П. Н. Демухаметов, Я. А. Зарембо, М. В. Аврискин
Тюменский государственный университет, г. Тюмень
УДК 004.89

ИССЛЕДОВАНИЕ МЕТОДОВ ГЕНЕРАЦИИ ТЕСТОВЫХ ВОПРОСОВ ПО ЛЕКЦИОННЫМ МАТЕРИАЛАМ ИТ-ДИСЦИПЛИН

Аннотация. В работе рассматриваются методы автоматизированной генерации тестовых вопросов, а также предложен метод с выбором предложений на основе вхождений терминов и индекса удобочитаемости Флеша. Для извлечения терминов сформирован датасет, на котором дообучены трансформерные энкодеры. Выполнено сравнение дообученных моделей и готовых решений. Также проведено сравнение предобученных моделей генерации вопросов.

Ключевые слова: автоматизированная генерация тестов, извлечение терминов, генерация вопросов, языковые модели, обработка естественного языка текста, тестирование знаний.

Введение. Тестирование остается одним из наиболее распространенных методов оценки знаний студентов. Они побуждают студентов более внимательно воспринимать и изучать лекционный материал. Кроме того, частое проведение даже небольших тестов помогает студентам лучше усваивать материал [1]. Из-за постоянного развития новых инструментов и методик возникает необходимость регулярной актуализации обучающих программ и бланка заданий. Это становится трудоемким процессом, требующим значительных временных затрат со стороны преподавателей. Поэтому многие преподаватели и исследователи интересуются автоматизированной генерацией тестовых заданий [2, 3].

В основном методы генерации вопросов можно разбить на две группы:

- методы, основанные на конструкции предложений с использованием правил. Эти правила, в свою очередь, делятся на три типа: шаблонные, синтаксические и семантические [5];
- методы, использующие нейронные сети для генерации предложений [5, 6].

Такие языковые модели, как T5 и BART, показывают результаты лучше в генерации вопросов на основе неструктурированного текста, чем методы по правилам [7, 8].

В статье [8] рассматривается подход с использованием дополнительной семантической разметки, позволяющей генерировать несколько вопросов на основе одного предложения. В работе [9] рассматривается подход с генерацией вопроса вместе с соответствующим ответом, а также использование дополнительной модели для генерации дистракторов (неверных вариантов ответов). Но в указанных статьях модели обучались на датасетах без конкретной тематики, и их дообучение на конкретной тематике может повысить качество работы [10, 11].

Существенным недостатком данных подходов является ограниченность объема контекста, который способны обрабатывать модели. Именно поэтому важным этапом является выбор целевых предложений для генерации вопросов. Это позволяет отбросить нерелевантную информацию и сосредоточиться на ключевых элементах исходного текста. Согласно исследованию М. А. Масловой [12] выделяют два основных метода выбора целевого предложения: метод суммаризации текста и метод ранжирования предложений по ключевым словам. Первый подход часто приводит к обобщенным результатам, что может способствовать утрате некоторых специфических деталей. А применение предобученной языковой модели для генерации вопросов из предложений, отобранных на основе ключевых слов, позволяет повысить качество итоговых результатов [7].

Проблема исследования. Дано $L=\{l_1,l_2,...,l_n\}$ — множество лекционных материалов, в котором каждый l_i представлен последовательностью токенов текста. Требуется найти множество тестовых вопросов Q, каждый из которых представляет тройку (q_i,a_i,d_i), где q_i — вопрос, a_i — верный ответ, d_i — неверные ответы, с помощью модели $f:L \to Q$,

Материалы и методы

Данные. Для дообучения и оценки моделей извлечения терминов были собраны статьи из российской научной электронной библиотеки «КиберЛенинка» из разделов «Компьютерные и информационные науки» и «Математика». Выбор обусловлен большим количеством статей в открытом доступе и стандартизированным форматом метаинформации (названия, ключевые слова, аннотации, полный текст), что позволяет автоматизировать сбор и последующую фильтрацию данных. Сбор осуществлялся по 1000 последних страниц для каждого раздела, охватывая публикации за период с 2011–2025 гг. для раздела «Математика» и 2016–2025 гг. для раздела «Компьютерные и информационные науки». Дата сбора данных 1 марта 2025. Всего было собрано 20021 статей.

У каждой статьи были извлечены: название, ключевые слова, аннотация, полный текст статьи, ссылка на статью. Были исключены следующие статьи: дубликаты, статьи без аннотаций, статьи не на русском языке. Также было удалено дублирование аннотаций на других языках.

Для дообучения и тестирования моделей извлечения терминов было размечено 13738 аннотаций: 7065 из раздела «Математика» и 6673 из раздела «Компьютерные и информационные науки». Эти статьи были случайно разбиты на обучающую и валидационную выборки в соотношении 80/20.

Разметка токенов осуществлялась с использованием схемы BIO. Разметка производилась при помощи GPT4o-mini с промтом, состоящего из трех частей: сама инструкция, пример и текст аннотации. Общая характеристика представлена в табл. 1.

Таблица 1

Общая характеристика датасета для извлечения терминов

Характеристика	*Значение*	
	Обучающая выборка	*Валидационная выборка*
Количество текстов	10990	2748
Средняя длина текстов в токенах (±σ)	102.56±74.30	105.43±74.88
Среднее количество терминов в текстах (±σ)	16.83±13.90	16.99±13.91

Оценка моделей генерации вопросов проводилась на датасете belebele rus_Cyrl [13]. Для оценки потребовались столбцы flores_passage и question, содержащие текст и вопрос по тексту соответственно.

Методы и метрики. Выделение терминов осуществлялось с использованием моделей с архитектурой трансформер, широко применяемых для задачи извлечения именованных сущностей [14, 15]. Были оценены энкодеры, предобученные на научно-технических текстах на

русском языке, такие как ai-forever/ruSciBERT [16] и mlsa-iai-msu-lab/sci-rus-tiny [17]. Эти модели были дообучены для задачи разметки последовательности токенов. Использовались следующие параметры: 5 эпох, скорость обучения — 2e-5, размер батча — 16, максимальная длина последовательности — 512, угасание весов — 0.01, оптимизатор — AdamW. Параметры взяты из примера в документации Transformers[1], количество эпох подобрано экспериментально. Для сравнения был рассмотрен метод извлечения ключевых слов rutermextract[2], основанный на морфологических правилах, а также проект TERMinator [15], использующий энкодер и эвристики для извлечения терминов. Оценка производилась с помощью метрик Recall, Precision и F1-score как на уровне сущностей, так и на уровне частичного совпадения.

Генерация вопросов выполнялась с использованием предобученных моделей с декодерами и большой языковой модели с промтом: hivaze/AAQG-QA-QG-FRED-T5-large[3], ai-forever/ruT5-base (SberQuad)[4], nbroad/mt5-base-qgen[5], lmqg/mt5-base-ruquad-qg[6], doc2query/msmarco-russian-mt5-base-v1[7], lmqg/mbart-large-cc25-ruquad-qg[8], GigaChat Lite[9].

Оценка результатов проводилась с помощью метрик ROUGE и BERTScore на основе bert-base-multilingual-cased, которые используются в задачах преобразования одного текста в другой, в том числе для преобразования текста в вопросы по тексту [6, 9]. При оценке производилось приведение к нижнему регистру, токенизация, удаление знаков препинания и стоп-слов, приведение к леммам по наличию или к стемме.

Для отбора предложений в тексте, по которым будут генерироваться вопросы можно использовать подход, предложенного в статье М. А. Масловой [12], в котором для каждого предложения считалась полезность и удобочитаемость с помощью индекса Флеша.

Полезность считается как:

$$\text{usefulness} = \frac{\textit{total word}\ \text{count} - \text{extra words count}}{\text{term count}} \quad (1)$$

где total word count — общее количество слов в предложении; extra words count — количество слов, которые не являются терминами в предложении; term count — количество терминов в предложении.

В отличие от предложенного подхода, мы предлагаем использовать именно термины, а не ключевые слова. Ключевые слова часто представляют собой общеупотребительные понятия и отражают только те элементы, которые встречаются достаточно часто в текстах. Это приводит к тому, что редкие, но важные для оценки знаний студента термины остаются незамеченными. Выделение специализированных терминов дает возможность охватывать охватывать все затронутые в тексте понятия.

[1] https://github.com/huggingface/notebooks/blob/main/examples/token_classification.ipynb
[2] https://github.com/igor-shevchenko/rutermextract
[3] https://huggingface.co/hivaze/AAQG-QA-QG-FRED-T5-large
[4] https://github.com/anaviel/SmartQuizGenerator
[5] https://huggingface.co/nbroad/mt5-base-qgen
[6] https://huggingface.co/lmqg/mt5-base-ruquad-qg
[7] https://huggingface.co/doc2query/msmarco-russian-mt5-base-v1
[8] https://huggingface.co/research-backup/mbart-large-cc25-ruquad-qg
[9] https://developers.sber.ru/docs/ru/gigachat/models

Результаты. Результат сравнения дообученных моделей для извлечения терминов представлен в таблице 2. Лучший результат показала модель ai-forever/ruSciBERT, имеющая наибольшее количество параметров. На частичном совпадении модель показала результат лучше, чем при полном совпадении. Это свидетельствует о проблеме с выделением границ терминов. Для задачи сортировки предложений по полезности (количеству терминов в предложении) нет необходимости получать четкие границы терминов.

Таблица 2

Сравнение моделей извлечения терминов

Модель	*Полное совпадение*			*Частичное совпадение*		
	Prec.	*Recall*	*F1*	*Prec.*	*Recall*	*F1*
Дообученные модели						
ai-forever/ruSciBert	70.73	70.99	69.90	86.36	86.57	85.20
mlsa-iai-msu-lab/sci-rus-tiny	56.05	57.76	55.02	78.85	80.83	77.03
Аналоги						
rutermextract	34.12	48.39	38.01	59.89	86.48	67.15
TERMinator	17.70	19.27	17.82	69.38	68.24	66.18

Результат сравнения предобученных моделей генерации вопросов представлен в таблице 3. Лучший результат показала модель ai-forever/ruT5-base, предобученная на датасете SberQuad. При этом эта модель может генерировать несколько различных вопросов для абзаца за один запрос. Модель Gigachat Light проигнорировала инструкцию о формате ответа в 13% случаев, что привело к снижению показателей, но был также представлен результат без учета ответов в неверном формате. При этом ее скорость работы значительно ниже по сравнению со всеми остальными моделями в связи с повторными попытками сгенерировать ответ в верном формате.

Таблица 3

Сравнение моделей генерации вопросов

Модель	*ROUGE-L*	*BERTScore*	*Время, сек*
GigaChat Ligh (Без учета ответов с неверным форматом)	17.265	69.776	-
ai-forever/ruT5-base (SberQuad)	15.783	69.181	669.094
doc2query/msmarco-russian-mt5-base-v1	15.015	69.427	597.159
GigaChat Light	14.944	60.395	2042.723
nbroad/mt5-base-qgen	14.166	68.665	577.413
hivaze/AAQG-QA-QG-FRED-T5-large (QG)	14.105	68.046	1126.046
lmqg/mbart-large-cc25-ruquad-qg	13.777	67.900	768.575
lmqg/mt5-base-ruquad-qg	13.526	68.132	641.309

Заключение. В ходе проведенной работы были рассмотрены методы для генерации тестовых вопросов. Предложен метод ранжирования предложений на основе терминов и индекса удобочитаемости, позволяющий отбирать участки текста для последующей генерации. Был размечен датасет из 13738 статей электронной библиотеки «Киберленинка» из разделов «Математика» и «Компьютерные и информационные технологии» с разметкой в формате BIO для задачи извлечения терминов. На этом датасете были дообучены и сравнены 2 модели извлечения терминов, основанные на энкодере архитектуры трансформер. Лучший результат показала модель ai-forever/ruSciBert со значением F1-score 85.20% по частичному совпадению, но 69.90% при полном совпадении. На датасете belebele rus_Cyrl, содержащем вопросы по текстам различных тематик, было оценено 6 предобученных моделей генерации вопросов, основанные на энкодере-декодере, и большая языковая модель Gigachat Light. Лучший результат показала модель ai-forever/ruT5-base, предобученная на датасете SberQuad, со значением метрик ROUGE-L 15.78% и BERTScore 69.18%.

С целью улучшения качества генерации вопросов по математическим и ИТ-текстам в следующих работах планируется собрать датасет вопросов по текстам статей раздела «Компьютерные науки и информационные технологии», чтобы дообучить модели генерации вопросов, ответов и дистракторов. Дополнительно планируется провести сравнение с результатами, полученными перед дообучением на тематическом датасете, чтобы выявить, насколько дообучение на тематических датасетах повысит качество по сравнению с общими датасетами, такими как SberQuad.

СПИСОК ЛИТЕРАТУРЫ

1. Murphy D. H., Little J. L., Bjork E. L. The value of using tests in education as tools for learning—not just for assessment // Educational Psychology Review. — 2023. — Т. 35, № 3. — С. 89.
2. Витвицкая А. А. Chat GPT как образовательный ресурс / А. А. Витвицкая // Современная психология образования: приоритеты и перспективы исследований и практик : сборник материалов Всероссийской научно-практической конференции, Казань, 08 декабря 2023 года. — Казань: Издательство федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет", 2024. — С. 14-16. — EDN LEHGFZ.
3. Сошников Д. В. Генерация учебных задач при помощи генеративных моделей / Д. В. Сошников, В. В. Буров, Е. Д. Патаракин // Информатизация образования и методика электронного обучения: цифровые технологии в образовании: Материалы VII Международной научной конференции, Красноярск, 19–22 сентября 2023 года. — Красноярск: Красноярский государственный педагогический университет им. В.П. Астафьева, 2023. — С. 1350-1354. — EDN LWEUHL.
4. Federiakin D. et al. Prompt engineering as a new 21st century skill //Frontiers in Education. — Frontiers Media SA, 2024. — Т. 9. — С. 1366434.
5. Zhang R. et al. A review on question generation from natural language text //ACM Transactions on Information Systems (TOIS). — 2021. — Т. 40, № 1. — С. 1-43.
6. GUO S. et al. A survey on neural question generation: Methods, applications, and prospects. — International Joint Conferences on Artificial Intelligence, 2024.
7. Васильев, А. А. Применение алгоритмов формирования вопросов по тексту для автоматической генерации тестов / А. А. Васильев, А. С. Нестеров // Вестник кибернетики. — 2023. — Т. 22, № 3. — С. 17-22. — DOI 10.35266/1999-7604-2023-3-17-22. — EDN RGEYPU.

8. Naeiji A. et al. Question generation using sequence-to-sequence model with semantic role labels // Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. — 2023. — С. 2830-2842.
9. Vachev K. et al. Leaf: Multiple-choice question generation //European Conference on Information Retrieval. — Cham : Springer International Publishing, 2022. — С. 321-328.
10. Wang B. et al. Domain-Adaptive Pre-training BERT Model for Test and Identification Domain NER Task // Journal of Physics Conference Series. — 2022. — Т. 2363, № 1. — С. 012019.
11. Guo K. et al. Fine-tuning Strategies for Domain Specific Question Answering under Low Annotation Budget Constraints //2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI). — IEEE, 2023. — С. 166-171.
12. Маслова М. А. Методы определения целевого предложения для автоматизированной генерации тестовых вопросов / М. А. Маслова // Инженерный вестник Дона. — 2022. — № 5(89). — С. 333-340. — EDN LWCDCR.
13. Bandarkar L. et al. The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants //Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2024. — С. 749-775.
14. Мельникова А. В. Сравнение предварительно обученных моделей для извлечения предметно-ориентированных сущностей из студенческих отчетных документов / А. В. Мельникова, М. С. Воробьева, А. В. Глазкова // Моделирование и анализ информационных систем. — 2025. — Т. 32, № 1. — С. 66-79. — DOI 10.18255/1818-1015-2025-1-66-79. — EDN HPYNWE.
15. Bruches E. et al. TERMinator: A System for Scientific Texts Processing //Proceedings of the 29th International Conference on Computational Linguistics. — 2022. — С. 3420-3426.
16. Gerasimenko N. A., Chernyavsky A. S., Nikiforova M. A. ruSciBERT: A transformer language model for obtaining semantic embeddings of scientific texts in Russian //Doklady Mathematics. — Moscow : Pleiades Publishing, 2022. — Т. 106. — № Suppl 1. — С. S95-S96.
17. Gerasimenko N. et al. SciRus: tiny and powerful multilingual encoder for scientific texts //Doklady Mathematics. — Moscow : Pleiades Publishing, 2024. — Т. 110. — № Suppl 1. — С. S193-S202.