Автоматизация разметки текстов о жизненных трудностях с использованием языковых моделей

А. А. Хлебникова[1], Е. В. Битюцкая[2], Г. В. Калачев[3], Э. Э. Гасанов[4]

Статья посвящена решению проблемы высокой трудоёмкости «ручного» кодирования качественных данных в психологических исследованиях, использующих контент-анализ. Оценивается эффективность методов автоматизированной разметки текстов с применением современных языковых моделей DeepSeek, GPT-4.1 и GPT-4.1-mini и разрабатываются пути повышения точности разметки. Материалом являются описания трудных жизненных ситуаций участников психологического исследования. Исследование подтверждает практическую целесообразность использования языковых моделей в качестве инструмента, значительно сокращающего временны́е затраты исследователя на первичный анализ текстовых данных.

Ключевые слова: контент-анализ, большая языковая модель, GPT-4.1, DeepSeek, трудная жизненная ситуация, копинг (совладание), восприятие ситуации.

[1] *Хлебникова Алёна Андреевна* — аспирант каф. общей психологии ф. психологии, МГУ имени М.В. Ломоносова, e-mail: alena.epochta@gmail.com.

Khlebnikova Alena Andreevna — PhD Student, Department of General Psychology, Faculty of Psychology, Lomonosov Moscow State University.

[2] *Битюцкая Екатерина Владиславовна* — канд. психол. наук, доцент кафедры общей психологии ф. психологии, МГУ имени М.В. Ломоносова, e-mail: bityutskaya_ew@mail.ru.

Bityutskaya Ekaterina Vladislavovna — Candidate of Psychological Sciences, Associate Professor, Department of General Psychology, Faculty of Psychology, Lomonosov Moscow State University.

[3] *Калачев Глеб Вячеславович* — канд. физ.-мат. наук, научный сотрудник кафедры математической теории интеллектуальных систем механико-математического ф. МГУ имени М.В. Ломоносова, e-mail: gleb.kalachev@yandex.ru.

Kalachev Gleb Viacheslavovich — Candidate of Physical and Mathematical Sciences, Researcher, Department of Mathematical Theory of Intellectual Systems, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University.

[4] *Гасанов Эльяр Эльдарович* — д-р физ.-мат. наук, заведующий кафедрой математической теории интеллектуальных систем механико-математического ф. МГУ имени М.В. Ломоносова, e-mail: el_gasanov@mail.ru.

Gasanov Elyar Eldarovich — Doctor of Physical and Mathematical Sciences, Professor, Head of the Department, Department of Mathematical Theory of Intelligent Systems, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University.

Введение

Применение контент-анализа в исследованиях копинга открывает возможности для глубокого анализа феноменологии и смыслов переживания людей. Однако ключевым ограничением метода является высокая трудоёмкость обработки качественных данных. При больших объёмах текстов этап кодирования (разметки) данных часто занимает наибольшее время по сравнению с другими его этапами (сбор данных, разработка кодировочной инструкции, интерпретация результатов). Процесс кодирования «является трудоёмким, дорогим, медленным» [3]. Поэтому с момента начала использования метода в 1960-х годах контент-анализ был тесно связан с разработкой компьютерных технологий для автоматизации кодирования [2].

Компьютерные средства, используемые для контент-анализа в современных исследованиях, можно разделить на программы для автоматической разметки (например, «LIWC», «Leximancer», «DICTION» и др.), программы для ручного и полуавтоматического кодирования (например, «NVivo», «MAXQDA», «ATLAS.ti» и др.) [6] и гибридные инструменты («WordStat», «Quanteda» и др.), сочетающие возможности автоматического кодирования и ручной обработки. Выбор инструмента зависит от объёма данных, типа контент-анализа, необходимости визуализации результатов. В последние годы нейросетевые методы (BERT, GPT и др.) активно применяются для автоматизации разметки [5] и в ряде задач показывают более высокую точность по сравнению с традиционными подходами («LIWC» и др.) [3, 7].

В наших исследованиях копинга и восприятия трудностей разметка используется для контент-анализа описаний трудных жизненных ситуаций, которые получены с помощью открытых вопросов «Методики структурированного описания ситуации» [1, 4]. Вопросы методики и пример случая представлены в Приложении А. Для кодирования описаний привлекаются независимые кодировщики и эксперты. Каждый случай, который используется в исследовании, закодирован двумя кодировщиками и одним экспертом. При этом фрагменты текста, которые закодированы по-разному или вызывают неоднозначные интерпретации, обсуждаются с достижением консенсуса. Кодировочная инструкция разработана совместно Е.В. Битюцкой и Н.Г. Малышевой и включает 187 единиц анализа — категорий и подкатегорий двух видов: относящихся к описанию ситуации в целом (1) и к отдельным вопросам (2). К категориям первого типа относятся эмоции, время, энергия, степень и суть трудности. Категориями второго типа выступают содержание ситуации, копинг, несколько категорий оценки, цели, возможности, ограничения и другие.

Разметка данных связана с выделением смысловых единиц текста описания ситуации, обозначением выделенного фрагмента квадратными скобками и проставлением кода подкатегории. Пример закодированного случая и перечень кодов, использованных в этом примере, представлены в Приложении Б. Размеченные тексты далее обрабатываются с помощью Python-приложения, что позволяет получить таблицу частот категорий и подкатегорий. Такой вариант разметки применялся для контент-анализа, проведённого на выборке 611 описаний трудных жизненных ситуаций (или случаев)[1]. В результате были описаны типы восприятия трудных жизненных ситуаций и проверены предположения о существовании различий между типами [1, 4].

В данной статье мы представляем результаты исследования, целью которого была оценка и повышение эффективности применения языковых моделей для автоматического кодирования текстов. Исследование включало пять последовательных этапов, каждый из которых решал конкретные задачи:

1) оценка базовой эффективности языковых моделей при разметке текстов;

2) оптимизация параметров взаимодействия с языковыми моделями;

3) дообучение GPT-4.1;

4) тестирование эффективности языковых моделей в условиях сокращённой кодировочной инструкции;

5) дообучение GPT-4.1 с применением сокращённой кодировочной инструкции.

1. Задача разметки и метрики качества

Для проведения экспериментов с языковыми моделями было выбрано 100 случаев описаний трудных жизненных ситуаций, которые образовали тестовую выборку. Пример одного из описаний приведён в Приложении А. Далее эти случаи были размечены кодировщиками и экспертом-психологом, и образовали эталонную размеченную выборку. Пример такой разметки приведён в Приложении Б. Тестовая выборка без разметки подавалась на языковую модель, и языковая модель осуществляла разметку

[1] В шестилетнем исследовании (выполнявшемся с ноября 2018 по октябрь 2024 г.) процесс кодирования данных (611 случаев), включая обучение независимых кодировщиков, занял четыре года.

по аналогии с разметкой эксперта-психолога. Полученная автоматическая разметка сравнивалась с эталонной. Сравнение осуществлялось с помощью следующих метрик:

- верные коды (%) — доля точных совпадений кодов и их позиций в тексте автоматической разметки с экспертной разметкой;
- смещённые коды (%) — доля верных кодов, поставленных в неверных фрагментах текста;
- пропущенные коды (%) — доля экспертных кодов, которые языковая модель не обнаружила;
- дубликаты кодов (%) — доля кодов, которые модель поставила избыточно, сверх экспертной разметки;
- лишние коды (%) — доля кодов, которые модель присвоила ошибочно (отсутствуют в экспертной разметке).

Определим более строго структуру документа, который мы размечаем, разметку и процедуру сравнения двух разметок.

Документ представляет собой набор из S *фрагментов*. S — это константа, в нашем случае равная 7 (формулирование трудной жизненной ситуации и ответы на 6 вопросов). Каждый фрагмент — это некоторый текст, который мы представляем как последовательность слов. Число слов в фрагменте f мы называем его *длиной* и обозначаем через $|f|$.

Основная задача состоит в разметке документа, что предполагает разделение каждого фрагмента на смысловые блоки и определение для каждого блока его смысла в терминах фиксированного множества категорий и подкатегорий, обозначаемых *кодами разметки*. Большинство кодов, которые фигурируют в разметке текстов, относятся к кодам подкатегорий. Код ставится в конце смыслового блока, к которому он относится. Во многих случаях встречаются смысловые блоки, закодированные с использованием двух или нескольких кодов. Множество всех кодов разметки обозначим через $\mathcal{L}$.

Роль языковой модели заключается в том, чтобы сделать первичную разметку, которую затем проверяет и корректирует эксперт. Соответственно, качество разметки определяет трудоёмкость проверки и исправления первичной разметки. Разного рода ошибки имеют разную сложность обнаружения и исправления, поэтому мы рассматриваем несколько категорий ошибок. Имея эталонную разметку документа, сделанную экспертом, и тестируемую разметку, сделанную языковой моделью, мы можем вычислить количество ошибок каждого типа.

Для определения необходимых видов ошибок в разметке нам понадобится формальное определение разметки документа. Данное выше

содержательное определение разметки можно формализовать следующим образом.

Разметкой фрагмента f мы называем произвольное множество $T \subseteq \mathbb{N} \times \mathcal{L}$ пар (p, ℓ), где $\ell \in \mathcal{L}$ интерпретируется как код разметки, $p \in \mathbb{N}$ интерпретируется как номер слова, после которого стоит данный код, $p \leq |f|$. *Разметка документа* $T^* = (T^1, \ldots, T^S)$ — набор разметок всех фрагментов этого документа.

Для разметки фрагмента T определим множество $L(T) = \{\ell \mid (p, \ell) \in T\}$, содержащее все метки, входящие в разметку T. Также для кода $\ell \in \mathcal{L}$ определим величину $N(T, \ell) = \big|\{p \in \mathbb{N} \mid (p, \ell) \in T\}\big|$ — количество вхождений кода ℓ в разметку T.

Для разметки документа T^* введём обозначение $N(T^*) = \sum_{i=1}^{S} |T^i|$ — общее число кодов в T^*.

Теперь мы можем определить наши основные метрики сходства тестируемой разметки фрагмента T с эталонной разметкой $\hat{T}$.

1) Количество *верных* кодов $v(\hat{T}, T) = |T \cap \hat{T}|$ — количество точных совпадений с тестируемой и эталонной разметках.

2) Количество *релевантных* кодов

$$r(\hat{T}, T) = \sum_{\ell \in L(\hat{T}) \cap L(T)} \min(N(T, \ell), N(\hat{T}, \ell)).$$

   Оно включает в себя все коды, которые присутствуют и в тестируемой, и в эталонной разметках, но, возможно, на разных позициях.

3) Количество *смещённых* кодов $s(\hat{T}, T) = r(\hat{T}, T) - v(\hat{T}, T)$ — коды, которые присутствуют в обеих разметках, но на разных позициях (не являются верными).

4) Количество *дубликатов*

$$d(\hat{T}, T) = \sum_{\ell \in L(T) \cap L(\hat{T})} \max\big(0, N(T, \ell) - N(\hat{T}, \ell)\big).$$

   Оно включает в себя коды, которые присутствовали в эталонной разметке, но в тестируемой разметке больше вхождений этих кодов.

5) Количество *лишних* кодов

$$e(\hat{T}, T) = \sum_{\ell \in L(T) \setminus L(\hat{T})} N(T, \ell).$$

   Оно включает в себя коды, которые присутствуют в тестируемой разметке, но их нет в эталонной.

6) Количество *пропущенных* кодов

$$m(\hat{T}, T) = \sum_{\ell \in L(\hat{T})} \max\bigl(0, N(\hat{T}, \ell) - N(T, \ell)\bigr).$$

Несложно убедиться, что для любых двух разметок фрагмента $\hat{T}$ и T выполнено

$$v(\hat{T}, T) + s(\hat{T}, T) + m(\hat{T}, T) = |\hat{T}|. \tag{1}$$

Для разметок документа можно определить все эти метрики как сумму соответствующих метрик для фрагментов, то есть метрика $h \in \{v, r, s, d, e, m\}$ для разметок документа определяется как

$$h(\hat{T}^*, T^*) = \sum_{i=1}^{S} h(\hat{T}^i, T^i).$$

Поскольку трудозатраты на исправление разметки мы сравниваем с трудозатратами разметки с нуля, то все посчитанные метрики мы нормируем на общее число кодов в эталонной разметке. В частности, при оценке качества разметки одного документа для метрики $h \in \{v, r, s, d, e, m\}$ мы можем определить нормированную метрику

$$\bar{h}(\hat{T}^*, T^*) = \frac{h(\hat{T}^*, T^*)}{N(\hat{T}^*)}.$$

Если у нас есть выборка $\mathcal{T} = (T_1, \ldots, T_n)$ разметок n документов и набор $\hat{\mathcal{T}} = (\hat{T}_1, \ldots, \hat{T}_n)$ соответствующих эталонных разметок, то мы определяем точно так же, как и для одного документа

$$h(\hat{\mathcal{T}}, \mathcal{T}) = \sum_{i=1}^{n} h(\hat{T}_i, T_i), \qquad \bar{h}(\hat{\mathcal{T}}, \mathcal{T}) = \frac{h(\hat{\mathcal{T}}, \mathcal{T})}{\sum_{i=1}^{n} N(\hat{T}_i)}.$$

В сводных таблицах и при обсуждении экспериментов приводятся именно нормированные метрики качества для всей тестовой выборки.

Учитывая (1), всегда выполнено равенство

$$\bar{v}(\hat{\mathcal{T}}, \mathcal{T}) + \bar{s}(\hat{\mathcal{T}}, \mathcal{T}) + \bar{m}(\hat{\mathcal{T}}, \mathcal{T}) = 1,$$

То есть в сумме нормированные метрики верных, смещённых и пропущенных кодов дают 100%.

Под *суммарным охватом релевантной разметки* в данной статье понимается доля релевантных кодов (сумма долей верных и смещённых кодов), то есть величина $(\bar{v}(\hat{\mathcal{T}}, \mathcal{T}) + \bar{s}(\hat{\mathcal{T}}, \mathcal{T})) \cdot 100\%$. Это связано с тем, что незначительные смещения могут возникать также при сравнении кодирования двух кодировщиков и не являются ошибкой. Трудозатраты эксперта на проверку верных кодов и смещённых кодов значительно ниже, чем на удаление лишних кодов и, в особенности, внесение пропущенных кодов.

2. Программа исследования

Модели. В исследовании были использованы следующие языковые модели:

1) DeepSeek — модель от DeepSeek AI, которая использовалась через веб-интерфейс DeepSeek Chat (https://chat.deepseek.com/). Особенности взаимодействия с моделью:

   - интерфейс чата не позволяет задавать параметры генерации, такие как «температура». Поэтому все эксперименты с данной моделью проводились при стандартных настройках.
   - интерфейс позволяет переключать обычный режим и «режим повышенной точности» DeepThink.

2) GPT-4.1 — модель от OpenAI версии GPT-4.1-2025-04-14, которая использовалась через OpenAI API. API позволяет задавать параметры генерации, в том числе температуру.

3) GPT-4.1-mini — облегчённый вариант GPT-4.1 версии GPT-4.1-mini-2025-04-14, которая также использовалась через OpenAI API.

Коды разметки. Для кодирования текстов использовалось 2 множества кодов:

1) полная кодировочная инструкция I, включающая множество $\mathcal{L}$ из 187 кодов — применялась на этапах 1, 2 и 3;

2) сокращённая кодировочная инструкция I', включающая подмножество $\mathcal{L}' \subseteq \mathcal{L}$ из 54 наиболее релевантных кодов — использовалась на этапах 4 и 5.

Кроме того использовался вспомогательный код «Другое», который здесь для краткости обозначим Λ.

Для удобства определим операцию *стандартной проекции* π, которая удаляет из разметки все коды, не входящие в $\mathcal{L}'$, а также операцию *расширенной проекции* $\tilde{\pi}$, которая заменяет в разметке все коды, не входящие в $\mathcal{L}'$ на Λ. Операции π и $\tilde{\pi}$ определены естественным образом на самих кодах, разметках фрагментов, разметках документов и обучающих/тестовых выборках. Заметим, что для любой разметки T выполнено

$$\pi(\pi(T)) = \pi(T), \qquad \tilde{\pi}(\tilde{\pi}(T)) = \tilde{\pi}(T), \qquad \pi(\tilde{\pi}(T)) = \pi(T).$$

Данные. Материалом исследования послужили описания 210 трудных жизненных ситуаций, полученные с помощью открытых вопросов «Методики структурированного описания ситуации» [1, 4] и размеченные экспертами-психологами.

Таким образом, для каждого случая имеется пара текстов: исходный текст и текст с эталонной разметкой — всего 210 пар текстов. Эти 210 пар текстов разделены на три части: S — обучающая выборка из 100 пар текстов, V — тестовая выборка из 100 пар текстов, а также выборка S_0 из 10 пар текстов, которая использовалась для включения в промпт примеров разметки[2] на этапах 1 и 2.

Также на этапах 4 и 5 использовались множества S', S'_0 и V', которые получены из S, S_0 и V применением расширенной проекции $\tilde{\pi}$, то есть заменой в текстах всех кодов, не входящих в сокращённую кодировочную инструкцию I', на код Λ.

Системное сообщение. В ходе экспериментов использовалось 3 различных варианта системных сообщений:

1) P_1 — включает полную кодировочную инструкцию I;
2) P_{23} — включает полную кодировочную инструкцию I и уточнённые правила разметки;
3) P_{45} — включает сокращённую кодировочную инструкцию I' и уточнённые правила разметки, а также правила использования кода Λ для маркировки смысловых фрагментов, нерелевантных 54 отобранным кодам.

Эксперименты. В каждом эксперименте обучающие примеры либо добавлялись к системному сообщению, либо использовались для дообучения модели GPT-4.1 через OpenAI API со значениями гиперпараметров по умолчанию.

Тестирование эффективности моделей проводилось на тестовой выборке следующим образом: для каждой пары $(s, \hat{t})$ из тестовой выборки модель получала на вход системное сообщение и исходный текст s и размечала его, добавляя в него коды разметки, генерируя на выходе некоторый текст t. При этом даже при строгом запрете на изменение исходного текста иногда модели меняли его, и текст t при обратном удалении разметки не совпадал с исходным текстом s.

В случае, если фрагмент в тексте t после удаления кодов не совпадал с соответствующим фрагментом в тексте s, то каждое такое изменение

[2] В промпт включался только второй элемент каждой пары — текст с эталонной разметкой.

фрагмента помечалось как *изменение исходного текста*. В этом случае применялась библиотека `difflib` из стандартной библиотеки Python для соотнесения позиций в изменённом тексте с позициями в исходном тексте. Далее, используя полученные позиции, формировалась разметка T^*, описанная в разделе 1. В случае, если текст на выходе модели после удаления кодов совпадал с исходным, то разметка T^* формировалась напрямую из позиций кодов в фрагментах текста t.

Размеченный экспертом текст $\hat{t}$ также сравнивался с s и вычислялись позиции всех кодов, формируя эталонную разметку $\hat{T}^*$. После этого вычислялись метрики качества разметки, описанные в разделе 1. Для каждой из метрик считался 95% доверительный интервал методом bootstrap с 2000 перезапусками, и в таблицах эта информация представлена в виде $N \pm \Delta/2$, где N — значение метрики, Δ — длина доверительного интервала. Следует отметить, что доверительные интервалы на самом деле расположены немного несимметрично относительно среднего значения, но сдвинуты незначительно (отношение величины сдвига к длине интервала не выше 0.1).

На этапах 4 и 5 для оценки качества тестируемой разметки $\mathcal{T}$ с помощью сокращённой схемы кодирования с эталонной разметкой $\hat{\mathcal{T}}$ использовались 2 режима вычисления метрик из раздела 1:

1) Сравнение расширенной проекции тестируемой разметки $\tilde{\pi}(\mathcal{T})$ с расширенной проекцией эталонной разметки $\tilde{\pi}(\hat{\mathcal{T}})$. Такой способ включает в сравнение те фрагменты, которым не соответствует ни один код из $\mathcal{L}'$, и которые в данном случае помечаются кодом Λ.

2) Сравнение стандартной проекции тестируемой разметки $\pi(\mathcal{T})$ с проекцией $\pi(\hat{\mathcal{T}})$ эталонной разметки. Такой способ отражает чистое сравнение на подмножестве кодов $\mathcal{L}'$.

Эти методы применялись как для оценки результатов этапов 4 и 5 — там, где разметка сразу производилась множеством кодов $\mathcal{L}' \cup \{\Lambda\}$, так и для разметок, полученных на этапах 2 и 3.

В таблице 1 дано описание этапов экспериментов, в котором указаны модели, системные сообщения, обучающие и тестовые выборки, использованные на каждом этапе.

Далее представлено описание задач и полученных результатов на каждом этапе исследования.

Таблица 1. Этапы экспериментов

Этап	Модели	Промпт	Обучение	Тест
1	DeepSeek, GPT-4.1-mini($t = 0.3$), GPT-4.1($t = 0.3$)	$P_1 + S_0$	нет	V
2	DeepSeek, DeepSeek(DeepThink), GPT-4.1($t = 0$)	$P_{23} + S_0$	нет	V
3	GPT-4.1($t = 0$)	P_{23}	S (3 эпохи)	V
4	DeepSeek, GPT-4.1($t = 0$)	$P_{45} + S'_0$	нет	V'
5	GPT-4.1($t = 0$)	P_{45}	S' (3 эпохи)	V'

3. Результаты и их анализ

Этап 1. Оценка базовой эффективности языковых моделей

Основной задачей первого этапа исследования стала оценка базовой эффективности языковых моделей DeepSeek, GPT-4.1-mini и GPT-4.1. На этом этапе был создан единый промпт с полной кодировочной инструкцией (187 кодов), правилами разметки и примерами размеченных кодировщиками и экспертом десяти текстов-описаний трудных жизненных ситуаций. Эти десять случаев не входили в тестовую выборку.

Результаты первого этапа представлены в таблице 2. Языковая модель DeepSeek продемонстрировала точность 34% верных кодов, однако пропустила 57% экспертных кодов. Суммарный охват релевантной разметки составил 43%. Языковая модель показала умеренный уровень избыточного кодирования (4% дубликатов кодов) и допустила 26% ошибочных присвоений кодов смысловым единицам текста.

GPT-4.1 показала схожий результат по основным метрикам (32% верных кодов, 57% пропусков), но продемонстрировала более высокую склонность к генерации ошибочных кодов (42% лишних кодов) при меньшем уровне дублирования (2%). Общий охват релевантной разметки (43%) подтверждает конкурентоспособность данной модели в задачах автоматического кодирования.

Наименее эффективной оказалась модель GPT-4.1-mini с показателем 14% верных кодов и 76% пропусков. При полном отсутствии дубликатов кодов модель допустила 22% ошибочных кодов, а суммарный охват релевантной разметки составил лишь 24%. Ключевым недостатком данной языковой модели стало неприемлемое количество изменений исходного

Таблица 2. Результаты первого этапа

Тип кодов, %	DeepSeek	GPT-4.1	GPT-4.1-mini
Верные	34.4 ± 2.8	32.3 ± 3.1	13.6 ± 1.9
Смещённые	8.5 ± 1.2	10.8 ± 1.6	10.6 ± 1.3
Пропущенные	57.1 ± 2.2	56.9 ± 3.1	75.8 ± 2.1
Дубликаты	3.6 ± 1.4	1.6 ± 0.6	0.0 ± 0.0
Лишние	25.6 ± 3.1	42.5 ± 4.8	21.7 ± 2.3

Примечание: здесь и далее в таблицах представлены доли всех типов кодов относительно экспертной разметки и 95% доверительный интервал.

текста (328 правок с сильным искажением исходного текста), что полностью исключает её использование в рамках задач автоматического кодирования. У остальных моделей количество внесённых изменений было незначительным (в среднем 4 правки с незначительным изменением исходного текста).

Полученные результаты позволили отобрать две наиболее перспективные модели — DeepSeek и GPT-4.1 — для дальнейшей оптимизации.

Этап 2. Оптимизация параметров взаимодействия с языковыми моделями

Основной задачей второго этапа исследования стала целенаправленная оптимизация параметров взаимодействия с языковыми моделями DeepSeek и GPT-4.1, продемонстрировавшими лучшие результаты на предыдущем этапе. По аналогии с первым этапом в промпт были включены десять случаев эталонной разметки. Меры оптимизации включали уточнение формулировки промпта, правил разметки и настройку параметра генерации «температура» ($t = 0$) для модели GPT-4.1. Снижение значения данного параметра до нуля было призвано минимизировать стохастичность выходных данных модели и максимизировать детерминированность её ответов. Параллельно проводилось тестирование режима повышенной точности DeepThink модели DeepSeek для сравнения эффективности различных подходов.

Результаты демонстрируют более высокие показатели эффективности языковых моделей по ключевым метрикам (таблица 3). Модель GPT-4.1 показала наиболее значительный прогресс: доля верных кодов возросла с 32% до 46%, а уровень пропусков снизился с 57% до 46%. Суммарный охват релевантной разметки достиг 54%. При этом отмечается увеличение доли лишних кодов с 42% до 63%, что свидетельствует о возросшей ак-

тивности языковой модели в присвоении кодов, часть которых не соответствует экспертной разметке. Показатель дублирования кодов увеличился с 2% до 8%, указывая на тенденцию к избыточному кодированию.

Языковая модель DeepSeek также показала улучшение результатов: точность верных кодов повысилась с 34% до 36%, а доля пропусков незначительно снизилась (с 57% до 56%). Суммарный охват релевантной разметки составил 45%. Количество лишних кодов осталось на сопоставимом уровне (27% против 26% на первом этапе), в то время как показатель дублирования увеличился с 4% до 5%.

Активация режима повышенной точности DeepThink языковой модели DeepSeek существенно не изменила показатели точности. Суммарный охват релевантной разметки составил 45% (37% верных кодов) при 55% пропусков. Модель продемонстрировала аналогичный уровень дублирования (5%) и незначительное снижение доли лишних кодов (26%).

Сравнительный анализ результатов двух языковых моделей выявляет различные стратегии поведения после оптимизации: GPT-4.1 продемонстрировала тенденцию к экспансии кодирования, характеризующуюся значительным увеличением количества присваиваемых кодов, что выразилось в возрастании числа как корректных, так и ошибочных кодов. В противоположность этому, модель DeepSeek сохранила консервативную стратегию кодирования, проявляющуюся в минимальном приросте ошибочных идентификаций кодов.

Такое различие в стратегиях может объясняться свойствами моделей и разной чувствительностью к оптимизации их параметров. Экспансивное поведение GPT-4.1 свидетельствует о её большей восприимчивости к изменениям промпта и параметров генерации, в то время как консервативность DeepSeek может указывать на бóльшую устойчивость к изменениям.

Полученные результаты подтверждают эффективность применяемых процедур оптимизации и обосновывают целесообразность их использования для повышения эффективности автоматизированного кодирования текстовых данных.

Этап 3. Дообучение модели GPT-4.1

Третий этап исследования был посвящён реализации более сложного подхода к автоматизации — проведению процедуры дообучения (fine-tuning) языковой модели GPT-4.1. Данный метод подразумевает адаптацию внутренних весов модели под специфику конкретной задачи. Для обучения использовался набор данных, состоящий из 100 новых эталонных примеров экспертной разметки, преобразованных в 100 текстовых пар формата

«исходный текст – текст с разметкой». Результаты представлены в таблице 3.

Языковая модель GPT-4.1 продемонстрировала 49% верных кодов при значительном снижении доли пропусков до 42%. Суммарный охват релевантной разметки составил 59%. Показатель дублирования кодов остался на умеренном уровне (8%), а количество лишних кодов составило 48%. Эти результаты свидетельствуют о высокой эффективности подхода с дообучением для задач автоматизированного кодирования.

Таблица 3. Результаты второго и третьего этапов

	Этап 2			Этап 3
Тип кодов, %	DeepSeek	DeepSeek (DeepThink)	GPT-4.1	GPT-4.1
Верные	36.2 ± 2.9	36.6 ± 2.6	46.1 ± 2.7	48.5 ± 2.7
Смещённые	8.3 ± 1.2	8.0 ± 1.3	8.3 ± 1.2	10.0 ± 1.5
Пропущенные	55.5 ± 2.5	55.4 ± 2.4	45.7 ± 2.4	41.5 ± 2.7
Дубликаты	4.6 ± 1.4	4.9 ± 1.4	8.3 ± 2.0	7.5 ± 1.7
Лишние	27.0 ± 3.2	26.4 ± 3.0	63.0 ± 6.5	47.7 ± 3.7

Этап 4. Применение сокращённой схемы кодирования

Четвёртый этап исследования был направлен на снижение сложности кодировочной инструкции, что, согласно нашему предположению, должно было повысить точность автоматизированной разметки. При этом мы исходили из результатов предыдущих исследований [4], которые показали, что для решения задачи классификации значимы лишь определённые коды, частота встречаемости которых в разметке позволяет выделять типы восприятия жизненных трудностей. Изначально мы предположили, что если в промпте останутся наиболее значимые для классификации коды (а коды, которые не используются для разделения на типы, применяться не будут), то это упрощение создаст условия для улучшения точности разметки.

В рамках данного этапа была выполнена сравнительная оценка эффективности моделей DeepSeek и GPT-4.1 с использованием промпта, адаптированного под сокращённую схему кодирования, включающую 54 наиболее релевантных кода. Подготовка обучающего материала состояла в том, что из исходного материала удалялись коды, которые не входили в множество выделенных 54 кодов. Вместо удалённых кодов вносился специальный код «Другое». Аналогичная замена кодов, не входящих в сокращённую схему, на код «Другое» была произведена и в инструкции для промпта. Если говорить формально, то к обучающему материалу

и к инструкции для промпта была применена операция расширенной проекции, которая описана в разделе 2.

Таблица 4. Результаты четвёртого и пятого этапов с учётом кода «Другое»

Тип кодов, %	Этап 4		Этап 5
	DeepSeek	GPT-4.1	GPT-4.1
Верные	45.6 ± 3.0	41.3 ± 3.0	49.2 ± 2.7
Смещённые	6.6 ± 1.3	8.3 ± 1.8	11.3 ± 1.7
Пропущенные	47.8 ± 2.8	50.4 ± 3.1	39.5 ± 2.4
Дубликаты	12.7 ± 2.9	9.6 ± 2.5	14.4 ± 3.3
Лишние	21.5 ± 2.9	64.1 ± 6.9	31.6 ± 3.5

Результаты четвёртого этапа, когда сравнение ведётся с учётом кода «Другое», представлены в таблице 4. Модель DeepSeek показала точность 46% верных кодов, 7% смещенных кодов при 48% пропусков. Количество лишних кодов снизилось до 22%, а показатель дублирования составил 13%. Суммарный охват релевантной разметки достиг 52%. Тем самым при сокращённой схеме DeepSeek показывает лучшие результаты по сравнению с результатами этапа 2.

Модель GPT-4.1 при сокращённой схеме демонстрирует ухудшение результатов по сравнению с этапом 2, а именно 41% верных кодов, 8% смещенных кодов при 50% пропусков. Количество лишних кодов составило 64%, а показатель дублирования — 10%. Суммарный охват релевантной разметки составил 50%.

Отметим, что доля кода «Другое» в экспертной разметке оказалась равна 46% от всех кодов, внесённых в тексты. При таком условии большое влияние на результаты может оказывать эффект угадывания кода «Другое». Чтобы устранить влияние этого кода на точность разметки, мы выполнили оценку точности разметки без учёта кода «Другое». Если говорить более формально, к результатам этапа 4 была применена операция стандартной проекции. Результаты этого сравнения приведены в таблице 5.

Заметим, что полученные результаты неоднозначны. Для модели DeepSeek мы ожидаемо видим уменьшение доли верных кодов до 28% (показатель релевантной разметки 34%). При этом возросла доля пропущенных кодов до 66%. Для модели GPT-4.1 мы неожиданно наблюдаем увеличение доли верных кодов до 47%, а доли смещённых кодов до 9% при уменьшении доли пропущенных кодов до 44%. Можно заметить, что эти результаты лучше, чем результаты этапа 2, но при этом доля лишних кодов становится слишком высокой — 112%. Можно предположить, что

Таблица 5. Результаты четвёртого и пятого этапов без учёта кода «Другое»

Тип кодов, %	Этап 4		Этап 5
	DeepSeek	GPT-4.1	GPT-4.1
Верные	28.1 ± 3.6	47.0 ± 3.7	30.3 ± 3.6
Смещённые	5.7 ± 1.5	9.1 ± 2.1	9.4 ± 2.1
Пропущенные	66.2 ± 3.5	44.4 ± 3.7	60.3 ± 3.2
Дубликаты	6.5 ± 3.0	11.5 ± 3.7	3.3 ± 2.0
Лишние	24.6 ± 4.9	112.3 ± 16.2	35.5 ± 5.5

языковая модель DeepSeek лучше приспособилась к угадыванию кода «Другое», тогда как модель GPT-4.1 показывает лучшие результаты при разметке релевантных кодов, но при этом избыточно добавляет лишние коды.

Чтобы понять, насколько на обучаемость моделей влияет именно сокращение числа кодов, были выполнены следующие действия. К результатам этапа 2 была применена операция расширенной и стандартной проекции. Т.е. в разметке, полученной на этапе 2, и в экспертной разметке все коды, не входящие в сокращённую схему, были заменены на код «Другое» в случае расширенной проекции, и удалены коды, не входящие в сокращённую схему — в случае стандартной проекции. Результаты сравнения расширенной и стандартной проекций представлены в таблицах 6 и 7.

Таблица 6. Сравнение расширенных проекций второго и третьего этапов на сокращённое множество кодов

Тип кодов, %	Разметка этапа 2		Разметка этапа 3
	DeepSeek	GPT-4.1	GPT-4.1
Верные	48.6 ± 3.7	52.6 ± 3.0	60.6 ± 3.1
Смещённые	8.3 ± 1.5	7.7 ± 1.6	8.6 ± 1.7
Пропущенные	43.1 ± 3.0	39.7 ± 2.7	30.8 ± 2.8
Дубликаты	10.8 ± 2.7	12.0 ± 2.8	17.5 ± 2.7
Лишние	22.5 ± 2.8	37.9 ± 4.0	29.5 ± 3.2

Анализируя полученные результаты, мы можем видеть, что как для модели DeepSeek, так и для модели GPT-4.1 результаты расширенной проекции этапа 2 лучше, чем результаты этапа 4. В то же время для

Таблица 7. Сравнение стандартных проекций второго и третьего этапов на сокращённое множество кодов

Тип кодов, %	Разметка этапа 2		Разметка этапа 3
	DeepSeek	GPT-4.1	GPT-4.1
Верные	33.6 ± 4.1	42.8 ± 3.5	43.7 ± 3.7
Смещённые	8.4 ± 2.1	7.2 ± 1.6	8.8 ± 2.1
Пропущенные	58.0 ± 3.4	50.0 ± 3.5	47.5 ± 3.8
Дубликаты	5.1 ± 2.7	9.2 ± 3.8	6.9 ± 3.0
Лишние	24.4 ± 4.7	56.3 ± 8.7	35.7 ± 4.7

стандартной проекции ситуация противоречивая: сокращённая схема даёт для модели GPT-4.1 преимущество, а для модели DeepSeek — ухудшение результатов. Таким образом, сокращённая схема не улучшает показатели эффективности разметки языковых моделей: они с примерно одинаковыми показателями могут обрабатывать как полное, так и сокращённое множество кодов.

Этап 5. Дообучение модели GPT-4.1 с применением сокращённой схемы кодирования

Пятый этап исследования был направлен на оценку влияния процедуры дообучения (fine-tuning) языковой модели GPT-4.1 на точность автоматизированного кодирования в условиях применения сокращённой схемы кодирования. Этап предусматривал оценку эффективности дообученной версии модели GPT-4.1 с использованием оптимизированного промпта, адаптированного под сокращённую схему кодирования (54 кода). Для обучения использовался тот же, что и на этапе 4, набор из 100 текстовых пар формата «исходный текст — текст с разметкой», в разметке которых были оставлены только коды из сокращённой схемы, а коды, не входящие в сокращённую схему, были заменены на код «Другое». Как и на предыдущем этапе, инструкция для промпта включала специализированный код «Другое» (для маркировки смысловых фрагментов, нерелевантных 54 отобранным кодам). Мы предположили, что явное задание правил обработки релевантных и нерелевантных смысловых элементов текста в сочетании с предварительным дообучением модели будет способствовать повышению эффективности разметки.

Результаты пятого этапа представлены в таблицах 4 и 5.

Применение дообученной модели GPT-4.1 с промптом, включающим код «Другое», позволило достичь высоких результатов. Модель показала

точность 49% верных кодов при доле пропусков 40%. Суммарный охват релевантной разметки составил 61%. Количество лишних кодов осталось на умеренном уровне (32%), а показатель дублирования увеличился до 14%, что свидетельствует о возросшей активности модели при сохранении высокой точности. Наблюдаемый рост доли смещённых кодов (11%) указывает на незначительные ошибки в определении границ смысловых единиц, которые, однако, не снижают общей эффективности. В то же время, если сравнивать результаты этапа 3 (показатель релевантной разметки 59%) и этапа 5 без учета кода «Другое» (40%), то можно наблюдать ухудшение качества получаемой разметки. Исходя из этого, можно сделать вывод, что модель GPT-4.1 научается правильно определять код «Другое», но содержательные коды различает хуже.

Сравнение результатов дообучения по сокращённой схеме (54 кода) с проекциями дообученной полной схемы (см. таблицы 6 и 7) показывает, что и стандартная, и расширенная проекции результатов этапа 3 дают лучшую разметку, чем дообученная модель этапа 5. Это позволяет сделать вывод о том, что использование сокращённой схемы не даёт улучшения точности разметки по сравнению с полной схемой при использовании языковых моделей.

4. Заключение

Проведённое исследование демонстрирует принципиальную возможность и практическую эффективность использования современных языковых моделей GPT-4.1 и DeepSeek для автоматизации трудоёмкого процесса кодирования текстовых данных в психологических исследованиях. На основе сравнительного анализа пяти последовательных этапов работы с моделями DeepSeek и GPT-4.1 были получены следующие выводы.

В задачах автоматического кодирования языковые модели демонстрируют различную эффективность. GPT-4.1 легче поддается оптимизации и дообучению, в то время как DeepSeek проявила стратегию кодирования с меньшим количеством ошибок, но и с меньшей полнотой охвата релевантной разметки (45%).

Оптимизация промптов и параметров генерации существенно улучшает качество разметки (GPT-4.1). Снижение параметра «температура» до нуля и уточнение формулировок правил кодирования позволили повысить точность релевантной разметки для GPT-4.1 с 43% до 54% на втором этапе исследования.

Дообучение на релевантных данных является эффективным методом повышения точности разметки. Процедура fine-tuning модели GPT-4.1 на 100 размеченных экспертами случаях позволила достичь наиболее высоких показателей на третьем этапе: 59% релевантных кодов.

Использование сокращённой схемы кодирования (54 кода) в целом не даёт преимуществ перед использованием полной схемы (187 кодов) для автоматизированной разметки с применением языковых моделей.

Полученные результаты обосновывают целесообразность использования языковых моделей в качестве инструмента, предназначенного для первичного анализа больших массивов текстовых данных, что значительно сокращает временны́е затраты исследователя. Говоря о первичном анализе, мы утверждаем, что важным условием применения языковых моделей для разметки данных является дальнейшая работа с текстами обученных кодировщиков и экспертов.

Перспективы исследования. Неожиданным итогом данного исследования являются схожие результаты эффективности разметки при использовании полной и сокращенной схемы кодирования (187 кодов и 54 кода). На первый взгляд, сокращение множества кодов должно было улучшить показатели языковой модели. Однако этот исследовательский ход в целом не дал преимуществ качества автоматизированной разметки. Одно из возможных объяснений такого эффекта связано с семантической или смысловой полнотой 187-элементного множества кодов. То есть данный набор кодов обеспечивает хорошую степень смыслового охвата (или «покрытия» текста кодами и стоящими за ними смысловыми единицами), тогда как результат кодирования на основе сокращенного множества кодов зияет смысловыми пробелами. Если принять гипотезу о семантической полноте, то в последующих исследованиях языковые модели можно использовать как инструмент измерения семантической полноты множеств кодов для контент-анализа текстов.

Предложенный подход может быть адаптирован для различных задач контент-анализа в психологии и смежных дисциплинах.

5. Благодарности

Благодарим доцента факультета психологии МГУ имени М.В. Ломоносова Н.Г. Малышеву за сотрудничество при разработке кодировочной инструкции для контент-анализа и аспирантку этого факультета А.Г. Докучаеву за помощь в обработке данных и экспертную работу. Выражаем признательность рецензентам статьи за ценные рекомендации.

Финансирование. Исследование выполнено за счет гранта Российского научного фонда № 25-18-00737, https://rscf.ru/project/25-18-00737/

Список литературы

[1] Е. В. Битюцкая, М. И. Кунашенко, “Стремление к трудности как тип восприятия жизненных ситуаций”, *Вестник Московского университета. Серия 14: Психология*, **47**:1 (2024), 56–87.

[2] Н. Н. Богомолова, Н. Г. Малышева, Т. Г. Стефаненко, “Контент-анализ”, *Социальная психология: практикум: учеб. пособие для студентов вузов*, ред. Т. В. Феломеева, Аспект Пресс, М., 2009, 131–162.

[3] J. Biggiogera, G. Boateng, P. Hilpert *et al.*, *BERT meets LIWC: Exploring State-of-the-Art Language Models for Predicting Communication Behavior in Couples' Conflict Interactions*, 2021, arXiv: 2106.01536.

[4] E. V. Bityutskaya, E. E. Gasanov, K. V. Khazova, N. A. Patrashkin, “Classifying the Perception of Difficult Life Tasks: Machine Learning and/or Modeling of Logical Processes”, *Psychology in Russia: State of the Art*, **17**:2 (2024), 64–84.

[5] T. B. Brown, B. Mann, N. Ryder, N. Subbiah *et al.*, *Language Models are Few-Shot Learners*, 2020, arXiv: 2005.14165.

[6] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 4th ed., Sage Publications, Inc., Thousand Oaks, CA, 2019.

[7] L. Tavabi, T. Tran, K. Stefanov *et al.*, “Analysis of Behavior Classification in Motivational Interviewing”, Proceedings of the Conference on Computational Linguistics and Clinical Psychology (CLPsych), 2021, 110–115.

Приложение А. Пример исходного описания трудной жизненной ситуации

Пример случая (мужчина, 38 лет). Курсивом даны инструкция и вопросы Методики структурированного описания ситуации.

Сформулируйте свою жизненную ситуацию, которая является трудной задачей, требующей решения в данный период времени.
Трудная финансовая ситуация. Высокая цена съёмной квартиры, больше половины зарплаты уходит на оплату аренды, плюс оплата кредита, плюс задолженность на кредитной карте. Естественно, ситуация накладывает отпечаток и на состояние.

1. Как Вы её воспринимаете, оцениваете, переживаете и преодолеваете? (Какие действия помогают вам преодолеть ситуацию или свое

состояние).
Присутствует недовольство, небольшой элемент угнетённости, невозможность в данный момент позволить себе то, что хочется, создает удручающие ощущения. Это произошло из-за нерационального распределения денег, желания позволить себе и своей семье больше. Небольшие накопления, практически все, ушли на то, чтобы погасить задолженность по кредитной карте. Помочь решить эту ситуацию может четко выверенный план. Составлен план по выходу из кризисной ситуации.

2. Каковы Ваши цели в этой ситуации?
Переезд в квартиру вдвое дешевле, полный подсчет необходимых расходов на жизнь, постоянные накопления для создания неприкасаемого запаса на форс-мажорные случаи.

3. Какие возможности и ограничения есть у Вас при достижении цели?
Возможности – зарплата. Ограничения – зарплата. Когда у меня получится найти дополнительный доход, будет проще преодолевать ситуацию.
4. Нужна ли Вам в этой ситуации помощь (поддержка) окружающих людей?
Да. Прежде всего моей семьи, моей жены. Вместе мы составили план и собираемся ему следовать. А если у неё получится найти работу, ту, которая ей понравится и будет приносить доход, будет еще лучше.

5. Если всё сложится очень плохо, то что это будет? (Максимальный неуспех).
Очередной уход в долговую яму, потеря единомыслия со своей супругой.

6. Опишите, что для Вас будет максимально успешным выходом, разрешением ситуации?
В данный момент – жизнь по средствам, выход из финансового кризиса, увеличение совокупного дохода семьи, накопление средств.

Приложение Б. Пример эталонной разметки, выполненной экспертом-психологом

Пример случая (мужчина, 38 лет). Курсивом даны инструкция и вопросы Методики структурированного описания ситуации. Коды разметки обозначены звёздочками.

Сформулируйте свою жизненную ситуацию, которая является трудной задачей, требующей решения в данный период времени.
[Трудная финансовая ситуация. *Ф1.1] [Высокая цена съёмной квартиры, больше половины зарплаты уходит на оплату аренды, плюс оплата кредита, плюс задолженность на кредитной карте. *Ф1.1.П] [Естественно, ситуация накладывает отпечаток и на состояние. *С8]

1. Как Вы её воспринимаете, оцениваете, переживаете и преодолеваете? (Какие действия помогают вам преодолеть ситуацию или свое состояние).
[Присутствует недовольство, *A4] [небольшой элемент угнетённости, *E2] [невозможность в данный момент позволить себе то, что хочется, *1K2, *Б1] [создает удручающие ощущения. *A3] [Это произошло из-за нерационального распределения денег, желания позволить себе и своей семье больше. Небольшие накопления, практически все, ушли на то, чтобы погасить задолженность по кредитной карте. *1B5, *1D7, *G1] [Помочь решить эту ситуацию может четко выверенный план. *1D4, *1D17] [Составлен план по выходу из кризисной ситуации. *1D4]

2. Каковы Ваши цели в этой ситуации?
[Переезд в квартиру вдвое дешевле, *2A2] [полный подсчет необходимых расходов на жизнь, *2A2] [постоянные накопления для создания неприкасаемого запаса на форс-мажорные случаи. *2A5]

3. Какие возможности и ограничения есть у Вас при достижении цели?
[Возможности – зарплата. *3A11] [Ограничения – зарплата. *3B9] [Когда у меня получится найти дополнительный доход, будет проще преодолевать ситуацию. *3A5]

4. Нужна ли Вам в этой ситуации помощь (поддержка) окружающих людей?
[Да. Прежде всего моей семьи, моей жены. *4A3] [Вместе мы составили план и собираемся ему следовать. *4B4] [А если у неё получится найти работу, ту, которая ей понравится и будет приносить доход, будет еще лучше. *4B5]

5. Если всё сложится очень плохо, то что это будет? (Максимальный неуспех).
[Очередной уход в долговую яму, *5B2, *G2] [потеря единомыслия со своей супругой. *5B2]

6. Опишите, что для Вас будет максимально успешным выходом, разрешением ситуации?
[В данный момент – жизнь по средствам, *6B4, *Б1] [выход из финансового кризиса, *6B2] [увеличение совокупного дохода семьи, накопление средств. *6B1]

Перечень применённых кодов

В таблице Б.1 приведён перечень кодов, которые использовались в разметке случая (в порядке их появления в тексте). Коды, которые начинаются с букв, относятся к описанию ситуации в целом, с цифр — к отдельным вопросам (первая цифра кода соответствует номеру вопроса).

Таблица Б.1. Коды, применявшиеся при разметке случая

Код	Подкатегория	Категория
Ф1.1	Материальная трудность	Жизненная сфера
Ф1.1.П	Подробности о материальной трудности	Жизненная сфера
C8	Собственное состояние и совладание с ним	Суть трудности
A4	Негативные неинтенсивные	Эмоции
E2	Низкий уровень энергии	Энергия
1K2	Неподконтрольность ситуации	Критерии оценки
Б1	Упоминание времени	Время
A3	Негативные интенсивные	Эмоции
1B5	Аргументы оценки	Основания оценки
1D7	Анализ опыта	Копинг
G1	Развитие ситуации	Динамика
1D4	Планомерный копинг	Копинг
1D17	Оценка действенности копинга	Копинг
2A2	Приближение (когнитивное фокусирование на трудной задаче)	Направленность цели
2A5	Развитие, увеличение	Направленность цели
3A11	Материальные возможности	Возможности
3B9	Материальные ограничения	Ограничения
3A5	Необходимость активности	Возможности
4A3	Уверенность в необходимости помощи	Необходимость помощи
4B4	Взаимодействие	Содержание помощи
4B5	Инструментальная помощь	Содержание помощи
5B2	Утрата чего-либо	Содержание неуспеха
G2	Повторяемость, цикличность	Динамика
6B4	Поддержание существующего положения	Содержание успеха
6B2	Избавление от чего-либо	Содержание успеха
6B1	Появление чего-то нового	Содержание успеха

Automated Markup of Texts Narrating about Life Difficulties Using Large Language Models

Khlebnikova A.A., Bityutskaya E.V., Kalachev G.V., Gasanov E.E.

The paper addresses the laborious nature of manual coding of qualitative data in psychological studies that use content analysis. The effectiveness of automated text markup methods utilizing modern language models such as DeepSeek, GPT-4.1, and GPT-4.1-mini is assessed, and approaches to improve markup accuracy are developed. The work is based on descriptions of difficult life situations experienced by participants in a psychological study. The study confirms the practical feasibility of using language models as a tool that significantly reduces the time spent by researchers on the initial analysis of text data.

Keywords: content analysis, large language model, GPT-4.1, DeepSeek, difficult life situation, coping, situation perception.

References

[1] E. V. Bityutskaya, M. I. Kunashenko, "Striving for Difficulty as a Type of Perception of Life Situations", *Lomonosov Psychology Journal*, **47**:1 (2024), 56–87 (In Russian).

[2] N. N. Bogomolova, N. G. Malysheva, T. G. Stefanenko, "Content Analysis", *Social Psychology: Practicum: A Textbook for University Students*, ed. T. V. Folomeeva, Aspect Press, Moscow, 2009, 131–162 (In Russian).

[3] J. Biggiogera, G. Boateng, P. Hilpert *et al.*, *BERT meets LIWC: Exploring State-of-the-Art Language Models for Predicting Communication Behavior in Couples' Conflict Interactions*, 2021, arXiv: 2106.01536.

[4] E. V. Bityutskaya, E. E. Gasanov, K. V. Khazova, N. A. Patrashkin, "Classifying the Perception of Difficult Life Tasks: Machine Learning and/or Modeling of Logical Processes", *Psychology in Russia: State of the Art*, **17**:2 (2024), 64–84.

[5] T. B. Brown, B. Mann, N. Ryder, N. Subbiah *et al.*, *Language Models are Few-Shot Learners*, 2020, arXiv: 2005.14165.

[6] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 4th ed., Sage Publications, Inc., Thousand Oaks, CA, 2019.

[7] L. Tavabi, T. Tran, K. Stefanov *et al.*, "Analysis of Behavior Classification in Motivational Interviewing", Proceedings of the Conference on Computational Linguistics and Clinical Psychology (CLPsych), 2021, 110–115.