

УДК 004.8+004.912

ТИПЫ ЭМБЕДДИНГОВ И ИХ ПРИМЕНЕНИЕ В ИНТЕЛЛЕКТУАЛЬНОЙ АКАДЕМИЧЕСКОЙ ГЕНЕАЛОГИИ

А. Х. Мариносян^[0000-0003-0577-2360]

Московский городской педагогический университет, г. Москва, Россия

a.marinosyan@yandex.ru

Аннотация

Рассмотрена проблема построения интерпретируемых векторных представлений научных текстов для задач интеллектуальной академической генеалогии. Предложена типология эмбедингов, включающая три класса: статистические, выученные нейросетевые и структурированные символьные. Обоснована необходимость объединения достоинств нейросетевых (высокая семантическая точность) и символьных (интерпретируемость измерений) подходов. Для реализации такого гибридного подхода предложен алгоритм построения выученных символьных эмбедингов путем регрессионного преобразования вектора внутреннего представления нейросетевой модели в интерпретируемый набор оценок.

Экспериментальная оценка алгоритма проведена на корпусе фрагментов авторефератов диссертаций по педагогическим наукам. Компактный трансформерный энкодер с регрессионной головой обучался воспроизводить тематические оценки, сгенерированные передовой генеративной языковой моделью. Сравнение шести режимов обучения (три типа регрессионной головы и два состояния энкодера) показало, что дообучение верхних слоев энкодера является ключевым фактором повышения качества. По результатам тестирования была выбрана наилучшая конфигурация, которая достигла коэффициента детерминации $R^2 = 0.57$ и точности определения трех наиболее релевантных концептов, равной 74%. Результаты подтверждают, что для определенного рода задач, в которых требуется формальное представление выходных данных, возможна аппроксимация поведения генеративной модели компактным энкоде-

© А. Х. Мариносян, 2026.

Данная статья распространяется на условиях международной лицензии Creative Commons License Attribution 4.0 International (CC BY 4.0).

ром с регрессионной головой при существенно меньших вычислительных затратах. В более широкой перспективе разработка алгоритмов построения выученных символьных эмбедингов будет способствовать созданию такой модели формальной репрезентации научного знания, в которой конвергенция нейросетевых и символьных методов обеспечит как масштабируемость обработки научных текстов, так и интерпретируемость векторных представлений, кодирующих содержание.

***Ключевые слова:** эмбединги, академическая генеалогия, трансформерный энкодер, регрессионная голова, символьные эмбединги, тематический профиль, обработка естественного языка, интерпретируемость, большие языковые модели, наукометрия.*

ВВЕДЕНИЕ

Академическая генеалогия – это междисциплинарная область исследований, изучающая структуру, динамику и эволюцию науки в контексте отношений научного руководства. Традиционно она опирается на формальные сведения о диссертациях (данные об авторе и руководителе, месте и годе защиты и т. д.) [1, 2]. Однако формальная связь «научный руководитель – ученик» не тождественна содержательной близости: ученик может существенно отклониться от тематики руководителя, тогда как исследователи, не связанные формально, нередко работают в едином проблемном поле. Переход от академической генеалогии формальных связей к интеллектуальной академической генеалогии, основанной на автоматизированном анализе содержания научных работ [3, 4], в своем логическом завершении предполагает построение эмбединг-пространства научного знания [5]. Под таким пространством понимается многомерная структура, в которой элементы знания представлены векторами, положение и свойства которых отражают их содержание, а топологические характеристики самого пространства несут информацию о структуре предметной области. К такому пространству могут быть применены математические методы для анализа динамики развития знания – от выявления концептуальной преемственности работ до отслеживания эволюции научных школ.

Существующие подходы к построению эмбедингов можно разделить на три основных типа, различающихся по способу получения и степени интерпретируемости.

1. Статистические эмбединги основаны на алгоритмах и методах математической статистики: TF-IDF [6, 7], матрицы совместной встречаемости слов (co-occurrence), модели «мешка слов» (bag-of-words). Их достоинство состоит в простоте реализации и интерпретации: каждое измерение вектора соответствует конкретному слову, что позволяет выявлять устойчивые группы часто встречающихся терминов, формирующих лексическое ядро текста. Ограничения, однако, существенны: эти методы игнорируют семантические связи между словами (омонимия, синонимия, полисемия), не учитывают порядок слов и контекст, а потому неспособны уловить смысловую близость текстов, различающихся лексикой. Кроме того, порождаемые ими разреженные высокоразмерные матрицы плохо масштабируются при росте словаря.

2. Выученные¹ (learned) нейросетевые эмбединги получают с помощью моделей, обученных на больших корпусах текстов. К ранним подходам относятся word2vec [8] и GloVe [9], порождающие статические векторы слов. Контекстуализированные модели: BERT [10], Sentence-BERT [11], SciBERT [12], E5 [13] – генерируют плотные векторы (размерностью от нескольких сотен до нескольких тысяч измерений), учитывающие контекст каждого слова в предложении. Достоинствами этих моделей являются высокая точность определения смысловой близости текстов, способность учитывать синонимию, полисемию, омонимию и ассоциативные связи, а также возможность работы с

¹Поясним выбор терминологии. Нейросетевые эмбединги второго типа нередко называют «семантическими» (semantic embeddings). Однако в статье отдано предпочтение термину «выученные» (learned). Определение «семантический» содержательно перегружено и не формулируемо строгим образом. С одной стороны, в широком смысле все три типа эмбедингов в той или иной мере кодируют семантику – статистические через лексические паттерны, нейросетевые через дистрибутивную гипотезу, символьные через экспертное определение шкал. С другой стороны, использование термина «семантический» несет в себе философско-филологическую нагрузку и ставит вопрос о том, что есть семантика. Термин же «выученный» точно указывает на ключевое различительное (и легко выявляемое) свойство – способ получения: эти представления *выучены* моделью из данных в ходе обучения на большом корпусе, а не сконструированы экспертом и не получены прямым подсчетом статистик.

нечеткими запросами и текстами на разных языках. Ключевое ограничение – это «непрозрачность» (black box): измерения векторного пространства не имеют явного символического значения, что затрудняет интерпретацию и вызывает обоснованные сомнения в применимости таких представлений в областях, требующих прозрачности, в частности в науке.

3. Структурированные символичные эмбединги – это такие представления, в которых каждое измерение вектора соответствует заранее определенной экспертной шкале, а значения задаются на интерпретируемой числовой шкале (например, от 0 до 10). В контексте анализа научных работ такой вектор может описывать *тематический профиль* исследования – формальное представление тематики работы, заданное иерархическим классификатором [14]. Был предложен алгоритм формирования тематических профилей диссертаций с помощью большой языковой модели (БЯМ): для каждой работы на основании иерархического классификатора БЯМ присваивает числовые оценки релевантности по каждому элементу классификатора, формируя тем самым *тематический вектор* – символичный эмбединг с интерпретируемыми координатами [14]. Следует отметить, что в общем случае символичные эмбединги могут отражать не только тематический профиль, но и более сложное формальное представление работы: например, результаты исследования, закодированные по правилам специализированной онтологии предметной области [15].

Вместе с тем опыт генерации символических эмбедингов напрямую с помощью БЯМ выявил ряд проблем. Процесс содержит высокую долю произвольности: результат зависит от формулировки промпта, версии модели, температуры и других факторов; воспроизводимость ограничена, а систематический контроль качества затруднен. Поэтому необходим инструментарий, который позволил бы получать символичные эмбединги более контролируемо и эффективно.

Таким образом, перед нами стоит следующая проблема: каким алгоритмическим образом можно создать формальное векторное представление научной работы, которое одновременно обладало бы высокой семантической точностью (свойством выученных нейросетевых эмбедингов) и интерпретируемостью координат (свойством символических эмбедингов), чтобы затем, применив различные метрики, выявить и оценить концептуальную преемствен-

ность научных работ. Цель настоящей статьи состоит не в полном решении этой проблемы, а в разработке и экспериментальной проверке подхода, являющегося компонентом ее решения: алгоритма построения **выученных символьных эмбеддингов** (learned symbolic embeddings), сочетающих достоинства типов представлений, описанных выше.

1. АЛГОРИТМ ПОСТРОЕНИЯ ВЫУЧЕННЫХ СИМВОЛЬНЫХ ЭМБЕДДИНГОВ

1.1. Обоснование

В основе современных подходов к обработке естественного языка лежит архитектура трансформера [16], преобразующая входную последовательность токенов в высокоразмерные векторные представления (эмбеддинги) в латентном пространстве. В классическом варианте трансформерной архитектуры, реализованном в большинстве БЯМ, механизм «выхода» из латентного пространства осуществляется через проекционный слой, который отображает латентный вектор в вектор логитов размерности словаря, после этого для получения вероятностного распределения следующего токена применяется функция Softmax [16].

Применительно к задачам извлечения формализованных знаний (например, оценки релевантности набора концептов) использование стандартной генеративной парадигмы сопряжено с рядом фундаментальных ограничений. Получение тематического вектора, описанного выше, происходит посредством генерации текста (например, в формате JSON), и процесс принимает вид двойной трансформации: сначала потенциально богатая семантика латентного вектора «сжимается» до одного дискретного токена естественного языка; затем сгенерированная последовательность токенов должна быть разделена на составные части, структурирована и преобразована обратно в числовой вектор.

Такой подход неэффективен с позиции генерации (инференса), поскольку требует авторегрессионной генерации последовательности токенов для описания структуры, которую можно выразить одним вектором. Кроме того, он неэффективен с позиции обучения: латентный вектор в принципе способен кодировать многомерные семантические конструкции, однако при авторегрессионной генерации весь обучающий сигнал на каждом шаге проходит через «узкое горлышко» (bottleneck) предсказания единственного токена, что ограничивает

объем структурированной информации, который модель может извлечь из обучающих данных за один шаг обратного распространения.

Альтернативным и потенциально более эффективным подходом для решения задачи создания формальной репрезентации текста является прямой выход из латентного слоя в целевое формальное пространство, минуя этап генерации естественного языка. Это можно достигнуть путем замены стандартного проекционного слоя на регрессионную голову (regression head) – выходной модуль, который с использованием регрессии преобразует вектор внутреннего (латентного) состояния в непрерывные числовые значения целевых признаков [17] (в нашем случае – в компоненты тематического вектора). Предлагаемый нами подход позволяет осуществлять предсказание всего вектора признаков за один проход (forward pass), существенно снижает вычислительные затраты и предоставляет более широкий инструментарий по контролю качества генерации.

Технически это может быть реализовано на базе трансформерного энкодера – архитектурного блока, использующего механизмы двунаправленного внимания (self-attention) для формирования целостного, контекстно-зависимого векторного представления входного текста [16]. Использование трансформерных энкодеров (самая известная их архитектура – BERT [10]) является стандартом в машинном обучении для задач извлечения признаков. Эффективность решения, когда регрессионная голова надстраивается над выходом энкодера, подтверждается рядом исследований. Например, BERT с регрессионной головой использовался для генерации оценок качества машинного перевода [17]. Мультиязычный энкодер mT0 с дополнительным предобучением на задачу различения близких и далеких пар предложений был дополнен регрессионной головой, преобразующей выходной семантический вектор напрямую в числовую оценку качества перевода [18]. Ансамбль энкодеров с регрессионным выходом применялся для предсказания сложности экзаменационных заданий [19].

1.2. Архитектуры регрессионной головы

Трансформерный энкодер обрабатывает входной текст потоково: каждому токenu (слову или его части) на выходе соответствует отдельный вектор. Таким образом, для текста из n токенов энкодер порождает n векторов раз-

мерности d . Для получения единого вектора предложения применено усреднение (mean pooling) этих n векторов. Поскольку тексты в наборе данных дополняются до одинаковой длины служебными «пустыми» токенами, усреднение произведено только по содержательным токенам (служебные исключаются маской). Полученный вектор $e \in \mathbb{R}^d$ подается на вход регрессионной головы, которая отображает его в вектор предсказанных оценок $\hat{y} \in \mathbb{R}^k$, где k – число компонент этого вектора (в случае академической генеалогии \hat{y} – это тематический вектор научной работы).

В алгоритме предусмотрены два варианта реализации регрессионной головы: посредством линейной регрессии и многослойного перцептрона (multilayer perceptron, MLP). Данные, полученные с использованием этих двух вариантов, будут сравниваться друг с другом на этапе интерпретации результатов.

Линейная регрессионная голова осуществляет аффинное преобразование

$$\hat{y} = We + b,$$

где $W \in \mathbb{R}^{k \times d}$ – обучаемая матрица весов, каждая строка которой задает «направление» в пространстве эмбедингов, соответствующее одной компоненте выходного вектора; $b \in \mathbb{R}^k$ – обучаемый вектор сдвига (bias). Число обучаемых параметров составляет $k \times d + k$, что обеспечивает устойчивость к переобучению при ограниченных объемах данных. Линейные пробы над предобученными энкодерами являются стандартным инструментом анализа латентных представлений в естественной обработке языка [20].

MLP-регрессионная голова в рамках исследования была реализована как двухслойный перцептрон с нелинейной активацией:

$$\hat{y} = W_2 \cdot \text{Dropout}(\text{ReLU}(W_1 e + b_1), p) + b_2,$$

где $W_1 \in \mathbb{R}^{h \times d}$ и $W_2 \in \mathbb{R}^{k \times h}$ – обучаемые матрицы весов первого и второго слоев соответственно, $b_1 \in \mathbb{R}^h$ и $b_2 \in \mathbb{R}^k$ – обучаемые сдвиги, h – размерность промежуточного скрытого слоя; ReLU – нелинейная функция активации, обнуляющая отрицательные значения; Dropout – механизм регуляризации, при котором случайная доля p нейронов обнуляется на каждом шаге обучения, что препятствует «соадаптации» нейронов и снижает переобучение [21]. С одной сторо-

ны, перцептрон MLP способен моделировать нелинейные зависимости между вектором энкодера и целевыми оценками. С другой стороны, если MLP-голова показывает существенно лучшие значения, чем линейный регрессор, то это означает, что сам вектор энкодера плохо уловил «семантику» исходного текста и «семантическая» связь между исходным текстом и оценками достраивается самим MLP.

1.3. Процесс обучения

Тренировочные данные разбиваются на тренировочную, валидационную и тестовую подвыборки в пропорции 70% – 15% – 15%. Целевые оценки $y_i \in \mathbb{R}^k$ (или, используя терминологию машинного обучения, оценки, рассматриваемые как эталонные данные (ground truth)) могут формироваться как экспертным образом, так и путем использования передовой БЯМ, которая считается достаточно квалифицированной для данной задачи (то есть разрыв в способностях между передовой (state-of-the-art) БЯМ и небольшим используемым энкодером настолько большой, что оценки, предоставленные БЯМ, могут приниматься за эталон (ground truth) для данной задачи).

Для определения влияния различных архитектурных решений предусмотрены шесть режимов обучения, образованных комбинацией по двум факторам.

Первый фактор – тип обучения регрессионной головы:

- *линейная голова с нахождением матрицы весов алгоритмическим путем.* Использована регуляризация Тихонова, известная в англоязычной литературе как гребневая регрессия (ridge regression) [22]²;
- *линейная голова с нахождением матрицы весов путем машинного обучения.* Параметры выучиваются итеративно посредством градиентного спуска³;

²В исследовании для реализации алгоритма гребневой регрессии использована функция Ridge python-библиотеки sklearn. URL: https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression

³Для построения линейного преобразования использован класс torch.nn.Linear библиотеки PyTorch. URL: <https://docs.pytorch.org/docs/stable/generated/torch.nn.Linear.html>

- *MLP-регрессионная голова*. Архитектура была описана выше⁴.

Вторым фактором является состояние энкодера:

- *замороженный энкодер*. Все параметры предобученного трансформерного энкодера фиксируются и не обучаются;
- *частично дообученный (fine-tuned) энкодер*. Основная часть параметров энкодера остается замороженной, но последние N слоев трансформера размораживаются и обучаются совместно с регрессионной головой.

Комбинация двух факторов (три типа обучения головы и два состояния энкодера) образует шесть режимов, что позволяет изолировать вклад каждого фактора: сравнение при фиксированном типе головы показывает эффект дообучения энкодера, сравнение при фиксированном состоянии энкодера – эффект выбора регрессионной головы.

Во всех режимах машинного обучения в качестве функции потерь выбрана среднеквадратичная ошибка⁵. Обучение контролируется механизмом ранней остановки: если значение функции потерь на валидационной выборке не улучшается в течение заданного числа последовательных эпох, обучение прекращается и восстанавливается состояние модели с наилучшим валидационным результатом. Для ускорения обучения использован алгоритм оптимизации Adam⁶.

1.4. Метрики оценки качества

Для оценки алгоритма использован набор метрик, каждая из которых характеризует определенный аспект качества регрессии. Все метрики вычислены на тестовой выборке в исходных единицах шкалы.

⁴Для реализации последовательной архитектуры слоев перцептрона использован класс Sequential python-библиотеки PyTorch. URL: <https://pytorch.org/docs/stable/generated/torch.nn.Sequential.html>

⁵Для реализации использован класс MSELoss python-библиотеки PyTorch. URL: <https://docs.pytorch.org/docs/stable/generated/torch.nn.MSELoss.html>

⁶Для реализации алгоритма использован класс Adam python-библиотеки PyTorch. URL: <https://pytorch.org/docs/stable/generated/torch.optim.Adam.html>

Средняя абсолютная ошибка (mean absolute error, MAE) принимает значения от 0 до $+\infty$; значение 0 соответствует идеальному предсказанию⁷. MAE измеряет среднее абсолютное отклонение предсказаний от истинных значений в единицах исходной шкалы. Эта метрика наиболее интуитивно интерпретируема и, в отличие от среднеквадратичной ошибки, одинаково учитывает все ошибки, не усиливая влияние выбросов.

Корень из среднеквадратичной ошибки (root mean square error, RMSE) принимает значения от 0 до $+\infty$ (0 – идеальное предсказание) и измеряется в тех же единицах, что и целевая переменная⁸. Метрика RMSE выбрана как дополнение к MAE, поскольку она непропорционально сильнее штрафует большие отклонения: возведение в квадрат делает вклад ошибки в 2 балла вчетверо большим, чем ошибки в 1 балл. Если RMSE существенно превышает MAE, это сигнализирует о наличии отдельных примеров с аномально большими ошибками.

Коэффициент детерминации (R^2) принимает значения от $-\infty$ до 1⁹. Значение 1 означает идеальное совпадение предсказаний с истинными значениями; 0 – то, что модель предсказывает не лучше, чем тривиальная модель, всегда возвращающая среднее значение по выборке; < 0 – то, что модель предсказывает хуже среднего. Коэффициент R^2 позволяет судить о качестве модели безотносительно к масштабу данных, что отличает его от MAE и RMSE.

Корреляция Спирмена (ρ) принимает значения от -1 до $+1$ ¹⁰. Значение 1 означает идеальное совпадение рангов (то есть порядка) предсказанных и истинных значений; 0 – отсутствие монотонной связи; -1 – полностью обратный порядок. Метрика вычисляется для каждой из k компонент выходного вектора отдельно и затем усредняется. Корреляция Спирмена выбрана потому, что она

⁷Для вычисления этой метрики использована функция `mean_absolute_error` python-библиотеки `sklearn`. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error

⁸Для реализации использована функция `mean_squared_error` python-библиотеки `sklearn`. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error

⁹При реализации использована функция `r2_score` python-библиотеки `sklearn`. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score

¹⁰При реализации использована функция `spearmanr` python-библиотеки `SciPy`. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html>

оценивает качество ранжирования, а не абсолютную точность (как последнее имеет место в корреляции Пирсона). Для анализа данных гуманитарных наук это важно, поскольку правильное определение порядка величин относительно друг друга зачастую имеет большее значение, чем точное предсказание абсолютной величины

Точность извлечения (Тор-К) принимает значения от 0 до 1 (от 0% до 100% совпадения). Для каждого примера определяются K компонент с наибольшими значениями в предсказанном и в целевом векторах, затем вычисляется доля пересечения этих двух множеств:

$$Top - K_i = \frac{|top_K(\hat{y}_i) \cap top_K(y_i)|}{K},$$

где $top_K(\mathbf{y})$ – множество индексов K наибольших компонент вектора \mathbf{y} , а итоговое значение метрики есть среднее $Top-K_i$ по всем примерам тестовой выборки. Эта метрика отвечает на практический вопрос: если из всех компонент тематического вектора необходимо отобрать K наиболее релевантных, то какая доля отобранных компонент (на основе предсказанных алгоритмом значений) совпадет с действительно наиболее значимыми (исходя из значений, принятых в исследовании за эталон)? Значение 1 означает, что модель безошибочно определяет все K ведущих компонент; 0 – ни одна из выбранных компонент не входит в число действительно наиболее значимых. В нашем исследовании были выбраны значения $K = 3$ и $K = 5$.

2. РЕЗУЛЬТАТЫ

Экспериментальная оценка была проведена на корпусе из 1105 текстов диссертаций по педагогическим наукам; каждый объект представляет собой объединенное текстовое поле, включающее название диссертации, объект и предмет исследования (ограниченность текстового поля объясняется тем, что контекст используемого трансформерного энкодера ограничен 512 токенами). Данные были разделены на три непересекающихся подвыборки: 773 текста в тренировочной выборке, 166 – в валидационной и 166 – в тестовой (соответственно 70%, 15% и 15%). Целевой вектор оценок $\mathbf{y}_i \in \mathbb{R}^{40}$ состоит из 40 компонент, из которых 15 отражают уровень образования, а оставшиеся 25 описывают предметную область исследования (простота компонентов была обусловле-

на тем, что целью оценки было проверить сам алгоритм, а не способность энкодеров различать сложную «семантику»). Оценки принимают целочисленные значения по шкале от 0 до 10, где 10 означает максимальную релевантность соответствующей компоненты, а 0 – ее отсутствие в тексте.

Эталонные оценки были сформированы с использованием одной генеративной модели, рассматриваемой как наилучшее доступное решение для данного типа задач, – Gemini 3.0 Flash [23], которой предъявлялись информация о диссертации (название, объект, предмет) и список из 40 концептов (взаимно соответствующим 40 компонентам целевого вектора). Модель получала инструкцию оценить релевантность каждого концепта по 11-балльной шкале (0–10), после этого ответы интерпретировались как компоненты вектора y_i . Такой дизайн эксперимента был специально выбран для проверки гипотезы: может ли компактный трансформерный энкодер с относительно небольшим числом параметров и регрессионной головой воспроизвести поведение передовой генеративной модели.

В качестве предобученного трансформерного энкодера была использована модель LaBSE-en-ru – компактная версия Language-Agnostic BERT Sentence Embedding, адаптированная для русского и английского языков. Модель генерирует эмбединги размерности $d = 768$ и содержит порядка 128 млн параметров, что существенно меньше, чем у современных генеративных моделей общего назначения. Выбор LaBSE-en-ru обусловлен, во-первых, ее хорошими показателями на задачах семантического сходства для русского языка, во-вторых, прагматическими соображениями: модель достаточно небольшая и может быть дообучена с использованием ограниченных вычислительных ресурсов.

Во всех нейросетевых режимах обучение проводилось в течение максимального количества эпох - 20, однако фактическое число эпох для каждой модели контролировалось механизмом ранней остановки: если значение функции потерь на валидационной выборке не улучшалось в течение трех последовательных эпох, обучение прекращалось и восстанавливалось состояние модели с наименьшим значением функции потерь. При дообучении энкодера пересчитывались веса последних трех слоев (всего в LaBSE-en-ru 12 слоев).

Табл. 1. Результаты шести режимов на тестовой выборке.

Режим	MAE	RMSE	R^2	ρ	Top-3	Top-5
Гребневая регрессия, замороженная голова	0.92	1.74	0.380	0.475	0.641	0.628
Гребневая регрессия, дообученная	0.66	1.45	0.562	0.524	0.735	0.680
Линейная голова, замороженная	0.94	1.75	0.363	0.452	0.635	0.608
Линейная голова, дообученная	0.65	1.45	0.569	0.540	0.744	0.666
MLP-голова, замороженная	0.87	1.71	0.411	0.477	0.647	0.629
MLP-голова, дообученная	0.68	1.48	0.538	0.538	0.719	0.671

Результаты обучения представлены в табл. 1. Дообучение энкодера дало наибольший эффект: сравнение пар режимов с одинаковой регрессионной головой, но с разным состоянием энкодера показывает, что размораживание трех верхних слоев и их совместное обучение с головой приводит к существенному улучшению всех метрик. Например, для линейной нейросетевой головы коэффициент детерминации R^2 на тестовой выборке возрастает с 0.36 до 0.57, средняя абсолютная ошибка снижается примерно с 0.94 до 0.65 балла, а доля правильно определенных трех наиболее релевантных концептов увеличивается с 0.63 до 0.75. Аналогичная картина наблюдается и для MLP-головой, а также для гребневой регрессии, где R^2 растет с 0.38 до 0.56, а MAE – уменьшается с 0.92 до 0.66. Эти результаты свидетельствуют о том, что предобученные эмбединги, хотя и содержат богатую «семантическую» информацию, неоптимальны для конкретной задачи тематической оценки и адаптация верхних слоев энкодера к этой задаче играет положительную роль.

Эффект архитектуры регрессионной головы оказался более умеренным и значительно менее выраженным, чем эффект дообучения энкодера: при замороженном энкодере MLP-голова демонстрирует небольшое преимущество над линейными вариантами по R^2 и ранжирующим метрикам (Спирмен, Тор-К), что указывает на способность нелинейного преобразования частично компенсировать ограниченность фиксированных эмбедингов. Однако после дообучения энкодера различия между типами голов существенно сглаживаются. Это пока-

зывает, что при дообученном пространстве представлений линейного отображения в целевое пространство в значительной степени достаточно, а усложнение головы дает минимальный выигрыш при повышенной вычислительной стоимости и риске переобучения.

Результат сравнения компактной регрессионной архитектуры с эталонным поведением Gemini 3.0 Flash можно оценить, по достигнутым значениям: дообученные режимы объясняют более 56% дисперсии целевых оценок и достигают средней корреляции по Спирмену около 0.54–0.56 между предсказаниями и эталонными оценками по 40-мерному пространству концептов. Три наиболее релевантных концепта определяются правильно примерно в 74% случаев, а пять – примерно в 67–68% случаев. Это указывает на то, что компактный энкодер с регрессионной головой способен достаточно точно при существенно меньших вычислительных затратах аппроксимировать оценки, выставленные передовой генеративной моделью. На практике это означает, что для описанного рода задач массовой автоматической разметки научных текстов можно использовать энкодер с регрессионной головой, сохраняя при этом приемлемый уровень качества и интерпретируемости

ЗАКЛЮЧЕНИЕ

Предложен подход к построению формальных тематических репрезентаций научных текстов, основанный на прямом отображении латентного пространства трансформерного энкодера в целевое метрическое пространство посредством регрессионной головы. Экспериментальная оценка на корпусе фрагментов авторефератов диссертаций по педагогическим наукам подтвердила работоспособность подхода: малый по числу параметров трансформерный энкодер воспроизводит сгенерированную передовой БЯМ оценку трех наиболее релевантных признаков (в рамках задач академической генеалогии) с точностью до 74%.

Практическая значимость предложенного подхода определяется возможностью его развертывания в локальных моделях, способных обрабатывать массивы научных текстов объемом в сотни тысяч и миллионы единиц – масштаб, типичный для библиотечных фондов и наукометрических баз данных. В отличие от обращений к генеративным языковым моделям, инференс с ис-

пользованием регрессионной головы выполняется за один прямой проход и не требует значительных вычислительных ресурсов. Результатом является интерпретируемая формальная репрезентация тематического профиля работы – числовой вектор в пространстве предметных концептов, к которому в дальнейшем могут быть применены разнообразные метрики для анализа динамики научного знания: выявление тематических трендов, кластеризация исследовательских направлений, измерение семантической близости между работами и предметными областями.

Исследование проводилось в условиях малой выборки и ограниченных вычислительных ресурсов, что подтвердило способность подхода функционировать при минимальных затратах. Вместе с тем именно эти ограничения указывают на перспективные направления масштабирования. Во-первых, перспективным является переход от плоского вектора оценок к более сложным формам целевого пространства, например к структурам, согласованным с онтологиями конкретных отраслей знания [24], где выходом модели является не вектор релевантностей, а фрагмент формального графа знаний. Во-вторых, представляет интерес интеграция регрессионной головы не с компактным энкодером, а с современными большими декодерными архитектурами со «смесью экспертов» (mixture-of-experts, MoE): замена выхода одного из экспертов на регрессионную голову позволила бы модели в процессе «рассуждения» порождать формальные репрезентации и тем самым «рассуждать» не только человеческим, но и строгим формальным языком предметной области. Совершенствование логики построения формализованного представления в сочетании с масштабированием на большие модели может дать не только повышение эффективности обработки, но и качественно новые возможности для формального анализа научных текстов.

В более широкой перспективе движение к построению эмбединг-пространства научного знания требует конвергенции нескольких традиционно развивающихся обособленно подходов: нейросетевых методов распределенных представлений и символьных методов формализации знания, экспертной работы по построению предметных онтологий и возможностей больших языковых моделей по обработке текстов, эпистемологических концепций структуры научного знания и инструментария машинного обучения. Для этого в том

числе необходимо совершенствование понимания того, какой может быть структура выученного символьного эмбединга-пространства и каким образом можно описывать динамические связи в этом пространстве.

СПИСОК ЛИТЕРАТУРЫ

1. *Mulcahy C.* The Mathematics Genealogy Project comes of age at twenty-one // *Notices of the AMS*. 2017. Vol. 64. No. 5. P. 466–470.
2. *David S.V., Hayden B.Y.* Neurotree: A Collaborative, Graphical Database of the Academic Genealogy of Neuroscience // *PLoS ONE*. 2012. Vol. 7. No. 10. e46608. <https://doi.org/10.1371/journal.pone.0046608>
3. *Лернер И.М., Мариносян А.Х., Григорьев С.Г., Юсупов А.Р., Аникиева М.А., Гарифуллина Г.А.* Подход к формированию интеллектуальной академической генеалогии с использованием больших языковых моделей // *Электромагнитные волны и электронные системы*. 2024. Т. 29. № 4. С. 108–120. <https://doi.org/10.18127/j5604128-202404-09>
4. *Григорьев С.Г., Лернер И.М., Мариносян А.Х., Григорьева М.А.* К вопросу отбора учебно-методической информации для реализации адаптивной системы управления обучением: алгоритм априорной классификации авторов // *Информатика и образование*. 2025. Т. 40. № 2. С. 66–78. <https://doi.org/10.32517/0234-0453-2025-40-2-66-78>
5. *Мариносян А.Х., Григорьев С.Г.* Научные публикации и эмбединга-пространство знаний // *Электронные библиотеки*. 2026. Т. 29. № 2 (в печати).
6. *Salton G., Buckley C.* Term-Weighting Approaches in Automatic Text Retrieval // *Information Processing & Management*. 1988. Vol. 24. No. 5. P. 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
7. *Sparck Jones K.* A Statistical Interpretation of Term Specificity and Its Application in Retrieval // *Journal of Documentation*. 1972. Vol. 28. No. 1. P. 11–21. <https://doi.org/10.1108/eb026526>
8. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space // *arXiv preprint*. 2013. arXiv:1301.3781.
9. *Pennington J., Socher R., Manning C.D.* GloVe: Global Vectors for Word Representation // *Proceedings of EMNLP*. 2014. P. 1532–1543.
10. *Devlin J., Chang M.-W., Lee K., Toutanova K.* BERT: Pre-training of Deep

Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019. Vol. 1. P. 4171–4186.

<https://doi.org/10.18653/v1/N19-1423>

11. *Reimers N., Gurevych I.* Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks // Proceedings of EMNLP. 2019. P. 3982–3992.

<https://doi.org/10.18653/v1/D19-1410>

12. *Beltagy I., Lo K., Cohan A.* SciBERT: A Pretrained Language Model for Scientific Text // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2019. P. 3615–3620.

<https://doi.org/10.18653/v1/D19-1371>

13. *Wang L., Yang N., Huang X., Yang L., Majumder R., Wei F.* Multilingual E5 Text Embeddings: A Technical Report // arXiv preprint. 2024. arXiv:2402.05672.

14. *Мариносян А.Х., Григорьев С.Г., Лернер И.М., Аникуева М.А.* Автоматизированное сравнение научных исследований на базе академической генеалогии // Информатика и образование. 2025. Т. 40. № 6. С. 16–27.

<https://doi.org/10.32517/0234-0453-2025-40-6-16-27>

15. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A., Solovyev V.D., Zhiltsov N.G.* Mathematical Knowledge Representation: Semantic Models and Formalisms // Lobachevskii Journal of Mathematics. 2014. Vol. 35. No. 4. P. 348–354. <https://doi.org/10.1134/S1995080214040143>

16. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I.* Attention Is All You Need // Advances in Neural Information Processing Systems. 2017. Vol. 30. P. 5998–6008.

17. *Shimanaka H., Kajiwara T., Komachi M.* Machine Translation Evaluation with BERT Regressor // arXiv preprint. 2019. arXiv:1907.12679.

18. *Viskov V., Kokush G., Larionov D., Eger S., Panchenko A.* Semantically-Informed Regressive Encoder Score // Proceedings of the Eighth Conference on Machine Translation (WMT). 2023. P. 815–821.

<https://doi.org/10.18653/v1/2023.wmt-1.69>

19. *Gombert S., Menzel L., Di Mitri D., Drachler H.* Predicting Item Difficulty and Item Response Time with Scalar-Mixed Transformer Encoder Models and Rational Network Regression Heads // Proceedings of the 19th Workshop on Innova-

tive Use of NLP for Building Educational Applications (BEA 2024). 2024. P. 483–492. URL: <https://aclanthology.org/2024.bea-1.40/> (дата обращения 02.02.2026).

20. *Alain G., Bengio Y.* Understanding Intermediate Layers Using Linear Classifier Probes // arXiv preprint. 2017. arXiv:1610.01644.

21. *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research. 2014. Vol. 15. No. 1. P. 1929–1958.

22. *Hoerl A.E., Kennard R.W.* Ridge Regression: Biased Estimation for Non-orthogonal Problems // Technometrics. 1970. Vol. 12. No. 1. P. 55–67. <https://doi.org/10.1080/00401706.1970.10488634>

23. *Pichai S., Hassabis D., Kavukcuoglu K.* A new era of intelligence with Gemini 3 // Google. The Keyword. URL: <https://blog.google/products-and-platforms/products/gemini/gemini-3/#note-from-ceo> (дата обращения 02.02.2026).

24. *Елизаров А.М., Кириллович А.В., Липачев Е.К., Невзорова О.А.* Цифровая экосистема OntoMath как подход к построению пространства математических знаний // Электронные библиотеки. 2023. Т. 26. № 2. С. 154–202. <https://doi.org/10.26907/1562-5419-2023-26-2-154-202>

TYPES OF EMBEDDINGS AND THEIR APPLICATION IN INTELLECTUAL ACADEMIC GENEALOGY

A. Kh. Marinosyan^[0000-0003-0577-2360]

Moscow City University, Moscow, Russia

a.marinosyan@yandex.ru

Abstract

The paper addresses the problem of constructing interpretable vector representations of scientific texts for intellectual academic genealogy. A typology of embeddings is proposed, comprising three classes: statistical, learned neural, and structured symbolic. The study argues for combining the strengths of neural embeddings (high semantic accuracy) with those of symbolic embeddings (interpretable dimensions). To operationalize this hybrid approach, an algorithm for learned symbolic embeddings is introduced, which utilizes a regression-based mapping from a model's internal representation to an interpretable vector of scores.

The approach is evaluated on a corpus of fragments from dissertation abstracts in pedagogy. A compact transformer encoder with a regression head was trained to reproduce topic relevance scores produced by a state-of-the-art generative language model. A comparison of six training setups (three regression-head architectures and two encoder settings) shows that fine-tuning the upper encoder layers is the primary driver of quality improvements. The best configuration achieves $R^2 = 0.57$ and a Top-3 accuracy of 74% in identifying the most relevant concepts. These results suggest that, for tasks requiring formalized output representations, a compact encoder with a regression head can approximate a generative model's behavior at substantially lower computational cost. More broadly, the further development of algorithms for constructing learned symbolic embeddings contributes to building a model of formal knowledge representation in which the convergence of neural and symbolic methods ensures both the scalability of scientific text processing and the interpretability of vector representations that encode their content.

Keywords: *embeddings, academic genealogy, transformer encoder, regression*

head, symbolic embeddings, topic profile, natural language processing, interpretability, large language models, scientometrics.

REFERENCES

1. *Mulcahy C.* The Mathematics Genealogy Project comes of age at twenty-one // *Notices of the AMS*. 2017. Vol. 64. No. 5. P. 466–470.
2. *David S.V., Hayden B.Y.* Neurotree: A Collaborative, Graphical Database of the Academic Genealogy of Neuroscience // *PLoS ONE*. 2012. Vol. 7. No. 10. e46608. <https://doi.org/10.1371/journal.pone.0046608>
3. *Lerner I.M., Marinosyan A.Kh., Grigoriev S.G., Yusupov A.R., Anikieva M.A., Garifullina G.A.* An Approach to the Formation of Intellectual Academic Genealogy Using Large Language Models // *Journal Electromagnetic Waves and Electronic Systems*. 2024. Vol. 29. No. 4. P. 108–120. <https://doi.org/10.18127/j5604128-202404-09> (In Russ.)
4. *Grigoriev S.G., Lerner I.M., Marinosyan A.Kh., Grigorieva M.A.* On the Issue of Educational and Methodological Information Selection for Implementing an Adaptive Learning Management System: Algorithm of A Priori Authors Classification // *Informatics and Education / Informatika i obrazovanie*. 2025. Vol. 40. No. 2. P. 66–78. <https://doi.org/10.32517/0234-0453-2025-40-2-66-78> (In Russ.)
5. *Marinosyan A.Kh., Grigoriev S.G.* Scientific Publications and the Embedding Space of Knowledge // *Electronic Libraries / Elektronnye biblioteki*. 2026. Vol. 29. No. 2. (In press.) (In Russ.)
6. *Salton G., Buckley C.* Term-Weighting Approaches in Automatic Text Retrieval // *Information Processing & Management*. 1988. Vol. 24. No. 5. P. 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
7. *Sparck Jones K.* A Statistical Interpretation of Term Specificity and Its Application in Retrieval // *Journal of Documentation*. 1972. Vol. 28. No. 1. P. 11–21. <https://doi.org/10.1108/eb026526>
8. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space // *arXiv preprint*. 2013. arXiv:1301.3781.
9. *Pennington J., Socher R., Manning C.D.* GloVe: Global Vectors for Word Representation // *Proceedings of EMNLP*. 2014. P. 1532–1543.
10. *Devlin J., Chang M.-W., Lee K., Toutanova K.* BERT: Pre-training of Deep

Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019. Vol. 1. P. 4171–4186.

<https://doi.org/10.18653/v1/N19-1423>

11. *Reimers N., Gurevych I.* Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks // Proceedings of EMNLP. 2019. P. 3982–3992.

<https://doi.org/10.18653/v1/D19-1410>

12. *Beltagy I., Lo K., Cohan A.* SciBERT: A Pretrained Language Model for Scientific Text // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2019. P. 3615–3620.

<https://doi.org/10.18653/v1/D19-1371>

13. *Wang L., Yang N., Huang X., Yang L., Majumder R., Wei F.* Multilingual E5 Text Embeddings: A Technical Report // arXiv preprint. 2024. arXiv:2402.05672.

14. *Marinosyan A.Kh., Grigoriev S.G., Lerner I.M., Anikieva M.A.* Automated Comparison of Scientific Research Based on Academic Genealogy // Informatics and Education / Informatika i obrazovanie. 2025. Vol. 40. No. 6. P. 16–27.

<https://doi.org/10.32517/0234-0453-2025-40-6-16-27> (In Russ.)

15. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A., Solovyev V.D., Zhiltsov N.G.* Mathematical Knowledge Representation: Semantic Models and Formalisms // Lobachevskii Journal of Mathematics. 2014. Vol. 35. No. 4. P. 348–354. <https://doi.org/10.1134/S1995080214040143>

16. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I.* Attention Is All You Need // Advances in Neural Information Processing Systems. 2017. Vol. 30. P. 5998–6008.

17. *Shimanaka H., Kajiwara T., Komachi M.* Machine Translation Evaluation with BERT Regressor // arXiv preprint. 2019. arXiv:1907.12679.

18. *Viskov V., Kokush G., Larionov D., Eger S., Panchenko A.* Semantically-Informed Regressive Encoder Score // Proceedings of the Eighth Conference on Machine Translation (WMT). 2023. P. 815–821.

<https://doi.org/10.18653/v1/2023.wmt-1.69>

19. *Gombert S., Menzel L., Di Mitri D., Drachler H.* Predicting Item Difficulty and Item Response Time with Scalar-Mixed Transformer Encoder Models and Rational Network Regression Heads // Proceedings of the 19th Workshop on Innova-

tive Use of NLP for Building Educational Applications (BEA 2024). 2024. P. 483–492. URL: <https://aclanthology.org/2024.bea-1.40/> (date accessed: 02.02.2026).

20. *Alain G., Bengio Y.* Understanding Intermediate Layers Using Linear Classifier Probes // arXiv preprint. 2017. arXiv:1610.01644.

21. *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research. 2014. Vol. 15. No. 1. P. 1929–1958.

22. *Hoerl A.E., Kennard R.W.* Ridge Regression: Biased Estimation for Non-orthogonal Problems // Technometrics. 1970. Vol. 12. No. 1. P. 55–67. <https://doi.org/10.1080/00401706.1970.10488634>

23. *Pichai S., Hassabis D., Kavukcuoglu K.* A new era of intelligence with Gemini 3 // Google. The Keyword. URL: <https://blog.google/products-and-platforms/products/gemini/gemini-3/#note-from-ceo> (date accessed: 02.02.2026).

24. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A.* Digital Ecosystem OntoMath as an Approach to Building the Space of Mathematical Knowledge // Electronic Libraries / Elektronnye biblioteki. 2023. Vol. 26. No. 2. P. 154–202. <https://doi.org/10.26907/1562-5419-2023-26-2-154-202> (In Russ.)

СВЕДЕНИЯ ОБ АВТОРЕ



МАРИНОСЯН Андреас Хачатурович – аспирант Института цифрового образования Московского городского педагогического университета. Область научных интересов: наукометрия, обработка естественных языков, архитектуры языковых моделей.

Andreas Khachaturovich MARINOSYAN – PhD Student at the Institute of Digital Education, Moscow City University. Research interests: scientometrics, natural language processing, language model architectures.

email: a.marinosyan@yandex.ru

ORCID: 0000-0003-0577-2360

Материал поступил в редакцию 10 декабря 2025 года