Russian Technological Journal.
2025;13(3):21–43

ISSN 2782-3210 (Print)
ISSN 2500-316X (Online)

УДК 004.8
https://doi.org/10.32362/2500-316X-2025-13-3-21-43
EDN QKUGFZ

ОБЗОР

Методы интеграции знаний для разработки вопросно-ответных систем

Д.В. Радюш [@]

Национальный исследовательский университет ИТМО, Санкт-Петербург, 197101 Россия
[@] *Автор для переписки, e-mail: daniil.radyush@gmail.com*

• **Поступила:** 26.06.2024 • **Доработана:** 13.02.2025 • **Принята к опубликованию:** 27.03.2025

Резюме

Цели. Несмотря на наблюдаемые в последние несколько лет успехи больших языковых моделей, которые способны решать широкий перечень задач, ряд практических проблем остается не до конца решенным. В контексте построения вопросно-ответных систем к таким проблемам можно отнести использование общих знаний и учет причинно-следственных связей. Целью статьи является рассмотрение методов интеграции знаний, которые способны усовершенствовать функционирование больших языковых моделей путем предоставления необходимых сведений и закономерностей из внешних источников.

Методы. В работе осуществляются классификация, анализ и сопоставление методов интеграции знаний, используемых в актуальных реализациях вопросно-ответных систем. В частности, рассматривается вовлечение вспомогательных сведений через самообучение, дообучение, механизм внимания и использование токенов взаимодействия, а также описываются соответствующие вспомогательные подходы для акцентирования наиболее релевантных сведений.

Результаты. Рассмотренные в обзоре вопросно-ответные системы непосредственно демонстрируют возрастание точности относительно базового решения на основе предобученной языковой модели за счет использования методов интеграции знаний на примере бенчмарка CommonsenseQA. При этом в целом более высокие результаты показывают методы интеграции знаний, основанные на использовании языковых моделей и механизма внимания.

Выводы. Представленный систематический обзор существующих методов интеграции знаний из внешних источников в работу вопросно-ответных систем фактически подтверждает эффективность и перспективность этого направления исследований. Данные методы демонстрируют не только возможность увеличить точность вопросно-ответных систем, но и в некоторой степени сгладить проблемы, связанные с интерпретируемостью результатов и устареванием знаний в предобученных моделях. Последующие изыскания способны как улучшить и оптимизировать отдельные аспекты существующих подходов, так и выработать концептуально новые.

Ключевые слова: глубокое обучение, обработка естественного языка, вопросно-ответная система, база знаний, графовые нейронные сети, интеграция знаний

Для цитирования: Радюш Д.В. Методы интеграции знаний для разработки вопросно-ответных систем. *Russian Technological Journal*. 2025;13(3):21–43. https://doi.org/10.32362/2500-316X-2025-13-3-21-43, https://www.elibrary.ru/QKUGFZ

Прозрачность финансовой деятельности: Автор не имеет финансовой заинтересованности в представленных материалах или методах.

Автор заявляет об отсутствии конфликта интересов.

REVIEW ARTICLE

Knowledge injection methods in question answering

Daniil V. Radyush [@]

ITMO University, Saint Petersburg, 197101 Russia
[@] *Corresponding author, e-mail: daniil.radyush@gmail.com*

• **Submitted:** 26.06.2024 • **Revised:** 13.02.2025 • **Accepted:** 27.03.2025

Abstract

Objectives. Despite the recent success of large language models, which are now capable of solving a wide range of tasks, a number of practical issues remain unsolved. For example, users of systems providing question answering (QA) services may experience a lack of commonsense knowledge and reasoning proficiency. The present work considers knowledge injection methods as a means of providing functional enhancements to large language models by providing necessary facts and patterns from external sources.

Methods. Knowledge injection methods leveraged in relevant QA systems are classified, analyzed, and compared. Self-supervised learning, fine-tuning, attention mechanism and interaction tokens for supporting information injection are considered along with auxiliary approaches for emphasizing the most relevant facts.

Results. The reviewed QA systems explicitly show the accuracy increase on the CommonsenseQA benchmark compared to pretrained language model baseline due to knowledge injection methods exploitation. At the same time, in general the higher results are related to knowledge injection methods based on language models and attention mechanism.

Conclusions. The presented systematic review of existing external knowledge injection methods for QA systems confirms the continuing validity of this research direction. Such methods are not only capable of increasing the accuracy of QA systems but also mitigating issues with interpretability and factual obsolescence in pretrained models. Further investigations will be carried out to improve and optimize different aspects of the current approaches and develop conceptually novel ideas.

Keywords: deep learning, nature language processing, question answering system, knowledge base, graph neural networks, knowledge injection

For citation: Radyush D.V. Knowledge injection methods in question answering. *Russian Technological Journal.* 2025;13(3):21–43. https://doi.org/10.32362/2500-316X-2025-13-3-21-43, https://www.elibrary.ru/QKUGFZ

Financial disclosure: The author has no financial or proprietary interest in any material or method mentioned.

The author declares no conflicts of interest.

ВВЕДЕНИЕ

Развитие вопросно-ответных систем в последние годы в существенной степени было обусловлено появлением и последующим совершенствованием предобученных (Pretrained) языковых моделей [1]. Эффективность таких моделей основывается на обработке большого корпуса текстов, содержащего разнородную информацию, что позволяет фиксировать в весах модели как определенные языковые закономерности, так и конкретные факты [2]. Тем не менее, в силу особенностей естественных языков значительное количество сведений об окружающем мире не всегда представлено в тексте в явном виде, что затрудняет их выявление на этапе обучения языковыми моделями. В первую очередь такие сведения касаются разного рода социальных взаимодействий, психологических аспектов и базовых физических законов, которые еще в раннем возрасте усваиваются человеком. В качестве примеров здесь можно привести понимание необходимости смотреть по сторонам при переходе через дорогу или остужать слишком горячую еду перед ее употреблением.

Для того чтобы нивелировать этот недостаток, могут использоваться разного рода источники сведений, в которых такие данные будут зафиксированы в однозначной форме. Одним из первых примеров базы знаний, разработка которой ставила своей целью сбор в т.ч. и общих представлений об окружающем мире, можно считать Cyc[1]. В ней подобные сведения фиксировались в виде логических правил, что соответствовало основному направлению развития приложений из области искусственного интеллекта того времени и должно было обеспечить их дальнейшее совершенствование. К настоящему времени получен целый ряд подобных источников, хотя и с несколько иными подходами к описанию знаний, таких как ATOMIC [3] и ConceptNet[2].

С формальной точки зрения можно выделить несколько доводов в пользу привлечения внешних источников знаний при разработке вопросно-ответных систем. Во-первых, вероятно, главным мотивом является непосредственно получение более точных и удовлетворяющих пользователя результатов. Предполагается, что за счет использования дополнительного контекста к запросу модель сможет ответить на ряд вопросов, для которых внутренних представлений предобученных языковых моделей может быть недостаточно. К подобным вопросам, с одной стороны, можно отнести такие, где опущены отдельные причинно-следственные связи, а, с другой стороны, существует неопределенность в плане идентификации семантики некоторых слов из-за их многозначности и недостаточности контекста. Во многом это определяется ограничениями, выявленными в рамках анализа применения моделей-трансформеров: они в основном используют только поверхностный, статистически наиболее популярный смысл отдельных слов [4], а при логическом выводе в значительной степени опираются на эвристики, усвоенные из обучающей выборки [5].

При этом даже языковые модели с большим количеством весов, демонстрирующие высокие результаты на множестве бенчмарков, могут не просто ошибаться, а еще и выдавать ответы, не имеющие отношения к действительности, – галлюцинации (Hallucinations) [6]. В связи с этим даже сформировалось отдельное направление исследований, посвященное методам извлечения релевантной для запроса информации и включению ее во входные данные для улучшения качества ответов [7] и уменьшения числа галлюцинаций – Retrieval Augmented Generation (расширенная поисковая генерация) [8].

Во-вторых, использование внешних источников знаний способно снизить требования к необходимым вычислительным ресурсам для использования предобученных языковых моделей. В частности, целенаправленное вовлечение дополнительных сведений может позволить применять модели с меньшим числом весов при сохранении точности системы на сопоставимом уровне [9]. Это может упростить работу с вопросно-ответными системами на практике, а также компенсировать издержки на извлечение и обработку вспомогательных данных.

В-третьих, не менее важным является использование баз знаний с позиции интерпретируемого искусственного интеллекта (Explainable Artificial Intelligence). Так, благодаря структурированности баз знаний, извлеченные из них сведения способны воспроизводить своего рода логические цепочки, которые можно предоставлять пользователю в качестве обоснования результата работы системы. Подобное свойство может быть крайне важным с точки зрения практического применения, т.к. нередко именно отсутствие интерпретируемости у результатов, полученных с помощью нейронных сетей, сдерживает их применение в областях, где риск ошибки и соответствующий потенциальный ущерб для общества достаточно велик. Помимо этого, и в целом для контроля за адекватностью функционирования системы желательно иметь наиболее полное представление о ее работе.

Наконец, существенной является проблема обновления фактов, зафиксированных в весах языковых

[1] Cycorp. https://cyc.com. Дата обращения 01.12.2024. / Accessed December 01, 2024.

[2] ConceptNet. An open, multilingual knowledge graph. https://conceptnet.io. Дата обращения 01.12.2024. / Accessed December 01, 2024.

моделей. Обучение таких моделей осуществляется на конкретных наборах данных и обычно занимает довольно длительный промежуток времени. В то же время каждый день происходит огромное количество событий, что приводит к изменению части знаний и появлению новых фактов. Одним из способов решения данной проблемы как раз и может быть извлечение такой информации из внешних баз данных.

В связи с этим достаточно актуальным становится рассмотрение способов использования вспомогательных общих сведений для решения конкретных задач, таких как разработка вопросно-ответных систем. В частности, для успешной работы системы требуется, чтобы полученная информация была достаточна, но при этом не избыточна, т.к. в противном случае это, наоборот, может затруднять ее функционирование и ухудшать соответствующие результаты. Также не меньшее значение имеет то, каким образом обрабатываются дополнительные знания, поскольку это будет во многом определять эффект от их использования в системе. Таким образом, т.к. на процедуру интеграции знаний может влиять существенное количество аспектов, в данной работе представлена попытка систематически проанализировать и сопоставить существующие подходы, чтобы составить полную картину соответствующих идей.

БАЗЫ ЗНАНИЙ

В целом в области разработки вопросно-ответных систем можно выделить несколько направлений в зависимости от особенностей привлечения дополнительных данных. Первое направление подразумевает отсутствие специализированной базы знаний и ориентировано на использование сведений из источников общего назначения – Open Domain Question Answering (поиск ответов на вопросы общего характера). Чаще всего в качестве такого источника может выступать Wikipedia[3], вследствие чего, в силу ее значительного объема и структурной неоднородности содержания, акцент существенно смещается в сторону методов поиска релевантной информации.

Также существует направление Closed Domain Question Answering (поиск ответов на специализированные вопросы), которое имеет дело с более узконаправленными запросами, в т.ч. усложненными необходимостью осуществления логического вывода и учета специфической информации. В связи с этим в качестве внешнего источника знаний могут выступать специализированные базы структурированных знаний, например, графы знаний, что в определенной степени упрощает поиск информации, а также осуществление логического вывода по данным запроса и вспомогательных операций. В качестве соответствующего примера можно привести граф знаний DBpedia[4].

В контексте разработки подходов к интеграции знаний в области вопросно-ответных систем интерес представляют в первую очередь базы структурированных знаний. В какой-то степени это можно обосновать текущим положением вещей в данной области. В частности, появление и последующее развитие предобученных языковых моделей позволило существенно снизить требования к предоставляемому к запросу контексту. В результате в настоящее время на ряде датасетов с учетом дообучения (Fine-tuning) некоторые модели способны показывать результаты, сопоставимые с результатами человека. Вследствие этого основной интерес приходится именно на анализ случаев, в которых человек превосходит существующие вопросно-ответные системы. И как раз основную часть таких случаев, как правило, и составляют запросы, требующие внеконтекстных общих представлений об устройстве окружающего мира, а также анализа причинно-следственных связей между отдельными фактами.

Именно поэтому особенно полезными в таких условиях могут считаться базы структурированных общих знаний. Во-первых, они непосредственно предоставляют системе отсутствующие факты. Во-вторых, эти факты могут быть извлечены с учетом существующих связей между собой и вместе с другой сопутствующей информацией. В-третьих, структурированность знаний значительно упрощает их машинную обработку и, следовательно, использование на практике. Таким образом, представляется возможным в определенной степени одновременно решить проблемы, связанные с частью запросов, которые могут считаться сложными для существующих вопросно-ответных систем.

Как уже отмечалось ранее, в качестве внешнего источника дополнительных данных может использоваться Wikipedia, что сохраняет определенную актуальность. Тем не менее, в силу отсутствия систематизации и большой избыточности информации как альтернатива начала также применяться структурированная база знаний, основанная на сведениях из Wikipedia – Wikidata[5]. Граф Wikidata состоит из более чем 100 млн записей, некоторым образом описывающих элементы человеческого знания. Каждому элементу графа соответствует определенный набор свойств, характеризующих его и устанавливающих

[3] https://www.wikipedia.org/. Дата обращения 01.12.2024. / Accessed December 01, 2024.

[4] The DBpedia Knowledge Base. https://www.dbpedia.org. Дата обращения 01.12.2024. / Accessed December 01, 2024.

[5] Wikidata. The free knowledge base. https://www.wikidata.org. Дата обращения 01.12.2024. / Accessed December 01, 2024.

его взаимосвязи с другими элементами. Таким образом, как и для других графов знаний, содержание Wikidata можно представить в виде набора так называемых триплетов «субъект-предикат-объект», где объект в данном случае является набором конкретных значений свойств, либо ссылкой на другую сущность.

Также к наиболее часто используемым источникам знаний в контексте построения вопросно-ответных систем относится база знаний ConceptNet. Данная база знаний, помимо уникальных сведений общего характера, частично включает в себя информацию и из других относительно часто используемых источников, таких как упомянутые ранее Cyc и DBpedia. В рамках ConceptNet слова и словосочетания группируются на основании нескольких десятков отношений. По сравнению с рассмотренной ранее Wikidata, ConceptNet содержит более 30 млн записей, хотя при этом надо учитывать, что значительная часть этой величины обусловлена наличием фактически дублирующих записей из-за существования аналогов на другом языке, однокоренных слов и симметричных взаимосвязей. Кроме того, чуть больший акцент в ConceptNet приходится на лингвистические свойства, например, за счет фиксирования для слова синонимов, антонимов и этимологически связанных слов. Наконец, особенностью ConceptNet является существование весов у каждого отношения между элементами, что эвристически отражает степень вероятности или важности данного отношения.

Из относительно недавно появившихся баз общих знаний можно также выделить ATOMIC, содержащую более 1 млн элементов. Особенностью ATOMIC является отражение сведений в виде абстрактных событий и их результатов, что позволяет делать акцент на комплексные причинно-следственные связи, существующие в окружающем мире. В частности, например, исходя из некоторого события в ATOMIC, возможно выявить какие-либо его последствия, а также намерение, желание или характеристику одного из участников, что способно предоставить модели потенциально отсутствующие знания о социальных взаимодействиях.

В табл. 1 представлены примеры информации, которая может быть извлечена из рассмотренных выше баз знаний, что в целом говорит об их определенной схожести, за исключением обладающей более специфичными целями базы ATOMIC.

Таблица 1. Примеры извлекаемой из баз знаний информации

База знания	Пример данных
DBpedia	DBpedia subject SemanticWeb (DBpedia субъект Семантической паутины)
Wikidata	Wikidata uses semantic technology (Wikidata использует семантические технологии)
ConceptNet	ConceptNet motivated by goal let computers understand what people already know (ConceptNet ставит задачей позволить компьютерам понимать то, что люди уже знают)
ATOMIC	Person X pays Person Y a compliment. Person X wanted to be nice (Персона X делает комплимент Персоне Y. Персона X хочет казаться приятной)

Рис. 1. Классификация методов интеграции знаний

МЕТОДЫ ИНТЕГРАЦИИ ЗНАНИЙ

Основываясь на анализе актуальных исследований по теме, автор разработал классификацию методов интеграции знаний, представленную на рис. 1. Согласно ей, возможно рассматривать основные идеи методов интеграции знаний в контексте разработки вопросно-ответных систем с приведением соответствующих примеров и с учетом особенностей конкретных выделенных классов методов. Основное место в данной классификации занимает разделение методов интеграции знаний по использованию баз знаний, при этом под использованием понимается вовлечение сведений непосредственно при получении ответов на запросы, что исключает случаи привлечения баз знаний в процессе предобучения моделей. В свою очередь, для извлечения признаков из данных баз знаний могут использоваться как языковые, так и графовые модели.

МЕТОДОЛОГИЧЕСКАЯ ОСНОВА

Несмотря на различия в использованных подходах к интеграции знаний, можно выделить также и общую методологическую основу в рассмотренных далее вопросно-ответных системах. В частности, это

касается как самой постановки задачи, так и применяемых вспомогательных методов.

Использование в работе системы дополнительного контекста создает определенную специфику с точки зрения функционирования. В связи с этим также начинают приобретать значительную важность такие вспомогательные этапы, как извлечение релевантных данных к запросу. В общем виде этот этап подразумевает определение в поступившем запросе некоторого количества сущностей n: $(q_1, \ldots, q_n)$. Для этого в настоящее время на практике преимущественно используются классические методы из области обработки естественного языка, такие как лемматизация и частеречная разметка. Последующая же часть процесса может варьироваться в зависимости от конкретной задачи.

Во многих работах, посвященных интеграции знаний в вопросно-ответных системах, подразумевается наличие у вопроса вариантов ответа. Соответственно, целью системы становится оценка вероятности каждого ответа и выбор наиболее вероятного. Это позволяет существенно упростить и унифицировать построение и оценивание систем. Поэтому в таких случаях будем считать, что выявленным n сущностям соответствуют m аналогично извлеченных сущностей из вариантов ответа: $(a_1, \ldots, a_m)$.

На следующем шаге в работу включается некоторая база знаний, которую можно формализовать как $G = (V, E)$, где V – набор сущностей в базе знаний, а $E \subseteq V \times R \times V$ – множество триплетов базы знаний вида «сущность-отношение-сущность». На практике устоявшейся формой представления подобных баз знаний является граф. На основании этого между сущностями, определенными в контексте используемой базы знаний, представляется возможным построить множество путей вида:

$$p = ((q_i, r_l, v_l), (v_l, r_{l+1}, v_{l+1}), \ldots, (v_{k-1}, r_k, a_j)),$$

где $i \in (1, \ldots, n), j \in (1, \ldots, m)$, k – длина пути в графе, $l \in (1, \ldots, k)$, q_i – i-я сущность из запроса, a_j – j-я сущность из ответа, а v_l и r_l – соответственно l-я по счету в пути сущность и отношение в графе.

Впоследствии подграф базы знаний или совокупность путей используются в качестве дополнительного контекста для определения наиболее вероятного ответа.

Одной из основных проблем при такой постановке является определение наиболее релевантной по отношению к запросу информации. Возможным инструментом для ее решения может служить механизм внимания (Attention) [10], который позволяет вычислять так называемые веса внимания (Attention Weights), количественно оценивающие степень значимости той или иной информации из контекста. Формально механизм внимания определяется выражением:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^{\mathrm{T}}}{\sqrt{d_{\text{model}}}}\right) \cdot \mathbf{V} = \\ = \textbf{Attention weights} \cdot \mathbf{V}, \quad (1)$$

где $\mathbf{Q}$ = Query = $\mathbf{X} \times \mathbf{W}_{\text{q}}$, $\mathbf{K}$ = Key = $\mathbf{X} \times \mathbf{W}_{\text{k}}$, $\mathbf{V}$ = Value = $\mathbf{X} \times \mathbf{W}_{\text{v}}$, $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{model}}}$ – векторное представление входных данных, $\mathbf{W}_{\text{q}} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_{\text{k}} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_{\text{v}} \in \mathbb{R}^{d_{\text{model}} \times d_v}$, N – количество векторов во входных данных, d_{model}, d_k, d_v – размерности векторных представлений в модели и матрицах $\mathbf{K}$ и $\mathbf{V}$, а $\text{softmax}(X_i) = \dfrac{\exp(X_i)}{\sum_{j=1}^{N} \exp(X_j)}$.

Таким образом, веса внимания при умножении на векторное представление контекста способны скорректировать нужным образом влияние отдельных его элементов на результат. На практике чаще всего для учета разных аспектов данных в рамках механизма применяется несколько групп различных матриц (так называемых *голов*), а результат их применения конкатенируется и проецируется в нужную размерность с помощью еще одной матрицы, что получило название Multi-Head Attention (многоголовое внимание):

$$\text{Multi-Head Attention} = \\ = \text{Concatenation}(\mathit{Attention}_1, \ldots, \mathit{Attention}_z) \times W_{\text{o}}, \quad (2)$$

где $\mathit{Attention}_i$ – i-й результат блока Attention, $W_{\text{o}} \in \mathbb{R}^{zd_v \times d_{\text{model}}}$, z – количество голов внимания.

Механизм внимания играет большую роль во многих моделях глубокого обучения и широко используется при разработке подходов для интеграции знаний. Также в связке с Multi-Head Attention часто используется Feed Forward Neural Network (нейронная сеть прямого распространения), что в совокупности составляет основную часть модели, называемой трансформером. В качестве конкретной реализации Feed Forward Neural Network может выступать, например, многослойный перцептрон, а с практической точки зрения роль этой составляющей трансформера рассматривается в контексте хранения и извлечения усвоенных в процессе обучения закономерностей.

МЕТОДЫ ИНТЕГРАЦИИ ЗНАНИЙ БЕЗ ИСПОЛЬЗОВАНИЯ БАЗ ЗНАНИЙ

В общем случае вовлечение вспомогательных знаний не обязательно подразумевает использование определенных баз знаний. Так, в качестве информации, способствующей получению более точного ответа на запрос, могут выступать похожие примеры с указанием правильного ответа. Например, в работах [11]

и [12] демонстрируется положительный эффект от добавления во входные данные запросов из обучающей выборки на основе близости их векторных представлений эмбеддингу[6] исходного запроса.

Другой тип подходов отталкивается от идеи непосредственно обращаться к усвоенным в процессе предобучения модели сведениям: извлекать их в зависимости от запроса и применять в качестве дополнительных входных данных. В [13] для этого предлагается задавать предобученной модели уточняющие вопросы с помощью шаблонов, а ответы на них использовать в качестве полезного контекста. Аналогичный подход в [14] также подразумевает вовлечение в работу вопросно-ответной системы сгенерированных по запросу вспомогательных данных. При этом для генерации знаний могут применяться и специально обученные модели, как это происходит в исследовании [15], в котором подобная модель создает структурированную информацию в формате путей по базе знаний. Таким образом, рассмотренные выше подходы отталкиваются от идеи предоставления на вход дополнительных сведений, для получения которых могут использоваться предобученные и другие вспомогательные модели, тогда как для работы вопросно-ответной системы в целом может не требоваться дообучение языковых моделей.

Количественно самая обширная группа подходов исходит из концепции предварительного обучения моделей. Множество экспериментальных результатов подтверждают идею, что модели с большим числом весов, обученные на как можно более значительном объеме разноплановой информации, способны показывать лучшие результаты при их последующей адаптации к конкретным условиям [16]. Данная концепция во многом опирается на методологию самообучения (Self-Supervised Learning), которая позволяет извлекать представления из корпусов текстов без необходимости их предварительной разметки. Для осуществления этого разрабатываются специальные задачи, в соответствии с которыми модель и обучается. В частности, в рамках разработки языковой модели BERT[7] [1] использовались две такие задачи. Первой задачей является предсказание слов в предложении, маскированных специальным токеном. В рамках этой задачи с вероятностью 15% выбираются некоторые токены из предложения, затем 80% из этих токенов маскируются, 10% – заменяются на случайный токен, а остальные 10% – остаются без изменений. В качестве меры ошибки при предсказании моделью маскированного токена может использоваться кросс-энтропия:

$$\text{Cross-entropy} = -\frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i \times \log(\overline{\mathbf{y}}_i), \qquad (3)$$

где N – общее число примеров; $\mathbf{y}_i$ – one-hot-вектор[8], кодирующий правильный ответ для i-го примера; $\overline{\mathbf{y}}_i$ – вектор-предсказание модели для i-го примера, обозначающий вероятность соответствия каждому возможному варианту ответа в рамках задачи.

Вторая задача касается определения правильного порядка двух предложений в тексте. Это реализуется через добавление во входные данные в процессе обучения специального токена [CLS], репрезентирующего информацию из предложения в целом, и сводится к задаче бинарной классификации. Цель данной классификации – определение того, является ли некоторое предложение *B* продолжением для предложения *A* на основе их итоговых векторных представлений на выходе модели для токенов [CLS]. При этом в рамках обучения в 50% случаев *B* является случайным предложением, а в других 50% – правильным продолжением, тогда как мерой ошибки для задачи также может служить кросс-энтропия. Итоговая ошибка модели при обучении рассматривается как сумма ошибок по каждой задаче. Общая схема обучения модели BERT представлена на рис. 2:

На этапе предобучения (Pretraining) часть токенов пары предложений (Unlabeled Sentence A and B Pair) маскируется, после чего эмбеддинги ($E_{[CLS]}$, E_1, …, E_N, $E_{[SEP]}$ и E_1', …, E_M') токенов (*Tok* 1, …, *Tok N* и *Tok* 1, …, *Tok M*) предложений A (Masked Sentence A) и B (Masked Sentence B) с добавлением обобщающего и разделительного токенов ([CLS] и [SEP]) поступают на вход трансформера BERT. Полученные итоговые эмбеддинги (C, T_1, …, T_N, T_{SEP}, T_1', …, T_M') используются для предсказания маскированных токенов (Mask LM, Language Modeling) и порядка предложений (NSP, Next Sentence Prediction). На этапе дообучения (Fine-Tuning) в зависимости от задачи (MNLI[9], NER[10], SQuAD[11])

[6] Embedding – векторное представление. [Embedding means a vector representation.]

[7] Bidirectional encoder representations from transformers.

[8] One-hot вектор – бинарный вектор, в котором только один элемент имеет значение 1, а остальные равны 0. [One-hot vector is a binary vector in which only one element has the value 1, and remaining elements are equal to 0.]

[9] Multi-Genre Natural Language Inference – датасет для задачи Natural Language Inference – установление логической взаимосвязи между фрагментами текста. [Multi-Genre Natural Language Inference – dataset for the Natural Language Inference task – establishes a logical relationship between the text fragments.]

[10] Named-entity recognition – задача распознавания в тексте именованных сущностей. [Named Entity Recognition is the task of recognizing named entities in the text.]

[11] Stanford Question Answering Dataset – вопросно-ответный датасет, подразумевает автоматическое получение ответов на вопросы на естественном языке. [Stanford Question Answering Dataset is a QA dataset, which implies automatic answers to questions in natural language.]

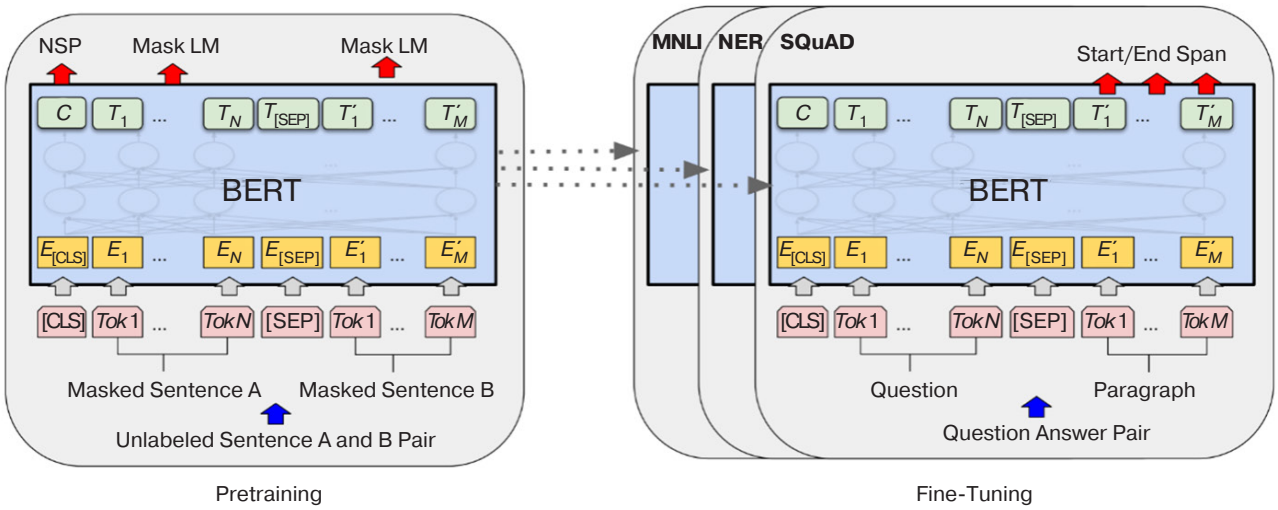

Рис. 2. Схема обучения модели BERT [1]

меняется формат входных данных и предсказываемого. В случае вопросно-ответного датасета SQuAD на вход подается вопрос (Question) и соответствующий контекст (Paragraph), а на выходе предсказывается положение в контексте правильного ответа (Start/End Span).

Впоследствии данная методология видоизменялась и адаптировалась в рамках предобучения и других языковых моделей. В контексте вопросно-ответных систем было разработано множество подходов, основанных на модификации и расширении задач для самообучения BERT или замене их на другие. В общем и целом, при создании такого рода моделей чаще всего модифицируется процедура маскирования за счет введения ограничений на то, что должно маскироваться в предложении, а также изменения параметров маскирования при обучении.

Одной из первых и наиболее значимых разработок в этом направлении стала модель Enhanced Language Representation with Informative Entities (ERNIE) [17], схема которой изображена на рис. 3. Основная ее идея заключается в том, что, если также предобучать предсказывать выявленные в тексте на основе базы знаний маскированные именованные сущности (Entities) в рамках дополнительной задачи для самообучения, то это способно улучшить понимание языка моделью, а также контекстуализировать ее определенные знания о мире. В частности, для этой цели в 5% случаев для токенов текста соответствующая им именованная сущность заменяется на случайную, а в 15% случаев – сущность маскируется, и ее требуется предсказать по текстовым токенам. Кроме того, в работе вводится механизм взаимодействия между эмбеддингами сущностей и соответствующих токенов текста, что позволяет привнести дополнительную информацию в оба векторных представления, увеличив тем самым точность предсказания правильных токенов. С этой целью вводится промежуточное векторное представление, объединяющее информацию на уровне токенов и именованных сущностей, за счет которого впоследствии обновляются исходные эмбеддинги токенов и сущностей, что задается выражениями:

$$\mathbf{h}_j = \sigma(\widetilde{\mathbf{W}}_\mathrm{t}\widetilde{\mathbf{w}}_j + \widetilde{\mathbf{W}}_\mathrm{e}\tilde{\mathbf{e}}_k + \tilde{\mathbf{b}}),$$

$$\mathbf{w}_j = \sigma(\mathbf{W}_\mathrm{t}\mathbf{h}_j + \mathbf{b}_\mathrm{t}), \qquad (4)$$

$$\mathbf{e}_k = \sigma(\mathbf{W}_\mathrm{e}\mathbf{h}_j + \mathbf{b}_\mathrm{e}),$$

где $\mathbf{h}_j$ – агрегированное векторное представление для токена с номером j, σ – заданная нелинейная функция активации, $\widetilde{\mathbf{w}}_j$ и $\mathbf{w}_j$ – векторные представления токена j до и после интеграции знаний, $\tilde{\mathbf{e}}_k$ и $\mathbf{e}_k$ – векторные представления соответствующей токену j именованной сущности k до и после интеграции знаний, $\widetilde{\mathbf{W}}$, $\mathbf{W}$, $\tilde{\mathbf{b}}$ и $\mathbf{b}$ – параметры модели.

В процессе обучения модели ERNIE поступившие на вход эмбеддинги токенов текста (Token Input) проходят через N слоев трансформера (T-Encoder), после чего вместе с эмбеддингами именованных сущностей (Entity Input) они обрабатываются M слоями агрегатора (K-Encoder). На каждом слое i агрегатора эмбеддинги сущностей (e_1 и e_2) и текста (w_1, …, w_n) проходят через свой блок Multi-Head Attention, а соответствующие обновленные эмбеддинги сущностей ($\tilde{e}_1$ и $\tilde{e}_2$) и текста ($\tilde{w}_1$, …, $\tilde{w}_n$) попадают в блок интеграции знаний (Information Fusion), на выходе которого, согласно формулам (4), получают эмбеддинги сущностей (Entity Output) и текста (Token Output) с учетом интеграции знаний.

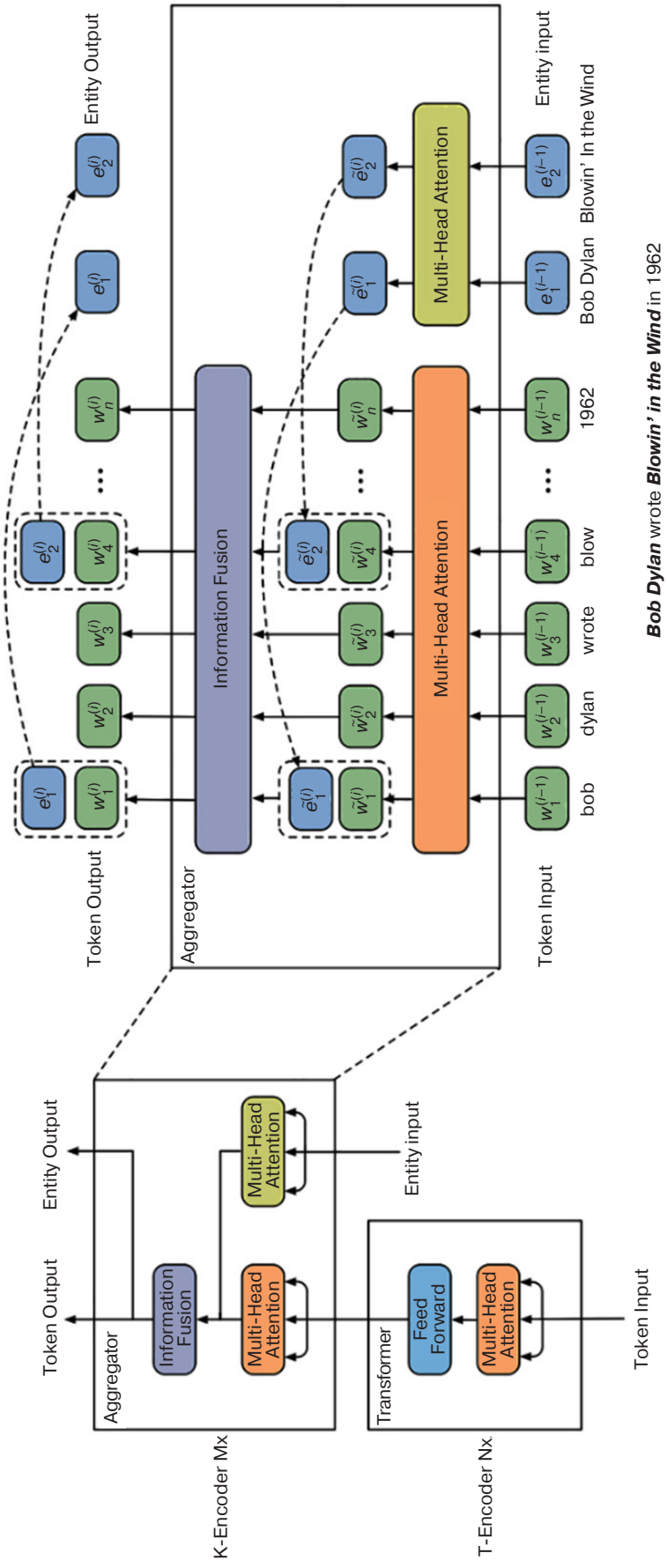

Рис. 3. Схема обучения модели ERNIE [17]: (а) архитектура модели, (б) архитектура агрегатора

Схожий метод лежит в основе модели KnowBERT [18], однако вовлечение внешней информации происходит на уровне векторных представлений сущностей, которые обновляются с помощью механизма внимания и за счет добавления предобученных эмбеддингов сущностей из базы знаний, что впоследствии оказывает влияние и на эмбеддинги всех токенов через механизм внимания, согласно формуле:

$$\mathbf{H}_i' = \text{MLP}(\text{MHA}(\mathbf{H}_i, \mathbf{S}'^e, \mathbf{S}'^e), \quad (5)$$

где $\mathbf{H}_i'$ – векторное представление токена *i* после интеграции знаний, MLP – многослойный перцептрон (Multi-layer Perceptron), MHA – Multi-Head Attention, $\mathbf{H}_i$ – векторное представление токена *i* до интеграции знаний, $\mathbf{S}'^e$ – обновленные векторные представления выявленных именованных сущностей.

Концептуально схожая с ERNIE архитектура предложена в работе [19], основным ее отличием является использование информации об отношениях между сущностями, предсказание которых представлено отдельной задачей для предобучения. В рамках метода Weakly Supervised Knowledge-Pretrained Language Model [20] вместо маскирования сущностей на этапе предобучения модели в качестве дополнительной задачи модель учат предсказывать, были ли сущности во входных данных заменены на другие такого же типа в рамках базы знаний Wikidata. Архитектура модели при этом соответствует BERT, однако маскирование токенов осуществляется только в 5%, а не в 15%, чтобы избежать маскирования слишком большого фрагмента контекста, т.к. сущности могут состоять из нескольких слов. В работе [21] вдобавок к маскированию слов и именованных сущностей применяется и маскирование словосочетаний, что улучшает понимание сочетаемости слов моделью, а интеграция осуществляется поэтапно: на каждом этапе модель, подобная BERT, тренируется только на одном типе маскирования. Использование нескольких режимов обучения, при которых модель переключается с предсказания слов на предсказание фраз в зависимости от того, в каком режиме между двумя последними последовательными итерациями наблюдалось наибольшее снижение ошибки модели относительно общего снижения ошибки за все итерации, является основным новшеством [22].

Авторы исследования [23] предобучают модель на основе BERT, ставя целью обучение предсказанию маскированных сущностей по их описанию, а также сближение векторных представлений описаний синонимичных сущностей и отдаление антонимичных, для чего используется специальная функция потерь:

$$L = -\sum \log \frac{f(\mathbf{h}_{\text{ori}}, \mathbf{h}_{\text{syn}})}{f(\mathbf{h}_{\text{ori}}, \mathbf{h}_{\text{syn}}) + f(\mathbf{h}_{\text{ori}}, \mathbf{h}_{\text{ant}})}, \quad (6)$$

где $f(\mathbf{h}_i, \mathbf{h}_j) = \exp(\mathbf{h}_i \mathbf{h}_j)$, $\mathbf{h}_{\text{ori}}$ – векторное представление описания маскированной сущности, $\mathbf{h}_{\text{syn}}$ – векторное представление описания синонимичной сущности, $\mathbf{h}_{\text{ant}}$ – векторное представление описание антонимичной сущности.

Для решения конкретных задач эта модель используется в паре с BERT в качестве дополнительного источника знаний в форме векторных представлений выявленных сущностей, а их инъекция осуществляется через конкатенацию выходов модели с выходами BERT с опциональным применением механизма внимания для учета важности данных по конкретным сущностям. При этом рассматривается использование механизма внимания как для выходных представлений с последних слоев моделей, так и для выходных представлений по слоям моделей с применением усреднения, а результат работы механизма внимания конкатенируется с выходом модели BERT вместо выхода вспомогательной модели. Лингвистические признаки играют важную роль и в работе [24], в которой дополнительной задачей для самообучения является определение семантической близости пары слов, тогда как обучается модель, чередуя задачи для самобучения BERT с дополнительной. Схожая идея представлена в статье [25], где на основе данных WordNet[12] модель тренируется классифицировать слова по группам со схожим значением.

Логичным развитием подходов с маскированием сущностей и отношений является использование на этапе предобучения моделей задачи предсказания структурированных единиц знания в форме триплетов, что может позволить усвоить более общие принципы и взаимосвязи, подобные содержащимся в базах знаний. Так, в модели Knowledge Embedding and Pretrained Language Representation [26] векторные представления элементов триплета рассматриваются как векторные представления их описаний из базы знаний, полученные с помощью той же модели, что используется для генерации векторных представлений токенов текста в задаче предсказания маскированных токенов. При этом для расчета ошибки в задаче предсказания триплетов применяется скоринговая функция из модели для получения эмбеддингов графов знаний TransE [27]:

[12] Лексическая база данных английского языка, разработанная в Принстонском университете. https://wordnet.princeton.edu/. Дата обращения 01.12.2024. [A lexical database of the English language developed at Princeton University. https://wordnet.princeton.edu/. Accessed December 01, 2024.]

$$d(\mathbf{h}, \mathbf{t}) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||, \quad (7)$$

где **h** – векторное представление субъекта в триплете, **r** – векторное представление отношения в триплете, **t** – векторное представление объекта в триплете.

В работе [28] на вход модели подается набор триплетов из одного подграфа, и поэтому в рамках механизма внимания дополнительно используется матрица смежности для учета существующих взаимосвязей, а обучение проводится в формате восстановления триплетов, что подразумевает составление триплетов из обновленных представлений вершин и использование скоринговой функции (7). Исследование [29] содержит идею предобучения трех функций на основе модели-энкодера для предсказания каждого элемента триплета по двум другим, что должно способствовать выучиванию возможных сочетаний. В этой постановке оценка ответа является произведением оценок схожести значений трех предобученных функций и соответствующих реальных элементов триплета, у которого субъект представляет собой контекст вопроса, отношение – сам вопрос, а объект – конкретный вариант ответа. Предобучение функции, которая будет участвовать в нахождении ответов на запросы, определяя наиболее вероятные взаимосвязи со вспомогательными данными из базы знаний, предлагается в статье [30]. С помощью этой функции извлеченные вспомогательные факты к каждому варианту ответа сопоставляются по степени схожести с фактами для вопроса, и более вероятным ответом считается тот, для которого эта схожесть фактов в среднем окажется выше.

В то же время триплеты из релевантного подграфа к запросу могут непосредственно подаваться на вход модели в рамках предобучения наравне с обычными текстовыми токенами с применением специальных эмбеддингов для указания типа токена, как это показано в [31]. В связи с этим при реализации механизма внимания в модели используется матрица-маска, ограничивающая взаимодействие несвязанных в подграфе вершин. В работе [32] было предложено усовершенствовать подход модели ERNIE путем модифицирования представления сущностей за счет учета их связей в соответствующем подграфе, а также использования механизма внимания для отфильтровывания потенциально нерелевантного контекста для запроса.

Другим способом использования баз знаний на этапе предварительного обучения модели может быть построение на их основе новых вопросно-ответных датасетов, с помощью которых система также определенным образом улучшает свои возможности находить верные ответы. Такой подход используется в [33], а в [34] его развивают с помощью концептуализации: конкретные факты рассматриваются в более общем ключе, благодаря чему можно охватить большее количество ситуаций и усовершенствовать способность различать схожие варианты. Например, с помощью базы ATOMIC «игра в футбол» может быть представлена как «утомительное событие».

Концепция модели Self-supervised Bidirectional Encoder Representation Learning of Commonsense [35] в большей степени акцентирована на количественном расширении числа задач для самообучения. Для того чтобы улучшить возможности системы при обработке трудных запросов, к задачам для самообучения BERT в ней было добавлено еще 3: первая нацелена на различение контекста с противоположным смыслом; вторая требует расстановки по порядку нескольких перемешанных предложений, взятых из одного абзаца; третья расширяет усвоение контекстных взаимосвязей через маскирование сущностей. По задумке авторов такой подход также позволит системе лучше улавливать языковые закономерности и обеспечит появление более универсальных реализаций.

Другая концепция интеграции знаний подразумевает в качестве дополнительного шага дообучение на основе соответствующих практической задаче существующих датасетов. Например, определенное распространение в связи с этим получило использование датасета SQuAD [36] из области построения вопросно-ответных систем. К его ключевым особенностям можно отнести относительно существенный размер (более 100000 запросов), при том, что к каждому запросу прилагается соответствующий контекст, взятый из Wikipedia. Таким образом, в результате обучения на данном датасете модель лучше адаптируется к постановке и формату задачи, а также в дополнение обрабатывает достаточно значительный объем данных, увеличивая тем самым количество усвоенных фактических сведений.

В качестве актуального и характерного примера в связи с этим можно упомянуть модель UnifiedQA [37], разработка которой строилась через дообучение языковой модели на 8 вопросно-ответных датасетах разного типа, что позволяет адаптироваться к существующим форматам бенчмарков и обеспечивает прирост точности модели на неиспользованных в процессе обучения вопросах, открывая также и новые возможности для последующего ее дообучения. Целесообразность подобного подхода была подтверждена и для модели Unicorn из [38], однако в данном случае область исследования была ограничена исключительно датасетами, подразумевающими использование общих знаний (Commonsense Question Answering).

Рассмотренные выше методы можно отнести к классу подходов без явного привлечения баз

знаний, т.к. при использовании соответствующих систем не подразумевается непосредственное извлечение контекста к запросу именно из баз знаний, а акцент создается на знаниях, которые были получены в процессе обучения. По сути, разработанные в последние несколько лет большие языковые модели в существенной степени основываются на схожей идее: обучение на большом количестве качественных данных, учитывающих различную специфику и человеческие предпочтения, способно повышать универсальность систем и оценку получаемых с их помощью результатов при достаточном масштабировании этого процесса.

Среди преимуществ данного класса можно назвать в некотором смысле большую универсальность благодаря независимости от использования баз знаний. Кроме того, существенная опора в архитектуре вопросно-ответной системы на предобученные и дообученные модели позволяет несколько упростить ее разработку, последующее использование и адаптацию к конкретным задачам.

В то же время этот класс подходов может считаться в определенной степени ограниченным в своих возможностях для дальнейшего развития. Дело в том, что в основном прирост эффективности моделей здесь связан с целенаправленным и точечным улучшением, расширением объема усвоенных сведений, тогда как никаких принципиально новых механизмов, улучшающих способности системы к рассуждению, не вводится. Кроме того, в рамках данного направления практически никак не решается проблема отсутствия интерпретируемости у полученных ответов и их обоснования системой, а также устаревания знаний.

МЕТОДЫ ИНТЕГРАЦИИ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ БАЗ ЗНАНИЙ

Основной единицей информации в базах знаний можно считать триплеты «сущность-отношение-сущность», тогда как более сложные взаимосвязи могут передаваться набором триплетов – путем. Помимо путей или вместо них могут использоваться также подграфы базы знаний, которые можно рассматривать как совокупность путей, имеющих общие элементы. В результате с точки зрения обработки информации в наличии у вопросно-ответной системы может оказаться 3 типа признаков в том или ином сочетании:

1) признаки, полученные обработкой контекста запроса языковой моделью;
2) признаки, основанные на извлеченных путях;
3) признаки, связанные с подграфами базы знаний.

Таким образом, одним из направлений для исследований являются способы эффективной обработки этих разных типов признаков и методы объединения соответствующих результатов.

Один из подходов к интеграции признаков в работу системы исходит из того, что триплеты и их совокупности зачастую достаточно просто могут быть транслированы в предложения на естественном языке, и в такой форме их представляется возможным подать на вход языковой модели в качестве вспомогательного контекста. При этом нужно отметить, что в такой форме они могут служить также и в качестве обоснования для получаемого ответа. Примером реализации подобного подхода может служить модель Knowledge-Augmented language model PromptING (KAPING) [39], а в работе [40] его эффективность исследуется в условиях использования предобученных на вспомогательных датасетах языковых моделей. В модели DEscriptive Knowledge for COmmonsense question answeRing (DEKCOR) [41], помимо извлеченных из ConceptNet триплетов, на вход подаются и словарные определения соответствующих сущностей, тогда как в Knowledgeable External Attention for commonsense Reasoning (KEAR) [42] (рис. 4) контекст к запросу расширяется за счет привлечения дополнительной информации из ряда вопросно-ответных датасетов, что позволяет использовать более специфические сведения.

В частности, в рамках архитектуры KEAR к конкатенации вопроса с одним из вариантов ответов (Question & Candidate) на вход модели добавляются соответствующие извлеченные вспомогательные данные (Knowledge Retrieval) из базы знаний ConceptNet, словаря Wiktionary (Definition) и дополнительных датасетов (Training Data). На вход трансформера подаются эмбеддинги токенов запроса ($E_{[CLS]}$, E_0, ..., E_N), к которым добавляется индикатор сегмента (S_0) и вспомогательного контекста (E_0^k, ..., $E_{N_k}^k$), к которым добавляется индикатор сегмента (S_1). Вероятность ответа (Score Prediction) определяется на основании итогового эмбеддинга вспомогательного токена $E_{[CLS]}$, полученного с помощью механизма внимания (Self-Attention / External Attention).

Преимуществом интеграции знаний через языковые модели является возможность в значительной степени опираться на эффективность предобученных моделей, при этом в некоторых случаях даже избегая необходимости изменять их веса (например, модель KAPING). Также вычислительные затраты на получение дополнительных признаков могут считаться относительно небольшими. В то же время, т.к. большинство предобученных моделей могут эффективно использовать только фиксированное количество информации из входных данных, возникает необходимость ограничивать количество менее релевантных сведений.

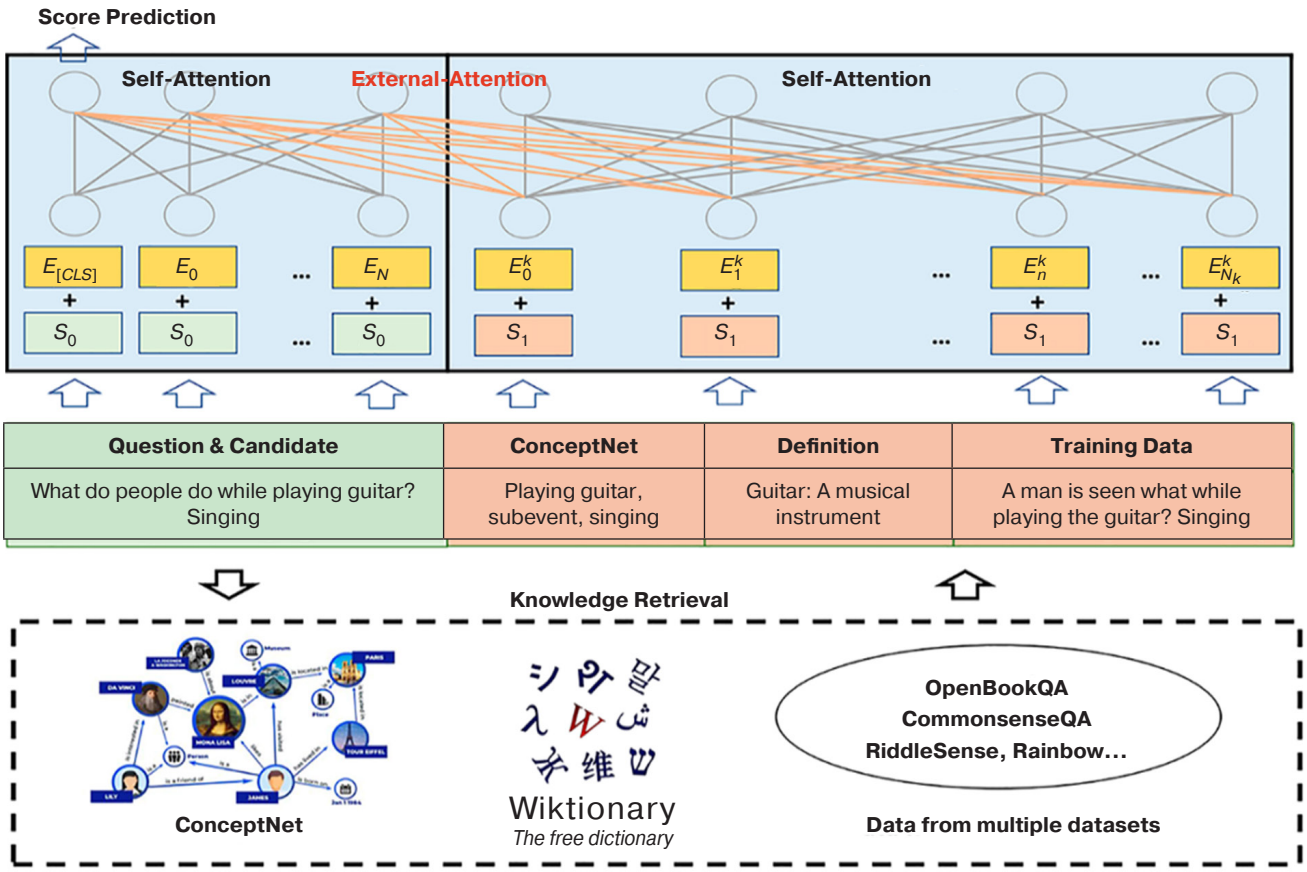

Рис. 4. Архитектура модели KEAR [42]

В простейшем случае для этого может использоваться ограничение на число триплетов (например, не более 3 последовательных) или набор некоторых эвристик, учитывающих особенности конкретной базы знаний. В [43] с целью вовлечения в работу модели только потенциально релевантных метаданных из Wikidata оценивается увеличение вероятности правильного ответа с учетом соответствующего прироста информации (Information Gain):

$$P(y \mid k_m) = 2^{\text{pmi}(y,k_m)} P(y), \qquad (8)$$

где y – правильный ответ; k_m – определенный паттерн, содержащий метаданные m;

$$\text{pmi} = \log\left(\frac{P(y,k_m)}{P(y)P(k_m)}\right).$$

Исследование [44] показывает целесообразность ранжирования извлеченных дополнительных сведений на основе определения их важности через вспомогательную предобученную модель, а в [45] в дополнение к ранжированию показан положительный эффект от использования взвешенного суммирования эмбеддингов знаний при их интеграции, что позволяет сделать акцент на более существенных фактах.

Также, для того чтобы интегрировать разнородные данные, может применяться механизм внимания, который позволяет агрегировать признаки с учетом их важности для задачи. Иначе говоря, более релевантными будут считаться извлеченные знания, которые имеют больше семантической связи с запросом, что определяется на основании операций над векторными представлениями. Чаще всего на практике на основе действий над эмбеддингами получают веса внимания, которые позволяют обновить соответствующие векторные представления с учетом конкретной задачи и ее контекста. В частности, подобный подход представлен в работах [46–48], а в статье [49] вспомогательные знания также фильтруются на основании частоты встречаемости сущностей и соответствующих путей, тогда как для интеграции знаний дополнительно используется сигмоидальная функция, которая регулирует то, как сильно они повлияют на обновление контекста для запроса в целом. В исследовании [50] помимо механизма внимания для отфильтровывания нерелевантных данных предложено использовать графовые подходы для определения важности отдельных вершин в извлеченном подграфе: расчет степени близости вершин, PageRank[13]

[13] Алгоритм ранжирования, который оценивает количество и качество ссылок, ведущих на веб-страницы. [A ranking algorithm that evaluates the number and quality of links leading to web pages.]

и его модификация, что позволяет учитывать только наиболее информативные пути.

Кроме того, недостатком интеграции знаний через языковые модели является ограниченное использование структурированности баз знаний, что может снижать потенциальную эффективность итоговой реализации. Для сохранения эффекта от учета связей при переводе триплетов в текст и предотвращения смешения информации в модели K-BERT [51] соответствующие позиционные данные включаются на этапе получения эмбеддингов, а в последующих вычислениях прибегают к специально вводимой в статье матрице видимости (Visible Matrix), элементы которой фиксируют то, с какими токенами должен взаимодействовать в заданном контексте конкретный токен.

В связи с этим необходимо упомянуть один из главных инструментов для обработки структурированных знаний – графовые нейронные сети. Этот инструмент позволяет получать и обновлять векторные представления вершин графов с помощью концепции передачи сообщений (Message Passing):

$$\mathbf{h}_u = \phi(x_u, \bigoplus_{v \in N_u} \psi(x_u, x_v, e_{uv})), \quad (9)$$

где $\mathbf{h}_u$ – векторное представление вершины u; x_u и x_v – признаки вершин u и v; e_{uv} – признаки ребра между вершинами u и v; ϕ и ψ – заданные дифференцируемые функции; $\bigoplus_{v \in N_u}$ – инвариантный к перестановкам оператор агрегирования, действующий на соседей вершины u.

Благодаря этому на практике вектор представления каждой сущности может использовать различные полученные из базы знаний контекстные данные, учитывая определенным образом информацию от своих соседей по графу. Примером используемой в контексте инъекции знаний модели такого рода является графовая сверточная нейронная сеть (Graph Convolutional Network) [52].

Полученные с помощью графовых нейронных сетей признаки впоследствии также могут интегрироваться в работу системы с помощью механизма внимания. Среди реализаций такого рода можно назвать архитектуру модели [53], изображенную на рис. 5. Здесь векторное представление каждой вершины вспомогательного подграфа перед его непосредственным использованием для получения ответа корректируется с учетом релевантности относительно имеющегося векторного представления запроса:

$$\alpha_i = \frac{\mathbf{h}^c \sigma(\mathbf{W}\mathbf{h}_i)}{\sum_{j \in N} \mathbf{h}^c \sigma(\mathbf{W}\mathbf{h}_j)}, \quad (10)$$

где α_i – степень релевантности вершины i; $\mathbf{h}^c$ – векторное представление контекста запроса; $\mathbf{W}$ – матрица весов; $\mathbf{h}_i$ – векторное представление вершины i; N – множество индексов вершин, соседних по отношению к вершине i.

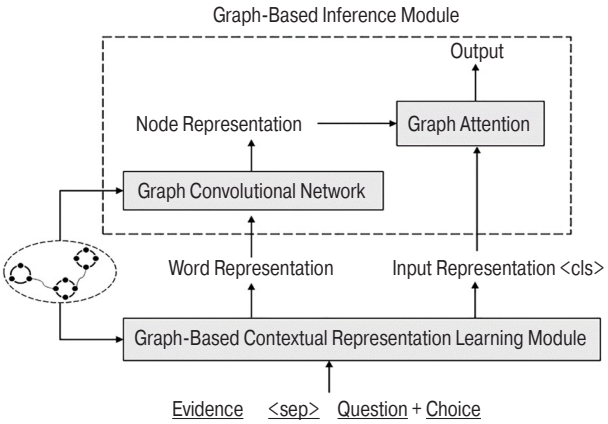

Рис. 5. Архитектура модели из [53]

В целом же схема работы модели устроена следующим образом: сначала по имеющемуся запросу, состоящего из вопроса (Question) и одного из вариантов ответа (Choice) извлекается подграф со вспомогательными сведениями. Эти сведения в форме текста (Evidence) присоединяются к запросу и подаются на вход языковой модели (Graph-Based Contextual Representation Learning Module). Полученные на выходе модели эмбеддинги токенов текста (Word Representation) попадают в модуль интеграции знаний (Graph-Based Inference Module), где используются для инициализации соответствующих вершин вспомогательного графа, которые впоследствии обновляются с помощью графовой сверточной сети (Graph Convolutional Network). Итоговые эмбеддинги вершин графа (Node Representation) затем агрегируются с помощью механизма внимания (Graph Attention) с учетом важности относительно эмбеддинга текстового контекста (Input Representation), а результирующий эмбеддинг графа вместе с текстовым эмбеддингом непосредственно используются для предсказания вероятности ответа с помощью многослойного перцептрона.

Схожим образом устроена модель Multi-hop Graph Relation Networks (MHGRN) [54], но ее ключевым отличием является рассмотрение вспомогательного подграфа в виде совокупности связывающих вершины путей, в соответствии с чем векторное представление каждой вершины обновляется на основании заданной длины путей из нее. Для агрегирования информации по путям вводятся специальные веса внимания, которые определяются как условная вероятность заданной последовательности триплетов с учетом имеющего контекста для запроса, тогда как для расчета вероятности конкретного ответа итоговые эмбеддинги для соответствующих ответу

сущностей агрегируются с помощью механизма внимания и вместе с эмбеддингом контекста запроса обрабатываются многослойным перцептроном. Таким образом, в рамках данного подхода учитывается и степень важности отношений между сущностями. В модели Joint reasoning with Language models and Knowledge graphs (JointLK) [55] используется отсечение наименее релевантных относительно запроса вершин вспомогательного подграфа, а также дополнительно вводится новое представление контекста запроса, которое учитывает степень его важности относительно подграфа и является третьей компонентой для получения оценки ответа наравне с исходным представлением контекста и эмбеддингом подграфа. В исследовании [56] в рамках механизма обмена сообщениями реализуется последовательное обновление и эмбеддингов сущностей, и отношений, которые в данном случае также используются для оценки вероятности ответа. При этом для формализации релевантности отношений между вершинами при заданном контексте запроса используется модифицированная матрица смежности, элементами которой являются соответствующие веса внимания. В модели Knowledge-aware graph Network (KagNet) [57] обновленные с помощью механизма обмена сообщениями векторные представления вершин вспомогательного графа рассматриваются в качестве элементов путей, соединяющих сущности из вопроса и одного из вариантов ответа. В итоге по каждой такой паре сущностей генерируется вектор структурированных признаков, как усреднение эмбеддингов соединяющих их путей, и вектор текстовых признаков, получаемый как результат применения многослойного перцептрона к конкатенации эмбеддингов запроса и каждой сущности из пары. Для оценки вероятности конкретного ответа реализуется усреднение по всем парам сущностей из запроса и ответа. Кроме того, в дополнение вместо усреднений авторами также предлагается использовать механизм внимания при агрегировании признаков.

Одним из определенных недостатков использования графовых нейронных сетей является увеличение числа параметров в модели и, соответственно, ресурсов на ее обучение и использование. В связи с этим в работе [58] предлагается упрощенный алгоритм получения эмбеддингов триплетов на основании one-hot-векторов, указывающих тип сущности в графе и определенное отношение в рамках базы ConceptNet. Для расчета итоговой вероятности ответа в модели используются две оценки: по текстовым и графовым признакам. Первая основывается на обработке многослойным перцептроном векторного представления запроса, а вторая – взвешенной суммы эмбеддингов путей, учитывающей частоту их встречаемости.

Несколько более сложным может представляться процесс интеграции знаний в случае наличия нескольких источников сведений и обучения на разных типах задач. В таких условиях приходится решать проблемы, связанные с необходимостью переучивать веса модели и вытеснением выученных фактов новыми, что способно вести к нестабильности результатов. Одним из возможных решений является использование адаптеров [59] – специальных модулей, ориентированных под конкретный источник данных или задачу, что позволяет не изменять веса основной модели и обучать только относительно небольшое число весов адаптера, а также тем самым избежать смешения знаний. При этом на практике обычно независимо обучается несколько различных адаптеров, которые затем уже используются совместно для решения определенной задачи. Так, в модели [60] применяется два вида адаптеров: первый ориентирован на усвоение общих фактов из баз знаний, тогда как второй – на лингвистические сведения. В рамках архитектуры каждый выход со слоя модели-трансформера подается на вход соответствующего слоя адаптера, в результате чего на последнем слое адаптера формируются определенные вспомогательные признаки, которые могут использоваться для предсказания ответа вместе с выходами последнего слоя трансформера. В работе [61] (рис. 6) реализован несколько иной подход, при котором веса предобученных на данных из баз знаний ATOMIC, ConceptNet, WikiData и WordNet адаптеров также не меняются при тренировке модели на конкретной задаче, а вместо этого интеграция знаний осуществляется за счет механизма внимания (формула (1)), где адаптеры формируют Value и Key, а предобученный трансформер – Query:

На каждом слое модели входные данные проходят через слой трансформера и попадают в блок интеграции знаний (Zero-shot Fusion) как напрямую (круг 4), так и после взаимодействия с моделями-адаптерами (круги 1, 2 и 3). В этом блоке происходит взаимодействие эмбеддингов в рамках механизма внимания (формула (1)): выходное представление из трансформера используются как query, тогда как выходы с адаптеров выступают в качестве Value и Key. Впоследствии результат блока интеграции знаний суммируется с выходом из блока Multi-Head Attention трансформера и нормализуется (Add & Norm). Целью обучения модели в рамках данной архитектуры является способность обращаться к более релевантному адаптеру, что в определенной степени напоминает концепцию смеси экспертов [62].

Отдельно в контексте инъекции знаний можно выделить группу подходов, опирающихся на использование так называемых токенов

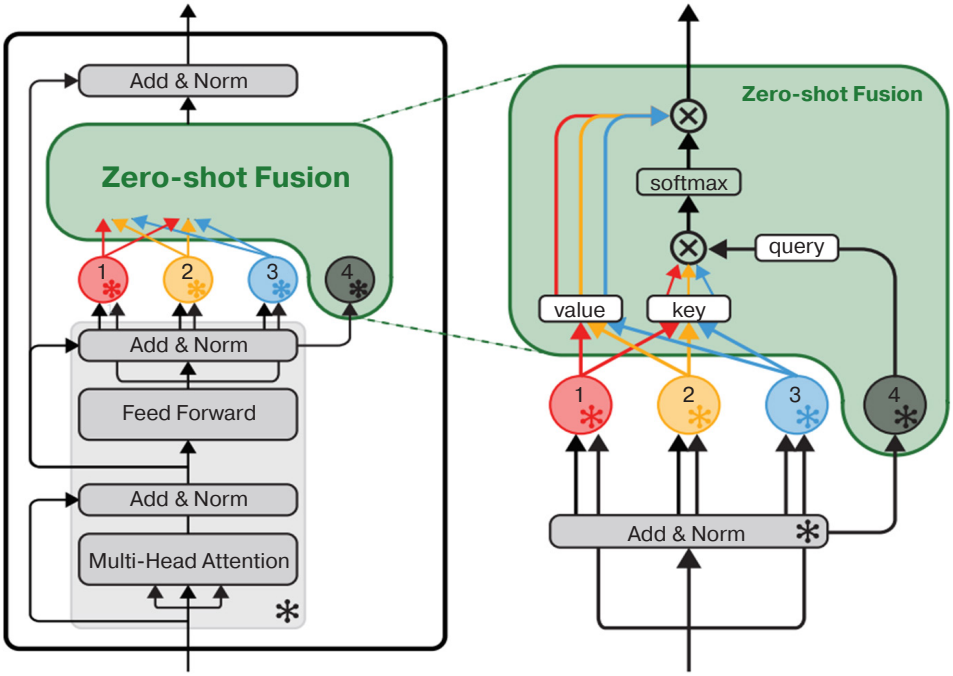

Рис. 6. Схема интеграции знаний с помощью адаптеров из [61]

взаимодействия (Interaction Token). Концепция токенов взаимодействия в значительной степени сходна с идеей применения специального токена [CLS] в языковых моделях, который может служить для классификации целого фрагмента текста. Аналогично токены взаимодействия в случае текстовой информации или вершины взаимодействия (Interaction Node) в случае графов способны выступить промежуточным вместилищем необходимых сведений для объединения разнородных данных. Пример соответствующей вопросно-ответной архитектуры можно увидеть на рис. 7: в рамках модели Graph REASoning Enhanced Language Model (GreaseLM) [63] текстовая и структурированная информация обрабатываются независимо, а их интеграция осуществляется за счет обновления векторных представлений токена взаимодействия и вершины взаимодействия посредством применения двуслойного перцептрона к их конкатенации:

$$[\mathbf{h}_{\text{int}};\mathbf{e}_{\text{int}}] = \text{MInt}([\tilde{\mathbf{h}}_{\text{int}};\tilde{\mathbf{e}}_{\text{int}}]) = \text{MLP}([\tilde{\mathbf{h}}_{\text{int}};\tilde{\mathbf{e}}_{\text{int}}]), \quad (11)$$

где $\tilde{\mathbf{h}}_{\text{int}}$ – векторное представление токена взаимодействия до интеграции знаний, $\tilde{\mathbf{e}}_{\text{int}}$ – векторное представление вершины взаимодействия до интеграции знаний, MInt (modality interaction layer) – слой взаимодействия модальностей.

При обучении GreaseLM конкатенация вопроса с одним из вариантов ответа обрабатывается с помощью N слоев модели-энкодера (Uni-modal Encoder) и вместе со вспомогательным графом (KG Retrieval) проходит через M слоев (GreaseLM Layer) блока интеграции знаний (Cross-modal Fuser). На каждом слое этого блока эмбеддинги текстовых токенов (h_1, …, h_T, h_{int}) и вершин графа (e_{int}, e_1, …, e_J) обрабатываются соответственно слоями языковой (LM Layer) и графовой моделей (GNN Layer), а сам процесс интеграции осуществляется через взаимодействие (MInt, формула (11)) эмбеддингов специальных токенов ($\tilde{h}_{\text{int}}$ и $\tilde{e}_{\text{int}}$). После завершения процесса интеграции знаний эмбеддинги специальных токенов с агрегированным с помощью механизма внимания эмбеддингом графа (Pooling) используются для выбора ответа (Answer Selection) посредством перцептрона (MLP).

В рамках модели DRAGON[14] [64] архитектура GreaseLM рассматривалась в контексте самообучения моделей: после слоя интеграции знаний полученные текстовые признаки служат для предсказания маскированных токенов текста, а графовые – для задачи предсказания связи (Link Prediction), которая подразумевает установление степени вероятности наличия связи между вершинами в графе с применением скоринговых функций, подобных (7).

В модели QA-GNN [65] применяется только вершина взаимодействия, инициализируемая векторным представлением текстового контекста из запроса, на основании близости к которой, определяемой с помощью предобученной модели, оценивается

[14] DRAGON — Deep Bidirectional Language-Knowledge Graph Pretraining.

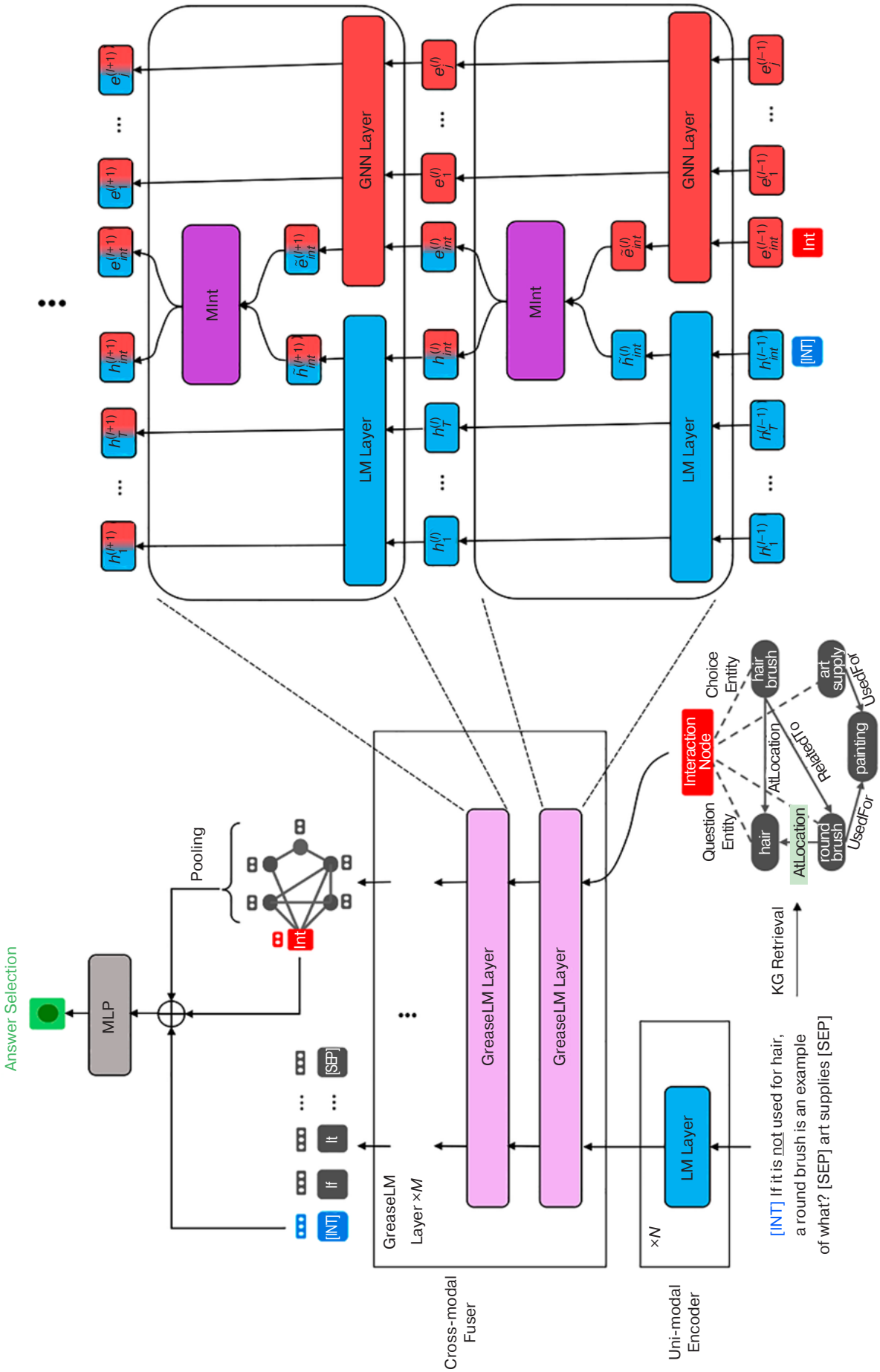

Рис. 7. Архитектура модели GreaseLM [63]

важность других вершин. Данные оценки вместе с признаками, репрезентирующими типы вершин и отношений в виде one-hot-кодирования, используются для расчета весов внимания, с учетом которых реализуется передача сообщений между вершинами и соответствующее обновление их эмбеддингов. Процесс выбора ответа при этом также по сути формулируется аналогично модели GreaseLM. В PipeNet [66] по сравнению с QA-GNN вычисление близости вершин вспомогательного графа контексту запроса в каком-то смысле заменяется алгоритмом отсечения нерелевантных вершин, основанном на определении кратчайшего расстояния между сущностями в рамках соответствующего запросу графа языковых зависимостей:

$$D(c_q) = -\frac{\sum_{i=1}^{|V_a|} Dist(c_q, c_a)}{|V_a|}, \quad (12)$$

где $D(c_q)$ – релевантность сущности c_q из запроса, $Dist(c_q, c_a)$ – кратчайшее расстояние между сущностью c_q из запроса и сущностью c_a из соответствующего варианта ответа, V_a – множество сущностей из варианта ответа для запроса.

Остальная же часть архитектуры QA-GNN по сути сохраняется, за исключением использования оценки важности вершин при расчете весов внимания.

Подводя некоторый итог методам интеграции знаний с помощью графовых моделей, можно констатировать, что они характеризуются наибольшим разнообразием используемых идей, которые демонстрируют широкие возможности для учета особенностей структурированных знаний и их включения в работу вопросно-ответных систем. Положительным аспектом можно также считать возможность повысить интерпретируемость модели за счет формирования цепочек фактов с помощью баз знаний, которые при этом могут своевременно отдельно обновляться в зависимости от происходящих событий. В то же время полноценная интеграция графовых признаков приводит к существенному усложнению архитектур моделей, а также, в зависимости от реализации, может требовать определенных дополнительных вычислительных ресурсов, вследствие чего польза от применения интеграции знаний становится менее однозначной.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ

Описанные выше подходы имеют расхождение также и с точки зрения постановки экспериментальной части. Так, в первую очередь, для тестирования эффективности реализаций могли использоваться разные бенчмарки. В результате сравнительный анализ было решено проводить относительно датасета, который встречался наиболее часто в рассмотренных работах – CommonsenseQA [67].

CommonsenseQA состоит из 12102 вопросов, предлагающих 5 вариантов ответов, один из которых является верным. Выбор в пользу этого набора данных может обосновываться его более высокой сложностью с точки зрения относительно невысоких результатов вопросно-ответных систем на нем по сравнению с аналогами. Эта сложность в данном случае обусловлена наличием акцента в вопросах на социальные и психологические аспекты и необходимостью установления причинно-следственных связей, а также отсутствием у вопросов какого-либо дополнительного контекста. Это, с одной стороны, усложняет эффективную реализацию предобученных языковых моделей в связи с меньшим числом входных данных. С другой же стороны, подобная постановка задачи как раз благоприятствует формированию такого контекста за счет внешних баз знаний.

В табл. 2 представлены сводные результаты моделей без использования ансамблирования (Ensemble) по тестовой выборке датасета CommonsenseQA. В контексте данного датасета для практического сравнения реализаций в качестве метрики принято использовать точность (Accuracy) – процент вопросов, на которые был получен правильный ответ. Также необходимо указать, что в качестве базового бенчмарка была выбрана одна из наиболее часто используемых языковых моделей-энкодеров – RoBERTa [68].

Приведенные результаты показывают, что любой из рассмотренных подходов позволяет увеличить точность вопросно-ответной системы относительно базового решения с помощью предобученной языковой модели, что подтверждает перспективность данного исследовательского направления. При этом использующие графовые эмбеддинги модели демонстрируют заметно более низкую точность, а лучший результат на датасете CommonsenseQA получен при использовании модели KEAR с привлечением сведений из базы знаний через текстовые эмбеддинги.

Тем не менее, с практической точки зрения необходимо принимать во внимание и существование других важных факторов при сравнении подходов. К примеру, основанные на самообучении и дообучении модели, несмотря на более низкую точность, требуют меньше дополнительных вычислений для получения ответа на запрос. В то же время сам процесс предобучения таких моделей подразумевает достаточно существенные затраты вычислительных ресурсов. Помимо этого, не во всех реализациях используются одинаковые языковые модели, что само по себе может давать разницу в итоговой точности. При этом можно учитывать и то, сколько времени в совокупности требуется модели для получения ответа.

Таблица 2. Сравнение эффективности методов интеграции знаний

Модель	Метод интеграции	Точность на тестовой выборке CommonsenseQA, %
RoBERTa [68] (2019)	–	68.7
Модель из [15] (2020)	Самообучение	75.6
Модель из [23] (2022)	Самообучение	78.5
UnifiedQA [37] (2020)	Дообучение	79.1
Модель из [44] (2023)	Текстовые эмбеддинги и механизм внимания	75.0
Модель из [47] (2020)	Текстовые эмбеддинги и механизм внимания	80.3
DEKCOR [41] (2021)	Текстовые эмбеддинги и механизм внимания	80.7
KEAR [42] (2022)	Текстовые эмбеддинги и механизм внимания	86.1
JointLK [55] (2022)	Графовые эмбеддинги и механизм внимания	74.4
Модель из [53] (2020)	Графовые эмбеддинги и механизм внимания	75.3
MHGRN [54] (2020)	Графовые эмбеддинги и механизм внимания	75.4
QA-GNN [65] (2021)	Токены взаимодействия	73.4
GreaseLM [63] (2022)	Токены взаимодействия	74.2
DRAGON [64] (2022)	Токены взаимодействия	76.0

Если же исходить исключительно из результатов на бенчмарке CommonsenseQA, то можно констатировать, что использование в целом более сложных с точки зрения архитектуры моделей не приносит достаточно существенного эффекта, чтобы конкурировать с более устоявшимися подходами, ориентированными исключительно на использование языковых моделей. Несмотря на это, необходимо отметить и целесообразность продолжения сравнительного анализа с применением других бенчмарков для более полной оценки реального положения вещей.

ЗАКЛЮЧЕНИЕ

Представленный обзор позволяет утверждать об эффективности использования методов интеграции знаний в области разработки вопросно-ответных систем. Уже существующие решения экспериментально подтверждают возможность одновременного достижения нескольких основных целей интеграции знаний в данном контексте.

При этом все еще существует значительное пространство для дальнейшего совершенствования множества аспектов данного процесса. Во-первых, в настоящее время преобладают относительно базовые и устоявшиеся в области обработки естественного языка методы извлечения данных из баз знаний по запросу. Только в нескольких работах предлагаются способы улучшения этого процесса, такие как перефразирование дополнительных знаний из базы для упрощения их обработки системой. В связи с этим, учитывая потенциальную важность извлечения релевантных сведений с точки зрения дальнейшей имплементации, представляется возможным также отдельно рассмотреть более специфические подходы и их влияние на результат.

Во-вторых, интерес представляет анализ потенциального влияния выбора определенной графовой модели для обработки структурированной информации, т.к. в существующих работах основной акцент смещен, прежде всего, на сравнение по критерию использованной языковой модели. В то же время за последние несколько лет появилось множество новых перспективных моделей векторных представлений графов знаний и графовых нейронных сетей, чьи возможности в рамках практических задач такого рода на данный момент не установлены, но могут существенным образом влиять на результаты работы системы в целом.

В-третьих, к настоящему моменту отсутствуют систематические исследования, касающиеся сопоставления методов объединения данных разных модальностей в контексте разработки вопросно-ответных систем. Этот вопрос также может считаться актуальным в силу возможности обобщения на более широкий спектр задач.

Наконец, в рамках текущего вектора развития области разработки вопросно-ответных систем, ведущего к преобладанию в приложениях универсальных генеративных языковых моделей-декодеров, таких как ChatGPT, имеет смысл сделать акцент на изучении особенностей способов инъекции знаний в модели такого типа.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019;1:4171–4186. https://doi.org/10.18653/v1/N19-1423
2. Petroni F., Rocktäschel T., Lewis P., et al. Language Models as Knowledge Bases? *Processing* (*EMNLP-IJCNLP*). 2019. P. 2463–2473. https://doi.org/10.18653/v1/D19-1250
3. Sap M., Le Bras R., Allaway E., et al. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019;33(1):3027–3035. https://doi.org/10.1609/aaai.v33i01.33013027
4. Niven T., Kao H.-Y. Probing Neural Network Comprehension of Natural Language Arguments. *arXiv preprint arXiv:1907.07355*. 2019. https://doi.org/10.48550/arXiv.1907.07355
5. McCoy R. T., Pavlick E., Linzen T. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. P. 3428–3448. http://doi.org/10.18653/v1/P19-1334
6. Li J., Chen J., Ren R., et al. The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models. *arXiv preprint arXiv:2401.03205*. 2024. https://doi.org/10.48550/arXiv.2401.03205
7. Wei J., Wang X., Schuurmans D., et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: *36th Conference on Neural Information Processing Systems*. 2022;35:24824–24837. https://doi.org/10.48550/arXiv.2201.11903
8. Lewis P., Perez E. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*. 2020;33:9459–9474. https://doi.org/10.48550/arXiv.2005.11401
9. Ye Zhi-Xiu, Chen Q., Wang W., Ling Zhen-Hua. Align, Mask and Select: A Simple Method for Incorporating Commonsense Knowledge into Language Representation Models. *arXiv preprint arXiv:1908.06725v5*. 2020. https://doi.org/10.48550/arXiv.1908.06725
10. Vaswani A., Shazeer N., Parmar N., et al. Attention Is All You Need. *Advances in Neural Information Processing Systems 30*. 2018. https://doi.org/10.48550/arXiv.1706.03762
11. Liu J., Shen D., Zhang Y., et al. What Makes Good In-Context Examples for GPT-3? *arXiv preprint arXiv:2101.06804*. 2021. https://doi.org/10.48550/arXiv.2101.06804
12. Gao T., Fisch A., Chen D. Making Pre-trained Language Models Better Few-shot Learners. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (*Volume 1: Long Papers*). 2021. P. 3816–3830. http://doi.org/10.18653/v1/2021.acl-long.295
13. Shwartz V., West P., Le Bras R., et al. Unsupervised Commonsense Question Answering with Self-Talk. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). 2020. P. 4615–4629. http://doi.org/10.18653/v1/2020.emnlp-main.373
14. Wang J., Zhao H. ArT: All-round Thinker for Unsupervised Commonsense Question-Answering. In: *Proceedings of the 29th International Conference on Computational Linguistics*. 2022. P. 1490–1501. https://doi.org/10.48550/arXiv.2112.13428
15. Wang P., Peng N., Ilievski F., et al. Connecting the Dots: A Knowledgeable Path Generator for Commonsense Question Answering. *arXiv preprint arXiv:2005.00691*. 2020. https://doi.org/10.48550/arXiv.2005.00691
16. Raffel C., Shazeer N., Roberts A., et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*. 2020;21(140):1–67. https://doi.org/10.48550/arXiv.1910.10683
17. Zhang Z., Han X., Liu Z., et al. ERNIE: Enhanced Language Representation with Informative Entities. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. P. 1441–1451. https://doi.org/10.18653/v1/P19-1139
18. Peters M.E., Neumann M., Logan IV R.L., et al. Knowledge enhanced contextual word representations. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*). 2019. P. 43–54. https://doi.org/10.18653/V1/D19-1005
19. He L., Zheng S., Yang T., Zhang F. KLMo: Knowledge Graph Enhanced Pretrained Language Model with Fine-Grained Relationships. In: *Findings of the Association for Computational Linguistics: EMNLP*. 2021. P. 4536–4542. https://doi.org/10.18653/v1/2021.findings-emnlp.384
20. Xiong W., Du J., Wang W.Y., Stoyanov V. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. *arXiv preprint arXiv:1912.09637*. 2019. https://doi.org/10.48550/arXiv.1912.09637

21. Sun Y., Wang S., Li Y., et al. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv preprint arXiv:1904.09223*. 2019. https://doi.org/10.48550/arXiv.1904.09223
22. Zhang D., Yuan Z., Liu Y., et al. E-BERT: A Phrase and Product Knowledge Enhanced Language Model for E-commerce. *arXiv preprint arXiv:2009.02835*. 2020. https://doi.org/10.48550/arXiv.2009.02835
23. Chen Q., Li F.-L., Xu G., et al. DictBERT: Dictionary Description Knowledge Enhanced Language Model Pre-training via Contrastive Learning. *arXiv preprint arXiv:2208.00635*. 2022. https://doi.org/10.48550/arXiv.2208.00635
24. Lauscher A., Vulić I., Ponti E.M., et al. Informing Unsupervised Pretraining with External Linguistic Knowledge. *arXiv preprint arXiv:1909.02339v1*. 2019. https://doi.org/10.48550/arXiv.1909.02339
25. Levine Y., Lenz B., Dagan O., et al. SenseBERT: Driving Some Sense into BERT. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020. P. 4656–4667. https://doi.org/10.18653/v1/2020.acl-main.423
26. Wang X., Gao T., Zhu Z., et al. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Trans. Assoc. Comput. Linguis*. 2021;9:176–194. https://doi.org/10.1162/tacl_a_00360
27. Bordes A., Usunier N., Garcia-Durán A., et al. Translating Embeddings for Modeling Multi-relational Data. *Advances in Neural Information Processing Systems*. 2013. P. 2787–2795.
28. He B., Zhou D., Xiao J., et al. BERT-MK: Integrating Graph Contextualized Knowledge into Pre-trained Language Models. *Findings of the Association for Computational Linguistics: EMNLP*. 2020. P. 2281–2290. https://doi.org/10.18653/v1/2020.findings-emnlp.207
29. Banerjee P., Baral C. Self-Supervised Knowledge Triplet Learning for Zero-shot Question Answering. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). 2020. P. 151–162. https://doi.org/10.18653/v1/2020.emnlp-main.11
30. Zhong W., Tang D., Duan N., et al. Improving Question Answering by Commonsense-Based Pre-training. In: Tang J., Kan M.Y., Zhao D., Li S., Zan H. (Eds.). *Natural Language Processing and Chinese Computing. NLPCC 2019. Lecture Notes in Computer Science*. Springer; 2019. V. 11838. P. 16–28. https://doi.org/10.1007/978-3-030-32233-5_2
31. Sun T., Shao Y., Qiu X., et al. CoLAKE: Contextualized Language and Knowledge Embedding. *arXiv preprint arXiv:2010.00309v1*. 2020. https://doi.org/10.48550/arXiv.2010.00309
32. Su Y., Han X., Zhang Z., et al. CokeBERT: Contextual knowledge selection and embedding towards enhanced pre-trained language models. *AI Open*. 2021;2:127–134. https://doi.org/10.1016/j.aiopen.2021.06.004
33. Ma K., Ilievski F., Francis J., et al. Knowledge-driven Data Construction for Zero-shot Evaluation in Commonsense Question Answering. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021;35(15):13507–13515. https://doi.org/10.1609/aaai.v35i15.17593
34. Wang W., Fang T., Ding W., et al. CAR: Conceptualization-Augmented Reasoner for Zero-Shot Commonsense Question Answering. *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023. P. 13520–13545. https://doi.org/10.18653/v1/2023.findings-emnlp.902
35. Zhan X., Li Y., Dong X., et al. elBERto: Self-supervised Commonsense Learning for Question Answering. *arXiv preprint arXiv:2203.09424v1*. 2022. https://doi.org/10.48550/arXiv.2203.09424
36. Rajpurkar P., Jia R., Liang P. Know What You Don't Know: Unanswerable Questions for SQuAD. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018;2(Short Papers):784–789. https://doi.org/10.18653/v1/P18-2124
37. Khashabi D., Min S., Khot T., et al. UnifiedQA: Crossing Format Boundaries with a Single QA System. In: *Findings of the Association for Computational Linguistics*. 2020. P. 1896–1907. https://doi.org/10.18653/v1/2020.findings-emnlp.171
38. Lourie N., Le Bras R., Bhagavatula C., Choi Y. UNICORN on RAINBOW: A Universal Commonsense Reasoning Model on a New Multitask Benchmark. *arXiv preprint arXiv:2103.13009v1*. 2021. https://doi.org/10.48550/arXiv.2103.13009
39. Baek J., Aji A.F., Saffari A. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In: *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations* (*NLRSE*). 2023. P. 78–106. https://doi.org/10.18653/v1/2023.nlrse-1.7
40. Pan X., Sun K., Yu D., et al. Improving Question Answering with External Knowledge. In: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. 2019. P. 27–37. https://doi.org/10.18653/v1/D19-5804
41. Xu Y., Zhu C., Xu R., et al. Fusing Context Into Knowledge Graph for Commonsense Question Answering. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021. P. 1201–1207. https://doi.org/10.18653/v1/2021.findings-acl.102
42. Xu Y., Zhu C., Wang S., et al. Human Parity on CommonsenseQA: Augmenting Self-Attention with External Attention. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence* (*IJCAI*). 2022. P. 2762–2768. https://doi.org/10.24963/ijcai.2022/383
43. Arora S., Wu S., Liu E., Ré C. Metadata Shaping: A Simple Approach for Knowledge-Enhanced Language Models. In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022. P. 1733–1745. https://doi.org/10.18653/v1/2022.findings-acl.137
44. Li S., Gao Y., Jiang H., et al. Graph Reasoning for Question Answering with Triplet Retrieval. In: *Findings of the Association for Computational Linguistics: ACL 2023*. 2023. P. 3366–3375. https://doi.org/10.18653/v1/2023.findings-acl.208
45. Mitra A., Banerjee P., Pal K.K., et al. How Additional Knowledge can Improve Natural Language Commonsense Question Answering? *arXiv preprint arXiv:1909.08855v3*. 2020. https://doi.org/10.48550/arXiv.1909.08855
46. Chen Q., Zhu X., Ling Z.-H., et al. Neural Natural Language Inference Models Enhanced with External Knowledge. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* 2018;1(Long Papers):2406–2417. https://doi.org/10.18653/v1/P18-1224

47. Chen Q., Ji F., Chen H., Zhang Y. Improving Commonsense Question Answering by Graph-based Iterative Retrieval over Multiple Knowledge Sources. In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020. P. 2583–2594. https://doi.org/10.18653/v1/2020.coling-main.232
48. Ma K., Francis J., Lu Q., et al. Towards Generalizable Neuro-Symbolic Systems for Commonsense Question Answering. In: *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*. 2019. P. 22–32. https://doi.org/10.18653/v1/D19-6003
49. Bauer L., Wang Y., Bansal M. Commonsense for Generative Multi-Hop Question Answering Tasks. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018. P. 4220–4230. https://doi.org/10.18653/v1/D18-1454
50. Paul D., Frank A. Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2019;1(Long and Short Papers):3671–3681. https://doi.org/10.18653/v1/N19-1368
51. Liu W., Zhou P., Zhao Z., et al. K-BERT: Enabling Language Representation with Knowledge Graph. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020;34(03):2901–2908. https://doi.org/10.1609/aaai.v34i03.5681
52. Kipf T.N., Welling M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*. 2017. https://doi.org/10.48550/arXiv.1609.02907
53. Lv S., Guo D., Xu J., et al. Graph-Based Reasoning over Heterogeneous External Knowledge for Commonsense Question Answering. In: *Proceedings of the AAAI Conference on Artificial Intelligence.* 2020;34(05):8449–8456. https://doi.org/10.1609/aaai.v34i05.6364
54. Feng Y., Chen Y., Lin B.Y., et al. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). 2020. P. 1295–1309. https://doi.org/10.18653/v1/2020.emnlp-main.99
55. Sun Y., Shi Q., Qi L., Zhang Y. JointLK: Joint Reasoning with Language Models and Knowledge Graphs for Commonsense Question Answering. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022. P. 5049–5060. https://doi.org/10.18653/v1/2022.naacl-main.372
56. Yan J., Raman M., Chan A., et al. Learning Contextualized Knowledge Structures for Commonsense Reasoning. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP*. 2021. P. 4038–4051. https://doi.org/10.18653/v1/2021.findings-acl.354
57. Lin B.Y., Chen X., Chen J., Ren X. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*). 2019. P. 2829–2839. https://doi.org/10.18653/v1/D19-1282
58. Jiang J., Zhou K., Zhao W.X., Wen J.-R. Great Truths are Always Simple: A Rather Simple Knowledge Encoder for Enhancing the Commonsense Reasoning Capacity of Pre-Trained Models. In: *North American Chapter of the Association for Computational Linguistics-Findings*. 2022. https://doi.org/10.48550/arXiv.2205.01841
59. Houlsby N., Giurgiu A., Jastrzebski S., et al. Parameter-Efficient Transfer Learning for NLP. In: *Proceedings of Machine Learning Research*. 2019;97:2790–2799. https://doi.org/10.48550/arXiv.1902.00751
60. Wang R., Tang D., Duan N., et al. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP.* 2021. P. 1405–1418. https://doi.org/10.18653/v1/2021.findings-acl.121
61. Kim Y.J., Kwak B., Kim Y., et al. Modularized Transfer Learning with Multiple Knowledge Graphs for Zero-shot Commonsense Reasoning. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022. P. 2244–2257. https://doi.org/10.18653/v1/2022.naacl-main.163
62. Jacobs R., Jordan M., Nowlan S., Hinton G. Adaptive Mixtures of Local Experts. *Neural Computation*. 1991;3(1):79–87. https://doi.org/10.1162/neco.1991.3.1.79
63. Zhang X., Bosselut A., Yasunaga M., et al. GreaseLM: Graph REASoning Enhanced Language Models for Question Answering. In: *The International Conference on Learning Representations* (*ICLR*). 2022. https://doi.org/10.48550/arXiv.2201.08860
64. Yasunaga M., Bosselut A., Ren H., et al. Deep Bidirectional Language-Knowledge Graph Pretraining. In: *36th Conference on Neural Information Processing Systems* (*NeurIPS*). 2022. https://doi.org/10.48550/arXiv.2210.09338
65. Yasunaga M., Ren H., Bosselut A., et al. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2021. P. 535–546. https://doi.org/10.18653/v1/2021.naacl-main.45
66. Su Y., Zhang J., Song Y., Zhang T. PipeNet: Question Answering with Semantic Pruning over Knowledge Graphs. *arXiv preprint arXiv:2401.17536v2*. 2024. https://doi.org/10.48550/arXiv.2401.17536
67. Talmor A., Herzig J., Lourie N., Berant J. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In*: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019;1(Long and Short Papers):4149–4158. https://doi.org/10.18653/v1/N19-1421

68. Liu Y., Ott M., Goyal N., et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*. 2019. https://doi.org/10.48550/arXiv.1907.11692
69. Robertson S.E., Walker S. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In: *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. 1994. P. 232–241. https://doi.org/10.1007/978-1-4471-2099-5_24

Об авторе

Радюш Даниил Валентинович, аспирант, факультет программной инженерии и компьютерной техники, ФГАОУ ВО «Национальный исследовательский университет ИТМО» (197101, Россия, Санкт-Петербург, Кронверкский пр., д. 49, лит. А). E-mail: daniil.radyush@gmail.com. Scopus Author ID 58234958500, https://orcid.org/0000-0001-8823-0609

About the Author

Daniil V. Radyush, Postgraduate Student, Faculty of Software Engineering and Computer Systems, ITMO University (49-A, Kronverkskii pr., Saint Petersburg, 197101 Russia). E-mail: daniil.radyush@gmail.com. Scopus Author ID 58234958500, https://orcid.org/0000-0001-8823-0609