Обзор алгоритмов семантического поиска по текстовым документам

Н.Д. Шалагин

***Аннотация*—Семантический поиск представляет собой современный подход к информационному поиску, основанный на понимании смысла и контекста запросов, что позволяет предлагать более релевантные результаты по сравнению с традиционными методами поиска основанными на совпадении ключевых слов. Использование технологий обработки естественного языка, таких как архитектура Transformer и большие предобученные языковые модели значительно улучшило качество семантического поиска. Данные модели продемонстрировали высокие результаты в различных бенчмарках, что привело к их широкому применению во множестве приложений.**

Основные преимущества семантического поиска включают более высокий уровень точности и релевантности результатов, улучшение пользовательского опыта и возможность свободного выражения запросов. Однако, несмотря на значительные достижения, существуют проблемы, связанные с вычислительной сложностью моделей, ограниченным размером обрабатываемого текста и временем отклика в режиме реального времени. В решениях требующих обработку пользовательских запросов в около-реальном времени разработчики часто вынуждены применять менее ресурсоемкие решения, что может снижать качество поиска. Частой дилеммой, встающей перед специалистом делающим реальное приложение, является компромисс между вычислительными затратами, скоростью работы и качеством поиска.

Данная работа направлена на обзор текущих методов семантического поиска и их сравнительный анализ. Особое внимание уделяется изучению преимуществ и недостатков различных подходов, а также анализу перспектив их дальнейшего развития и применения в различных областях.

***Ключевые слова*—семантический поиск, поиск по текстовым документам, обработка естественного языка, NLP**

I. Введение

Семантический поиск — это подход к поиску информации, основанный на понимании смысла и контекста поискового запроса, а не просто на обнаружении совпадений по ключевым словам. Данный подход использует технологии обработки естественного языка, чтобы понять намерения пользователя и предложить наиболее релевантные результаты.

Семантический поиск имеет несколько ключевых преимуществ по сравнению с традиционными методами поиска. Прежде всего, он позволяет достигать более высокого уровня точности и релевантности результатов поиска, поскольку учитывает контекст запроса и взаимосвязь между словами. Такой подход обеспечивает более глубокое понимание запроса, что позволяет предоставлять более релевантные результаты.

Кроме того, семантический поиск способствует улучшению пользовательского опыта, так как он позволяет пользователям более естественным образом формулировать свои запросы. Они не обязаны использовать определенные ключевые слова или фразы и могут свободно выражать свои запросы, как если бы они общались с человеком.

Чем быстрее и точнее сервис может отвечать на информационные запросы, тем выше будет уровень удовлетворенности пользователей. Семантический поиск применим во многих сферах: интернет-магазины, электронные библиотеки, онлайн-кинотеатры и многие другие сервисы, где объекты поиска имеют описание на естественном языке.

Усовершенствование технологий обработки естественного языка, в частности появление архитектуры Transformer [1] и больших генеративых предобученных языковых моделей на ее основе [2][3][4][5] дало значительный толчок к развитию семантического поиска. Как демонстрирует ряд бенчмарков[6][7][8][9][10], данные модели способны качественно извлекать смысл из текста и понимать его контекст.

Семантический поиск на основе вышеописанных моделей установил новый SOTA-уровень качества, и был внедрен многими крупными компаниями, такими как Google[11], Facebook[12], Huawei[13]. Также, семантический поиск необходим для решения ряда более узких задач, среди которых можно выделить вопросно-ответные ситемы [14] и генеративные модели с памятью [15].

Однако, несмотря на значительное увеличение качества, системы реализующие семантический поиск часто сталкиваются с целым рядом проблем, обусловленными большой вычислительной сложностью исходных моделей. Среди данных проблем можно выделить ограниченный размер текста для обработки и время отклика при функционировании в режиме реального времени, а также надежность инфраструктуры обеспечивающей данные вычисления. В связи с этим, компании как правило внедряют менее ресурсоемкие решение с меньшим качеством поиска [12][16].

Целью данной работы является обзор существующих методов семантического поиска и их сравнительный анализ.

II. Метрики оценки качества поиска

Для оценки качества поиска могут применяться следующие метрики:

1) Recall@N:

$$\text{Recall@N} = \frac{|\text{Relevant Items} \cap \text{Retrieved Items@N}|}{|\text{Relevant Items}|}$$

Recall@N определяет полноту результатов поиска. Данная метрика часто используется для оценки

Статья получена 14 августа 2024.

Никита Дмитриевич Шалагин, МГУ им. М.В. Ломоносова, (email: shalaginnd@my.msu.ru).

качества подсистем поиска чья роль сводится к отбору кандидатов для последующего ранжирования.
Преимущества:

- Интуитивно понятная и простая в вычислении.
- Дает понимание какая доля результатов упущена поисковым алгоритмом.

Недостатки:

- Не учитывает порядок документов среди первых N.
- Не учитывает релевантность документов, которые находятся ниже N.

2) Precision hit@N:

$$Precision@N = \frac{|Relevant\ Documents\ in\ Top\ N|}{N}$$

Precision hit@N определяет, сколько из N верхних документов являются релевантными.
Преимущества:

- Интуитивно понятная и простая в вычислении.

Недостатки:

- Не учитывает порядок документов среди первых N.
- Не учитывает релевантность документов, которые находятся ниже N.

3) Discount Cumulative Gain (DCG):

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

где rel_i - релевантность i-го документа, а p - количество рассматриваемых документов.
Преимущества:

- Учитывает порядок документов.
- Учитывает релевантность документов.

Недостатки:

- Не нормирована, то есть не удобна для сравнения результатов на разных поисковых выборках.

4) Normalized DCG (nDCG):

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

где $IDCG_p$ - идеальный DCG, который представляет собой максимально возможное значение DCG для данного набора документов.
Преимущества:

- Нормирована, то есть удобна для сравнения результатов на разных наборах данных.
- Учитывает порядок документов.
- Учитывает релевантность документов.

Недостатки:

- Может быть сложнее в вычислении, поскольку требует знания идеального DCG.

5) Mean Average Precision (MAP):

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{M_q} \sum_{k=1}^{n} Precision@k \cdot rel(k)$$

где Q - количество запросов, M_q - количество релевантных документов для q-го запроса, n - количество документов в ответе на запрос, а $rel(k)$ - индикаторная функция, которая равна 1, если k-й документ является релевантным, и 0 в противном случае.
Преимущества:

- Учитывает и порядок документов, и их релевантность.
- Сводит результаты по всем запросам в одно число, что удобно для сравнения.

Недостатки:

- Не учитывает относительную важность различных релевантных документов (все релевантные документы считаются равно важными).

6) Mean Reciprocal Rank (MRR):

$$MRR = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{\text{rank}_q}$$

где Q - количество запросов, а rank_q - позиция первого релевантного документа в списке результатов для q-го запроса.
Преимущества:

- Фокусируется на наиболее релевантных результатах (то есть на тех, которые находятся в верхней части списка).
- Простая в вычислении.

Недостатки:

- Не учитывает релевантные документы, которые находятся ниже первого релевантного в списке.
- Не учитывает общее количество релевантных документов в списке.

Важным свойством, объединияющим все вышеупомянутые метрики, является тот факт, что для их подсчета необходима человеческая оценка релевантности.

III. Наборы данных для оценки качества поиска и обучения моделей

Для оценки качества и обучения моделей существует ряд бенчмарков и наборов данных. Их можно категоризировать по нескольким параметрам:

1) Применимые метрики. Различные наборы данных имеют различный формат разметки и потому к ним могут применимы не все метрики. Так например большинство датасетов не имеют полной разметки релевантности между всеми запросами и всеми документами в силу квадратичной сложности такой операции. Данный метод делает затруднительным качественную оценку по метрикам наподобие nDCG, требующими оценку релевантности для любого наперед заданного запроса и подмножества документов.
2) Источник данных. Все датасеты можно разделить на естественные - собранные на основе поведения реальных пользователей в существующий поисковых системах и курируемые - размеченные специальной группой людей на заранее выбранном наборе документов.
3) Тип данных. Многие датасеты применяющиеся для задачи поиска имеют в себе несколько отличающиеся от реальных сценариев данные. Так, вместо

стандартных пар короткий запрос - длинный документ, некоторые из них состоят из пар вопрос-вопрос-метка синонимичности или пар вопрос-ответ, что в свою очередь при обучении и оценке качества несколько смещает исходный домен задачи.

4) Язык. Разные датасеты содержат разные наборы языков, что может сказываться на качестве обучения и итоговой оценке

В Таблице I организован список наиболее часто встречающихся наборов данных для задачи поиска.

IV. Обзор существующих методов

A. Полнотекстовый поиск

Полнотекстовый поиск - это метод поиска, который позволяет найти все документы, содержащие определенные слова или фразы. Он является основой поисковых систем и баз данных. Далее представлены основные методы полнотекстового поиска:

1) Поиск на основе инвертированного индекса: этот метод является одним из самых эффективных для полнотекстового поиска. Инвертированный индекс - это структура данных, которая отображает каждое слово на список документов, в которых это слово встречается. Это делает поиск очень быстрым, так как достаточно просмотреть индекс по слову, а не все документы.
2) TF-IDF (Term Frequency-Inverse Document Frequency) - это статистический метод для оценки важности слова в документе, основанный на количестве раз, когда слово встречается в документе, скорректированном на частоту появления слова во всех документах. Это позволяет системе выделять документы, которые наиболее релевантны запросу.
3) BM25: Это улучшение TF-IDF, которое также учитывает длину документа и эффект насыщения терминов. BM25 широко используется в современных поисковых системах. BM25 (Best Matching 25) - это алгоритм ранжирования, используемый поисковыми системами для улучшения точности полнотекстового поиска. Он был разработан в 1990-х годах и представляет собой усовершенствование классического алгоритма TF-IDF.

BM25[17] BM25 использует модель вероятностного ранжирования, которая учитывает терминологические веса и длину документа для ранжирования документов по отношению к запросу. Специфически, BM25 модифицирует TF (частоту терминов) компонент TF-IDF, чтобы учесть эффект насыщения - когда повышение частоты термина в документе в меньшей степени влияет на релевантность, если частота уже достаточно высока.

$$\text{BM25}(q,d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,d)\cdot(k_1+1)}{f(q_i,d)+k_1\cdot(1-b+b\cdot\frac{|d|}{\text{avgdl}})} \quad (1)$$

,где
q - запрос,
d - документ,
n - количество терминов в запросе q,
q_i - i-й термин запроса q,

$f(q_i,d)$ - частота термина q_i в документе d,
k_1 - параметр регулирования влияния частоты термина (обычно $1.2 \leq k_1 \leq 2.0$),
b - параметр регулирования влияния длины документа (обычно $0.5 \leq b \leq 1.0$),
$|d|$ - длина документа d (количество слов),
avgdl - средняя длина документа в коллекции.

$\text{IDF}(q_i)$ - обратная частота документов для термина q_i,

$$\text{IDF}(t) = \log \frac{N - n(t) + 0.5}{n(t) + 0.5} \quad (2)$$

,где
t - термин,
N - общее количество документов в коллекции,
$n(t)$ - количество документов в коллекции, содержащих термин t.

Преимущества BM25:

1) BM25 учитывает длину документа и обратную частоту документа, что помогает улучшить релевантность результатов.
2) BM25 лучше, чем TF-IDF, когда дело доходит до обработки длинных документов, которые могут содержать много нерелевантной информации.
3) BM25 учитывает насыщение термов, что улучшает его способность ранжировать документы.

Недостатки BM25:

1) BM25 не учитывает семантическую связь между словами, что может привести к пропуску релевантных документов, которые используют синонимы или связанные термины.
2) BM25 предполагает независимость слов в запросе, что может быть неправильным в случае фраз или запросов, где порядок слов важен.

Улучшения BM25:

1) Добавление потенциальных запросов к индексу[18]. В современный исследованиях встречается прием, при котором с помощью языковых моделей по документу создается ряд потенциальных запросов, и затем полученные запросы индексируются вместе с исходным содержимым документа.
2) Модификация запросов с помощью языковых моделей [19] - запрос пользователя модифицируется языковой моделью, и по измененному запросу осуществляется поиск
3) Нормализация [20]. При данном подходе в индекс включаются исключительно начальные формы слов (инфинитивы), а при поиске все слова в пользовательском запросе также приводятся к начальной форме. Данный подход особенно сильно сказывается на обработке языков, пользующихся порождающими механизмами, такими как агглютинация и компаундирование (например, русский и немецкий языки). Так, например, без нормализации, слова ”яблоко”и ”яблоки несмотря на очевидное обозначение одного предмета, будут восприняты алгоритмом как разные, никак не связанные

термины. Также нормализация включает в себя перевод численных обозначений в словесную форму (или наоборот).

4) Словари синонимов. Данный подход позволяет учитывать особенность естественного языка, что один и тот же предмет может обозначаться разными словами. При добавлении словаря синонимов, при подсчете релевантности, за совпадение термов также считается совпадение терма запроса с синонимом терма из документа.

Несмотря на эти недостатки, BM25 остается широко используемым алгоритмом ранжирования из-за своей простоты и эффективности во многих сценариях поиска.

B. Поиск в векторном пространстве

Данный подход опирается на концепт изучения представлений[21]. Идея семантического поиска по векторному пространству заключается в том, чтобы каждому документу в коллекции присвоить свое вектор-представление. Во время поиска запрос встраивается в то же векторное пространство, ищутся топ-k ближайших соседей из индекса. Эти записи должны иметь высокое семантическое совпадение с запросом.

Основные этапы векторного семантического поиска включают:

1) Векторизация: документы и преобразуются в векторы фиксированной размерности с использованием моделей, предобученных обученных на больших наборах данных, и далее настроенных на генерацию векторов-представлений (эмбеддингов) чья близость отражает семантическую схожесть между исходными отрывками текста. Примеры таких моделей включают BERT[2], coCondenser[22], RetroMAE[23], CoTMAE[24]
2) Индексация полученных векторов. При поиске по большим корпусам документов, как правило, не используется наивный подход, подразумевающий вычисление N метрик близости для корпуса из N документов на каждый запрос. Как правило используются иерархические структуры, такие как [25] и и различные квантизации исходных эмбедингов [26]
3) Поиск ближайших соседей: вектор-представление запроса сравнивается с эмбеддингами документов в коллекции, чтобы найти наиболее близкие или похожие документы. Для оценки близости векторов представлений, как правило, используются классические метрики: евклидово расстояние, косинусная метрика или скалярное произведение. Документы с наивысшей близостью принимаются наиболее релевантными и возвращаются пользователю.

Достоинства:

1) Возможность быстро осуществлять поиск по большому корпусу данных.
2) Учитывается семантика содержимого документа, из текста извлекаются более высокоуровневые концепции

Недостатки:

1) Низкая чувствительность к ключевым словам. Как правило, эмбеддинги отражают общий смысл исходной фразы и плохо передают имена, топонимы и прочие специфические слова.

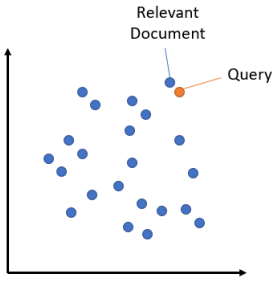

Рис. 1. Отбор результатов в векторном пространстве построенном bi-encoder моделью[27]

2) Качество поиска ниже, чем при попарной оценке релевантности.
3) С увеличением длины исходного документа снижается репрезентативность эмбеддинга. Иными словами, чем больше контекст для построения представления, тем более общие концепции будут передаваться итоговым вектором.

C. Попарная оценка релевантности

Существуют различные подходы к попарной оценке релевантности, и одним из наиболее эффективных является использование Cross-encoder[28]. Cross-encoder - это модель машинного обучения, которая обрабатывает пару элементов (запрос и документ) как единую сущность и выдаёт оценку релевантности. Стоит отметить, что использование Cross-encoder в семантическом поиске требует больших вычислительных ресурсов.

Процесс поиска при использовании Cross-encoder выглядит следующим образом:

1) Входные данные (пары запрос-документ) подаются на вход Cross-encoder.
2) Cross-encoder обрабатывает пары и выдаёт оценку релевантности для каждой пары.
3) Эти оценки затем используются для ранжирования документов по их релевантности к запросу.

Основное преимущество использования Cross-encoder вместе с попарной оценкой релевантности заключается в том, что он может учитывать контекст и семантику в обоих элементах пары без потери информации, как это происходит при построении эмбеддингов. Это может значительно улучшить качество ранжирования и точность поиска.

В дальнейшем, оригинальная идея использовать BERT[2] в качестве модели для попарной оценки релевантности получила развитие в следующих моделях: MaxP [29], CEDR [30], Birch [31], PARADE [32].

D. Retrieve& Re-rank

Подход Retrieve & Re-rank используется в семантическом поиске для улучшения качества результатов по сравнению с векторным поиском и для снижения вычислительных затрат по сравнению с попарной оценкой релевантности. Он состоит из двух основных этапов:

1) Retrieve: на данном этапе система быстро сканирует огромное количество документов или записей в поисках тех, которые могут быть релевантны запросу пользователя. Здесь обычно используются более

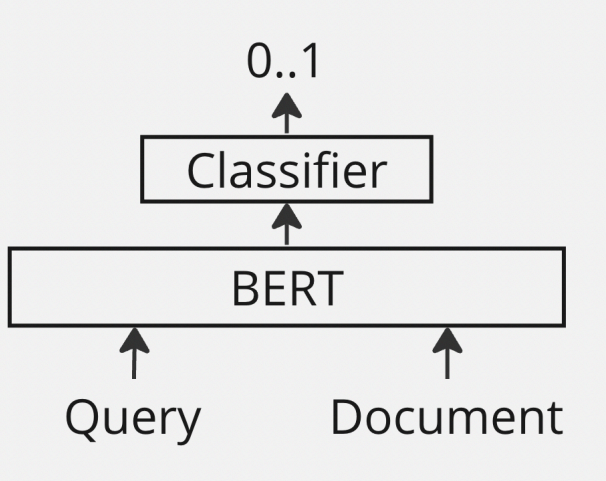

Рис. 2. Процесс оценки семантической близости двух документов с помощью модели Cross-encoder[27]

простые и быстрые алгоритмы или методы, такие как обратный индекс, BM25 или поиск ближайших соседей в векторном пространстве, чтобы уменьшить исходный объем данных до более управляемого числа потенциально релевантных документов.

2) Re-rank: после извлечения потенциально релевантных документов система использует более сложные алгоритмы или модели, обычно основанные на глубоком обучении, чтобы повторно ранжировать эти документы. Эти модели могут оценивать не только совпадение ключевых слов между запросом и документами, но и глубину семантического смысла, обеспечивая более точные и релевантные результаты.

Данный подход сочетает в себе быстроту и эффективность простых методов извлечения с точностью и глубиной анализа более сложных моделей машинного обучения, являясь компромиссом между качеством поиска и вычислительной сложностью.

В дальнейшем, данный метод получил развитие в виде механизма HLATR[13], применяющего трансформерные слои к промежуточным представлениям из модели Cross-Encoder, что позволило получить лучший на момент написания работы результат на наборе данных MS MARCO[33]

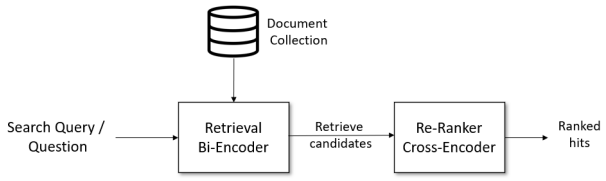

Рис. 3. Общая схема процесса Retrieve&re-rank[27]

E. Transformer как дифференцируемый поисковый индекс

Предложенная модель Differentiable Search Index (DSI)[34] представляет собой полностью параметризованный подход к традиционным поиску и ранжированию, интегрированный в одну нейронную модель. В основе данного подхода лежит способность генеративных моделей архитектуры Transformer запоминать факты и данные использованные в процессе предобучения и в дальнейшем их выдавать во время генерации текста[5][35]. На рисунке 4 изображена схема подхода DSI по сравнению с традиционными подходами к поиску.

Для достижения желаемого результата модель обучают на двух задачах:

1) Индексирование: модель должна учиться связывать содержимое каждого документа с соответствующим идентификатором документа. В этой статье для этого используется простой подход sequence-to-sequence (seq2seq[36]), который принимает токены документа на входе и генерирует идентификаторы на выходе.
2) Поиск: при получении входного запроса модель DSI должна возвращать ранжированный список кандидатов-идентификаторов документов.

После этих двух операций, полученная модель DSI может индексировать корпус документов, чтобы затем использоваться для извлечения релевантных документов - все в рамках одной, единой модели.

Процесс генерации идентификаторов на инференсе не отличается от процесса генерации токенов в языковой модели: для генерации идентификатора используется beam search, а за релевантность принимается произведение оценок токенов составляющих финальный идентификатор.

Данная работа получила развитие в двух направлениях:

1) Neural Corpus Indexer [37] - Внесены изменения в архитектуру, добавлен PAWA Encoder, позволяющий качественнее генерировать идентификаторы, токены на каждом иерархическом уровне сделаны уникальными (в оригинальной работе они переиспользовались)
2) Listwise Generative Retrieval [38] - К функции потерь добавлена часть учитывающая весь список документов и позволяющая различать релевантность нескольких документов относящихся к запрос.

Наиболее значительным недостатком данного подхода для поиска является непостоянство данных при обновлении индекса, также описываемое в литературе как ”критическое забывание”[39] - явление, при котором в процессе добавления к индексу новых данных посредством частичного повторения обучения теряются накопленные знания о некоторой части старых документов.

Также подобные модели сильно зависят от качества построения иерархической структуры идентификаторов. В оригинальной статье показано, что семантическая кластеризация документов и последующее построение идентификаторов в соответствии с полученной структурой значительно увеличивает качество результата.

F. Анализ

В таблице III приведен общий анализ методов. В таблице II приведен анализ показателей эффективности описанных методов из оригинальных работ.

V. Заключение

В процессе изучения различных методов семантического поиска становится очевидной их важность в современной информационной эпохе. С учетом растущего

объема данных, доступ к релевантной, точной и полной информации становится все более критичным. Традиционные методы поиска, основанные на прямом совпадении ключевых слов, не могут обеспечивать глубокого понимания контента. Семантический поиск, в свою очередь, анализирует смысл запросов и предоставляемой информации, что позволяет достигать высокой степени релевантности результатов. В ходе работы произведен анализ бенчмарков и методов для оценки качества поиска, описаны текущие SOTA решения в области, описаны их преимущества и недостатки. Произведен сравнительный анализ качественных показателей существующих моделей. В заключение, семантический поиск представляет собой передовой подход в области информационного поиска, который может революционизировать способы, которыми мы взаимодействуем с данными. Ожидается, что с усовершенствованием технологий и методик, эффективность и точность семантического поиска будут только расти, предоставляя пользователям еще более качественные и релевантные результаты.

Список литературы

[1] Vaswani Ashish, Shazeer Noam, Parmar Niki et al. Attention is all you need. — 2017. — URL: https://arxiv.org/abs/1706.03762.

[2] Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding. — 2018. — URL: https://arxiv.org/abs/1810.04805.

[3] Liu Yinhan, Ott Myle, Goyal Naman et al. Roberta: A robustly optimized bert pretraining approach. — 2019. — URL: https://arxiv.org/abs/1907.11692.

[4] Language models are unsupervised multitask learners / Alec Radford, Jeff Wu, Rewon Child et al. — 2019.

[5] Brown Tom B., Mann Benjamin, Ryder Nick et al. Language models are few-shot learners. — 2020. — URL: https://arxiv.org/abs/2005.14165.

[6] Wang Alex, Singh Amanpreet, Michael Julian et al. Glue: A multi-task benchmark and analysis platform for natural language understanding. — 2018. — URL: https://arxiv.org/abs/1804.07461.

[7] Rajpurkar Pranav, Zhang Jian, Lopyrev Konstantin, Liang Percy. Squad: 100,000+ questions for machine comprehension of text. — 2016. — URL: https://arxiv.org/abs/1606.05250.

[8] Lai Guokun, Xie Qizhe, Liu Hanxiao et al. Race: Large-scale reading comprehension dataset from examinations. — 2017. — URL: https://arxiv.org/abs/1704.04683.

[9] Zellers Rowan, Holtzman Ari, Bisk Yonatan et al. Hellaswag: Can a machine really finish your sentence? — 2019. — URL: https://arxiv.org/abs/1905.07830.

[10] Position-aware attention and supervised data improve slot filling / Yuhao Zhang, Victor Zhong, Danqi Chen et al. // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. — Copenhagen, Denmark : Association for Computational Linguistics, 2017. — . — P. 35–45. — URL: https://aclanthology.org/D17-1004.

[11] Yang Yinfei, Cer Daniel, Ahmad Amin et al. Multilingual universal sentence encoder for semantic retrieval. — 2019. — 1907.04307.

[12] Embedding-based retrieval in facebook search / Jui-Ting Huang, Ashish Sharma, Shuying Sun et al. // CoRR. — 2020. — Vol. abs/2006.11632. — arXiv : 2006.11632.

[13] Zhang Yanzhao, Long Dingkun, Xu Guangwei, Xie Pengjun. Hlatr: Enhance multi-stage text retrieval with hybrid list aware transformer reranking. — 2022. — 2205.10569.

[14] Karpukhin Vladimir, Oğuz Barlas, Min Sewon et al. Dense passage retrieval for open-domain question answering. — 2020. — 2004.04906.

[15] Borgeaud Sebastian, Mensch Arthur, Hoffmann Jordan et al. Improving language models by retrieving from trillions of tokens. — 2022. — 2112.04426.

[16] Thakur Nandan, Reimers Nils, Daxenberger Johannes, Gurevych Iryna. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. — 2021. — 2010.08240.

[17] Okapi at trec-6 automatic ad hoc, vlc, routing, filtering and qsdr / Steve Walker, Stephen E Robertson, Mohand Boughanem et al. // NIST SPECIAL PUBLICATION SP. — 1998. — P. 125–136.

[18] Penha Gustavo, Palumbo Enrico, Aziz Maryam et al. Improving content retrievability in search with controllable query generation. — 2023. — 2303.11648.

[19] Jagerman Rolf, Zhuang Honglei, Qin Zhen et al. Query expansion by prompting large language models. — 2023. — 2305.03653.

[20] Zhang Yang, Bartley Travis M., Graterol-Fuenmayor Mariana et al. A chat about boring problems: Studying gpt-based text normalization. — 2024. — 2309.13426.

[21] Bengio Yoshua, Courville Aaron, Vincent Pascal. Representation learning: A review and new perspectives. — 2014. — 1206.5538.

[22] Gao Luyu, Callan Jamie. Unsupervised corpus aware language model pre-training for dense passage retrieval. — 2021. — 2108.05540.

[23] Xiao Shitao, Liu Zheng, Shao Yingxia, Cao Zhao. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. — 2022. — 2205.12035.

[24] Wu Xing, Ma Guangyuan, Lin Meng et al. Contextual masked auto-encoder for dense passage retrieval. — 2022. — 2208.07670.

[25] Malkov Yu. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. — 2018. — 1603.09320.

[26] Guo Ruiqi, Sun Philip, Lindgren Erik et al. Accelerating large-scale inference with anisotropic vector quantization. — 2020. — 1908.10396.

[27] Retrieve re-rank. — https://www.sbert.net/examples/applications/retrieve_rerank/README.html#retrieve-re-rank. — Accessed: 2022-12-21.

[28] Nogueira Rodrigo, Cho Kyunghyun. Passage re-ranking with bert. — 2020. — 1901.04085.

[29] Dai Zhuyun, Callan Jamie. Deeper text understanding for IR with contextual neural language modeling // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. — ACM, 2019. — jul. — URL:

[30] CEDR / Sean MacAvaney, Andrew Yates, Arman Cohan, Nazli Goharian // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. — ACM, 2019. — jul. — URL:

[31] Cross-domain modeling of sentence-level evidence for document retrieval / Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, Jimmy Lin // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). — Hong Kong, China : Association for Computational Linguistics, 2019. — . — P. 3490–3496. — URL: https://aclanthology.org/D19-1352.

[32] Li Canjia, Yates Andrew, MacAvaney Sean et al. Parade: Passage representation aggregation for document reranking. — 2021. — 2008.09093.

[33] Bajaj Payal, Campos Daniel, Craswell Nick et al. Ms marco: A human generated machine reading comprehension dataset. — 2018. — 1611.09268.

[34] Tay Yi, Tran Vinh Q., Dehghani Mostafa et al. Transformer memory as a differentiable search index. — 2022. — URL: https://arxiv.org/abs/2202.06991.

[35] Raffel Colin, Shazeer Noam, Roberts Adam et al. Exploring the limits of transfer learning with a unified text-to-text transformer. — 2019. — URL: https://arxiv.org/abs/1910.10683.

[36] Sutskever Ilya, Vinyals Oriol, Le Quoc V. Sequence to sequence learning with neural networks. — 2014. — URL: https://arxiv.org/abs/1409.3215.

[37] Wang Yujing, Hou Yingyan, Wang Haonan et al. A neural corpus indexer for document retrieval. — 2023. — 2206.02743.

[38] Tang Yubao, Zhang Ruqing, Guo Jiafeng et al. Listwise generative retrieval models via a sequential learning process. — 2024. — 2403.12499.

[39] Mehta Sanket Vaibhav, Gupta Jai, Tay Yi et al. Dsi++: Updating transformer memory with new documents. — 2022. — 2212.09744.

[40] Searching for answers in a pandemic: An overview of trec-covid / Ellen M. Voorhees, Ian Soboroff, Kirk Roberts et al. // Journal of Biomedical Informatics. — 2021. — Vol. 121. — URL: https://doi.org/10.1016/j.jbi.2021.103865.

[41] Natural questions: a benchmark for question answering research / Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield et al. // Transactions of the Association for Computational Linguistics. — 2019. — Vol. 7. — P. 452–466.

[42] TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension / Mandar Joshi, Eunsol Choi, Daniel Weld, Luke Zettlemoyer // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Vancouver, Canada : Association for Computational Linguistics, 2017. — . — P. 1601–1611. — URL: https://aclanthology.org/P17-1147.

[43] Nentidis Anastasios, Krithara Anastasia, Paliouras Georgios, Bougiatiotis Konstantinos. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. — urlhttp://participants-area.bioasq.org/. — 2021. — Accessed: 2024-07-17.

[44] Quora. Quora question pairs. — 2017. — Accessed: 2024-07-17. URL: https://www.kaggle.com/c/quora-question-pairs.

[45] FEVER: a large-scale dataset for fact extraction and VERification / James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Arpit Mittal // NAACL-HLT. — New Orleans, Louisiana : Association for Computational Linguistics, 2018. — P. 809–819. — URL: https://aclanthology.org/N18-1074.

[46] HotpotQA: A dataset for diverse, explainable multi-hop question answering / Zhilin Yang, Peng Qi, Saizheng Zhang et al. // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing / Association for Computational Linguistics. — 2018. — P. 2369–2380. — URL: https://arxiv.org/abs/1809.09600.

[47] Www'18 open challenge: Financial opinion mining and question answering / Saulo Macedo Maia, Siegfried Handschuh, André Freitas et al. // Companion Proceedings of the The Web Conference 2018. — 2018. — URL: https://github.com/dayanfcosta/fiqa-2018-task1/blob/master/datasets/Readme_task1.pdf.

[48] Fact or fiction: Verifying scientific claims / David Wadden, Shanchuan Lin, Kyle Lo et al. // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Online : Association for Computational Linguistics, 2020. — . — P. 7534–7550. — URL: https://aclanthology.org/2020.emnlp-main.609.

[49] Cohan Arman, Feldman Sergey, Beltagy Iz et al. SciDocs: A Benchmark Suite for Document-Level Representation Learning. — https://allenai.org/data/scidocs. — 2020. — Version 1.0.

Таблица I
Современные наборы данных для информационного поиска.

Название набора данных	**Метод сбора данных**	**Языки**	**Последнее обновление**	**Формулировка задачи**
MS MARCO [33]	Запросы поиска Bing и страницы результаты	Английский, многоязычный	2020	Вопрос-Ответ, Ранжирование пассажей, Ранжирование документов
TREC-COVID [40]	Научные статьи о COVID и вручную написанные запросы	Английский	В процессе	Отбор документов
Natural Questions [41]	Поисковые запросы Google	Английский	2019	Вопрос-Ответ
TriviaQA [42]	Викторины и сайты с вопросами	Английский	2017	Вопрос-Ответ
BioASQ [43]	Биомедицинские статьи	Английский	2020	Биомедицинский Вопрос-Ответ
Quora [44]	Пары вопросов Quora	Английский	2017	Классификация дублирующихся вопросов
FEVER [45]	Википедия	Английский	2018	Проверка фактов
HotpotQA [46]	Википедия	Английский	2019	Вопрос-Ответ с подтверждающими фактами
FiQA-2018 [47]	Финансовые данные и статьи	Английский	2018	Финансовый Вопрос-Ответ
SciFact [48]	Научные статьи	Английский	2020	Проверка научных утверждений
SciDocs [49]	Научная литература	Английский	2020	Поиск научных документов

Таблица II
Таблица с метриками для различных моделей на различных наборах данных

Модель	**MS MARCO**				**NQ320k**				
	MRR@3	**MRR@10**	**Recall@100**	**HITS@10**	**MRR@3**	**MRR@10**	**MRR@100**	**Recall@100**	**HITS@10**
ListGR	0.4656	0.4901	×	0.6471	0.6019	0.7723	×	×	0.8412
NCI	×	×	×	×	×	×	0.7312	0.9622	0.9176
DSI	×	×	×	×	×	×	×	×	0.703
coCondenser	×	0.382	0.984	×	×	×	×	0.89	×
RetroMAE	×	0.3822	0.9074	×	×	×	×	0.8942	×
CoTMAE	×	0.394	×	×	×	×	×	0.893	×
HLATR	×	0.426	×	×	×	×	×	×	×

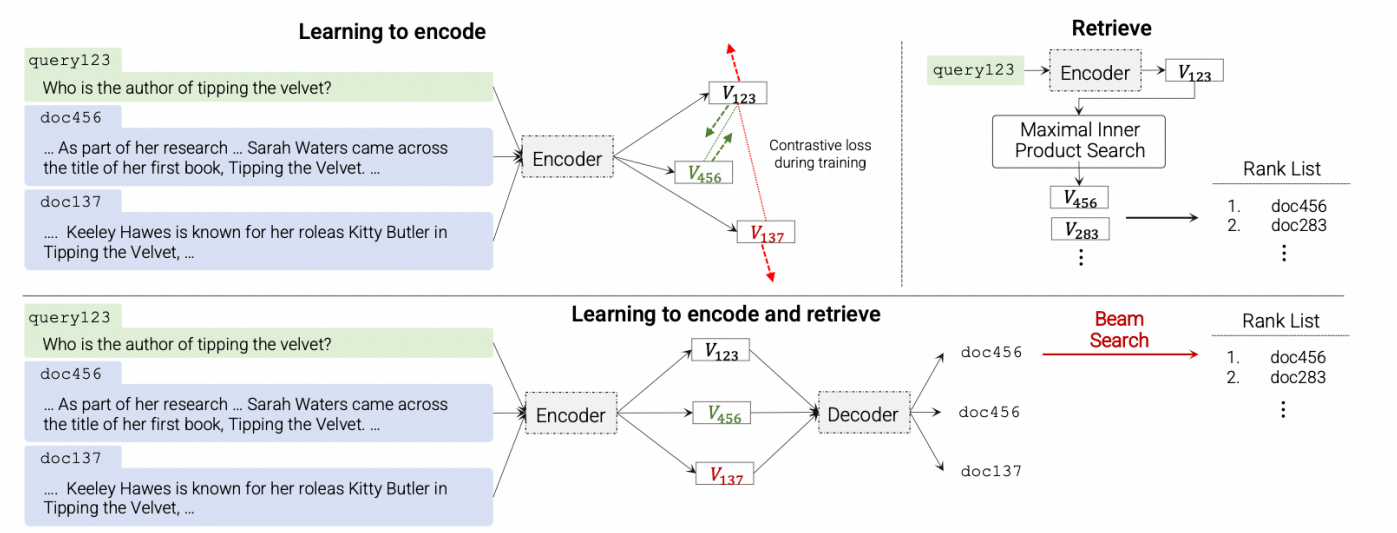

Рис. 4. Сравнение поиска на основе векторного отбора и поиска на основе DSI[34]

Таблица III
Сравнительный анализ алгоритмов поиска

Алгоритм	**Преимущества и недостатки**
Полнотекстовый поиск	**Преимущества:** Быстрый и эффективный для поиска по ключевым словам. Легко реализуется и имеет низкие требования к ресурсам. Хорошо работает с большими наборами данных. **Недостатки:** Не понимает контекст запроса и не учитывает семантику документов. Может не выявлять важные документы, если запрос не содержит точных совпадений с ключевыми словами.
Поиск по векторному пространству	**Преимущества:** Лучше понимает контекст запроса и документа, учитывает семантику. Способен улавливать неявные и тонкие взаимосвязи между словами и предложениями. **Недостатки:** По сравнению с Cross-Encoder'ами качество поиска может быть ниже. Плохо ищет по специфическим словам и именам собственным.
Cross-Encoder	**Преимущества:** Одна из лучших техник для семантического поиска. Одновременно принимает на вход запрос и документ, работает на уровне токенов и улавливает очень тонкие взаимосвязи. **Недостатки:** Из-за сложности обработки имеет большое время работы на больших корпусах. Не подходит для работы в системах реального времени.
Retrieve&Re-rank	**Преимущества:** Гибридный подход более качественный, чем обычный векторный отбор, и значительно более быстрый, чем Cross-Encoder. **Недостатки:** Стадия отбора может пропускать релевантные документы.
Transformer memory as Differentiable search index	**Преимущества:** Относительно небольшая вычислительная сложность при потенциальном качестве сопоставимом с Cross-Encoder. **Недостатки:** Необходимо увеличивать размер модели с увеличением количества данных. Отсутствие возможности оперативно обновлять индекс новыми документами. Проблема забывания: документы внутри индекса забываются в процессе дообучения для добавления новых документов.

A survey on natural language semantic search algorithms

Nikita Shalagin

***Abstract*—Semantic search represents a modern approach to information retrieval based on understanding the meaning and context of queries, enabling more relevant results compared to traditional keyword-based search methods. The use of natural language processing technologies, such as the Transformer architecture and large pre-trained language models, has significantly improved the quality of semantic search. These models have demonstrated high performance in various benchmarks, leading to their widespread application in numerous areas.**

The main advantages of semantic search include a higher level of accuracy and relevance of results, enhanced user experience, and the ability to freely express queries. However, despite significant

achievements, there are challenges related to the computational complexity of models, the limited size of the text that can be processed, and the response time in real-time modes. In solutions requiring near-real-time processing of user queries, developers often need to apply less resource-intensive methods, which can reduce search quality. A common dilemma for specialists developing practical applications is the trade-off between computational costs, processing speed, and search quality.

This work aims to review current semantic search methods and provide a comparative analysis. Special attention is given to examining the advantages and disadvantages of various approaches, as well as analyzing the prospects for their further development and application in different fields.

***Keywords*—Natural language processing, NLP, semantic search, text documents search**

References

[1] Vaswani Ashish, Shazeer Noam, Parmar Niki et al. Attention is all you need. — 2017. — URL: https://arxiv.org/abs/1706.03762.

[2] Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding. — 2018. — URL: https://arxiv.org/abs/1810.04805.

[3] Liu Yinhan, Ott Myle, Goyal Naman et al. Roberta: A robustly optimized bert pretraining approach. — 2019. — URL: https://arxiv.org/abs/1907.11692.

[4] Language models are unsupervised multitask learners / Alec Radford, Jeff Wu, Rewon Child et al. — 2019.

[5] Brown Tom B., Mann Benjamin, Ryder Nick et al. Language models are few-shot learners. — 2020. — URL: https://arxiv.org/abs/2005.14165.

[6] Wang Alex, Singh Amanpreet, Michael Julian et al. Glue: A multi-task benchmark and analysis platform for natural language understanding. — 2018. — URL: https://arxiv.org/abs/1804.07461.

[7] Rajpurkar Pranav, Zhang Jian, Lopyrev Konstantin, Liang Percy. Squad: 100,000+ questions for machine comprehension of text. — 2016. — URL: https://arxiv.org/abs/1606.05250.

[8] Lai Guokun, Xie Qizhe, Liu Hanxiao et al. Race: Large-scale reading comprehension dataset from examinations. — 2017. — URL: https://arxiv.org/abs/1704.04683.

[9] Zellers Rowan, Holtzman Ari, Bisk Yonatan et al. Hellaswag: Can a machine really finish your sentence? — 2019. — URL: https://arxiv.org/abs/1905.07830.

[10] Position-aware attention and supervised data improve slot filling / Yuhao Zhang, Victor Zhong, Danqi Chen et al. // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. — Copenhagen, Denmark : Association for Computational Linguistics, 2017. — . — P. 35–45. — URL: https://aclanthology.org/D17-1004.

[11] Yang Yinfei, Cer Daniel, Ahmad Amin et al. Multilingual universal sentence encoder for semantic retrieval. — 2019. — 1907.04307.

[12] Embedding-based retrieval in facebook search / Jui-Ting Huang, Ashish Sharma, Shuying Sun et al. // CoRR. — 2020. — Vol. abs/2006.11632. — arXiv : 2006.11632.

[13] Zhang Yanzhao, Long Dingkun, Xu Guangwei, Xie Pengjun. Hlatr: Enhance multi-stage text retrieval with hybrid list aware transformer reranking. — 2022. — 2205.10569.

[14] Karpukhin Vladimir, Oğuz Barlas, Min Sewon et al. Dense passage retrieval for open-domain question answering. — 2020. — 2004.04906.

[15] Borgeaud Sebastian, Mensch Arthur, Hoffmann Jordan et al. Improving language models by retrieving from trillions of tokens. — 2022. — 2112.04426.

[16] Thakur Nandan, Reimers Nils, Daxenberger Johannes, Gurevych Iryna. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. — 2021. — 2010.08240.

[17] Okapi at trec-6 automatic ad hoc, vlc, routing, filtering and qsdr / Steve Walker, Stephen E Robertson, Mohand Boughanem et al. // NIST SPECIAL PUBLICATION SP. — 1998. — P. 125–136.

[18] Penha Gustavo, Palumbo Enrico, Aziz Maryam et al. Improving content retrievability in search with controllable query generation. — 2023. — 2303.11648.

[19] Jagerman Rolf, Zhuang Honglei, Qin Zhen et al. Query expansion by prompting large language models. — 2023. — 2305.03653.

[20] Zhang Yang, Bartley Travis M., Graterol-Fuenmayor Mariana et al. A chat about boring problems: Studying gpt-based text normalization. — 2024. — 2309.13426.

[21] Bengio Yoshua, Courville Aaron, Vincent Pascal. Representation learning: A review and new perspectives. — 2014. — 1206.5538.

[22] Gao Luyu, Callan Jamie. Unsupervised corpus aware language model pre-training for dense passage retrieval. — 2021. — 2108.05540.

[23] Xiao Shitao, Liu Zheng, Shao Yingxia, Cao Zhao. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. — 2022. — 2205.12035.

[24] Wu Xing, Ma Guangyuan, Lin Meng et al. Contextual masked auto-encoder for dense passage retrieval. — 2022. — 2208.07670.

[25] Malkov Yu. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. — 2018. — 1603.09320.

[26] Guo Ruiqi, Sun Philip, Lindgren Erik et al. Accelerating large-scale inference with anisotropic vector quantization. — 2020. — 1908.10396.

[27] Retrieve re-rank. — https://www.sbert.net/examples/applications/retrieve_rerank/README.html#retrieve-re-rank. — Accessed: 2022-12-21.

[28] Nogueira Rodrigo, Cho Kyunghyun. Passage re-ranking with bert. — 2020. — 1901.04085.

[29] Dai Zhuyun, Callan Jamie. Deeper text understanding for IR with contextual neural language modeling // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. — ACM, 2019. — jul. — URL:

[30] CEDR / Sean MacAvaney, Andrew Yates, Arman Cohan, Nazli Goharian // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. — ACM, 2019. — jul. — URL:

[31] Cross-domain modeling of sentence-level evidence for document retrieval / Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, Jimmy Lin // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-

IJCNLP). — Hong Kong, China : Association for Computational Linguistics, 2019. — . — P. 3490–3496. — URL: https://aclanthology.org/D19-1352.
[32] Li Canjia, Yates Andrew, MacAvaney Sean et al. Parade: Passage representation aggregation for document reranking. — 2021. — 2008.09093.
[33] Bajaj Payal, Campos Daniel, Craswell Nick et al. Ms marco: A human generated machine reading comprehension dataset. — 2018. — 1611.09268.
[34] Tay Yi, Tran Vinh Q., Dehghani Mostafa et al. Transformer memory as a differentiable search index. — 2022. — URL: https://arxiv.org/abs/2202.06991.
[35] Raffel Colin, Shazeer Noam, Roberts Adam et al. Exploring the limits of transfer learning with a unified text-to-text transformer. — 2019. — URL: https://arxiv.org/abs/1910.10683.
[36] Sutskever Ilya, Vinyals Oriol, Le Quoc V. Sequence to sequence learning with neural networks. — 2014. — URL: https://arxiv.org/abs/1409.3215.
[37] Wang Yujing, Hou Yingyan, Wang Haonan et al. A neural corpus indexer for document retrieval. — 2023. — 2206.02743.
[38] Tang Yubao, Zhang Ruqing, Guo Jiafeng et al. Listwise generative retrieval models via a sequential learning process. — 2024. — 2403.12499.
[39] Mehta Sanket Vaibhav, Gupta Jai, Tay Yi et al. Dsi++: Updating transformer memory with new documents. — 2022. — 2212.09744.
[40] Searching for answers in a pandemic: An overview of trec-covid / Ellen M. Voorhees, Ian Soboroff, Kirk Roberts et al. // Journal of Biomedical Informatics. — 2021. — Vol. 121. — URL: https://doi.org/10.1016/j.jbi.2021.103865.
[41] Natural questions: a benchmark for question answering research / Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield et al. // Transactions of the Association for Computational Linguistics. — 2019. — Vol. 7. — P. 452–466.
[42] TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension / Mandar Joshi, Eunsol Choi, Daniel Weld, Luke Zettlemoyer // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Vancouver, Canada : Association for Computational Linguistics, 2017. — . — P. 1601–1611. — URL: https://aclanthology.org/P17-1147.
[43] Nentidis Anastasios, Krithara Anastasia, Paliouras Georgios, Bougiatiotis Konstantinos. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. — urlhttp://participants-area.bioasq.org/. — 2021. — Accessed: 2024-07-17.
[44] Quora. Quora question pairs. — 2017. — Accessed: 2024-07-17. URL: https://www.kaggle.com/c/quora-question-pairs.
[45] FEVER: a large-scale dataset for fact extraction and VERification / James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Arpit Mittal // NAACL-HLT. — New Orleans, Louisiana : Association for Computational Linguistics, 2018. — P. 809–819. — URL: https://aclanthology.org/N18-1074.
[46] HotpotQA: A dataset for diverse, explainable multi-hop question answering / Zhilin Yang, Peng Qi, Saizheng Zhang et al. // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing / Association for Computational Linguistics. — 2018. — P. 2369–2380. — URL: https://arxiv.org/abs/1809.09600.
[47] Www'18 open challenge: Financial opinion mining and question answering / Saulo Macedo Maia, Siegfried Handschuh, André Freitas et al. // Companion Proceedings of the The Web Conference 2018. — 2018. — URL: https://github.com/dayanfcosta/fiqa-2018-task1/blob/master/datasets/Readme_task1.pdf.
[48] Fact or fiction: Verifying scientific claims / David Wadden, Shanchuan Lin, Kyle Lo et al. // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Online : Association for Computational Linguistics, 2020. — . — P. 7534–7550. — URL: https://aclanthology.org/2020.emnlp-main.609.
[49] Cohan Arman, Feldman Sergey, Beltagy Iz et al. SciDocs: A Benchmark Suite for Document-Level Representation Learning. — https://allenai.org/data/scidocs. — 2020. — Version 1.0.