Итеративная подборка коллекции релевантных документов

Никишкина Евгения Геннадьевна

Май 2024

Аннотация

В условиях постоянно растущего объема данных очевидна важность эффективного поиска и фильтрации информации. Методы итеративной подборки релевантных документов становятся все более актуальными в контексте потребности пользователей в быстром и точном доступе к необходимой информации. Учитывая разнообразие задач и сфер применения, такие методы обладают широким потенциалом, от справочного поиска до научного анализа. Результатами данной работы являются создание нового итеративного метода подборки коллекции релевантных документов, а также автоматизация процесса оценивания данного алгоритма без задействования реальных пользователей. Помимо этого, результаты экспериментов демонстрируют, что комбинирование тренировочных данных позволяет улучшить метрики качества полученной модели. Разработанный метод демонстрирует эффективность в процессе поиска информации, что делает его перспективным инструментом для систем информационного поиска и анализа текстовых данных. Код проекта расположен по ссылке: NLP_project.

1 Введение

Быстрое развитие технологий приводит к стремительному увеличению объёма информации. Исследователи используют разнообразные техники анализа больших данных для поиска и фильтрации различной информации из поисковых систем, таких как Google, Bing и др. Fahad et al. (2014), Xia et al. (2018), Liu et al. (2018). Помимо поиска информации некоторые исследователи делятся своими научными результатами и публикациями посредством цифровых платформ Sun et al. (2013). Избыток данных вызывает информационное перенасыщение и затрудняет исследователям поиск релевантных документов Miah et al. (2017). Создание системы подборки коллекции документов актуально, поскольку в академических исследованиях она способствует упрощению и усовершенствованию процедуры поиска релевантных статей.

Задача подбора коллекции документов тесно связана с задачей информационного поиска и созданием рекомендательных систем. Все эти задачи связаны с тем, как эффективно организовать и предоставить пользователю доступ к нужной информации из огромного объема данных. Информационный поиск заключается в нахождении информации в больших наборах данных в ответ на запрос пользователя. Рекомендательные системы используются для предоставления персонализированных рекомендаций, основанных на предпочтениях и поведении пользователя. Подбор коллекции документов включает в себя выбор, категоризацию и организацию документов таким образом, чтобы пользователь мог легко находить необходимую информацию. При этом учитываются различные факторы, такие как актуальность, достоверность, релевантность и интересы пользователя.

Классическими методами для решения задачи информационного поиска являются такие подходы, как использование ключевых слов Chor et al. (1997), различные статистические методы Sparck Jones et al. (2000) и т.д. Однако данные процедуры нередко сталкиваются с проблемой недостаточной точности и эффективности в условиях изменяющейся информационной среды Hassan (2017), что в свою очередь требует от пользователей дополнительной фильтрации результатов поиска для получения релевантных документов. Также существенным недостатком таких подходов является отсутствие персонализации. В то же время продвинутые системы рекомендаций статей учитывают интересы исследователей, отношения с соавторами и отношения цитирования для разработки итоговых алгоритмов.

По сравнению с традиционной техникой информационного поиска по ключевым словам, современные рекомендательные системы более персонализированы и эффективны для обработки больших объемов данных Miah et al. (2017), Liu et al. (2015). С момента появления систем рекомендаций возникло множество алгоритмов рекомендаций Xia et al. (2016). Техники рекомендаций можно разделить на четыре основные категории: фильтрация на основе контента (CBF), коллаборативная фильтрация (CF), методы на основе графов (GB) и гибридные методы рекомендаций. Каждый метод имеет свою собственную логику, лежащую в основе рекомендации интересных для исследователей статей Xia et al. (2016), Sugiyama and Kan (2011). Фильтрация на основе контента (CBF) в основном учитывает исторические предпочтения пользователей и их личную библиотеку для извлечения и построения модели интересов пользователей, которая называется профилем пользователя Sun et al. (2013). Затем в данном методе извлекаются ключевые слова из «документов-кандидатов» и вычисляется их сходство. В результаты ранжирования (под ранжированием понимается процесс упорядочивания объектов по определенным критериям) статей по уровню сходства пользователь получит рекомендации. Коллаборативная фильтрация (CF) в основном сосредотачивается на действиях или оценках пользователей по отношению к документам других пользователей, чьи профили схожи с профилем пользователя, называемых «соседними пользователями» Pera and Ng (2013). Предполагается, что, если у пользователей были схожие интересы в прошлом, они будут совпадать и в будущем. Существует немало ис-

следований, описывающих метод на основе графов (GB) Zhao et al. (2016). Многие исследования основаны на построении графа, в котором авторы и статьи рассматриваются в качестве узлов. Отношение между статьями, отношение между пользователями и отношение между пользователями и статьями рассматриваются как рёбра. Затем на графе используются случайное блуждание или другие алгоритмы для вычисления соответствия между пользователями и статьями. Для гибридного метода системы рекомендаций обычно используют методы фильтрации на основе контента и коллаборативной фильтрации для формирования рекомендаций, тем самым сочетая преимущества каждого из методов. Методы фильтрации на основе контента и коллаборативной фильтрации дополняют друг друга, и системы рекомендаций с их комбинацией обычно более точны, чем система, которая работает только с одним из данных алгоритмов. К преимуществам рекомендательных систем относятся простота реализации, относительная скорость работы, применимость к ограниченным данным. Однако существует немало недостатков такого рода подходов: ограниченная точность, неэффективность при учете контекста, необходимость ручной настройки, а также чувствительность к качеству входных данных.

Ввиду озвученных выше недостатков исследователи стали все чаще полагаться на сложные инструменты обнаружения и рекомендации для поиска релевантной литературы и научных работ Ammar et al. (2018), Bonifacio et al. (2022). Одним из ключевых преимуществ, которое отличает нейронные сети от многих других стратегий обучения, применяемых ранее, является их способность работать с сырыми входными данными. Например, при наличии достаточного количества обучающих данных хорошо спроектированные сети могут самостоятельно извлекать признаки, включая основные характеристики входных данных, такие как частота терминов (tf) и значимость терминов (idf) - которые ранее рассчитывались заранее. Ранее проектирование признаков было ключевым аспектом и вкладом во вновь предложенные подходы к информационному поиску, однако сейчас основной акцент переместился на разработку и использование нейронных сетей. Также развитие информационного поиска посредством нейронных сетей Lin et al. (2020) привело к потребности поиска представлений документов, которые лучше улавливают семантику документов по сравнению с предыдущим поколением методов поиска. Такие представления зачастую получаются путем добавления стадии дообучения предварительно обученных больших языковых моделей, ориентированных на задачу информационного поиска.

Термин «итеративность» в контексте итеративной подборки релевантных документов означает процесс многократного повторения определённых действий с целью уточнения и улучшения результатов на каждом шаге. В данном случае итеративность связана с подбором документов таким образом, чтобы каждый следующий этап был основан на обратной связи, полученной от человека. Однако описанные выше методы не подразумевают процесс итеративности, поэтому целью данной работы является исследование и разработка методов для решения задачи итеративной подборки коллекции релевантных документов, а также оценка их эффективности. В работе

представлены описание разработанных методов поиска и оценки полученной процедуры, а также результаты экспериментов, подтверждающие их эффективность. В работе представлен обзор и анализ существующих методов в разделе 3, предложен новый подход к решению поставленной задачи 2 и описаны предлагаемые методики 4. Были проведены эксперименты на данных, описанных в 5, и осуществлён анализ полученных результатов 7.

2 Постановка задачи

Дана коллекция документов $D = \{d_1, d_2, ..., d_n\}$, где каждый документ d_i представлен в виде текстовой строки. Пусть также имеется запрос пользователя q, который представлен в виде текстовой строки.

Требуется разработать алгоритм, который соответствует итеративному процессу отбора подмножества документов $R \subseteq D$, наиболее релевантных с точки зрения пользователя по отношению к запросу q, а также разработать методику оценивания такого рода алгоритма, не требующую проведения тестирования с участием реальных пользователей

Итеративный процесс предполагает следующие шаги:

1. **Первоначальный этап:** Начальное подмножество R_0 документов выбирается исключительно на основе оценивания релевантности документов к запросу. Вычисление релевантности каждого документа $d_i \in D$ относительно запроса q осуществляется за счет использования некоторой функции $f(q, d_i)$, в данном случае в качестве такой функции выступает косинусная мера между векторными представлениями документа d_i и запроса q.

2. **Промежуточный этап:** На каждой итерации t алгоритма происходит следующее:

   - Из полученного на предыдущем этапе подмножества документов R_{t-1} пользователем отбираются субъективно понравившиеся документы, векторные представления которых впоследствии будут агрегироваться с запросом q. Итеративность процесса заключается в том, что на каждой итерации алгоритма производится уточнение выданных релевантных документов за счет получения обратной связи от пользователя о ранее полученных системой документов.
   - Вычисление релевантности каждого документа $d_i \in D$ относительно запроса q, агрегированного с понравившимися ранее пользователю документами, с использованием функции релевантности $f(q, d_i)$.

3. **Заключительный этап:** Процесс завершается после получения информации от пользователя о его остановке.

3 Известные результаты

В данном разделе будут рассмотрено несколько ключевых подходов и результатов, на которые опирается данная работа.

3.1 Модель Contriever

В качестве основной модели в данной работе была использована модель Contriever Izacard et al. (2021). Данная модель предназначена для поиска соответствующих документов в большой коллекции для заданного запроса. Модель принимает на вход набор документов и запрос, и выдает оценку релевантности для каждого документа. Архитектура модели состоит из двух моделей-кодировщиков (encoder model), в которых векторные представления для запроса и документа получаются независимо. Оценка релевантности между запросом и документом задается скалярным произведением их векторных представлений после применения кодировщика. Запрос q и документ d кодируются независимо с использованием одной и той же модели f_θ, параметризованной θ. Оценка релевантности $s(q, d)$ между запросом q и документом d представляет собой скалярное произведение полученных представлений:

$$s(q, d) = \langle f_\theta(q), f_\theta(d) \rangle.$$

В работе Izacard et al. (2021) в качестве f_θ используется сеть трансформер, чтобы получить векторные представления как для запросов, так и для документов. В качестве альтернативы могут использоваться два различных кодировщика для кодирования запросов и документов, как предлагается в Karpukhin et al. (2020). Авторы статьи Izacard et al. (2021) отметили, что использование одного и того же кодировщика в целом улучшает устойчивость модели. Представление $f_\theta(q)$ (соответственно $f_\theta(d)$) для запроса (соответственно документа) получается путем усреднения скрытых представлений последнего слоя.

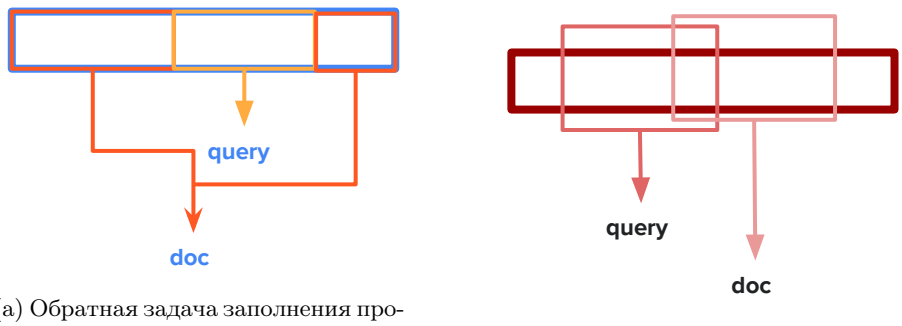
В данной работе используется предварительно обученная с использованием контрастивной функции потерь (contrastive loss) He et al. (2019) на задаче обучения без учителя модель. Контрастное обучение — это подход, основанный на том факте, что каждый документ в некотором роде уникален. Контрастивная функция потерь используется для обучения путем нахождения отличия между документами. Эта функция сравнивает либо положительные (из одного и того же документа), либо отрицательные (из разных документов) пары представлений документов. Формально, учитывая запрос q, набор документов $\{d_1, d_2, ..., d_K\}$, а также релевантный документ (обозначенный как d_+), который соответствует q. Контрастивная потеря InfoNCE определяется как:

$$\mathcal{L}(q, d_+) = -\frac{\exp(s(q, d_+)/\tau)}{\exp(s(q, d_+)/\tau) + \sum_{i=1}^{K} \exp(s(q, d_i)/\tau)},$$

где τ - это гиперпараметр температуры Wu et al. (2018). Значение данной

функции потерь мало, когда q похож на релевантный документ d_+ и отличается от всех остальных документов (рассматриваемых как нерелевантные документы для q). Другая интерпретация этой функции потерь следующая: учитывая представление запроса q, цель состоит в том, чтобы восстановить или извлечь представление d_+, соответствующее положительному документу, среди всех отрицательных d_i.

Ключевым элементом контрастного обучения является способ создания положительных пар из одного входного набора данных. Обратная задача заполнения пропусков (Inverse Cloze Task, ICT) рис. 1a — это аугментация данных, которая генерирует два взаимоисключающих представления входных данных, введенная в контексте извлечения информации в Lee et al. (2019). Первое представление данных получается случайной выборкой фрагмента лексем из сегмента текста, в то время как дополнение к фрагменту формирует второе представление. Формально, учитывая последовательность текста $(w_1, ..., w_n)$, ICT выбирает фрагмент $(w_a, ..., w_b)$, где $1 \leq a \leq b \leq n$, и использует лексемы фрагмента в качестве запроса, а дополнение $(w_1, ..., w_{a-1}, w_{b+1}, ..., w_n)$ в качестве документа. Также еще одним способом является независимое обрезание рис. 1b, в котором представления генерируются независимо путем обрезки входных данных. В контексте текста обрезка эквивалентна выборке фрагмента лексем. Таким образом, эта стратегия независимо выбирает два фрагмента из документа, чтобы сформировать положительную пару. В отличие от обратной задачи заполнения пропусков, при обрезке оба представления соответствуют непрерывной подпоследовательности входных данных. Независимая обрезка также приводит к перекрытию между двумя представлениями, тем самым стимулируя сеть учиться точным совпадениям между запросом и документом.



(a) Обратная задача заполнения пропусков

(b) Независимое обрезание

Рис. 1: Способ создания положительных пар

Также важным аспектом контрастивного обучения является выборка большого количества негативных примеров. Существует большое число подходов для обработки негативных примеров, в работе Izacard et al. (2021) используется MoCo He et al. (2019). Данный подход заключается в том,

что существует очередь, в которой хранятся представления из предыдущих пакетов для последующего использования их в качестве отрицательных примеров в функции потерь. Это позволяет использовать меньший размер пакета, но немного изменяет потери, делая их асимметричными между запросами (одно из представлений, сгенерированное из элементов текущего пакета) и документами (элементы, хранящиеся в очереди). Градиент передается только через запросы, а представления ключей рассматривается как фиксированные. На практике представления, хранящиеся в очереди из предыдущих пакетов, формируются в предыдущих итерациях сети, что приводит к снижению производительности, когда сеть быстро меняется во время обучения. Вместо этого в He et al. (2019) было предложено генерировать представления ключей из второй сети, которая обновляется медленнее. Этот подход, называемый MoCo, рассматривает две сети: одну для ключей, параметризованную θ_k, и одну для запросов, параметризованную θ_q. Параметры сети ключей обновляются от параметров сети запросов с использованием экспоненциального скользящего среднего:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q,$$

где m - параметр импульса, принимающий значения из $[0, 1]$.

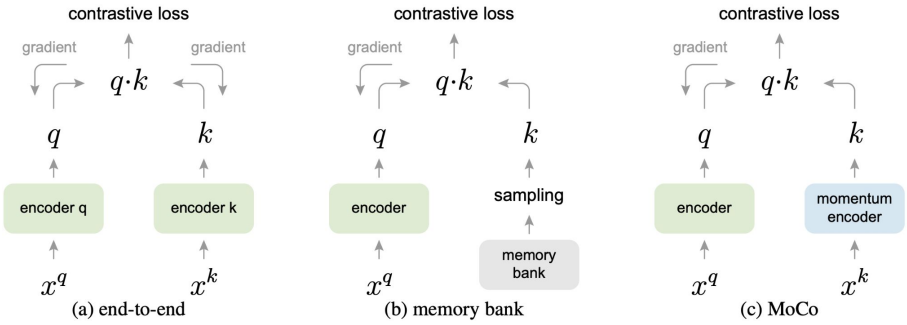


Рис. 2: Построение негативных примеров

3.2 Модель BM25

BM25 (Best Match 25) представляет собой вероятностный алгоритм ранжирования, применяемый для оценки степени релевантности документов по отношению к поисковому запросу. Широко применяется в поисковых системах и системах управления информацией Robertson and Zaragoza (2009). Данный алгоритм действует как поисковая функция на неупорядоченном множестве терминов (так называемом «мешке слов») и множестве документов. Он основан на анализе встречаемости слов запроса в каждом документе и не учитывает взаимоотношений между ними. BM25 представляет собой семейство функций с различными компонентами и параметрами.

Пусть дан запрос q, содержащий слова $w_1, ..., w_n$, тогда функция BM25 даёт следующую оценку релевантности документа d запросу q:

$$\mathrm{BM}(d,q) = \sum_{i=1}^{n} \mathrm{IDF}\left(w_i\right) \cdot \frac{f\left(w_i,d\right)\cdot\left(k_1+1\right)}{f\left(w_i,d\right)+k_1\cdot\left(1-b+b\cdot\frac{|d|}{\mathrm{avgdl}}\right)},$$

где $f\left(w_i,d\right)$ – частота слова (term frequency, TF) w_i в документе D, $|D|$ – длина документа (количество слов в нём), avgdl – средняя длина документа в коллекции. k_1 и b - свободные коэффициенты, обычно выбираются как $k_1 = 2.0$ и $b = 0.75$.

$\mathrm{IDF}\left(q_i\right)$ — обратная документная частота (inverse document frequency, IDF) слова w_i, которая представляет собой меру, используемую для оценки важности данного слова в контексте корпуса документов. Существует несколько интерпретаций IDF и незначительных вариаций его формулы. В классическом понимании IDF определяется как:

$$\mathrm{IDF}\left(w_i\right) = \log\frac{N}{n\left(w_i\right)},$$

где N — общее количество документов в коллекции, а $n\left(w_i\right)$ – количество документов, содержащих w_i. Однако чаще используются «сглаженные» варианты данной формулы, такие как:

$$\mathrm{IDF}\left(w_i\right) = \log\frac{N-n\left(w_i\right)+\frac{1}{2}}{n\left(w_i\right)+\frac{1}{2}}.$$

Упомянутая выше формула IDF имеет следующий недостаток: для слов, встречающихся в более чем половине документов из коллекции, значение IDF становится отрицательным. В результате, при наличии двух почти идентичных документов, где один содержит слово, а другой - нет, второй может получить более высокую оценку.

Другими словами, часто встречающиеся слова могут искажать окончательную оценку документа. Это нежелательно, поэтому во многих приложениях вышеупомянутая формула может быть скорректирована следующими способами:

- Игнорирование всех отрицательных слагаемых в сумме (что эквивалентно исключению слов из стоп-листа и игнорированию всех соответствующих высокочастотных слов);
- Установка нижней границы ε для IDF: если IDF меньше ε, то его значение считается равным ε;
- Использование альтернативной формулы IDF, не принимающей отрицательных значений.

Принимая во внимание описанные недостатки, полученная итоговая формула имеет следующий вид:

$$\mathrm{BM}(d,q) = \sum_{i=1}^{n} \max\left\{\log\frac{N-n\left(w_i\right)+\frac{1}{2}}{n\left(w_i\right)+\frac{1}{2}}, \varepsilon\right\}\cdot\frac{f\left(w_i,d\right)\cdot\left(k_1+1\right)}{f\left(w_i,d\right)+k_1\cdot\left(1-b+b\cdot\frac{|d|}{\mathrm{avgdl}}\right)}.$$

3.3 Перенос обучения

Перенос обучения (transfer learning) — метод обучения глубоких нейронных сетей, основанный на использовании заранее обученных моделей для решения новых задач. Он позволяет передать знания, полученные при решении одной задачи, на другую задачу, связанную с ней. Этот подход стал широко распространен благодаря своей эффективности и способности ускорить процесс обучения моделей на новых наборах данных. Идея переноса обучения основана на предположении о том, что признаки, полученные моделью при решении одной задачи, часто могут быть полезны при решении других задач. Это особенно актуально в случае, когда у нас есть ограниченное количество размеченных данных для новой задачи.

В Zhuang et al. (2019) отмечается, что использование предварительно обученных моделей позволяет извлечь общие признаки из данных и переиспользовать их для решения новых задач. Этот метод позволяет значительно снизить объем данных, необходимых для обучения модели, и улучшить ее обобщающую способность. В другой работе Pan and Yang (2010) представлен обзор основных теоретических концепций и методов переноса обучения. В ней подчеркивается, что перенос обучения позволяет эффективно использовать знания, накопленные при решении одной задачи, для улучшения производительности модели на связанных задачах, даже если эти задачи имеют разные распределения данных.

Подход, основанный на признаках, (feature-based) и подход, основанный на дообучении, (fine tuning) – два основных подхода к применению переноса обучения для тренировки глубоких нейронных сетей. Feature-based подход заключается в том, что предварительно обученная модель используется для извлечения признаков, которые затем подаются на вход другой модели для решения конкретной задачи. Этот подход основывается на предположении, что признаки, полученные предварительно обученной моделью, содержат в себе общую информацию о данных, которую можно использовать для решения новой задачи. Fine Tuning подход, с другой стороны, заключается в том, что предварительно обученная модель адаптируется под новую задачу путем дообучения на новых данных, относящихся к этой задаче. Этот подход особенно эффективен, когда у нас есть небольшое количество размеченных данных для новой задачи, так как он позволяет эффективно использовать знания, накопленные предварительно обученной моделью, для улучшения производительности на новой задаче.

Таким образом, перенос обучения позволяет улучшить производительность моделей на новых задачах за счет использования знаний, полученных на ранних этапах обучения.

4 Предложенное решение

Выше была описана модель Contriever, которая используется в предложенном решении в качестве базовой. Для достижения высокой производитель-

ности и точности в реальных сценариях эффективным подходом является дообучение базовой модели на конкретных данных. Непосредственно для дообучения базовой модели использовались контрастивная функция потерь InfoNCE и различные техники комбинирования тренировочных данных, описанные подробно ниже. А для оценки итеративной процедуры подборки коллекции релевантных документов был использован статистический алгоритм BM25, который позволяет провести тестирование алгоритма без участия реальных пользователей.

4.1 Обучение

В подразделе 3.1 представлены подробное описание модели Contriever и методика её обучения. В данной работе указанная модель применялась в качестве предварительно обученной и затем дополнительно обучалась на специфических данных. Под процессом дообучения понимается, описанная выше процедура переноса обучения, в частности, fine tuning подход.

Для дополнительного обучения модели были использованы наборы данных, состоящие из троек вида: $(q, d_+, d_-$, где d_+ и d_- — соответственно релевантные и нерелевантные документы для запроса q. При возможности заголовок и аннотация документа объединены следующим образом: $f_\theta(q) = f_\theta(d_{\texttt{title}}[SEP]d_{\texttt{abstract}})$.

Пусть дан закодированный запрос q и набор закодированных документов $\{d_1, d_2, d_3, ...\}$, пусть среди них есть релевантный документ (обозначенный как d_+), который соответствует q. С учетом сходства, измеряемого скалярным произведением, в данной работе используется контрастивная функция потерь InfoNCE, подробно описанная выше:

$$\mathcal{L}_q = -\log \frac{\exp\left(s\left(q, d_+\right)/\tau\right)}{\sum_{i=1}^{K} \exp\left(s\left(q, d_i\right)/\tau\right)}$$

.

Сумма берется по одному релевантному и K нерелевантным документам. В интуитивном плане эта функция потерь представляет собой кросс-энтропийную ошибку классификатора на основе софтмакс с $K+1$ классом, который стремится классифицировать q как d_+.

4.2 Методы комбинирования тренировочных данных

Этот подраздел описывает три способа объединения тренировочных данных, которые используются в процессе обучения модели.

Смешивание внутри пакета (in-batch mixing).

Один из методов агрегации данных заключается в их интеграции внутри пакета: данные из различных наборов случайным образом комбинируются в пределах одного пакета. Каждому из N наборов данных присваивается вес w_i ($\sum_{n=1}^{N} w_i = 1$). Пакет, состоящий из B экземпляров, в среднем содержит $w_i \times B$ примеров из набора данных i. Поскольку негативные примеры

документов общие в пределах одного пакета, интеграция отрицательных примеров становится более разнообразной по сравнению с двумя другими методами агрегации наборов данных.

Смешивание в альтернативных пакетах (alt-mixing).

Другой метод агрегации данных предполагает изменение данных в процессе последовательных итераций обучения. При данном подходе тренировочные данные равномерно распределяются по наборам, однако отсутствует смешивание внутри пакета. В отличие от метода смешивания внутри пакета, где набор негативных документов берется из разных источников, здесь он выбирается из одного и того же.

Дополнительное обучение.

Последний рассмотренный метод в данном исследовании представляет собой дополнительный этап обучения. На первом этапе модель обучается на наборе данных A до достижения сходимости, после чего происходит дополнительное обучение на наборе данных B. Процесс обучения на втором наборе данных обычно короче и использует более низкий темп обучения.

4.3 Итеративная процедура

После дополнительного этапа обучения модели существенным аспектом становится разработка итеративной процедуры, направленной на улучшение точности и релевантности предоставляемых документов путем получения обратной связи от пользователя. Этот процесс позволяет модели адаптироваться к индивидуальным предпочтениям и потребностям конкретного пользователя, что является важным аспектом в контексте персонализированного информационного поиска.

На первом этапе пользователь формулирует запрос q, по которому модель выдает набор документов R, наиболее релевантных данному запросу с учетом своего текущего состояния после дообучения. Пользователь, в свою очередь, осуществляет отбор из этого набора документов $\widetilde{R}$, выбирая те, которые соответствуют его запросу и наиболее интересны или полезны.

Выбранные пользователем документы $\widetilde{R}$ объединяются во временный вектор $\hat{d}$, который представляет собой агрегацию информации из этих документов. Затем этот временный вектор $\hat{d}$ усредняется с вектором запроса q, формируя вектор $\hat{q}$, который отражает обновленные предпочтения пользователя и уточненный запрос.

Полученный вектор $\hat{q}$ затем нормализуется для использования в качестве запроса на следующей итерации. Этот процесс итеративного уточнения позволяет модели постепенно адаптироваться к предпочтениям и потребностям конкретного пользователя, повышая качество и точность предоставляемых результатов информационного поиска.

4.4 Методика оценивания

В данной работе возникает существенная проблема, связанная с необходимостью оценки эффективности предложенного метода на реальных пользователях. Отсутствие подходящих датасетов, способных адекватно моделировать итеративное взаимодействие пользователей, делает данную задачу сложной и затратной. Поэтому возникает необходимость разработки методики оценивания, способной моделировать поведение реальных пользователей.

Предлагается следующий подход к оценке эффективности метода. На каждом шаге t имеется подмножество документов R_t, где $R_t \in D$), а также множество документов, которые пользователь уже оценил как понравившиеся. С целью упрощения процесса будем оценивать только одну итерацию алгоритма.

Процесс оценки проводится следующим образом. Предположим, что имеется выдача алгоритма R_0, а также реальный документ, который пользователь хочет найти $\widetilde{d}$. Предполагается, что пользователь отметит в качестве понравившегося тот документ из множества R_0, который наиболее похож на документ $\widetilde{d}$. Будем использовать для моделирования выбора пользователя статистический алгоритм BM25 Robertson and Zaragoza (2009). Такой выбор алгоритма осуществляется для минимизации вмешательства в результаты влияния нейросетевых контриверсий. Затем осуществляется расчет итоговой метрики.

Предложенная методика оценивания позволяет моделировать поведение реальных пользователей и оценивать эффективность алгоритма на основе их предпочтений.

5 Данные

5.1 Тренировочные данные

Неразмеченные данные, сгенерированные с использованием метода независимого обрезания Izacard et al. (2021) на научном корпусе. Независимое обрезание (independent cropping, IC) подразумевает выбор двух независимых отрезков сегментов (которые могут пересекаться), формирующих тем самым запрос и положительный документ рис. 1b. Отрицательный документ выбирается аналогично из другого документа в корпусе. Авторы статьи Izacard et al. (2021) предполагают, что перекрытие между запросом и документами способствует обучению модели лексическому сопоставлению между запросом и документом.

MS MARCO (Microsoft MAchine Reading COmprehension Bajaj et al. (2016) — это набор данных, который был создан для оценки алгоритмов машинного чтения на основе журналов поисковых запросов Bing. MS MARCO содержит большое количество запросов пользователей и соответствующие им документы, а также релевантные параграфы в этих документах. Главная задача MS MARCO состоит в том, чтобы разработать и оценить модели,

способные понимать и отвечать на вопросы на естественном языке на основе предоставленной информации.

В отличие от некоторых других наборов данных, MS MARCO фокусируется на задаче получения ответа на вопросы с использованием реальных данных из веб-поиска. Это делает его особенно ценным для разработки и оценки алгоритмов, которые могут быть применены к реальным сценариям веб-поиска и информационного поиска.

Каждый запрос в датасете MS MARCO сопровождается несколькими документами, из которых пользователь предположительно может получить ответ на свой вопрос. Это делает задачу еще более реалистичной, так как информационный поиск обычно предоставляет несколько потенциально релевантных документов для любого данного запроса.

В Sentence-BERT Devlin et al. (2019) предоставляется набор данных, состоящих из негативных примеров для этого набора данных. Для обучения были сформированы тройки вида: (q, d_+, d_-), где q - это запрос, d_+ - релевантный документ, а d_- - нерелевантный документ, выбранный на основе оценок модели с ранним связыванием документа с запросом. Для каждой пары «запрос – ответ» было отобрано 5 нерелевантных документов с оценкой модели 3.0 или ниже.

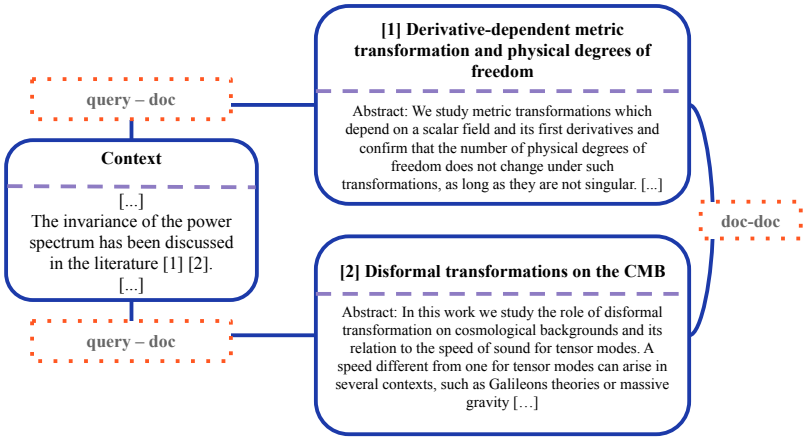


Рис. 3: Описания двух способов извлечения соответствующих пар в UnarXiv.

UnarXiv Saier and Färber (2020) — это крупный научный набор данных с аннотированными цитатами в тексте. Из него были извлечены все контексты, состоящие из одного предложения и содержащие как минимум две статьи arXiv, причем были выбраны только те контексты, для которых процитированные в них статьи размещены на arXiv. Итоговый набор данных содержит коллекцию из 343578 предложений, цитирующих в об-

щей сложности 300736 статей arXiv, соответствующих различным научным областям. Контексты и документы содержат в среднем 29.8 и 152.5 слов соответственно. UnarXiv используется для двух задач:

1. Во-первых, используются совместно цитируемые документы в качестве положительных примеров в настройке поиска «документ-документ» (эта цель называется doc-doc);

2. Во-вторых, используется контекст в качестве запроса для любого из документов, на которые он ссылается. Эта цель называется query-doc, то есть «запрос-документ».

InPars Bonifacio et al. (2022) — это метод генерации синтетических наборов данных для задач информационного поиска. Идея заключается в использовании больших языковых моделей, таких как GPT-3 Brown et al. (2020), для создания запросов, которые соответствуют данному документу. В Brown et al. (2020) были созданы синтетические данные для всех наборов данных, представленных в наборе данных BEIR Thakur et al. (2021).

5.2 Тестовые данные

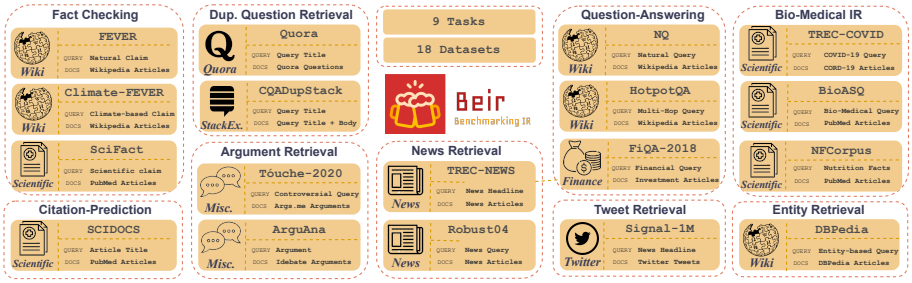


Рис. 4: Обзор различных задач и поднаборов данных в наборе данных BEIR

В качестве данных для оценки был использован набор данных **BEIR** Thakur et al. (2021). Этот набор содержит 18 поднаборов данных для решения задачи информационного поиска. Были выбраны 3 открытых поднабора: SciDocs Cohan et al. (2020); SciFact Wadden et al. (2020) - набор научных данных для проверки фактов из 300 запросов и 5183 документов; ArguAna Wachsmuth et al. (2018) - набор данных контраргументации из 1406 запросов и 8670 документов. За исключением ArguAna, где запросы имеют в среднем 192.98 слов, все остальные выбранные наборы данных BEIR представляют собой задачи по поиску коротких запросов к документам. Выбранный набор данных с самыми длинными запросами - это SciFact с 12.37 слов в среднем на запрос.

6 Вычислительный эксперимент

6.1 Метрика

Методика оценивания была подробно описана в подразделе 4.4. В качестве итоговой метрики в данной задаче используется nDCG@10 (Normalized discounted cumulative gain) Wang et al. (2013), на основе выдачи, сформированной по запросу с учетом отмеченного пользователем документа:

$$DCG@k(q) = \sum_{i=1}^{k} \frac{2^{y\left(q, d_q^{(i)}\right)} - 1}{\log(i+1)},$$

$$nDCG@k(q) = \frac{DCG@k(q)}{\max DCG@k(q)},$$

где $\max DCG@k(q)$ — значение DCG (Discounted cumulative gain) при идеальном ранжировании.

6.2 Постановка эксперимента

Для анализа эффективности предложенного решения требуется провести следующие эксперименты:

1. Дообучение на одну цель: проведение процедуры дополнительного обучения базовой модели Contriever на одном из вспомогательных наборов данных;
2. Дообучение на все цели: проведение процедуры дополнительного обучения базовой модели Contriever на всех вспомогательных наборах данных с использованием различных техник комбинирования данных;
3. Предварительное обучение на неразмеченных данных с последующим дополнительным обучением полученной модели на одном из вспомогательных наборов данных;
4. Предварительное обучение на неразмеченных данных с последующим дополнительным обучением полученной модели на всех вспомогательных наборах данных с использованием различных техник комбинирования данных;
5. Дообучение на синтетических данных: проведение процедуры дополнительного обучения отдельно на наборе данных MS MARCO и на всех данных с использованием техники комбинирования alt mixing.

Гиперпараметры в ходе обучения:

- размер пакета = 32;
- число шагов = 100000;
- при обучении использовалась половинная точность.

6.3 Базовый эксперимент

В качестве базового эксперимента рассматривается генерация ответов с использованием модели Contriever без проведения дополнительного обучения на вспомогательных наборах данных и без применения итеративной процедуры. Результаты данного эксперимента представлены в таблице ниже:

model	scidocs	scifact	arguana	avg
1) Contriever	0.1096	0.5714	0.3353	0.3388

Таблица 1: Базовый эксперимент (ndcg@10)

7 Результаты

7.1 Дообучение на одну цель

model	scidocs	scifact	arguana	avg
1) Contriever	0.1096	0.5714	0.3353	0.3388
Contriever (iterative)	0.1228	0.6957	0.3781	0.3989
2) Contriever + IC	0.1470	0.5703	0.3313	0.3495
Contriever + IC (iterative)	0.1709	0.7159	0.3528	0.4132
3) Contriever + UnarXiv query-doc	0.1520	0.5560	0.3030	0.3370
Contriever + UnarXiv query-doc (iterative)	0.1740	0.7112	0.3437	0.4096
4) Contriever + UnarXiv doc-doc	0.1526	0.4960	0.2753	0.3079
Contriever + UnarXiv doc-doc (iterative)	0.1781	0.6241	**0.3817**	0.3946
5) Contriever + MS MARCO	0.1623	0.6359	0.3657	0.3879
Contriever + MS MARCO (iterative)	**0.1909**	**0.8008**	0.3737	**0.4551**

Таблица 2: Дополнительное обучение модели Contriver на одну цель (ndcg@10)

Можно заметить, что модели, включающие итеративную процедуру для подборки коллекции документов, демонстрируют улучшенное качество по сравнению с аналогичными методами, обученными на тех же наборах данных без использования данной процедуры. Этот факт подтверждает возможность применимости предложенного итеративного подхода. Кроме того, наблюдается явное превосходство модели, обученной на наборе данных MS MARCO, по сравнению с другими моделями на большинстве валидационных наборов. Это объясняется тем, что MS MARCO содержит реальные

запросы пользователей, в отличие от синтетических данных или контекстов, что способствует более точному моделированию реальных сценариев использования. Также стоит отметить, что модель, дообученная на UnarXiv doc-doc и использующая итеративную процедуру, демонстрирует наилучшее качество на поднаборе ArguAna.

7.2 Дообучение на все цели

model	scidocs	scifact	arguana	avg
6) Contriever + alt mixing	0.1819	0.6683	0.3610	0.4037
Contriever + alt mixing (iterative)	**0.2129**	**0.7997**	**0.3715**	**0.4613**
7) Contriever + in-batch mixing	0.1446	0.6466	0.3288	0.3733
Contriever + in-batch mixing (iterative)	0.1612	0.7866	0.3540	0.4339

Таблица 3: Дополнительное обучение Contriver сразу на все цели (ndcg@10)

Отмечается, что использование техники множественного обучения alt-mixing в сочетании с итеративным подходом к обучению модели приводит к лучшим результатам, хотя по некоторым показателям валидации результаты немного уступают модели, обученной с использованием одной цели (MS MARCO). Важно отметить, что данная техника превосходит in-batch mixing в категории обучения без использования итеративной процедуры, что указывает на более подходящий характер метода множественного обучения для решения поставленной задачи. Этот эффект можно объяснить вариативностью обучающих данных и тем самым их оптимальным слиянием не в пределах одного пакета, а по мере прохождения итераций обучения.

7.3 Предобучение на IC + дообучение на одну цель

Данный эксперимент нацелен на иллюстрацию эффекта предварительного обучения модели на неразмеченных данных. В рамках этого эксперимента модель сначала обучается на IC данных, сформированных при помощи независимой обрезки, а затем подвергается дополнительному обучению на целевых задачах. Из результатов экспериментов следует, что добавление дополнительного этапа обучения на неразмеченных данных способствует улучшению качества модели (за исключением поднабора SciFact) по сравнению с отсутствием данной процедуры. Однако результаты данного эксперимента ограничены, поскольку модель Contriever предварительно обучалась на неразмеченных данных, полученных аналогичным образом, что может объяснить относительно невысокий эффект. В случае использования в качестве базовой модели модели, не обученной на подобного рода данных, эффект мог бы быть более выраженным. Как и в предыдущих экспериментах, итеративная процедура демонстрирует улучшение качества модели.

model	scidocs	scifact	arguana	avg
8) Pretrain + UnarXiv query-doc	0.1747	0.5866	0.3372	0.3661
Pretrain + UnarXiv query-doc (iterative)	0.2019	0.7373	0.3471	0.4288
9) Pretrain + UnarXiv doc-doc	0.1629	0.4499	0.2705	0.2944
Pretrain + UnarXiv doc-doc (iterative)	0.1902	0.5671	0.3581	0.3718
10) Pretrain + MS MARCO	0.1759	0.6217	0.3662	0.3879
Pretrain + MS MARCO (iterative)	**0.2066**	**0.7821**	**0.3738**	**0.4542**

Таблица 4: Предварительное обучение модели Contriver на IC, а затем дополнительное обучение на одну цель (ndcg@10)

7.4 Предобучение на IC + дообучение на все цели

model	scidocs	scifact	arguana	avg
11) Pretrain + alt mixing	0.1877	0.6604	**0.3703**	0.4061
Pretrain + alt mixing (iterative)	**0.2196**	**0.8065**	0.3659	**0.4640**
12) Pretrain + in-batch mixing	0.1833	0.6399	0.3557	0.3929
Pretrain + in-batch mixing (iterative)	0.2110	0.7928	0.3445	0.4494

Таблица 5: Предварительное обучение модели Contriver на IC, а затем дополнительное обучение сразу на все цели (ndcg@10)

Аналогично предыдущему эксперименту внедрение дополнительной стадии предварительного обучения на неразмеченных данных приводит к улучшению качества модели. В данном эксперименте использование итеративного процесса и объединение данных с помощью метода alt-mixing показывает наилучшие результаты на поднаборе SciDocs.

7.5 Дообучение на синтетических данных

В настоящем исследовании был проведен аналогичный эксперимент, описанный в работе Bonifacio et al. (2022), с использованием синтетического набора данных InPars. Результаты данного эксперимента свидетельствуют об улучшении качества по сравнению с моделями, обученными на данных из набора MS MARCO, а также на комбинированных данных, сформированных с использованием метода alt-mixing. Возможным направлением для будущих исследований является проведение сравнительного анализа с использованием метода дообучения на синтетических данных, сгенерированных с применением более современных моделей, таких как GPT-4 OpenAI et al. (2023). Также стоит отметить, что использование итеративного про-

model	scidocs	scifact	arguana	avg
15) Pretrain on MS MARCO	0.1671	0.6590	0.3753	0.4004
Pretrain on MS MARCO (iterative)	0.1921	**0.8193**	0.3804	0.4639
16) Pretrain on alt mixing	0.1901	0.6833	0.3710	0.4148
Pretrain on alt mixing (iterative)	**0.2145**	0.8094	**0.3810**	**0.4683**

Таблица 6: Дополнительное обучение на синтетических данных (ndcg@10)

цесса, предварительного обучения на MS-MARCO и последующего дообучения на синтетических данных демонстрирует наилучшие результаты на поднаборе SciFact.

8 Заключение

Результаты данной работы заключаются в следующем:

1. Предложен новый метод для итеративной подборки коллекции релевантных документов;
2. Предложена модель, автоматизирующая процесс генерации информации от пользователя на каждой итерации;
3. Объединение указанных двух предложений привело к возможности преобразования методики оценивания в неитеративную процедуру, что в свою очередь позволяет оценивать модель без привлечения реальных пользователей и представляет собой значимый результат;
4. Продемонстрировано улучшение качества по сравнению с отсутствием итеративной процедуры;
5. Проведены эксперименты по применению различных методов комбинирования тренировочных данных различной структуры, проанализированы их результаты и продемонстрировано улучшение качества.

Список литературы

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V., Kinney, R., Kohlmeier, S., Lo, K., Murray, T., Ooi, H.-H., Peters, M., Power, J., Skjonsberg, S., Wang, L., Willhelm, C., Yuan, Z., Zuylen, M., and oren (2018). Construction of the literature graph in semantic scholar. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), Association for Computational Linguistics.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., and Wang, T. (2016). Ms marco: A human generated machine reading comprehension dataset.

Bonifacio, L., Abonizio, H., Fadaee, M., and Nogueira, R. (2022). Inpars: Data augmentation for information retrieval using large language models.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Chor, B., Gilboa, N., and Naor, M. (1997). Private information retrieval by keywords.

Cohan, A., Feldman, S., Beltagy, I., Downey, D., and Weld, D. S. (2020). Specter: Document-level representation learning using citation-informed transformers.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., and Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, 2(3):267–279.

Hassan, H. A. M. (2017). Personalized research paper recommendation using deep learning. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2019). Momentum contrast for unsupervised visual representation learning.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. (2021). Unsupervised dense information retrieval with contrastive learning.

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering.

Lee, K., Chang, M.-W., and Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Lin, J., Nogueira, R., and Yates, A. (2020). Pretrained transformers for text ranking: Bert and beyond.

Liu, H., Yang, Z., Lee, I., Xu, Z., Yu, S., and Xia, F. (2015). Car: Incorporating filtered citation relations for scientific article recommendation. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 513–518.

Liu, J., Kong, X., Xia, F., Bai, X., Wang, L., Qing, Q., and Lee, I. (2018). Artificial intelligence in the 21st century. *IEEE Access*, 6:34403–34421.

Miah, S. J., Vu, H. Q., Gammack, J., and McGrath, M. (2017). A big data analytics method for tourist behaviour analysis. *Information amp; Management*, 54(6):771–785.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, , Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Kondraciuk, , Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo,

R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., Peres, F. d. A. B., Petrov, M., Pinto, H. P. d. O., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2023). Gpt-4 technical report.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pera, M. S. and Ng, Y.-K. (2013). Exploiting the wisdom of social connections to make personalized recommendations on scholarly articles. *Journal of Intelligent Information Systems*, 42(3):371–391.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Saier, T. and Färber, M. (2020). unarxive: a large scholarly data set with publications' full-text, annotated in-text citations, and links to metadata. *Scientometrics*, 125:3085 – 3108.

Sparck Jones, K., Walker, S., and Robertson, S. (2000). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing amp; Management*, 36(6):779–808.

Sugiyama, K. and Kan, M.-Y. (2011). Serendipitous recommendation for scholarly papers considering relations among researchers. JCDL '11: Joint Conference on Digital Libraries, ACM.

Sun, J., Ma, J., Liu, Z., and Miao, Y. (2013). Leveraging content and connections for scientific article recommendation in social computing contexts. *The Computer Journal*, 57(9):1331–1342.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. (2021). Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models.

Wachsmuth, H., Syed, S., and Stein, B. (2018). Retrieval of the best counterargument without prior topic knowledge. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

Wadden, D., Lin, S., Lo, K., Wang, L. L., van Zuylen, M., Cohan, A., and Hajishirzi, H. (2020). Fact or fiction: Verifying scientific claims. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Wang, Y., Wang, L., Li, Y., He, D., Liu, T.-Y., and Chen, W. (2013). A theoretical analysis of ndcg type ranking measures.

Wu, Z., Xiong, Y., Yu, S., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance-level discrimination.

Xia, F., Liu, H., Lee, I., and Cao, L. (2016). Scientific article recommendation: Exploiting common author relations and historical preferences. *IEEE Transactions on Big Data*, 2(2):101–112.

Xia, F., Wang, J., Kong, X., Wang, Z., Li, J., and Liu, C. (2018). Exploring human mobility patterns in urban scenarios: A trajectory data perspective. *IEEE Communications Magazine*, 56(3):142–149.

Zhao, W., Wu, R., and Liu, H. (2016). Paper recommendation based on the knowledge gap between a researcher's background knowledge and research target. *Information Processing amp; Management*, 52(5):976–988.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2019). A comprehensive survey on transfer learning.