Параметры эксперимента:

— базовая модель для генерации плотных векторных представлений: sentence-transformers/msmarco-distilbert-dot-v5;
— размер плотного вектора: 768;
— размер разреженного вектора: 3000;
— количество ненулевых компонент для функции TopK: 600 (что соответствует уровню разреженности 80 % — такому же, как и в версии алгоритма [7]).

Важно отметить, что основная задача заключается в разреживании векторных представлений с минимальной потерей качества поиска по сравнению с исходной плотной моделью. Идеальным результатом было бы максимальное приближение к качеству поиска оригинальной плотной модели, но с преимуществами разреженных представлений в скорости и эффективности поиска благодаря применения инвертированного индекса.

Результаты оценки качества представлены в табл. 1.

Анализ результатов показывает, что предложенная новая версия алгоритма с K-sparse энкодером демонстрирует лучшее качество поиска (0,57 против 0,51 по метрике NDCG@10), приближаясь к 96,6 % от качества исходной плотной модели.

Основными факторами улучшения являются: упрощение архитектуры модели благодаря замене iVAE на обычный автоэнкодер; устранение необходимости постобработки с занулением значений, близких к нулю; явное ограничение разреженности с помощью функции активации TopK.

Выполнен анализ влияния уровня разреженности на качество поиска. В табл. 2 представлены значения метрики NDCG@10 в зависимости от k — количества ненулевых компонент разреженного вектора.

Результаты показывают, что увеличение уровня разреженности приводит к снижению качества поиска. Наиболее значительная потеря качества наблюдается при увеличении уровня разреженности с 93 % до 97 %. С другой стороны, повышение разреженности приводит к снижению требований к памяти за счет хранения меньшего числа ненулевых компонент, а также к ускорению поиска, поскольку при уменьшении числа термов в запросе инвертированные индексы работают быстрее. Таким образом, выбор уровня разреженности представляет собой компромисс между качеством и ресурсоемкостью поиска.

Таблица 1. Значения NDCG@10 для разных версий предложенного метода и эмбеддинг модели
Table 1. NDCG@10 scores for different versions of the proposed method and the embedding model

Алгоритм	NDCG@10	Относительное качество, %
Алгоритм, предложенный в работе [7] (с iVAE)	0,51	86,4
Алгоритм, предложенный в настоящей работе (с K-sparse)	0,57	96,6
Эмбеддинг модель	0,59	100

Таблица 2. Значения NDCG@10 при различном уровне разреженности вектора
Table 2. NDCG@10 values at different sparsity levels of the vector

k	Уровень разреженности, %	NDCG@10
100	97	0,41
200	93	0,50
300	90	0,53
400	87	0,54
500	83	0,56
600	80	0,57

Таблица 3. Значения NDCG@10 при различных коэффициентах функции потерь
Table 3. NDCG@10 values for different loss function coefficient settings

α_1	α_2	NDCG@10
0	1	0,5720
1	0	0,0062
0,5	1	0,5606
1	0,5	0,5694
1	1	0,5696

Исследовано влияние компонентов функции потерь, которое регулируется с помощью коэффициентов формулы (3) (табл. 3).

Полученные значения NDCG@10 показывают, что основное влияние на качество поиска оказывает предложенная функция потерь (2), направленная на сохранение относительных расстояний. Данный результат ожидаем, поскольку в задаче ранжирования относительный порядок документов является ключевым фактором, используемым при обучении моделей ранжирования. Однако важно отметить, что включение ошибки восстановления в общую функцию потерь не приводит к значительным потерям качества, но может быть полезным при интерпретации координат разреженного пространства за счет сохранения семантической составляющей векторов. Однако этот аспект не рассматривается в рамках данной работы и является предметом дальнейших исследований.

Все остальные эксперименты в работе выполнялись с коэффициентами $\alpha_1 = \alpha_2 = 1$.

Для оценки эффективности предложенного алгоритма в качестве базового алгоритма для сравнения был выбран графовый Hierarchical Navigable Small World (HNSW) [17], являющийся одной из самых эффективных реализаций алгоритмов приближенного векторного поиска. Благодаря высокой скорости и качеству поиска, HNSW используется в качестве алгоритма по умолчанию во многих современных векторных базах данных. Его преимущество заключается в использовании внутренней графовой структуры, обеспечивающей высокую производительность при сохранении точности. Однако существенным недостатком HNSW являются высокие требования к объему памяти, необходимой для хранения этой структуры. Предложенный алгоритм реализован с использованием библиотеки Lucene, которая является стандартом в области полнотекстового поиска, а также с применением модуля PyLucene, поскольку реализация выполнена на языке программирования Python. Для обеспечения корректного сравнения реализация HNSW также использовалась в рамках библиотеки Lucene. Это важно, так как в обоих случаях учитываются накладные расходы, связанные с вызовами из Python в Java, а также работа Java Virtual Machine, включая JIT-компиляцию (JIT — just-in-time). Для оценки эффективности были применены подмножества набора данных MS MARCO. При измерении времени выполнения учитывались как этап кодирования запроса в векторное представление, так и последующий поиск по индексу. Эксперименты проводились на персональном компьютере с процессором Intel Core i7-11850H, графическим процессором NVIDIA RTX A4000 Mobile и 32 ГБ оперативной памяти. Сравнение охватывает следующие метрики: объем используемой дисковой памяти, время выполнения запросов и качество поиска на подмножестве набора данных. Результаты приведены в табл. 4.

Из результатов сравнения видно, что предложенный алгоритм во всех вариантах превосходит HNSW по используемой памяти, но уступает по времени выполнения. Однако при увеличении разреженности время выполнения приближается к показателям HNSW, при этом затраты на память значительно снижаются. Однако, это негативно сказывается на качестве поиска.

Обсуждение

В результате экспериментального исследования было показано, что предложенная модель позволяет снизить ресурсоемкость векторного поиска при сохранении высокого качества результатов.

Установлено, что при увеличении разреженности векторов существенно снижаются требования к памяти, а производительность поиска приближается к уровню

Таблица 4. Сравнение предложенного алгоритма и HNSW на подмножествах набора данных MS MARCO
Table 4. Comparison of the proposed algorithm and HNSW on subsets of the MS MARCO dataset

Количество документов в подмножестве	Предложенный алгоритм			HNSW
	$k = 100$	$k = 200$	$k = 600$	
Объем используемой дисковой памяти, МБ				
25 000	13	25	60	75
50 000	24	43	113	149
100 000	46	85	224	298
Время выполнения запроса, мс				
25 000	37,3	57,5	225,8	30,6
50 000	48,1	97,0	366,5	33,6
100 000	58,0	128,1	755,6	38,0
Качество поиска, NDCG@10				
25 000	0,2268	0,2451	0,2535	0,2410
50 000	0,2183	0,2973	0,3404	0,3513
100 000	0,2887	0,3485	0,3616	0,3710

высокоэффективного алгоритма HNSW. Замедление работы при более высоком уровне разреженности объясняется необходимостью перебора большего числа постинг-листов, соответствующих термам запроса. Дополнительным фактором замедления является использование энкодера, обеспечивающего разреживание — в отличие от исходной модели, не имеющей аналогичного компонента, он вносит дополнительную вычислительную нагрузку.

Показано, что коэффициент разреженности представляет собой компромисс между качеством поиска и его скоростью. Таким образом, в задачах, где приоритет отдается точности, целесообразно использовать менее разреженные векторы, тогда как в сценариях, критичных к времени отклика, — более разреженные. При этом на всем диапазоне исследованных коэффициентов разреженности предложенный алгоритм демонстрирует существенно более низкие требования к объему памяти по сравнению с HNSW, что особенно важно при работе с большими объемами данных. Кроме того, использование инвертированного индекса — отраслевого стандарта в системах полнотекстового поиска — обеспечивает предложенному методу потенциал к лучшему масштабированию в сравнении с HNSW.

Тем не менее важным направлением дальнейшей работы остается повышение выразительности векторных представлений при высокой степени разреженности, что может позволить улучшить качество поиска без потери производительности.

Заключение

В рамках данного исследования разработана и экспериментально проверена усовершенствованная версия алгоритма разреживания плотных векторных представлений для задач информационного поиска. Ключевым архитектурным изменением стало использование K-sparse автоэнкодера вместо идентифицируемого вариационного автоэнкодера.

Такой подход позволил устранить необходимость постобработки векторов путем зануления. Уменьшена сложность модели при сохранении ее эффективности. Появилась возможность напрямую контролировать уровень разреженности получаемых представлений.

В результате проведенных экспериментов было достигнуто улучшение качества поиска по сравнению с предыдущей версией модели для метрики NDCG@10 с 0,51 до 0,57, что соответствует 96,6 % от качества оригинальной плотной модели. Это существенно приближает разреженное представление к качеству плотного при $k = 600$, при этом сохраняя преимущества разреженных векторов в виде возможности использования с инвертированным индексом для высокоэффективного поиска.

Экспериментальная оценка показала, что K-sparse индексы занимают значительно меньше памяти, чем HNSW, при умеренной потере скорости и незначительном падении качества.

Полученные результаты подтверждают перспективность перехода от двухстадийной архитектуры поисковых систем к одностадийной с использованием полученных разреженных представлений, способных обеспечить как низкие требования к ресурсам, так и качество поиска.

В дальнейших исследованиях планируется углубленная работа по интерпретации полученного разреженного векторного представления, изучение способов повышения выразительности векторных представлений с большим коэффициентом разреженности, масштабирование обучения на больших наборах данных.

Литература

1. Chen R., Gallagher L., Blanco R., Culpepper J.S. Efficient cost-aware cascade ranking in multi-stage retrieval // Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017. P. 445–454. https://doi.org/10.1145/3077136.3080819
2. Liu S., Xiao F., Ou W., Si L. Cascade ranking for operational E-commerce search // Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017. P. 1557–1565. https://doi.org/10.1145/3097983.3098011
3. Furnas G.W., Landauer T.K., Gomez L.M., Dumais S.T. The vocabulary problem in human-system communication // Communications of the ACM. 1987. V. 30. N 11. P. 964–971. https://doi.org/10.1145/32206.32212
4. Zhao L., Callan J. Term necessity prediction // Proc. of the 19th ACM International Conference on Information and Knowledge Management. 2010. P. 259–268. https://doi.org/10.1145/1871437.1871474
5. Zamani H., Dehghani M., Croft W.B., Learned-Miller E., Kamps J. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing // Proc. of the 27th ACM International Conference on Information and Knowledge Management. 2018. P. 497–506. https://doi.org/10.1145/3269206.3271800
6. Formal T., Piwowarski B., Clinchant S. SPLADE: Sparse lexical and expansion model for first stage ranking // Proc. of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021. P. 2288–2292. https://doi.org/10.1145/3404835.3463098
7. Dobrynin V., Sherman M., Abramovich R., Platonov A. A sparsifier model for efficient information retrieval // IEEE 18th International Conference on Application of Information and Communication Technologies (AICT). 2024. P. 1–4. https://doi.org/10.1109/aict61888.2024.10740301
8. Dobrynin V.Yu., Abramovich R.K., Platonov A.V. Efficient sparse retrieval through embedding-based inverted index construction // Scientific and Technical Journal of Information Technologies, Mechanics and Optics. 2025. V. 25. N 1. P. 61–67. https://doi.org/10.17586/2226-1494-2025-25-1-61-67
9. Khemakhem I., Kingma D.P., Monti R.P., Hyvarinen A. Variational autoencoders and nonlinear ICA: A unifying framework // Proc. of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS). 2020. V. 108. P. 2207–2216.
10. Louizos C., Welling M., Kingma D.P. Learning sparse neural networks through L0 regularization // arXiv. 2017. arXiv:1712.01312. https://doi.org/10.48550/arXiv.1712.01312
11. Makhzani A., Frey B. k-Sparse autoencoders // arXiv. 2013. arXiv:1312.5663. https://doi.org/10.48550/arXiv.1312.5663
12. Gao L., Tour T.D., Tillman H., Goh G., Troll R., Radford A., Sutskever I., Leike J., Wu J. Scaling and evaluating sparse autoencoders // arXiv. 2024. arXiv:2406.04093. https://doi.org/10.48550/arXiv.2406.04093
13. Bricken T., Templeton A., Batson J., Chen B., Jermyn A., Conerly T., et al. Towards monosemanticity: decomposing language models with dictionary learning // Transformer Circuits Thread. 2023.
14. Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., et al. PyTorch: An imperative style, high-performance deep learning library // arXiv. 2019. arXiv:1912.01703. https://doi.org/10.48550/arXiv.1912.01703
15. Bajaj P., Campos D., Craswell N., Deng L., Gao J., Liu X., et al. MS MARCO: A human generated MAchine Reading COmprehension dataset // arXiv. 2016. arXiv:1611.09268. https://doi.org/10.48550/arXiv.1611.09268
16. Wadden D., Lin S., Lo K., Wang L.L., van Zuylen M., Cohan A., Hajishirzi H. Fact or fiction: verifying scientific claims // Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020. P. 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609
17. Malkov Y., Yashunin D.A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020. V. 42. N 4. P. 824–836. https://doi.org/10.1109/TPAMI.2018.2889473

References

1. Chen R., Gallagher L., Blanco R., Culpepper J.S. Efficient cost-aware cascade ranking in multi-stage retrieval. *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 445–454. https://doi.org/10.1145/3077136.3080819
2. Liu S., Xiao F., Ou W., Si L. Cascade ranking for operational E-commerce search. *Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1557–1565. https://doi.org/10.1145/3097983.3098011
3. Furnas G.W., Landauer T.K., Gomez L.M., Dumais S.T. The vocabulary problem in human-system communication. *Communications of the ACM*, 1987, vol. 30, no. 11, pp. 964–971. https://doi.org/10.1145/32206.32212
4. Zhao L., Callan J. Term necessity prediction. Proc. of the 19th ACM *International Conference on Information and Knowledge Management*, 2010, pp. 259–268. https://doi.org/10.1145/1871437.1871474
5. Zamani H., Dehghani M., Croft W.B., Learned-Miller E., Kamps J. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. *Proc. of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 497–506. https://doi.org/10.1145/3269206.3271800
6. Formal T., Piwowarski B., Clinchant S. SPLADE: Sparse lexical and expansion model for first stage ranking. *Proc. of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2288–2292. https://doi.org/10.1145/3404835.3463098
7. Dobrynin V., Sherman M., Abramovich R., Platonov A. A sparsifier model for efficient information retrieval. *IEEE 18th International Conference on Application of Information and Communication Technologies (AICT)*, 2024, pp. 1–4. https://doi.org/10.1109/aict61888.2024.10740301
8. Dobrynin V.Yu., Abramovich R.K., Platonov A.V. Efficient sparse retrieval through embedding-based inverted index construction. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2025, vol. 25, no. 1, pp. 61–67. https://doi.org/10.17586/2226-1494-2025-25-1-61-67
9. Khemakhem I., Kingma D.P., Monti R.P., Hyvarinen A. Variational autoencoders and nonlinear ICA: A unifying framework. *Proc. of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020, vol. 108, pp. 2207–2216.
10. Louizos C., Welling M., Kingma D.P. Learning sparse neural networks through L0 regularization. *arXiv*, 2017, arXiv:1712.01312. https://doi.org/10.48550/arXiv.1712.01312
11. Makhzani A., Frey B. k-Sparse autoencoders. *arXiv*, 2013, arXiv:1312.5663. https://doi.org/10.48550/arXiv.1312.5663
12. Gao L., Tour T.D., Tillman H., Goh G., Troll R., Radford A., Sutskever I., Leike J., Wu J. Scaling and evaluating sparse autoencoders. *arXiv*, 2024, arXiv:2406.04093. https://doi.org/10.48550/arXiv.2406.04093
13. Bricken T., Templeton A., Batson J., Chen B., Jermyn A., Conerly T., et al. Towards monosemanticity: decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.
14. Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., et al. PyTorch: An imperative style, high-performance deep learning library. *arXiv*, 2019, arXiv:1912.01703. https://doi.org/10.48550/arXiv.1912.01703
15. Bajaj P., Campos D., Craswell N., Deng L., Gao J., Liu X., et al. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv*, 2016, arXiv:1611.09268. https://doi.org/10.48550/arXiv.1611.09268
16. Wadden D., Lin S., Lo K., Wang L.L., van Zuylen M., Cohan A., Hajishirzi H. Fact or fiction: verifying scientific claims. *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7534–7550. https://doi.org/10.18653/v1/2020.emnlp-main.609
17. Malkov Y., Yashunin D.A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, vol. 42, no. 4, pp. 824–836. https://doi.org/10.1109/TPAMI.2018.2889473

Автор

Добрынин Вячеслав Юрьевич — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 57223099701, https://orcid.org/0009-0004-3056-8403, vidobrynin@itmo.ru

Author

Viacheslav Yu. Dobrynin — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 57223099701, https://orcid.org/0009-0004-3056-8403, vidobrynin@itmo.ru

Статья поступила в редакцию 05.05.2025
Одобрена после рецензирования 10.06.2025
Принята к печати 20.07.2025

Received 05.05.2025
Approved after reviewing 10.06.2025
Accepted 20.07.2025