MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 32, NO. 3, 2025
journal homepage: www.mais-journal.ru

ARTIFICIAL INTELLIGENCE

Hierarchical Multi-task Learning Methodology for ERNIE-3-Type Neural Networks in Russian-language Text Analysis and Generation

E. V. Totmina[1], I. Bondarenko[1], A. V. Seredkin[1]

DOI: 10.18255/1818-1015-2025-3-282-297

[1]Novosibirsk National Research State University, Novosibirsk, Russia

MSC2020: 68T50
Research article
Full text in Russian

Received June 6, 2025
Revised August 26, 2025
Accepted August 27, 2025

The article addresses the development of a methodology for hierarchical multi-task learning of neural networks, inspired by the ERNIE 3 architecture, and its experimental validation using the FRED-T5 model for Russian-language text analysis and generation tasks. Hierarchical multi-task learning represents a promising approach for creating universal language models capable of efficiently solving a variety of natural language processing (NLP) tasks. The proposed methodology integrates specialized encoder blocks for natural language understanding (NLU) tasks with a shared decoder for natural language generation (NLG) tasks, thus improving model performance and reducing computational costs. This paper presents a comparative analysis of the developed methodology's performance using the open Russian SuperGLUE benchmark and the pre-trained Russian-language model FRED-T5-1.7B. Experimental results confirm a significant improvement in model quality in both zero-shot and few-shot scenarios compared to the baseline configuration. Additionally, the paper explores practical applications of the developed approach in real NLP tasks and provides recommendations for further advancement of the methodology and its integration into applied systems for processing Russian-language texts.

Keywords: hierarchical multi-task learning; FRED-T5; natural language processing; neural networks; text generation; text analysis; zero-shot learning; few-shot learning; seq2seq models

INFORMATION ABOUT THE AUTHORS

Totmina, Ekaterina V. (corresponding author)	ORCID iD: 0009-0000-7413-1503. E-mail: e.totmina@g.nsu.ru Master's Student, Faculty of Information Technology
Bondarenko, Ivan	ORCID iD: 0000-0002-0457-0698. E-mail: i.bondarenko@g.nsu.ru Senior Lecturer, Department of Fundamental and Applied Linguistics, Humanitarian Institute
Seredkin, Aleksandr V.	ORCID iD: 0009-0007-6873-7868. E-mail: a.seredkin@g.nsu.ru Junior Researcher, Laboratory of Applied Digital Technologies

For citation: E. V. Totmina, I. Bondarenko, and A. V. Seredkin, "Hierarchical multi-task learning methodology for ERNIE-3-Type neural networks in Russian-language text analysis and generation", *Modeling and Analysis of Information Systems*, vol. 32, no. 3, pp. 282–297, 2025. DOI: 10.18255/1818-1015-2025-3-282-297.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 32, № 3, 2025
сайт журнала: www.mais-journal.ru

ARTIFICIAL INTELLIGENCE

Методология иерархического многозадачного обучения нейронных сетей типа ERNIE 3 для анализа и генерации русскоязычных текстов

Е. В. Тотмина[1], И. Бондаренко[1], А. В. Середкин[1]

DOI: 10.18255/1818-1015-2025-3-282-297

[1]Новосибирский национальный исследовательский государственный университет, Новосибирск, Россия

УДК 004.852
Научная статья
Полный текст на русском языке

Получена 6 июня 2025 г.
После доработки 26 августа 2025 г.
Принята к публикации 27 августа 2025 г.

Статья посвящена разработке методологии иерархического многозадачного обучения нейронных сетей, основанной на принципах архитектуры ERNIE 3, и экспериментальной апробации данной методологии на базе модели FRED-T5 для задач анализа и генерации текстов на русском языке. Иерархическое многозадачное обучение является перспективным подходом к созданию универсальных языковых моделей, способных эффективно решать разнообразные задачи обработки естественного языка (NLP). Предложенная методология объединяет преимущества специализированных энкодерных блоков для задач понимания текста (NLU) и общего декодера для генеративных задач (NLG), что позволяет повысить производительность модели и снизить вычислительные затраты. В работе проведён сравнительный анализ эффективности разработанной методологии на открытом бенчмарке Russian SuperGLUE с использованием предварительно обученной русскоязычной модели FRED-T5-1.7B. Экспериментальные результаты подтвердили существенное улучшение качества модели в режимах zero-shot и few-shot по сравнению с базовой конфигурацией. Дополнительно рассмотрены возможности практического применения разработанного подхода в решении реальных NLP-задач, а также даны рекомендации по дальнейшему развитию методологии и её интеграции в прикладные системы обработки русскоязычных текстов.

Ключевые слова: иерархическое многозадачное обучение; FRED-T5; обработка естественного языка; нейронные сети; генерация текста; анализ текста; zero-shot обучение; few-shot обучение; seq2seq модели

ИНФОРМАЦИЯ ОБ АВТОРАХ

Тотмина, Екатерина Вадимовна (автор для корреспонденции)	ORCID iD: 0009-0000-7413-1503. E-mail: e.totmina@g.nsu.ru Магистрант факультета информационных технологий
Бондаренко, Иван	ORCID iD: 0000-0002-0457-0698. E-mail: i.bondarenko@g.nsu.ru Старший преподаватель кафедры фундаментальной и прикладной лингвистики Гуманитарного института
Середкин, Александр Валерьевич	ORCID iD: 0009-0007-6873-7868. E-mail: a.seredkin@g.nsu.ru Младший научный сотрудник лаборатории прикладных цифровых технологий

Для цитирования: E. V. Totmina, I. Bondarenko, and A. V. Seredkin, “Hierarchical multi-task learning methodology for ERNIE-3-Type neural networks in Russian-language text analysis and generation”, *Modeling and Analysis of Information Systems*, vol. 32, no. 3, pp. 282–297, 2025. DOI: 10.18255/1818-1015-2025-3-282-297.

Введение

В последнее время, большие предобученные языковые модели (англ. *Large Language Models*, LLMs) такие, как модели семейства GPT [1], LLaMA [2], PaLM [3], T5 [4] и др., продемонстрировали значительные успехи в решении многих задач обработки естественного языка (NLP). Они показали, что увеличение масштабов и параметров существенно улучшает качество выполнения задач, связанных с генерацией текста и его пониманием [2], [5]. Однако традиционные методы разработки и дообучения таких моделей, основанные на использовании простых текстов, обладают рядом ограничений.

Недавние исследования [1, 4—6] показывают, что большие языковые модели (LLMs) сталкиваются с рядом серьёзных ограничений при решении задач, требующих глубокого понимания языка и интеграции внешних знаний. Во-первых, модели испытывают трудности с обобщением на новые данные, отличные от обучающих [7], что приводит к снижению эффективности при работе в нестандартных контекстах и ситуациях. Во-вторых, обобщение на генеративных моделях вроде LLaMA и других, построенных на автогрессивных языковых моделях (англ. *AutoRegressive Language Models*), сталкивается с ключевыми ограничениями: односторонний контекст мешает задачам понимания естественного языка (англ. *Natural Language Understanding*, NLU), вся модель используется целиком даже для «лёгких» классификаций, а единая LM-голова порождает конфликт градиентов между генерацией и классификацией. Именно такие узкие места мы устраняем, вводя двухблочный энкодер с отдельными головами и разграниченным путём градиента. В-третьих, их способность решать сложные задачи ограничена: хотя модели типа T5 или семейства GPT показывают высокие результаты на стандартных тестах, они часто не справляются с задачами, требующими глубокого анализа и интеграции знаний [8]. Кроме того, традиционные подходы используют универсальные стратегии обработки данных по типу «один подход для всех» (англ. *one-size-fits-all*), что снижает эффективность моделей в многозадачных и контекстно сложных сценариях. Наконец, обучение sequence-to-sequence (seq2seq) моделей оказывается более сложным процессом, поскольку возникают трудности с передачей градиента от декодера к энкодеру, что может привести к расходимости в процессе обучения [9].

Целью данной статьи является улучшение обобщающей способности фундаментальных языковых моделей за счёт разработки и реализации собственной методологии иерархического многозадачного обучения, основанной на принципах, предложенных авторами ERNIE 3.0 Titan [10], а также экспериментальная оценка эффективности полученной многозадачной модели FRED-T5 [11] на задачах обработки естественного языка для русского языка. Предлагаемая методология предполагает применение специальных и общих энкодерных слоёв, что позволяет повысить эффективность передачи знаний между задачами. В работе подробно рассмотрен принцип иерархического многозадачного обучения (англ. *Hierarchical Multi-Task Learning*) [12], согласно которому задачи понимания естественного языка выполняются специализированными энкодерами, а задачи генерации текста (англ. *Natural Language Generation*, NLG) — единым декодером. В рамках задач NLU разработанная по представленной методологии система обеспечивает эффективное решение таких задач, как: (а) распознавание именованных сущностей (Named Entity Recognition, NER), (б) многоклассовая классификация (Multiclass Classification), (в) распознавание подразумеваемых отношений (Natural Language Inference, NLI). Среди задач NLG, поддерживаемых системой, можно выделить следующие: (а) генерация парафразов (Paraphrase Generation), (б) генерация вопросов (Question Generation, QG), (в) генерация ответов на вопросы (Question Answering, QA), (г) суммаризация текста (Text Summarization), (д) генерация заголовков (Title Generation). Такое архитектурное решение позволяет существенно сократить время инференса за счёт использования только части архитектуры для задач NLU, в то время как для задач NLG задействуется вся модель целиком. В результате модель демонстрирует не только

более компактную структуру по сравнению с традиционными крупными языковыми моделями, такими как GPT, но и лучшие показатели устойчивости и обобщения на различных типах задач.

1. Обзор научной литературы

Концепция многозадачного обучения (англ. *Multi-Task Learning*, MTL) рассматривает параллельное формирование модели на совокупности задач с целью индуктивного переноса знаний и повышения обобщающей способности алгоритма [1]. Фундамент современных языковых моделей составляют трансформеры, впервые описанные в работе [13]; ключевой их компонент — механизм самовнимания (англ. *self-attention*), обеспечивающий извлечение глобальных зависимостей и формирование контекстно-чувствительных представлений текста. Реализация многоголового внимания позволяет модели одновременно учитывать несколько проекций контекста, что способствует построению универсальных лингвистических признаков.

Увеличение масштаба моделей — например, GPT-3 с 175 млрд параметров [1] — ведёт к существенному росту качества, однако сопровождается резким увеличением вычислительных затрат и требует специализированных схем распределённого обучения, таких как разбиение модели (англ. *pipeline parallelism*) и использование разреженных экспертных моделей (англ. *Mixture-of-Experts*). Одним из инструментов решения указанных проблем является многозадачное обучение, позволяющее сформировать устойчивые представления.

Современные архитектуры, ориентированные на MTL, — T5 [4], T0 [14], UL2 [15] — предлагают унифицированные text-to-text формулировки заданий и гибридные предобучающие режимы, что упрощает совместное обучение широкого спектра NLP-задач. Тем не менее простое агрегирование разнородных данных оказывается недостаточно эффективным из-за интерференции задач. Эту проблему адресуют иерархические MTL-подходы, группирующие задания по уровню сложности либо семантической близости, минимизируя негативный перенос и повышая качество обобщения [16], [17]. Представительным примером является ERNIE 2.0 [18], где последовательное введение задач различной сложности позволило достичь высоких показателей без катастрофического забывания.

Дальнейшее развитие MTL предполагает использование модульных структур — адаптеров и специализированных подсетей — которые обеспечивают гибкое управление параметрами и снижают риск переобучения. Перспективными также являются методы динамического взвешивания и адаптивной балансировки градиентов, автоматизирующие распределение внимания модели между задачами.

Отдельное направление развития MTL и языковых моделей в целом связано с интеграцией внешних знаний (графов знаний, энциклопедических корпусов), расширяющих семантическое пространство модели и повышающих достоверность генерации. В модели ERNIE 3.0 Titan [10] сочетание многоуровневых предобучающих заданий и knowledge-aware-механизмов обеспечило рекордные результаты на ряде бенчмарков, что подтверждает эффективность комплексного, структурированного MTL. Ожидается, что дальнейшее исследование данных подходов приведёт к созданию более универсальных и экономичных языковых моделей, способных адаптироваться к широкому кругу типов задач.

Особый интерес в контексте русскоязычных NLP-задач представляет семейство моделей FRED-T5, разработанное командой SberDevices. Базовая версия FRED-T5-1.7B (1.7 млрд параметров), представленная в 2022 году, стала одной из первых крупных seq2seq-моделей, предобученных на обширном русскоязычном корпусе текстов. Её ключевыми особенностями являются расширенный токенизатор, оптимизированный для морфологически богатой русской лексики, и высокая эффективность в задачах генерации, классификации и анализа текста.

На сегодняшний день основными конкурентами FRED-T5, основанными на архитектуре T5, являются такие модели, как ruT5 и более крупные многоязычные модели — mT5 [19]. FRED-T5 проде-

монстрировал высокий уровень качества среди специализированных моделей для русского языка. В настоящее время в открытом доступе представлен ряд дообученных вариантов, оптимизированных под прикладные задачи:

1. По результатам на Russian SuperGLUE по состоянию на май 2025 года, два доработанных варианта модели демонстрируют конкурентоспособные результаты в данном бенчмарке: (а) модель FRED-T5 1.7B (only encoder 760M) finetune показывает средний балл (total score) 0.694, что соответствует 12-й позиции в рейтинге (лидерборде); (б) модель FRED-T5 large finetune достигает более высокого результата — 0.706 балла, занимая 4-е место в лидерборде. Последняя из моделей демонстрирует существенное улучшение производительности на комплексных задачах, требующих глубокого анализа контекста и логических рассуждений.
2. Исследовательским сообществом и разработчиками были созданы многочисленные производные модели, полученные путем тонкой настройки базового FRED-T5 на специфических датасетах. Например, FRED-T5-Summarizer, sage-fredt5-distilled-95m показывают гибкость архитектуры и являются прямыми аналогами для сравнения в задачах предметной адаптации больших языковых моделей.

Таким образом, семейство FRED-T5, включающее его масштабированные и инструктивные вариации, формирует актуальный state-of-the-art для русскоязычных seq2seq-моделей. Сопоставление эффективности и новизны предлагаемого в работе решения необходимо проводить именно с этими специализированными версиями, а не только с базовой моделью, чтобы корректно позиционировать свой вклад в современное состояние области.

2. Развитие концепции ERNIE 3.0 Titan

Модель ERNIE 3.0 Titan, представленная в 2021 году [10], стала значительным развитием концепции иерархического многозадачного обучения в обработке естественного языка. Основной особенностью данной модели является чёткое разделение на универсальное ядро и специализированные подсети, которые отдельно отвечают за задачи NLU и NLG.

Архитектура ERNIE 3.0 Titan состоит из нескольких слоёв. Базовый уровень модели образует мощный трансформер, предназначенный для извлечения общих семантических признаков из текстовых данных. Этот универсальный модуль включает от 12 до 24 слоёв трансформера, в зависимости от конфигурации модели, и формирует общее языковое представление текста. Верхние слои модели разделены на две специализированные ветви: первая решает задачи анализа и понимания текста (классификация, извлечение информации, вопросы и ответы), а вторая предназначена для генеративных задач, таких как создание связного текста и ведение диалогов [10], [20].

ERNIE 3.0 Titan вводит также интеграцию внешних знаний, полученных из баз знаний и графов знаний, прямо в процесс обучения. Это позволяет модели не только предсказывать слова или фразы, но и восстанавливать отношения между сущностями, такими как «столица — страна». В процессе обучения Titan динамически обогащается новыми задачами, начиная с более простых (лексический уровень) и постепенно переходя к более сложным задачам синтаксического и семантического уровней, что предотвращает потерю усвоенных ранее навыков.

Для контроля качества генерации модель использует самосостязательный подход (англ. *self-adversarial*), при котором обучается дополнительный модуль-критик (критик-ранкер, англ. *critic-ranker*), оценивающий правдоподобность сгенерированного текста. Также реализована управляемая генерация (англ. *controllable generation*), которая позволяет задавать конкретные параметры текста, такие как стиль и длина. Дополнительно применяется механизм онлайн-дистилляции, обеспечивающий параллельное обучение облегченных версий модели, что позволяет использовать рассматриваемый в статье [10] подход в практических задачах с меньшими затратами ресурсов.

Модель ERNIE 3.0 Titan была выбрана в качестве референсной точки для сравнения, поскольку её архитектурная философия наиболее точно соответствует ключевой идее данного исследования —

созданию унифицированной модели для совместного обучения как пониманию (NLU), так и генерации текста (NLG). В то время как другие методологии (такие как AdapterFusion [21], Mixture-of-Experts [22] или Flan-T5 [23]) предлагают высокоэффективные решения для отдельных аспектов многозадачности — например, параметрической эффективности или масштабируемости, — их основные цели и архитектурные принципы отличаются от нашей конкретной задачи. ERNIE 3.0, напротив, явно демонстрирует реализацию именно той двунаправленной синергии между NLU и NLG, которую стремится развить наша работа, и поэтому служит наиболее релевантным ориентиром для сравнения.

Таким образом, ERNIE 3.0 Titan иллюстрирует эффективность современных подходов к многозадачному обучению в NLP. Иерархическое построение модели, поддержка множества задач и парадигм, а также встроенная работа с внешними знаниями позволяют преодолевать ограничения предыдущих поколений языковых моделей. Дальнейшие исследования в этом направлении будут способствовать созданию ещё более мощных универсальных моделей, способных учиться быстрее, объясняться лучше и безопасно применяться в различных областях.

3. Этапы реализации разрабатываемой методологии

В качестве базовой модели была выбрана открытая модель FRED-T5-1.7B. Выбор основывался на предварительном сравнении нескольких доступных в открытом доступе моделей семейства T5: ruT5-Large (774 млн параметров), FRED-T5-1.7B (1.7 млрд параметров) и mT5-XL (3 млрд параметров). Основным критерием выбора выступала возможность обучения на имеющихся вычислительных ресурсах: модель с 1.7 млрд параметров обеспечивала оптимальный баланс между размером, допустимым для эффективного обучения на доступных ресурсах, и качеством — на момент мая 2025 года её дообученный вариант FRED-T5 1.7B finetune занял 4 место на бенчмарке Russian SuperGLUE со средней оценкой 0.762. Дополнительными факторами в пользу выбора послужили наличие расширенного словаря русскоязычных токенов, открытая лицензия и изначальная специализация под задачи обработки русского языка. Однако ввиду отсутствия модели FRED-T5 1.7B finetune в открытом доступе ее использование в настоящем исследовании не представлялось возможным.

Следующий этап — создание собственного корпуса. Данный шаг обусловлен отсутствием в открытом доступе русскоязычных многозадачных больших (размером свыше 1 млн примеров) наборов данных, подходящих для одновременного обучения модели на задачах понимания и генерации текста. В рамках задач NLU модель обучалась на таких задачах, как: (а) распознавание именованных сущностей (NER), (б) многоклассовая классификация (MC), (в) распознавание подразумеваемых отношений (NLI). Среди задач NLG модель обучалась на: (а) генерации парафразов (Paraphrase generation), (б) генерации вопросов (QG), (в) генерации ответов на вопросы (QA), (г) суммаризации текста (Text summarization), (д) генерации заголовков (Title Generation).

На финальном этапе была реализована многозадачная модель на основе архитектуры FRED-T5 с применением принципа иерархической многозадачности. Разрабатываемая архитектура включает специализированные энкодеры, отдельно предназначенные для каждой задачи понимания текста, а также единый общий декодер, отвечающий за выполнение задач генерации текста. Архитектура модели была организована с использованием модульного подхода, что существенно упрощает её последующее тестирование и расширение для решения новых задач. Схематические представления реализованной методологии и базовой архитектуры модели FRED-T5-1.7B представлено на рис. 1.

Архитектура модели состоит из двух последовательных энкодерных блоков и одного декодерного блока.

Первый энкодерный блок (Encoder-Block-1, 6 слоёв) формирует первичное («поверхностное») представление, общее для всех решаемых задач. С данного уровня представлений берутся признаки для следующих двух задач классификации: а) тематическая классификация (*MC-head*, линейный классификатор на 17 классов); б) токенная классификация именованных сущностей (*NER-head*,

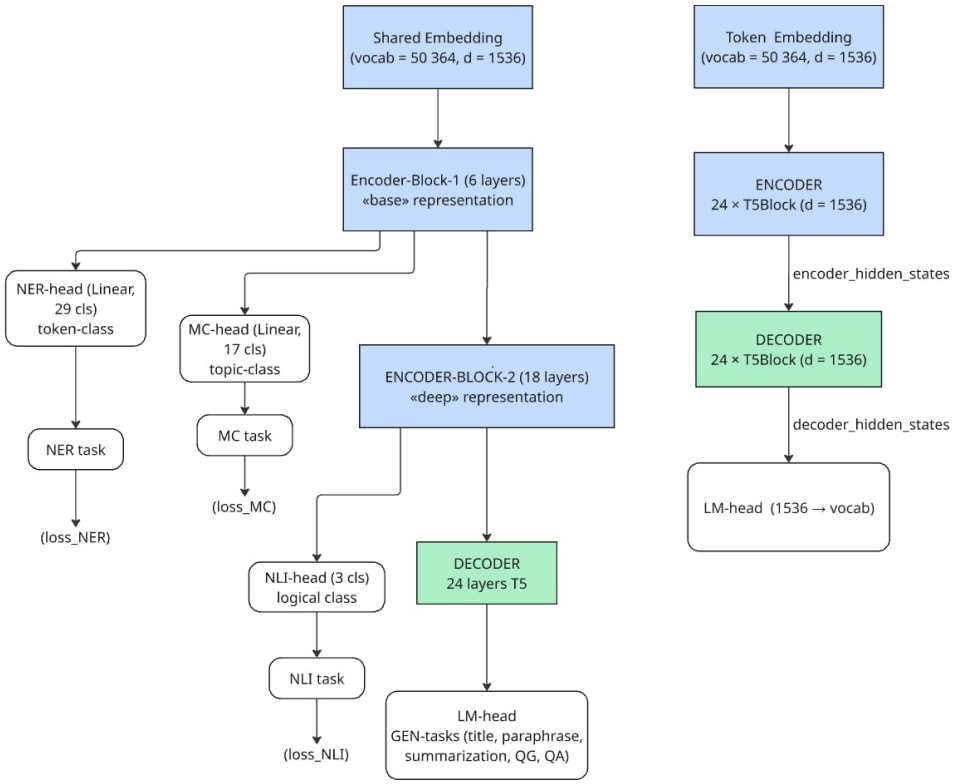

Fig. 1. Structure of the implemented hierarchical approach based on the FRED-T5-1.7B model (left) compared to the original model architecture (right)

Рис. 1. Структура реализованного иерархического подхода на основе модели FRED-T5-1.7B (слева) в сравнении с исходной архитектурой модели (справа)

линейный классификатор на 29 классов). К данным головам подключён компактный линейный адаптер (классический bottleneck-adapter, аналогичный описанному в статье [24]), позволяющий стабилизировать обучение и обеспечить эффективный перенос знаний между задачами без значительного увеличения числа параметров. Использование только первого блока энкодера для этих задач позволяет снизить вычислительную нагрузку и ускорить инференс, поскольку им не требуется глубокий контекст.

Второй энкодерный блок (*Encoder-Block-2*, дополнительные 18 слоёв) активируется при решении задач, требующих более глубоких контекстных представлений. Получаемые на выходе признаки используются для задачи определения логических отношений между предложениями (*NLI-head*, линейный классификатор на 3 класса) и для последующей передачи в декодер при решении генеративных задач. Аналогично предыдущим задачам, для NLI-головы также подключён компактный линейный адаптер (англ. *bottleneck-adapter*), способствующий стабилизации обучения и переносу знаний.

Декодерный блок (*Decoder*, 24 слоя T5) в сочетании с линейной головой генерации (*LM-head*) решает набор генеративных задач (генерация заголовков, перефразирование, суммаризация, генерация вопросов и ответов). Для этой головы также применяется компактный линейный адаптер (bottleneck-adapter, аналогичный описанному выше [24]), позволяющий стабилизировать обучение и обеспечить эффективный перенос знаний между генеративными сценариями без значительного увеличения числа параметров. Предложенная архитектура обеспечивает каждой задаче необходимую глубину вычислений: задачи MC и NER ограничены первым блоком энкодера, задача NLI использует оба блока энкодера, а генеративные задачи задействуют всю структуру (*Encoder-Block-1*,

Encoder-Block-2 и *Decoder*). Такое разделение градиентных потоков позволяет снизить время обучения и инференса на этих задачах без ухудшения качества модели.

Модель обучалась одновременно на всех задачах в рамках единого цикла, без этапного распределения (англ. *curriculum*). Для оптимизации применялся адаптивный оптимизатор AdamW совместно с циклическим расписанием шага обучения CyclicLR [25] (режим *exp_range*, $\gamma = 0.9$): в течение цикла шаг обучения монотонно возрастал в заданном диапазоне, при этом на каждую итерацию накладывалось дополнительное экспоненциальное масштабирование γ^t, выполняющее роль регуляризующей мажоранты и ускоряющее сходимость.

Для стабилизации обучения имитировался крупный размер батча за счёт накопления градиентов: физический размер батча составлял 12 примеров с шагом накопления, равным 2, что эквивалентно эффективному размеру батча из 24 примеров. Для предотвращения возможного «взрыва» значений градиентов применялась их обрезка по норме (англ. *gradient clipping*) [26]. В совокупности, перечисленные изменения позволили создать более компактную и гибкую модель, демонстрирующую устойчивую динамику обучения на смешанных корпусах задач NLU и NLG и упрощающую последующую адаптацию к новым сценариям в парадигме few-shot tuning.

4. Создание единого корпуса данных

В качестве базового корпуса данных для реализации предлагаемой методологии был выбран открытый датасет RULM, охватывающий тексты различных жанров и тематик, включая разговорные тексты, новостные статьи и другие типы данных, всего свыше 75 миллионов текстов. Для повышения качества данных была проведена их предварительная очистка, направленная на удаление зашумлённых и некачественных текстов, содержащих эмодзи, специальные символы и другие нежелательные элементы. После очистки была сформирована выборка из примерно 870 тысяч текстов различной длины и тематик, что обеспечивает необходимое разнообразие для эффективного обучения многозадачной модели. Дополнительно использовались уже размеченные датасеты, такие как NEREL, а также специализированные корпуса с конференций по компьютерной лингвистике, включая RuSimpleSentEval, что позволило значительно повысить качество разметки и разнообразие типов задач. Итоговый объём сформированного набора данных составляет около 950 тыс. примеров.

Для формирования разметки по каждой из рассматриваемых задач понимания и генерации естественного языка была применена техника автоматической аннотации данных с использованием предварительно обученных моделей из открытых источников, адаптированных под конкретные NLP-задачи. В качестве примера, для задач генерации вопросов и ответов (QA и QG соответственно) использовалась модель mdeberta-v3-base-squad2.

В результате интеграции описанных выше источников данных и использования стратегий автоматической аннотации с помощью адаптированных предобученных моделей и последующей постобработки был сформирован итоговый корпус данных, содержащий примерно 950 тыс. примеров. Данный корпус обеспечивает разнообразие и высокое качество данных для эффективного многозадачного обучения модели.

5. Обучение модели и оценка её эффективности

Обучение модели проводилось в едином мультизадачном цикле на узле с одной видеокартой GPU NVIDIA A100 80 GB. Для каждой итерации формировался батч из 12 последовательностей; за счёт шага накопления градиентов (англ. *accumulation step*) равного 2 эффективный размер батча составлял 24 примера. Обучение выполнялось в режиме смешанной точности (англ. *mixed precision*): основные матричные операции и активации вычислялись в *bfloat16*, тогда как мастер-копии весов и накопление градиентов велись в *fp32*, что обеспечивало как экономию памяти, так и числовую стабильность с включённым градиентным чекпоинтингом (англ. *gradient-checkpointing*), что позво-

ляло удерживать последовательности длиной до 512 токенов в памяти и при этом укладываться в заданный объём видеопамяти.

На каждой итерации обрабатывались батчи данных всех восьми задач, при этом градиенты аккумулировались на протяжении двух шагов перед обновлением параметров. В качестве основной функции потерь для задач NER, MC и NLI использовалась классическая кросс-энтропия (англ. *Cross-Entropy Loss*) (1).

$$L = -\sum_{i=1}^{C} y_i \log(p_i), \tag{1}$$

где

- C — общее число классов;
- y_i — истинная метка для i-го класса (обычно представляется в виде one-hot вектора, где значение равно 1 для правильного класса и 0 для остальных);
- p_i — предсказанная вероятность принадлежности i-му классу.

Для задач генерации текста (QG, QA, Paraphrase, Summarization, Title Generation) использовалась функция потерь token-level Cross-Entropy Loss (Language Modeling Loss) (2), вычисляемая на основе вероятностей генерации каждого токена относительно эталонной последовательности.

$$L = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{C} y_{t,i} \log(p_{t,i}), \tag{2}$$

где

- T — общее число токенов (например, в последовательности или батче);
- C — число классов (обычно размер словаря);
- $y_{t,i}$ — индикатор истинной метки: равен 1, если для токена на позиции t правильный класс имеет индекс i, иначе 0 (one-hot представление);
- $p_{t,i}$ — предсказанная моделью вероятность того, что токен на позиции t принадлежит классу i.

Для оценки динамики обучения использовались следующие метрики:

- Average Train Loss — средняя величина функции потерь по всем задачам на тренировочной выборке в пределах каждой эпохи;
- Average Validation Loss — аналогичный показатель, рассчитываемый на валидационной выборке.

В качестве оптимизатора использовался AdamW в связке с циклическим расписанием CyclicLR (*exp_range*, $\gamma = 0.9$). Параметр скорости обучения (англ. *learning rate*) монотонно возрастал от 1×10^{-6} до 1×10^{-4}, после чего на каждой итерации масштабировался экспоненциальным множителем γ^t, выступая дополнительным регуляризатором и ускоряя сходимость. Числовую стабильность обеспечивали обрезка по норме с порогом $\|g\| \leqslant 1.0$, обучение в режиме смешанной точности и контрольные точки активаций, что позволило экономить память без потери точности и снижать риск переобучения.

Динамика обучения на выборке, содержащей 10 тыс. примеров демонстрирует устойчивое снижение средних значений функции потерь на протяжении 4 эпох как на валидационной, так и на тренировочной выборках (рис. 2). Отсутствие расхождения между кривыми потерь свидетельствует о стабильности обучения и отсутствии переобучения, что подтверждает обоснованность выбранной методики.

Полученные результаты позволили перейти к масштабному обучению модели на расширенном корпусе, включающем около 900 тыс. примеров для обучения и примерно 50 тыс. примеров для валидации. Обучение проводилось в течение 4 эпох. Рис. 3 демонстрирует динамику средней валидационной функции потерь: значение снизилось с 425,9 на первой эпохе до 420,2 к концу четвёртой. Для обучающей выборки потери уменьшились с 427 до 371 к концу четвёртой эпохи.

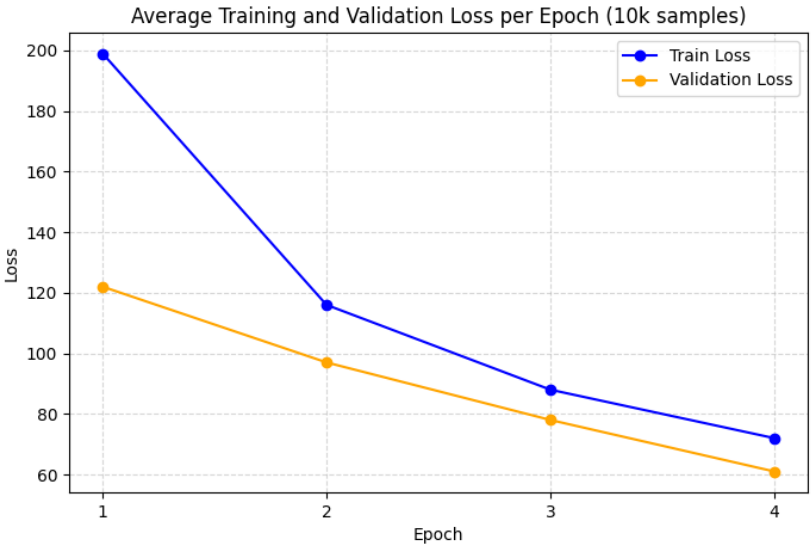

Fig. 2. Plot of the average train (blue) and validation (orange) losses for the model trained on 10 000 train and 1 000 validation samples

Рис. 2. График средней оценки тренировочной (голубая линия) и валидационной (оранжевая линия) функции потерь обучения модели на 10 тыс. примерах в тренировочной выборке и 1 тыс. примеров — в валидационной

Одновременное и последовательное снижение обеих кривых указывает на корректную работу процесса оптимизации: обучение протекало стабильно, без явных признаков дивергенции или переобучения. Несмотря на то, что уменьшение валидационной метрики невелико (порядка 1 %), оно согласуется с положительной динамикой модели на внешнем бенчмарке Russian SuperGLUE, что косвенно подтверждает её способность извлекать полезные закономерности из данных. Выбранные гиперпараметры не подбирались с целью оптимизации, а были заданы исходя из теоретических соображений и допустимых вычислительных ограничений; при необходимости они могут быть изменены для дальнейших экспериментов.

6. Анализ эффективности предложенной методологии на бенчмарке Russian SuperGLUE

Тестирование полученной модели на бенчмарке Russian SuperGLUE было направлено на объективную оценку эффективности разрабатываемой методологии. Учитывая различия в вычислительных ресурсах и методах обучения существующих моделей (в частности, модель от SberDevice под названием «FRED-T5 1.7B finetune», опубликованная на бенчмарке и на май 2025 года имеющая среднюю оценку 0.762, была обучена по закрытой схеме с использованием крупного GPU-кластера), для обеспечения сопоставимости результатов и исключения влияния дополнительных факторов было принято решение сравнивать предлагаемые модели с открытой базовой моделью FRED-T5-1.7B, размещённой на платформе HuggingFace. Подобный подход к организации экспериментов позволил минимизировать влияние посторонних факторов и объективно подтвердить, что улучшение результатов обусловлено предложенным подходом. Серия сравнительных экспериментов проводилась в двух режимах:

1. Zero-shot (без предварительного обучения на конкретных задачах) — обе модели (базовая и предлагаемая) без какого-либо дообучения выполняли предсказания для девяти тестовых задач из набора Russian SuperGLUE: LiDiRus, RCB, PARus, MuSeRC, TERRa, RUSSE, RWSD, DaNetQA

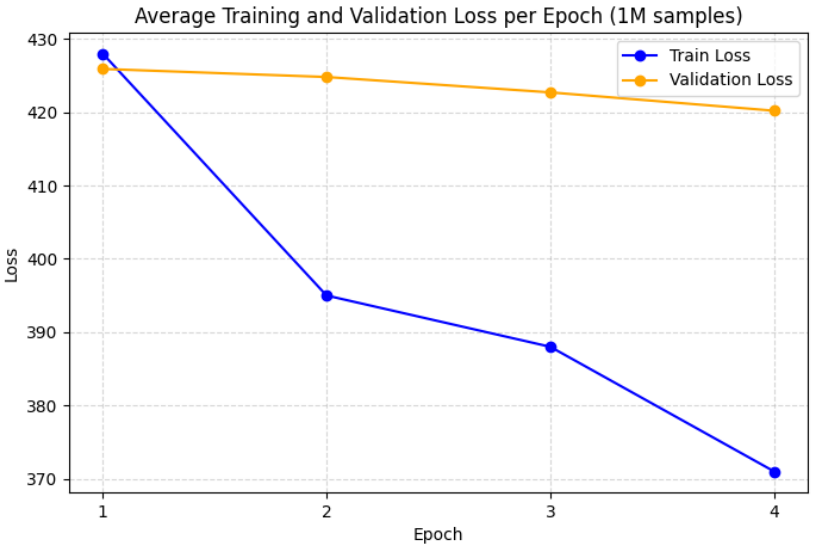

Fig. 3. Plot of the average train (blue) and validation (orange) losses for the model trained on a dataset with 900,000 train samples and 50,000 validation samples

Рис. 3. График средней оценки тренировочной (голубая линия) и валидационной (оранжевая линия) функции потерь обучения модели на корпусе данных с 900 тыс. примеров в тренировочной и 50 тыс. примерами в валидационной выборках

и RuCoS. После получения предсказаний результаты подавались на бенчмарк, и выполнялось сравнение по средним итоговым показателям точности.

2. Few-shot (20-shot) — обе модели предварительно обучались непосредственно на тренировочных выборках указанных задач в идентичных условиях. После этапа дообучения аналогичным образом публиковались результаты на тестовом наборе задач, и модели сравнивались по итоговым средним показателям.

Ниже в таблице 1 приведены результаты проведённых экспериментов, демонстрирующие полученные средние оценки. Сравнение средней точности базовой модели FRED-T5-1.7B и моделей, дообученных с помощью предложенной методологии на корпусах разного размера представлены в таблице 2.

Результаты экспериментов показали, что дополнительное обучение FRED-T5-1.7B на корпусе из 10 тыс. примеров в течение четырёх эпох повышает её средний балл на Russian SuperGLUE:

1. В zero-shot режиме метрика увеличилась с 0,432 до 0,506 (примерно на 17% относительно базовой модели).
2. В 20-shot режиме — с 0,504 до 0,518 (прирост около 3 %).

При расширении обучающей выборки до 950 000 примеров, также за четыре эпохи, прирост оказался немного заметнее:

1. В zero-shot результат вырос с 0,432 до 0,551 (около 28 %);
2. В 20-shot — с 0,504 до 0,637 (около 26 %).

Примечательно, что значительный прирост в режиме zero-shot был достигнут без дополнительного целевого обучения под конкретные задачи. Это указывает на улучшение обобщающей способности модели и её повышенную чувствительность к формулировкам задач и контексту даже в условиях отсутствия примеров для настройки.

Итого, модель с лучшими результатами (названная при подаче на бенчмарк «20_mfredt5_2bl»), основанная на архитектуре FRED-T5-1.7B, и обученная на корпусе данных объёмом 900 тыс. при-

Table 1. Average scores obtained from comparative testing of the baseline and fine-tuned FRED-T5-1.7B models on the Russian SuperGLUE benchmark

Таблица 1. Средние оценки, полученные после проведения сравнительного тестирования базовой и дообученных моделей FRED-T5-1.7B на бенчмарке Russian SuperGLUE

Модель	Количество эпох обучения на датасетах Russian SuperGLUE	Название сабмита при подаче на бенчмарк	Средняя оценка на бенчмарке
FRED-T5-1.7B	0	zero_fredt5	0.432
FRED-T5-1.7B	20	20_fredt5	0.504
Наша FRED-T5-1.7B (10 тыс. примеров, 4 эпохи)	0	zero_mfredt5	0.506
Наша FRED-T5-1.7B (10 тыс. примеров, 4 эпохи)	20	20_mfredt5	0.518
Наша FRED-T5-1.7B (950 тыс. примеров, 4 эпохи)	0	zero_mfredt5_2bln	0.551
Наша FRED-T5-1.7B (950 тыс. примеров, 4 эпохи)	20	20_mfredt5_2bln	0.637

Table 2. Results of comparative testing of the baseline and fine-tuned FRED-T5-1.7B models on the Russian SuperGLUE benchmark

Таблица 2. Результаты сравнительного тестирования базовой и дообученных моделей FRED-T5-1.7B на бенчмарке Russian SuperGLUE

Модель	Средняя точность по эпохам обучения	
	0 эпох	20 эпох
FRED-T5-1.7B	0.432	0.504
Наша FRED-T5-1.7B (10 тыс. примеров, 4 эпохи)	0.506 (+17 %)	0.518 (+3 %)
Наша FRED-T5-1.7B (950 тыс. примеров, 4 эпохи)	0.551 (+28 %)	0.637 (+26 %)

меров (тренировочная выборка) и 50 тыс. примеров (валидационная выборка) показала на бенчмарке итоговую среднюю оценка 0.637. Детальные результаты по отдельным задачам, полученных для двух моделей под названиями «20_mfredt5_2bl» и «20_fredt5», представлены в таблице 3.

При сравнении результатов моделей «20_fredt5» (базовая модель FRED-T5-1.7B, обученная на задачах Russian SuperGLUE в течение 20 эпох) и «20_mfredt5_2bl» (аналогичная модель, дообученная по предложенной методологии, также на 20 эпохах), наблюдается небольшое преимущественное улучшение показателей второй модели, прошедшей дополнительное предварительное обучение. На рис. 4 представлены скриншоты данных оценок с сайта Russian SuperGLUE[1] для рассматриваемых моделей.

Наибольшие улучшения наблюдаются на задачах, предполагающих чёткие формулировки и наличие явных текстовых признаков. Так, на задачах PARus и TERRa предложенная модель достигла значительного прироста точности: на PARus точность возросла с 0.68 до 0.876, а на TERRa — с 0.685 до 0.875. Это свидетельствует о повышенной способности модели эффективно идентифицировать

[1] https://russiansuperglue.com/

Table 3. Table 3. Final evaluation scores of the multitask model "20_mfredt5_2bl" and the baseline model "20_fredt5" on the Russian SuperGLUE benchmark.

Таблица 3. Итоговые оценки мультитасдной модели на бенчмарке Russian SuperGLUE «20_mfredt5_2bl» и базовой модели «20_fredt5»

Задача	Оценки модели «20_fredt5»	Оценки модели «20_mfredt5_2bl»	Прирост качества (%)
LiDiRus	0.367 (Matthew's Corr)	0.587 (Matthew's Corr)	+59.95 %
RCB	0.381 / 0.447 (F1 / Accuracy)	0.371 / 0.434 (F1 / Accuracy)	-2.62 % / -2.91 %
PARus	0.68 (Accuracy)	0.876 (Accuracy)	+28.82 %
MuSeRC	0.79 / 0.498 (F1a / EM)	0.924 / 0.771 (F1a / EM)	+16.96 % / +54.82 %
TERRa	0.685 (Accuracy)	0.875 (Accuracy)	+27.74 %
RUSSE	0.563 (Accuracy)	0.641 (Accuracy)	+13.85 %
RWSD	0.338 (Accuracy)	0.338 (Accuracy)	0 %
DaNetQA	0.513 (Accuracy)	0.513 (Accuracy)	0 %
RuCoS	0.34 / 0.325 (F1 / EM)	0.66 / 0.64 (F1 / EM)	+94.12 % / +96.92 %

Total score: 0.637

Dataset	Score	Metric
LiDiRus	0.587	Matthew`s Corr
RCB	0.371 / 0.434	F1/Acc
PARus	0.876	Accuracy
MuSeRC	0.924 / 0.771	F1a/Em
TERRa	0.875	Accuracy
RUSSE	0.641	Accuracy
RWSD	0.338	Accuracy
DaNetQA	0.513	Accuracy
RuCoS	0.66 / 0.64	F1/EM

Diagnostic (Matthew`s Correlation): 0.587

Category	Score
LOGIC	0.5281737413938106
KNOWLEDGE	0.5602388460037974
PREDICATE-ARGUMENT STRUCTURE	0.5926390590535277
LEXICAL SEMANTICS	0.6623102679914405

Total score: 0.504

Dataset	Score	Metric
LiDiRus	0.367	Matthew`s Corr
RCB	0.381 / 0.447	F1/Acc
PARus	0.68	Accuracy
MuSeRC	0.79 / 0.498	F1a/Em
TERRa	0.685	Accuracy
RUSSE	0.563	Accuracy
RWSD	0.338	Accuracy
DaNetQA	0.513	Accuracy
RuCoS	0.34 / 0.325	F1/EM

Diagnostic (Matthew`s Correlation): 0.367

Category	Score
LOGIC	0.29263376623373305
KNOWLEDGE	0.3812602941508865
PREDICATE-ARGUMENT STRUCTURE	0.36209829585555675
LEXICAL SEMANTICS	0.36344245337393355

Fig. 4. Average scores obtained on the Russian SuperGLUE benchmark for the model "20_mfredt5_2bl" — the baseline FRED-T5-1.7B model in the 20-shot setting (right), and "20_fredt5" — the FRED-T5-1.7B model trained using the developed methodology in the 20-shot setting (left)

Рис. 4. Средние оценки, полученные на бенчмарке Russian SuperGLUE, для модели «20_mfredt5_2bl» — базовая модель FRED-T5-1.7B в режиме 20-shot (справа), «20_fredt5» — обученная по разрабатываемой методологии FRED-T5-1.7B в режиме 20-shot (слева)

причинно-следственные связи и фактологические отношения entailment-типа, которые явно выражены текстовыми маркерами.

Аналогичная позитивная динамика отмечена на задаче MuSeRC: показатель F1a увеличился с 0.79 до 0.924, а метрика полного соответствия EM выросла с 0.496 до 0.771. Несмотря на рост показателей, сохраняется существенный разрыв между F1a и EM, обусловленный строгими требованиями к полному совпадению всех ответов. На задачах RUSSE и RuCoS также был отмечен значительный

прирост качества: для RUSSE точность повысилась с 0.563 до 0.641, а для RuCoS результаты улучшились особенно существенно — метрика F1 выросла почти вдвое (с 0.34 до 0.66), а EM — с 0.325 до 0.64. Улучшение на задаче RuCoS подтверждает, что предложенная модель лучше справляется с анализом сложных квазикореферентных связей и восстановлением пропущенных сущностей.

На задачах, ранее демонстрировавших наибольшие трудности, также наблюдаются улучшения, хотя в целом показатели остаются умеренными. В LiDiRus коэффициент корреляции Мэтьюса вырос с 0.367 до 0.587, что говорит о лучшей способности модели определять эмоциональную полярность текста. На задаче RCB, несмотря на сложность распознавания отношений между текстами, показатель F1 практически не изменился (с 0.381 до 0.371), однако точность несколько снизилась (с 0.447 до 0.434). Результат на RWSD не изменился и остался на уровне 0.338, что подтверждает сложность задачи для данной модели. В задаче DaNetQA точность также не изменилась и составила 0.513, что может быть связано с трудностями модели в распознавании тонких семантических конструкций.

Таким образом, проведённое сравнение показывает, что дообучение модели по предложенной методологии способствует улучшению её показателей на большинстве задач бенчмарка Russian SuperGLUE, особенно на задачах с чётко выраженными текстовыми признаками и относительно простыми формулировками.

Заключение

В работе предложена и реализована методология иерархического многозадачного обучения нейронных сетей, основанная на принципах архитектуры ERNIE 3.0 Titan, и проведена её экспериментальная апробация на русскоязычной модели FRED-T5-1.7B. Разработанная методология эффективно объединяет специализированные энкодерные модули для задач понимания естественного языка и общий декодер для задач генерации текста, что позволило повысить производительность модели и сократить вычислительные затраты при инференсе.

Экспериментальные исследования на открытом бенчмарке Russian SuperGLUE показали, что предложенный подход демонстрирует положительное влияние на качество работы модели как в режиме zero-shot, так и при дообучении на ограниченном числе примеров (few-shot) по сравнению с исходной конфигурацией. При этом наблюдается рост качества по ряду метрик, тогда как в некоторых случаях улучшение оказывается незначительным или отсутствует. Полученные результаты свидетельствуют о применимости методологии для дообучения модели и её способности извлекать полезные закономерности из данных, особенно по задачам, требующим глубокого анализа текста, однако не претендуют на универсальное превосходство над базовой архитектурой FRED-T5.

Таким образом, представленная методология демонстрирует высокую эффективность и универсальность, позволяя успешно решать широкий спектр русскоязычных задач обработки естественного языка. Дальнейшие направления исследований включают оптимизацию архитектуры, расширение многоязычных и мультимодальных возможностей модели, а также интеграцию разработанного подхода в прикладные решения и промышленные продукты.

References

[1] T. Brown *et al.*, “Language models are few-shot learners”, *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[2] H. Touvron *et al.*, *LLaMA: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].

[3] A. Chowdhery *et al.*, “Palm: Scaling language modeling with pathways”, *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[4] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer", *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[5] Y. Zhu *et al.*, "Can large language models understand context?", in *Findings of the Association for Computational Linguistics: EACL 2024*, 2024, pp. 2004–2018.

[6] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends, challenges", *Multimedia Tools, Applications*, vol. 82, pp. 3713–3744, 2023. DOI: 10.1007/s11042-022-13428-4.

[7] D. Hupkes *et al.*, "A taxonomy, review of generalization research in NLP", *Nature Machine Intelligence*, vol. 5, pp. 1161–1174, 2023. DOI: 10.1038/s42256-023-00729-y.

[8] P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope", *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023.

[9] Y. Yang and Z. Xue, "Training heterogeneous features in sequence to sequence tasks: Latent enhanced multi-filter Seq2Seq model", in *Intelligent Systems, Applications*, 2023, pp. 103–117.

[10] Y. Sun *et al.*, *ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation*, 2021. arXiv: 2107.02137 [cs.CL].

[11] D. Zmitrovich *et al.*, "A family of pretrained transformer language models for Russian", in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 507–524.

[12] M. Song and Y. Zhao, "Enhance RNNLMs with hierarchical multi-task learning for ASR", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 6102–6106.

[13] A. Vaswani *et al.*, "Attention is all you need", in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[14] V. Sanh *et al.*, *Multitask prompted training enables zero-shot task generalization*, 2022. arXiv: 2110.08207 [cs.LG].

[15] Y. Tay *et al.*, *UL2: Unifying language learning paradigms*, 2023. arXiv: 2205.05131 [cs.CL].

[16] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning", in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 41–48.

[17] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[18] Y. Sun *et al.*, "ERNIE 2.0: A continual pre-training framework for language understanding", in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 34, 2020, pp. 8968–8975.

[19] L. Xue *et al.*, "MT5: A massively multilingual pre-trained text-to-text transformer", in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41.

[20] S. Wang *et al.*, *ERNIE 3.0 Titan: Exploring larger-scale knowledge enhanced pre-training for language understanding and generation*, 2021. arXiv: 2112.12731 [cs.CL].

[21] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "AdapterFusion: Non-destructive task composition for transfer learning", in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 487–503. DOI: 10.18653/v1/2021.eacl-main.39.

[22] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity", *Journal of Machine Learning Research*, vol. 23, no. 120, pp. 1–39, 2022.

[23] S. Longpre *et al.*, "The flan collection: Designing data and methods for effective instruction tuning", in *Proceedings of the International Conference on Machine Learning*, PMLR, 2023, pp. 22 631–22 648.

[24] N. Houlsby *et al.*, "Parameter-efficient transfer learning for NLP", in *Proceedings of the International Conference on Machine Learning*, 2019, pp. 2790–2799.

[25] H. A. Al-Khamees, M. E. Manaa, Z. H. Obaid, and N. A. Mohammedali, "Implementing cyclical learning rates in deep learning models for data classification", in *Proceedings of the International Conference on Forthcoming Networks and Sustainability in the AIoT Era*, 2024, pp. 205–215.

[26] A. Koloskova, H. Hendrikx, and S. U. Stich, "Revisiting gradient clipping: Stochastic bias and tight convergence guarantees", in *Proceedings of the International Conference on Machine Learning*, 2023, pp. 17 343–17 363.