Исследование подходов для построения векторных представлений текстов в задаче сопоставления вакансий и резюме

A Preprint

Петрова Александра Сергеевна
Московский государственный университет имени М. В. Ломоносова
Научный руководитель: Майсурадзе Арчил Ивериевич
Научный консультант: Колосов Алексей Михайлович

Abstract

Задача сопоставления вакансий и резюме связана с необходимостью работодателей отбирать кандидатов на основе большого количества резюме. Эта задача стала особенно актуальной с развитием онлайн-платформ для поиска работы, где количество резюме может быть значительным. В данной статье рассматривается метод сопоставления вакансий и резюме с использованием векторных представлений, полученных на основе различных моделей архитектуры Transformer. В данной статье предлагается несколько подходов, направленных на повышение качества сопоставления: Fine-tuning моделей и линейное преобразование векторных представлений.

Keywords Job–candidate matching · рекомендательные модели · предобученные NLP модели · embedding · cosine similarity

1 Введение

Современные технологии электронного рекрутинга значительно изменили процесс найма, создавая большие объемы вакансий и резюме, из-за этого появилась необходимость разработки и внедрения эффективных систем рекомендаций.

Семантические технологии продемонстрировали свою эффективность в задачах ранжирования документов в смежных областях, таких как поиск научных публикаций [Latard et al., 2017], информационный поиск в новостных агрегаторах [Kumar et al., 2022]. В области подбора персонала они также показали высокую результативность, позволяя улучшать сопоставление вакансий и резюме за счет учета скрытых семантических связей между текстами [Assia and Zizette, 2022, Thali et al., 2023, Tayade et al., 2021, Huang, 2023]. Благодаря использованию векторных представлений текстов, основанных на современных языковых моделях, такие технологии обеспечивают высокую точность рекомендаций даже при отсутствии точного совпадения ключевых слов. Преимущества векторного семантического поиска включают возможность измерять семантическую близость между текстами и выявлять схожие кандидаты без необходимости полного тематического совпадения. Исследования показывают, что использование векторных представлений позволяет улучшить качество рекомендаций в задачах сопоставления вакансий и резюме [Kurek et al., 2024]. В это работе предложено использование Zero-Shot Learning для адаптации рекрутинговых систем к новым вакансиям и резюме. В качестве моделей ZSL используются предварительно обученные модели архитектуры Transformer, такие как BERT [Devlin et al., 2019, Reimers and Gurevych, 2019] и GPT [Yenduri et al., 2023].

В данной статье рассматривается методика ранжирования вакансий и резюме на основе векторной близости. Для создания векторных представлений используются модели из семейства BERT и GPT, а также другие модели, оптимизированные для работы с текстами на русском языке. Оценка близости

вакансий и резюме проводится с применением косинусного сходства векторов. Предлагаются методы для повышения качества рекомендаций, направленные на сближение верных пар вакансий и резюме в общем векторном пространстве.

2 Постановка задачи

Задача сопоставления вакансии и резюме представляется как задача ранжирования элементов множества, где элементами выступают резюме, а запросом — вакансия.

В этом исследовании мы ограничиваемся предположением, что и вакансии, и резюме представляют собой текстовые данные. Таким образом, задача сопоставления вакансии и резюме сводится к ранжированию текстов в ответ на текстовый запрос.

Входные данные:

- Множество вакансий $V = \{v_1, v_2, \ldots, v_n\}$, где каждая вакансия v_i описана текстом $T(v_i)$.
- Множество резюме $R = \{r_1, r_2, \ldots, r_m\}$, где каждое резюме r_j описано текстом $T(r_j)$.
- Множество релевантных пар $P = \{(v_i, r_j)\}$, где v_i и r_j представляют собой корректно сопоставленную пару вакансии и резюме.

Необходимо построить ранжирующее отображение из множества V в множество упорядоченных списков резюме $r \in R$ длины n.

3 Методология

Для оценки схожести текстов вакансий и резюме предлагается строить их векторные представления и использовать cosine similarity в качестве метрики близости.

3.1 Токенизация

Перед векторизацией текстов вакансий и резюме необходимо разделить их на отдельные единицы, или токены, которые будут являться входными данными для языковых моделей. Пусть текст t состоит из последовательности символов. Токенизация g — это процесс, который преобразует текст t в множество токенов:

$$g(t) = \{t_1, t_2, \ldots, t_n\},$$

где t_i — отдельный токен, а n — общее количество токенов в тексте.

Токенизация проводится с использованием подсловных единиц (subword units), таких как байт-пейр кодирование (BPE) или SentencePiece, чтобы эффективно представлять как часто встречающиеся, так и редкие слова. Это позволяет избежать проблемы "неизвестных слов"(out-of-vocabulary), возникающей при обработке текстов, содержащих специализированные термины и аббревиатуры, характерные для резюме и вакансий.

3.2 Векторизация

Векторное представление текста (эмбеддинг) — это преобразование множества текстов T в многомерное векторное пространство:

$$f : T \to E \subseteq \mathbb{R}^d.$$

Эмбеддинг текста $t \in T$, обозначаемый как $f(t)$, описывает его семантическое содержание. Используя эмбеддинги $f(t_1)$ и $f(t_2)$, можно оценить степень семантической близости между текстами t_1 и t_2.

3.3 Ранжирование

После того как тексты вакансий и резюме преобразованы в векторные представления, для каждой вакансии вычисляется косинусное сходство с каждым из резюме. Пусть $\mathbf{v}_q$ — векторное представление вакансии, а $\mathbf{v}_{r_i}$ — векторное представление i-го резюме из множества резюме $R = \{r_1, r_2, \ldots, r_m\}$. Косинусное сходство вычисляется по формуле:

$$\text{cosine\_similarity}(\mathbf{v}_q, \mathbf{v}_{r_i}) = \frac{\mathbf{v}_q \cdot \mathbf{v}_{r_i}}{\|\mathbf{v}_q\| \cdot \|\mathbf{v}_{r_i}\|}, \quad \forall i \in \{1, 2 \ldots, m\}.$$

Результатом является набор значений $S = \{\text{cosine\_similarity}(\mathbf{v}_q, \mathbf{v}_{r_1}), \ldots, \text{cosine\_similarity}(\mathbf{v}_q, \mathbf{v}_{r_m})\}$, который затем сортируется по убыванию.

Итоговый ранжированный список резюме определяется как:

$$R_{\text{sorted}} = \text{sort}(R, \text{key} = S, \text{descending=True}).$$

Таким образом, резюме с наибольшим значением метрики cosine similarity располагаются в начале списка.

3.4 Оценка качества

Для оценки качества сопоставления вакансий и резюме предлагается использовать стандартную метрику для задачи ранжирования - mean average precision at K.

Допустим, алгоритм ранжирования выдал ранжированный список L_v^K длины K объектов $r \in R$ для элемента $v \in V$. Тогда precision at K (P@K) - это величина, равная сумме P@k по индексам k от 1 до K только для релевантных элементов, деленной на мощность множества R_v:

$$AP_v@K = \frac{1}{|R_v|} \sum_{k=1}^{K} 1\left[L_v^K[k] \in R_v\right] P_v@k$$

В average precision at K качество ранжирования оценивается для отдельно взятого объекта. Идея mean average precision at K (MAP@K) заключается в том, чтобы посчитать AP@K для каждого объекта и усреднить.

Mean average precision at K - это усредненная по всем объектам AP@K:

$$MAP@K = \frac{1}{N} \sum_{v=1}^{N} AP_v@K$$

Идея усреднения логична, если все объекты одинаково важны. В случае если это не так, вместо простого усреднения можно использовать взвешенную сумму, домножив AP@K каждого объекта на вес, соответствующий его важности. В данной работе будет принято предположение, что все вакансии имеют одинаковый вес.

4 Эксперименты

4.1 Описание данных

Эксперименты проводились на двух наборах данных: реальных и синтетических. Необходимость генерации синтетических данных связана с ограниченностью реальных данных.

Реальные данные включают набор вакансий и резюме на русском языке, предоставленный HR-отделом компании ACD/Labs. Большинство вакансий и резюме относятся к сфере IT.

Для каждой вакансии известен перечень резюме кандидатов, приглашённых на собеседование. В выборке отсутствуют резюме, не связанные с конкретными вакансиями, и каждое резюме привязано к единственной вакансии. В Таблице 1 представлено распределение вакансий по числу соответствующих им резюме.

Синтетический набор данных был сформирован путем генерации текстов резюме на основе описаний вакансий, полученных с платформы HH.ru. Для генерации текстов использовалась модель GPT-4. В результате было создано 600 пар текстов, где каждая пара представляет собой описание вакансии и соответствующее ей резюме. Качество данных было проверено выборочно вручную.

Количество резюме	Количество вакансий	Описание
1	1	1 вакансия - 1 резюме
2	2	1 вакансия - 2 резюме
3	1	1 вакансия - 3 резюме
4	2	1 вакансия - 4 резюме
5	1	1 вакансия - 5 резюме
8	1	1 вакансия - 8 резюме
9	1	1 вакансия - 9 резюме
11	2	1 вакансия - 11 резюме
13	2	1 вакансия - 13 резюме
17	1	1 вакансия - 17 резюме
90	13	Total

Таблица 1: Распределение вакансий по количеству резюме

4.2 Базовый подход: предварительно обученные модели

Модели NLP, обученные на обширных и разнообразных наборах данных, имеют широкое понимание естественного языка. Суть эксперимента заключается в использовании этих моделей для сопоставления описаний вакансий и резюме без предварительного обучения на этой задаче.

4.3 Архитектура Transformer

В основе предварительно обученных моделей лежат архитектуры на основе Transformer, которые произвели революцию в обработке естественного языка. Такие модели, как BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pretrained Transformer) и их производные продемонстрировали исключительную способность понимать и генерировать человекоподобный текст.

4.4 Обоснование выбора моделей

Для экспериментов были выбраны следующие эмбеддеры:

- BERT multilingual: Эта версия модели BERT обучена на множестве языков, включая русский, что позволяет применять её в мультиязычной среде и обрабатывать тексты с элементами иностранных языков.
- BERT Russian: Эта версия BERT специализирована на русском языке и обучена на корпусах, включающих русские тексты.
- MiniLM Sentence-Transformer: Компактная и высокопроизводительная модель Wang et al. [2020], оптимизированная для вычисления семантического сходства текстов. Благодаря небольшому числу параметров, она требует минимальных вычислительных ресурсов и обеспечивает высокую скорость обработки.
- RuGPT2 Large: Эта модель, основанная на архитектуре GPT-2 и адаптированная для русского языка, обучена на крупном корпусе русскоязычных текстов.
- RuGPT3 Large: Модель RuGPT3 представлена как более мощная версия GPT-3, обученная на русском языке.
- text-embedding-ada-002: Модель от OpenAI, предоставляющая эффективные эмбеддинги для текстов на разных языках, включая русский. Text-embedding-ada-002 создана для извлечения обобщённых текстовых представлений. Её способность к генерации высококачественных эмбеддингов особенно ценна для задач ранжирования и сравнения текстов.

Основные характеристики моделей приведены в Табл.2

Итоговое векторное представление текста формируется путём агрегации векторов токенов с использованием pooling-методов (mean pooling или CLS pooling) для BERT. В случае с GPT векторизация проводится на основе использования скрытых слоёв модели. MiniLM и text-embedding-ada-002 выдают векторное представление всего текста сразу.

Использование нескольких моделей позволяет провести сравнительный анализ их эффективности в задаче сопоставления вакансий и резюме и выбрать наиболее подходящую.

Model	Parameters	Layers	Languages
bert-base-multilingual-cased	110M	12	104
bert-base-ru-cased	110M	12	Russian
paraphrase-MiniLM-L6-v2	22M	6	Multilingual
rugpt2large	774M	48	Russian
rugpt3large	760M	96	Russian
text-embedding-ada-002	-	-	Multilingual

Таблица 2: Сравнение характеристик различных моделей векторизации

4.5 Результаты экспериментов

Результаты базовых экспериментов приведены в Табл. 3. Модели на основе архитектур BERT и GPT показали схожие показатели качества, но BERT обладает меньшим количеством параметров и более высокой эффективностью. Sentence-Transformer превзошел базовые трансформеры, демонстрируя лучшие результаты, что объясняется его адаптацией для задач сравнения текстов через обучение на парах предложений. Модель text-embedding-ada-002 продемонстрировала наилучшие результаты на тестовой выборке.

Metric	Random	BERT Multilingual	BERT Russian	RuGPT2 Large	RuGPT3 Large	MiniLM	text-embedding-ada-002
MAP@10	0.03	0.1	0.12	0.11	0.13	0.33	0.58
MAP@20	0.04	0.14	0.15	0.14	0.15	0.41	0.66

Таблица 3: Результаты моделей по метрикам MAP@10 и MAP@20.

4.6 Разрыв модальностей

Предположим, что вакансии и резюме - это две различные модальности мультимодального объекта «найм», то есть «найм» определяется двумя «состояниями»: вакансия и резюме. Мультимодальным назовем объект, который имеет несколько состояний - модальностей.

Возможным недостатком базового подхода является наличие разрыва модальностей - несовпадения областей, в которые попадают вектора разных модальностей, в общем векторном пространстве. Существование этого разрыва делает сопоставление вакансий и резюме на основе косинусного сходства некорректным: для нескольких вакансий ближайшим может оказаться одно и то же резюме - резюме, вектор которого находится ближе всего к области вакансий.

В идеальном случае, соответствующие друг другу вакансии и резюме должны отображаться в совпадающие векторы. Если это условие выполняется, сопоставление вакансий и резюме с помощью косинусного сходства становится корректной операцией. Исходя из этих наблюдений, возникает гипотеза: после преодоления разрыва модальностей, качество сопоставления вакансий и резюме вырастет.

4.6.1 Проверка существования разрыва: Maximum mean discrepancy

Метрика Maximum Mean Discrepancy измеряет расхождение между распределениями $P(X)$ и $Q(Y)$ на основе расстояния между их средними в отображенном пространстве. Для двух модальностей — вакансий (X) и резюме (Y) — MMD определяется как:

$$\mathrm{MMD}^2(\mathcal{F}, P, Q) = \|\mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{y\sim Q}[f(y)]\|^2_{\mathcal{F}},$$

где: $P(X)$ и $Q(Y)$ — распределения данных двух модальностей; f — функция в пространстве $\mathcal{F}$.

На практике часто используется ядровая версия MMD, где функция $f(x)$ заменяется ядром $k(x, y)$.

Ядровая версия MMD:

Для конечных выборок $X = \{x_1, \ldots, x_m\}$ и $Y = \{y_1, \ldots, y_n\}$, вычисление MMD через ядро $k(x, y)$ выглядит так:

$$\text{MMD}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}k(x_i, x_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}k(y_i, y_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}k(x_i, y_j).$$

где: $k(x, y)$ — функция ядра.

4.6.2 Линейное преобразование

Чтобы перевести множество вакансий во множество резюме, предлагается сделать выравнивание эмбеддингов с помощью линейного преобразования с нулевым смещением. Подобный подход использовался в [5] в задаче машинного перевода. Функцию потерь определим, как среднеквадратическую ошибку между преобразованными векторами вакансий и истинными векторами резюме:

$$\text{Loss}_{\text{Linear}} = \frac{1}{l}\sum_{i=1}^{l}\|r_i - Wv_i\|^2$$

где r_i - вектор резюме, v_i - вектор вакансии, W - обучаемая матрица весов, l - количество пар (вакансия, резюме) в обучающей выборке.

4.7 Fine-tuning

Другим подходом к повышению качества сопоставления вакансий и резюме является fine-tuning предварительно обученной модели для создания векторных представлений текстов. В данной работе предлагается проводить дообучение на синтетических парах текстов, представляющих собой описания соответсвующих друг другу вакансий и резюме.

Для дообучения модели предлагается использовать функцию потерь MultipleNegativesRankingLoss:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_i))}{\sum_{j=1}^{N}\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_j))},$$

где $\mathbf{u}_i$ и $\mathbf{v}_i$ — векторные представления текста вакансии и резюме соответственно, $\text{sim}(\cdot, \cdot)$ обозначает косинусное сходство, а N — количество пар в батче. Эта функция потерь учит модель увеличивать сходство между релевантными парами $(\mathbf{u}_i, \mathbf{v}_i)$ и уменьшать его для нерелевантных $(\mathbf{u}_i, \mathbf{v}_j,\ i \neq j)$.

В качестве базовой модели предлагается использовать MiniLM, так как она содержит меньшее количество параметров по сравнению с более крупными моделями, что снижает объем занимаемой памяти и ускоряет процесс обучения.

4.8 Результаты экспериментов

Эксперименты выполнялись на синтетически сгенерированных данных, поскольку данные, размеченные экспертами, обладают иной природой. Выборка была разделена на обучающую и тестовую части в соотношении 5:1. Для обучения линейного преобразования использовалась модель text-embedding-ada-002, для fine-tuning применялась модель MiniLM.

Параметры обучения:

1. Линейное преобразование:
   (a) Функция потерь: MSE
   (b) Количество эпох: 50
2. Fine-tuning:
   (a) Функция потерь: MultipleNegativesRankingLoss
   (b) Количество эпох: 20
   (c) warmup_steps: 300

	Вакансии и резюме	Вакансии	Резюме
MMD	0.087	0.003	0.002

Таблица 4: До преобразования

	Вакансии и резюме	Вакансии	Резюме
MMD	0.003	0.003	0.002

Таблица 5: После линейного преобразования

Таблица 6: Сравнение MMD до и после преобразования

Линейное преобразование позволило устранить разрыв модальностей, что подтверждается данными в Табл.6, где статистика была рассчитана с использованием косинусного ядра $k(x, y)$.

Оба подхода продемонстрировали улучшение качества сопоставления Табл.7, подтверждая их эффективность. Fine-tuning обеспечивает значительное повышение качества за счет точной адаптации весов модели к специфике данных, что позволяет достичь существенного прироста метрик. Линейное преобразование, хотя и улучшает результаты, демонстрирует меньший абсолютный прирост, что объясняется уже изначально высоким базовым качеством модели до преобразования.

Metric	text-embedding-ada-002	text-embedding-ada-002 linear	MiniLM	MiniLM fine-tuning
MAP@10	0.9	0.95	0.24	0.91
MAP@20	0.9	0.95	0.25	0.91

Таблица 7: Результаты моделей по метрикам MAP@10 и MAP@20 на синтетической выборке

5 Выводы

Задача сопоставления вакансий и резюме была формализована как задача ранжирования элементов, где резюме рассматриваются как элементы множества и вакансия как запрос. Исследование предполагало, что как вакансии, так и резюме представляют собой тексты, и поэтому задача сводилась к ранжированию текстов по текстовому запросу.

Для оценки качества использовались метрики MAP@K, которые дают среднюю эффективность ранжирования по выборке. Обзор литературы показал, что современные методы ранжирования основываются на построении векторных представлений объектов. В нашем исследовании рассматривались различные подходы к созданию векторных представлений, согласованные с косинусным сходством. Использовались модели различных архитектур Transformer, такие как BERT и GPT.

Было показано, что базовые архитектуры BERT и GPT-2 не справляются с задачей ранжирования, показывая точность MAP@10 около 0.1. Модель из семейства Sentence-Transformer MiniLM лучше справляется с задачей, показывая точность около 0.4. Наилучшее качество (0.6) показала модель text-embedding-ada-002, которая имеет наиболее сложную архитектуру и наибольшее число параметров.

Между векторными представлениями вакансий и резюме существует разрыв, который может быть вызван их структурными и смысловыми различиями. Для того чтобы его устранить, было предложено обучить линейное преобразование над замороженными векторными представлениями модели text-embedding-ada-002. Оно увеличило точность MAP@10 рекомендаций на синтетической выборке с 0.9 до 0.95. Также с целью приближения векторов верных пар вакансий и резюме, был проведен fine-tuning модели MiniLM, который улучшил точность MAP@10 с 0.24 до 0.91. В дальнейшем, можно рассматривать разные комбинации этих двух методов на различных архитектурах моделей.

Список литературы

Bastien Latard, Jonathan Weber, Germain Forestier, and Michel Hassenforder. Towards a Semantic Search Engine for Scientific Articles, page 608–611. Springer International Publishing, 2017. ISBN 9783319670089. doi:10.1007/978-3-319-67008-9_54. URL http://dx.doi.org/10.1007/978-3-319-67008-9_54.

Anshul Kumar, Abhinav Panwar, and Anurag Rawat. Research paper on question answering system using bert, 12 2022.

Brek Assia and Boufaida Zizette. Semantic approaches survey for job recommender systems. 08 2022.

Raj Thali, Suyog Mayekar, Shubham More, Sanjana Barhate, and Sangeetha Selvan. Survey on job recommendation systems using machine learning. In 2023 International Conference on

Innovative Data Communication Technologies and Application (ICIDCA), pages 453–457, 2023. doi:10.1109/ICIDCA56705.2023.10100122.

Tanvi Tayade, Rutuja Akarte, Gayatree Sorte, Rohit Tayade, and Priti Khodke. Data Mining Approach to Job Recommendation Systems, pages 503–509. 01 2021. ISBN 978-3-030-49794-1. doi:10.1007/978-3-030-49795-8_48.

Ran Huang. Improved content recommendation algorithm integrating semantic information. Journal of Big Data, 10, 05 2023. doi:10.1186/s40537-023-00776-7.

Jarosław Kurek, Tomasz Latkowski, Michał Bukowski, Bartosz Świderski, Mateusz Łępicki, Grzegorz Baranik, Bogusz Nowak, Robert Zakowicz, and Łukasz Dobrakowski. Zero-shot recommendation ai models for efficient job–candidate matching in recruitment process. Applied Sciences, 14(6):2601, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions, 2023. URL https://arxiv.org/abs/2305.10435.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. URL https://arxiv.org/abs/2002.10957.