Полнота поиска по продуктам: сравнение встраиваемых кодеров

Ф.В. Краснов

***Аннотация*—Стремительное развитие нейронных сетей на основе архитектуры трансформеров не оставляет времени для глубокого анализа их эффективности в задачах продуктового поиска. Новые исследования трансформеров появляются достаточно часто, но для проверки их превосходства над существующими решениями, например на основе fastText, DSSM или ELMO, не всегда есть время и ресурсы. Сравнения в академических статьях обычно производятся на публичных наборах данных, которые значительно отличаются от используемых в промышленных условиях. Переход на новые модели машинного обучения в промышленных системах поиска сопряжён со значительными затратами времени и ресурсов из-за необходимости индексации каталога продуктов, поэтому цена ошибки достаточно высока. В настоящем исследовании проанализированы сильные и слабые стороны различных моделей применительно к задаче продуктового поиска, проведена экспериментальное сравнение различных моделей для кодеров.**

Был создан набор данных для обучения на основании журналов поисковых запросов пользователей, произведено обучение моделей с оптимальными гипер-параметрами, даны подробные определения автономных показателей для измерения эффективности, собран проверочный набор данных на текстовых представлениях продуктов, сделаны сравнения автономных показателей для задачи продуктового поиска в электронной коммерции и сделана оценка показателей асессорами.

Для кодеров рассматривались предобученные модели и модели, обученные «с нуля». При сравнении для каждой модели кодера посчитаны значения показателя пороговой полноты R@k для набора поисковых запросов. Было выявлено, что кодеры эффективны в разной степени в зависимости от количества слов в поисковом запросе.

Данное исследование не ставит задачи получить наилучшую модель для решения общих задач получения текстовых эмбеддингов. Напротив, авторы осознают узость своей задачи, но в условиях ограниченных ресурсов, хотят быть уверенными в методике выбора наиболее эффективной модели машинного обучения для её решения. Поэтому основным вкладом данного исследования является методика проведения сравнения моделей машинного обучения для кодера в прикладной задаче увеличения полноты продуктового поиска.

***Ключевые слова* — сравнение моделей, показатели эффективности продуктового поиска, нейросетевые архитектуры для поиска, кодеры, продуктовый поиск.**

I. Введение

Задача сравнения моделей машинного обучения, работающих с текстовыми данными, стала насущной в связи с ростом количества академических исследований таких моделей. Упростить задачу выбора лучшей модели для конкретной задачи предназначены соревнования, такие как Kaggle или Massive Text Embedding Benchmark (MTEB). Например, MTEB [1] выявляет лидеров соревнования на 52 наборах данных для 12 показателей и 8 типов задач. Для русского языка есть отдельный конкурс RU-MTEB [2] из 20 заданий. Модели-победители хороши для задач проверки концепций (Proof of Concept, PoC), но обычно требуют избыточных вычислительных ресурсов, являются слишком общими и нуждаются в настройке на предметную область. Например, на электронную коммерцию.

Одним из подходов к моделированию информации о продуктах для поиска стало извлечение информации на основании эмбеддингов (Embedding-based Retrieval, EBR). Эмбеддинги фрагментов текста получаются в результате работы кодера на основе моделей машинного обучения. Для обучения кодеров в MTEB представлены наборы данных TREC Robust [3], MSMARCO [4] и набор данных о кликах [5]. Но несмотря на разнообразие и количество публичных наборов данных, участвующих в MTEB, наборы данных по электронной коммерции с текстами в нем не представлены.

В академических исследованиях представлены несколько наборов данных для электронной коммерции. Наиболее популярные из них собраны в таблице 1.

Таблица 1: Наборы данных по электронной коммерции

Название	Ссылка	Количество запросов
ESCI	[6]	130 652
WANDS	[7]	480
Home Depot	[8]	11 795
Crowdflower	[9]	261

Следует обратить внимание, на то что количество и разнообразие данных не велико. Например, в наборе данных WANDS из всего разнообразия продуктов представлена только мебель и товары для дома.

Совсем другое качество у наборов данных для соревнований по рекомендательным системам в электронной коммерции. Например, в соревновании компании OTTO [10] представлены наборы данных с миллионами примеров. Но в связи с

Статья получена 1 октября 2024. Ф.В.Краснов, Исследовательский центр ООО "ВБ СК" на базе Инновационного Центра Сколково. krasnov.fedor2@wb.ru, http://orcid.org/0000-0002-9881-7371.

конфиденциальностью информации в наборе данных OTTO нет названий товаров, представлены только цифровые идентификаторы товаров, поэтому нет возможности применить инструменты анализа текста, которые по мнению автора дают максимальный результат.

В соревновании компании H&M [11] предложен набор данных из 45875 продуктов с названиями и принадлежностью к таксономии, но нет поисковых запросов и все тексты только на английском. В исследовании [12] так же отмечено, что ни в соревновании [10], ни в [11] победители не использовали искусственные нейронные сети для достижения лучших результатов. Таким образом, существует определённый дефицит наборов данных на русском языке для оценки и обучения моделей кодеров для продуктового поиска.

II. Полнота поиска по продуктам

Пороговая полнота поисковой выдачи R@k - это общепринятый показатель качества продуктового поиска на ряду с пороговой точностью [16, 13]. Падение полноты продуктового поиска негативно влияет на узкие поисковые запросы. Во-первых, приводит к появлению нулевых результатов поиска. Во-вторых, уменьшает количество найденных продуктов до критических величин, вызывающих недоверие пользователей и снижающих количество кликов, приводящих к покупке.
В исследованиях встречаются значения @k=100 [13] и @k=1000 [15] для порога полноты (@k). Хотя, например, для запроса «юбка» в каталоге продуктов интернет-магазина женской одежды будут сотни тысяч продуктов, соответствующих данному запросу, и оценивать полноту по порогу @k=1000 не представляется продуктивным согласно [13]. С другой стороны для поисковых запросов с длинной выше среднего значения в три слова пороговая полнота будет иметь значительное влияние на качество поиска при сравнении с лексическими методами извлечения информации о продуктах и кодерами на основе разряженных данных.

III. Методы сравнения продуктов

Большинство моделей для поиска информации изучались в однородных и узких условиях, что значительно ограничивало понимание их возможностей обобщения на новые данные. Чтобы решить эту проблему и помочь исследователям в широкой оценке эффективности своих моделей в [13] был разработан инструмент оценки Benchmarking-IR (BEIR). Инструмент BEIR позволяет оценивать пороговые значения показателей Полноты (Recall@k, R@k) и Средней точности (mAP@k) в соответствии с алгоритмами, реализованными в [14]. Но в методике, изложенной в [13] не фиксируется как сравнивать продукты в условиях мультимодальности. Например, если совпало название продукта или картинка с изображением продукта, то этого достаточно или нет? Другими словами для вычисления показателей необходимо определить функцию совпадения двух продуктов. К сожалении в академических исследованиях редко уделяется внимание этому крайне важному аспекту оценки результатов поиска.

Научно-исследовательский вопрос данной статьи состоит в том, чтобы определить наилучшую по совокупности показателей модель для работы с данными из журналов пользовательских поисковых запросов с целью эффективного продуктового поиска.

В результате проведения данного исследования получены следующие результаты:
1. Определили архитектуру и конфигурацию модели кодера, дающую наибольшую пороговую полноту.
2. Определили сильные и слабые месте нескольких наиболее современных нейросетевых архитектур в применении к задаче увеличения пороговой полноты в продуктовом поиске.
3. Экспериментально определено пороговое количество токенов в поисковом запросе, при котором лексические методы становятся менее эффективны, чем нейросетевые..

IV. Контекстно-зависимые векторные представления токенов

Одним из преимуществ, выявленных в исследованиях ELMO и BERT является возможность получения контекстно-зависимых векторных представлений токенов. Другими словами, например, токены *кондиционер, замок, молния* будут представлен различными векторами в зависимости от того в каком предложении они использованы. В моделях дистрибутивной семантики fastText, glove и моделях DSSM считается, что токены имеют единственное векторное представление вне зависимости от возможного наличия разных контекстов употребления.
Рассмотрим применение контекстно-зависимых векторных представлений токенов в задаче поиска по продуктам. Пусть $T^q = t_i^q, i \in [1, N]$ – это токены поискового запроса, а $T^p = t_j^p, j \in [1, K]$ - это токены текстового представления продукта, например, название продукта и его описание. В случае модели BERT, позволяющей создавать контекстно-зависимые векторные представления токенов, векторное представление токена $BERT(t_i^q)$ поискового запроса будет зависеть от контекстаT^q , то есть от других токенов поискового запроса. Вектор $BERT(T^q)$ вычисляется как векторное представление на специальном токене CLS через усреднённое взвешивание всех токенов $FF(Attention(T^q))$.

В случае модели fastText, для которой токены имеют единственное векторное представление, независящее от контекста, можем записать векторное представление токена в виде $fastText(t_i^q)$, а для всего поискового запроса $fastText(T^q)$, где $fastText(T^q)$ вычисляется по формуле (3).

$$fastText(T^q) = \frac{\sum_{i \in N} fastText(t_i^q)}{N} \quad (3)$$

Приведём пример для поискового запроса T^q = «дешёвый кондиционер» N = 2 и продукта T^p = «недорогой кондиционер» K=2. В таком случае лексические модели поиска, такие как BM25, будут иметь низкую полноту, так как не смогут обойтись без информации о синонимах: «дешёвый», «недорогой». Векторные модели BERT и fastText лучше справятся с такой задачей, если в обучающей выборке были представлены такие примеры. Для модели fastText(«дешёвый кондиционер») получим средний вектор от двух токенов fastText(«дешевый») и fastText(«кондиционер»). В случае BERT(«дешёвый кондиционер») получим вектор этого предложения на токене CLS.

Для задачи поиска нам важно найти продукты, векторные представления которых наиболее близки векторному представлению поискового запроса по косинусной метрике:

$$\text{argmax (cos (fastText}(T^q), \text{fastText}(T^p)) \quad (1)$$
$$\text{argmax (cos (BERT}(T^q), \text{BERT}(T^p)) \quad (2)$$

Формулы 1-2 показывают, что в задаче продуктового поиска не используется явное векторное представление отдельных токенов как поискового запроса, так и текстового представления продукта. Поэтому наличие в модели BERT возможности контекстно-зависимого векторного представления токенов не даёт такого явного преимущества, как в задачах классификации токенов и определения частей речи. Как видно из приведенного примера, полисемия в продуктовом поиске имеет ограниченное влияние на результат. Таким образом, в задаче поиска по продуктам модели с контекстно-независимым представлением токенов, например, fastText и модели с контекстно-зависимым представлением токенов, такие как BERT, не обладают явными методически обоснованными преимуществами. Этот факт служит основанием для проведения дальнейшего изучения.

V. Функции потерь

Важным преимуществом модели BERT является возможность дообучения на доменные задачи, например на задачу семантического сходства текстов (CCT). Первичное обучение языковой модели производится с помощью методики маскирования токенов MLM, а в дальнейшем используют методики предсказания следующего фрагмента текста NSP [17], DOC-SENTENCES[18], SOP[19] для дообучения.

В исследовании [17] обучение BERT, как модели языка, производится на задаче классификации токенов и текстов с помощью функций потерь на основе перекрёстной энтропии. Другой подход, продемонстрирован в исследовании [15], когда обучение модели BERT сразу выполняется с помощью контрастной функции потерь (contrastive learning) для задачи CCT. В результате исследователи [15] пишут следующее «E5 - первая модель, которая превзошла базовую версию BM25 без использования каких-либо размеченных данных». Это утверждение нуждается в пояснении, так как имеется в виду только разметка асессорами.

Важно отметить, что BM25 не является моделью машинного обучения, оперирует токенами без плотного векторного представления и имеет невысокие значения показателя полноты поиска, так как не использует информацию о семантике текста.

В моделях дистрибутивной семантики fastText, glove, word2vec используют функции потерь на основе векторного произведения, а не косинусной метрики, по которой оценивается близость поискового запроса и текстового представления продукта в задаче продуктового поиска.

Представляется рациональным оценивать модель с помощью той метрики, которая в дальнейшем будет участвовать в применении этой модели. Для MLM [17] и NSP [17] и SOP[19], которые обучаются на перекрёстной энтропии, такой метрикой является точность классификации, например F1-score.

В задаче продуктового поиска косинусная близость используется для поиска продуктов соответствующих поисковому запросу, поэтому при дообучении моделей кодеров предлагается изучить косинусную близость на основании эмбеддингов в качестве метрики для оценки и функции потерь.

Обучение языковой модели BERT с помощью маскирования токенов (MLM) происходит медленнее из-за правила маскирования токенов с вероятностью 15%. В случае текстов с длинной 512 токенов более такой подход оправдан. Но в случае поисковых запросов, длинна которых в среднем 2.5 токена и не более 15 токенов, MLM значительно теряет свою эффективность. Например, для поискового запроса «черная юбка», если будет замаскирован токен «юбка», то на основании токена «черный» сложно однозначно предсказать, что имеется ввиду. Поэтому на коротких поисковых запросах не целесообразно использовать MLM, и даже на поисковых запросах средней длинны маскирование более одного токена может снижать эффективность обучения. Подход к обучению BERT как языковой модели у четом особенностей продуктового поиска назван автором CTX.

VI. Выбор архитектуры кодера

Со времени первой статьи [17] архитектура и способы обучения моделей с использованием трансформеров претерпели ряд изменений. Поэтому с настоящем исследовании были рассмотрены модели BERT [17], RoBERTa[18], ALBERT[19], DeBERTa[20]. Наиболее подходящей с методической точки зрения является DeBERTa из-за механизм внимания более нацеленного

на относительные позиции токенов, а не на абсолютные как в других архитектурах. Основным преимуществом ALBERT авторы выделяют уменьшение количества параметров за счёт разделения размерности таблицы эмбеддингов от измерения скрытых слоёв. Данное преимущество ALBERT не представляет большой ценности для решения исследовательской задачи настоящей статьи. RoBERTa по мнению авторов является подходом к конфигурации и способу обучения, а не самостоятельной архитектурой модели.

VII. Данные

Для обучения моделей было создано три набора данных: 1. Журнал поисковых запросов за 11 месяцев – D_Q. 2. Журнал покупок и помещений продукта в корзину за 3 месяца – D_{QP}. 3. Размеченный асессорами набор пар «поисковый запрос» и «текстовое представление продукта» с метками соответствует или нет – D_{QPL}. Наборы данных обладают следующими размерностями:D_Q = 265 млн. строк, D_{QP} = 29 млн. строк, D_{QPL} = 339 тысяч пар.

VIII. Эксперимент

По трем наборам данных было построено распределение частот количества токенов в поисковых запросах (Рис.1).

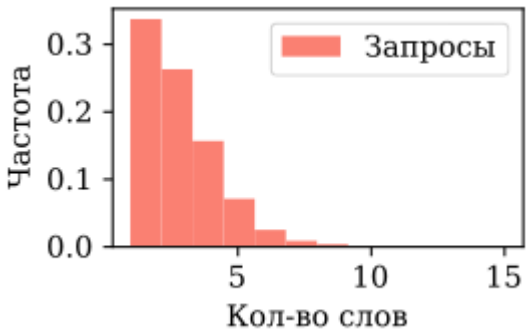

Рисунок 1: Распределение количества слов в поисковых запросах

Распределение на Рис. 1. подтверждает, что поисковые запросы в среднем состоят из 2-3 слов и не превышают 11 слов. Разница в словах и токенах не существенна, хотя для поисковых запросов с опечатками количество токенов становится больше чем количество слов, разделенных пробелом для SentencePiece токенизации.

IX. Обучение моделей

В соответствии с выбранными архитектурами и способами обучения было произведено обучение моделей (Таблица 2.).

Таблица 2-- Параметры моделей кодеров

Название модели	Детали обучения	Размер, млн.	Тип
fastText	Обучение произведено на наборе данных D_Q с размерностью векторов 64, окном контекста 21 и суб-словарными n-граммами от 3 до 6 букв.	147	CPU
fastText_wiki	Взята обученная модель [22]	600	CPU
BERT_MLM_NSP	Создана модель токенизации SentencePiece с размером словаря 32 тыс токенов, обученная на наборе данных D_Q. Создана модель BERT с конфигурацией: hidden_size = 64 , intermediate_size = 256, num_attention_heads = 2, num_hidden_layers = 8. Обучение по методике MLM на наборе данных D_Q. Обучение по методике NSP на наборе данных D_{QP}. Финальная метрика F1-score 0.98	4.2	GPU
sBERT	Взята обученная модель ai-forever/sbert_large_nlu_ru [21]	427	GPU
ALBERT	Взята обученная модель albert/albert-base-v2 [19]	11.8	GPU
e5_CLS	Взята обученная модель intfloat/multilingual-e5-small [15]	117	GPU
e5_LARGE_CLS	Взята обученная модель intfloat/multilingual-e5-large [15]	335	GPU
electra	Взята обученная модель ai-forever/ruElectra-small [21]	42	GPU
DE_BERT	Создана модель токенизации SentencePiece с размером словаря 32 тыс токенов, обученная на наборе данных D_Q. Создана модель DeBERTa с конфигурацией: hidden_size = 64 , intermediate_size = 256, num_attention_heads = 8, num_hidden_layers = 8. Обучение DeBERTa как языковой модели по методике CTX на наборе данных D_Q. Обучение по методике DualEncoder [16] на наборе данных D_{QP} с косинусной функции потерь. Финальная метрика F1-score 0.99	9.0	CPU/GPU

X. Оценка эффективности

Для оценки эффективности обученных моделей использована метрика пороговой полноты по методике

описанной в работе [16]. Оценка произведена на наборе данных D_{QPL}.

Зависимость пороговой полноты от количества слов в запросе проанализирована для двух значений порога k равного 100 и 1000. Результаты анализа представлены на Рис 3. и 4. соответственно. Лучше всех себя показала модель E5. Для k=100 у модели e5_CLS для всех длин поисковых запросов полнота выше чем у других моделей. Следующей идет модель fastText и DE_BERT обученные «с нуля» в отличие от предобученной модели E5. Модель DE_BERT обучалась аналогично E5 на функции потерь, отражающей ее дальнейшее применение.

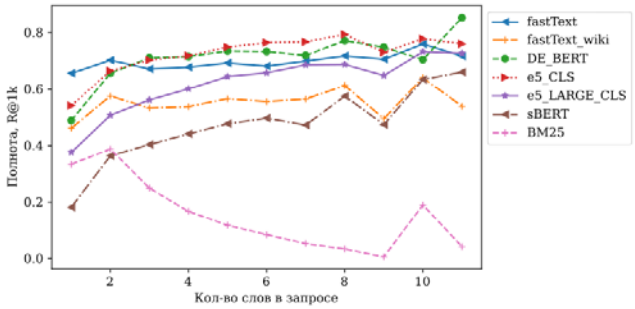

Рисунок 2-- Зависимость полноты поисковой выдачи для порога 1000 от длинны поискового запроса

Сравнивая поведение зависимостей на Рис.2 для лексической модели BM25 и sBERT, можно отметить, что они одинаково неэффективны на коротких поисковых запросах, но sBERT имеет значительно преимущество на длинных поисковых запросах. А полнота поисковой выдачи лексической модели BM25 падает с ростом длинны поискового запроса. Уже при трех словах в поисковом запросе все векторные модели показывают более высокую пороговую полноту, чем лексическая модель BM25.

Отдельно отметим, что предобученные модели ALBERT, electra, sBERT показали значительно более низкие значения пороговой полноты для всех длин поисковых запросов. Таким образом, количество параметров модели и размер данных для обучения не являются преимуществом (Рис. 3) для решения увеличения полноты продуктового поиска.

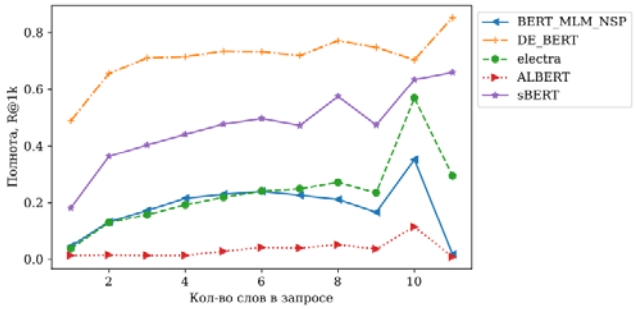

Рисунок 3-- Зависимость полноты поисковой выдачи для порога 1000 от длинны поискового запроса

В таблице 3 представлены результаты измерения полноты поисковой выдачи в зависимости от модели для кодера. Для увеличения полноты продуктового поиска наибольшую важность представляет значение полноты при пороге 1000. Меньшие значения порога более подходят для оценки точности поисковой выдачи. Из таблицы 3 видно, что наилучшие значения показывает обученная на общих данных модель E5 с 117 млн. параметров. На 0.007 от E5 отстает модель fastText, обученная на данных D_Q с 147 млн параметров, которые состоят 2.3 млн. контекстно-независимых эмбеддингов с размерностью 64. И на 0.009 от E5 отстает модель DE_BERT, созданная и обученная на данных D_{QP} с 9.0 млн параметров, из которых 32 тысячи составляют эмбеддинги с размерностью 64.

Таблица 3: Зависимость средней по всем запросам пороговой полноты поисковой выдачи от модели кодера

Название	**R@10**	**R@50**	**R@100**	**R@200**	**R@300**	**R@500**	**R@1k**
fastText	0.129	0.289	0.374	0.466	0.522	0.592	**0.683**
fastText_wiki	0.127	0.263	0.329	0.395	0.433	0.480	0.543
BERT_MLM_NSP	0.042	0.073	0.089	0.108	0.121	0.139	0.168
DE_BERT	0.110	0.274	0.366	0.463	0.521	0.592	**0.681**
electra	0.024	0.046	0.062	0.082	0.097	0.119	0.158
ALBERT	0.003	0.005	0.006	0.008	0.010	0.012	0.018
e5_CLS	**0.157**	**0.341**	**0.427**	**0.513**	**0.561**	**0.619**	**0.690**
e5_LARGE_CLS	0.124	0.253	0.319	0.389	0.430	0.482	0.553
sBERT	0.061	0.136	0.181	0.235	0.271	0.320	0.393
BM25	0.075	0.163	0.193	0.217	0.229	0.241	0.253

Высокий результат модели DE_BERT обусловлен тем, что модель обучалась на парах поисковый запрос и текстовое представление продукта с косинусной функцией потерь.

XI. Заключение

В статье проведено исследование моделей для кодеров в задаче продуктового поиска способствующих увеличению метрики пороговой полноты. Получен ответ на исследовательский вопрос о том, какая модель для работы с данными из журналов пользовательских поисковых запросов с целью эффективного продуктового поиска является наилучшей.

В случае создания прототипа или офлайн модели с возможностью запуска на GPU лидером стала модель E5_SMALL. Объяснением является способ обучения модели максимально соответствующий ее использованию в качестве кодера. Так же было выяснено, что специфика поисковых запросов не позволяет улучшать показатели пороговой полноты с помощью увеличения размеров модели. В эксперименте модель E5_LARGE показала себя хуже, чем модель E5_SMALL для всех значений порога.

Наиболее универсальной, эффективной по ресурсам и значением показателей пороговой полноты показала

себя модель DE_BERT для запросов более 10 слов с метрикой R@1k = 0.845.

Библиография

[1] Muennighoff N. et al. MTEB: Massive text embedding benchmark //arXiv preprint arXiv:2210.07316. – 2022.

[2] Snegirev A. et al. The Russian-focused embedders' exploration: ruMTEB benchmark and Russian embedding model design //arXiv preprint arXiv:2408.12503. – 2024.

[3] Ellen Voorhees. 2004. Overview of the TREC 2004 Robust Retrieval Track. In TREC.

[4] Nguyen, Tri, et al. "MS MARCO: A human generated machine reading comprehension dataset." choice 2640 (2016): 660.

[5] Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 18 Million Clicked Query-Document Pairs for Analyzing Search. arXiv preprint arXiv:2006.05324 (2020).

[6] Reddy C. K. et al. Shopping queries dataset: A large-scale ESCI benchmark for improving product search //arXiv preprint arXiv:2206.06588. – 2022.

[7] Chen Y. et al. Wands: Dataset for product search relevance assessment //European Conference on Information Retrieval. – Cham: Springer International Publishing, 2022. – С. 128-141.

[8] Choi, J.I., Kallumadi, S., Mitra, B., Agichtein, E., Javed, F.: Semantic product search for matching structured product catalogs in e-commerce. In: https://arxiv.org/pdf/2008.08180.pdf (2020)

[9] Crowdflower search results relevance. https://www.kaggle.com/c/crowdflower-search-relevance/overview

[10] Wand An., Normann Ph., Baumeister S., Wilm T., Reade W., Demkin M. OTTO Recommender Systems Dataset: A real-world e-commerce dataset for session-based recommender systems research (2022).

[11] Carlos García Ling, Elizabeth HMGroup, Frida Rim, inversion, Jaime Ferrando, Maggie, neuraloverflow, xlsrln // H&M Personalized Fashion Recommendations. Kaggle. https://kaggle.com/competitions/h-and-m-personalized-fashion-recommendations, 2022.

[12] Краснов Ф. В. Управление разнообразием товаров в рекомендательных моделях на основе архитектуры с механизмом внимания (трансформерах) //International Journal of Open Information Technologies. – 2024. – Т. 12. – №. 1. – С. 68-75.

[13] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv preprint arXiv:2104.08663 (4 2021). https://arxiv.org/abs/2104.08663

[14] Voorhees E. M. et al. (ed.). TREC: Experiment and evaluation in information retrieval. – Cambridge : MIT press, 2005. – Т. 63.

[15] Wang L. et al. Multilingual e5 text embeddings: A technical report //arXiv preprint arXiv:2402.05672. – 2024.

[16] Краснов Ф.В. Пороговые показатели полноты и точности для оценки системы извлечения информации о товарах на основе эмбеддингов // Бизнес-информатика. 2024. Т. 18. № 2. С. 22–34. DOI: 10.17323/2587-814X.2024.2.22.34

[17] Devlin, Jacob et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. // North American Chapter of the Association for Computational Linguistics – 2019.

[18] Liu, Yinhan et al. : RoBERTa: A Robustly Optimized BERT Pretraining Approach.// ArXiv abs/1907.11692 – 2019.

[19] Lan Z.: ALBERT. A lite BERT for self-supervised learning of language representations //arXiv preprint arXiv:1909.11942. – 2019.

[20] He, Pengcheng, et al. "Deberta: Decoding-enhanced bert with disentangled attention." arXiv preprint arXiv:2006.03654 (2020).

[21] Zmitrovich, Dmitry, et al. "A family of pretrained transformer language models for Russian." arXiv preprint arXiv:2309.10931 (2023).

[22] Bojanowski P. et al. Enriching word vectors with subword information //Transactions of the association for computational linguistics. – 2017. – Т. 5. – С. 135-146.

Product Search Recall: Comparison of Embedding Encoders

F.V. Krasnov

***Abstract* – The rapid development of neural network architectures based on the transformer model leaves little time for in-depth evaluation of their efficacy in product search applications. New research into transformers is published frequently, but there may not always be sufficient time and resources available to test their performance against existing solutions, such as those based on FastText, DSSM, or ELMO, for example. Comparisons in academic publications are typically made using public datasets that can differ significantly from those employed in industrial settings. Transitioning to new machine learning models within industrial search systems involves significant time and resource investment due to the necessity of indexing product catalogs, resulting in a high cost of error. In this study, we analyze the strengths and limitations of various models in relation to product search and conduct an experimental evaluation of different models for coders.**

A training dataset was generated based on user search logs. Models with optimal hyperparameters were trained using this dataset. Detailed definitions of autonomous metrics for measuring efficiency were provided. A validation dataset was collected for textual representations of products. Comparisons of autonomous metrics were made for the task of product search in e-commerce, and the metrics were evaluated by experts.

Pre-trained models and models trained from scratch were both considered for coders. When comparing models, the values of the recall R@k were calculated for each model for a set of search queries. It was found that the effectiveness of encoders varied depending on the number of words in the search query.

This study does not seek to develop the best possible model for solving the common challenges associated with obtaining textual representations. Instead, the authors acknowledge the limitations of their work, but given the constraints of available resources, they wish to establish a reliable methodology for selecting the most appropriate machine learning model for addressing this specific task. Therefore, the primary contribution of this research is the methodology employed for comparing various machine learning approaches within the context of the encoder component in the application of enhancing the comprehensiveness of product searches.

***Keywords* — comparison of models, performance indicators of product search, neural network architectures for search, encoders, product search**.

REFERENCES

[1] Muennighoff N. et al. MTEB: Massive text embedding benchmark //arXiv preprint arXiv:2210.07316. – 2022.

[2]Snegirev A. et al. The Russian-focused embedders' exploration: ruMTEB benchmark and Russian embedding model design //arXiv preprint arXiv:2408.12503. – 2024.

[3] Ellen Voorhees. 2004. Overview of the TREC 2004 Robust Retrieval Track. In TREC.

[4]Nguyen, Tri, et al. "MS MARCO: A human generated machine reading comprehension dataset." choice 2640 (2016): 660.

[5]Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 18 Million Clicked Query-Document Pairs for Analyzing Search. arXiv preprint arXiv:2006.05324 (2020).

[6]Reddy C. K. et al. Shopping queries dataset: A large-scale ESCI benchmark for improving product search //arXiv preprint arXiv:2206.06588. – 2022.

[7]Chen Y. et al. Wands: Dataset for product search relevance assessment //European Conference on Information Retrieval. – Cham: Springer International Publishing, 2022. – S. 128-141.

[8]Choi, J.I., Kallumadi, S., Mitra, B., Agichtein, E., Javed, F.: Semantic product search for matching structured product catalogs in e-commerce. In: https://arxiv.org/pdf/2008.08180.pdf (2020)

[9]Crowdflower search results relevance. https://www.kaggle.com/c/crowdflower-search-relevance/overview

[10] Wand An., Normann Ph., Baumeister S., Wilm T., Reade W., Demkin M. OTTO Recommender Systems Dataset: A real-world e-commerce dataset for session-based recommender systems research (2022).

[11] Carlos García Ling, Elizabeth HMGroup, Frida Rim, inversion, Jaime Ferrando, Maggie, neuraloverflow, xlsrln // H&M Personalized Fashion Recommendations. Kaggle. https://kaggle.com/competitions/h-and-m-personalized-fashion-recommendations, 2022.

[12] Krasnov F. V. Upravlenie raznoobraziem tovarov v rekomendatel'nyh modeljah na osnove arhitektury s mehanizmom vnimanija (transformerah) //International Journal of Open Information Technologies. – 2024. – T. 12. – #. 1. – S. 68-75.

[13] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv preprint arXiv:2104.08663 (4 2021). https://arxiv.org/abs/2104.08663

[14] Voorhees E. M. et al. (ed.). TREC: Experiment and evaluation in information retrieval. – Cambridge : MIT press, 2005. – T. 63.

[15] Wang L. et al. Multilingual e5 text embeddings: A technical report //arXiv preprint arXiv:2402.05672. – 2024.

[16] Krasnov F.V. Porogovye pokazateli polnoty i tochnosti dlja ocenki sistemy izvlechenija informacii o tovarah na osnove jembeddingov // Biznes-informatika. 2024. T. 18. # 2. S. 22–34. DOI: 10.17323/2587-814X.2024.2.22.34

[17] Devlin, Jacob et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. // North American Chapter of the Association for Computational Linguistics – 2019.

[18] Liu, Yinhan et al. : RoBERTa: A Robustly Optimized BERT Pretraining Approach.// ArXiv abs/1907.11692 – 2019.

[19] Lan Z.: ALBERT. A lite BERT for self-supervised learning of language representations //arXiv preprint arXiv:1909.11942. – 2019.

[20] He, Pengcheng, et al. "Deberta: Decoding-enhanced bert with disentangled attention." arXiv preprint arXiv:2006.03654 (2020).

[21] Zmitrovich, Dmitry, et al. "A family of pretrained transformer language models for Russian." arXiv preprint arXiv:2309.10931 (2023).

[22] Bojanowski P. et al. Enriching word vectors with subword information //Transactions of the association for computational linguistics. – 2017. – T. 5. – S. 135-146.