МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА МОДЕЛИ НА ОСНОВЕ ТРАНСФОРМЕРОВ ДЛЯ АНАЛИЗА РЕЗЮМЕ И ВЫЯВЛЕНИЯ КОМПЕТЕНЦИЙ КАНДИДАТОВ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Кудряшова Александра Сергеевича

Научный руководитель
доцент, к.ф.-м.н. _____ А. С. Иванова

Заведующий кафедрой
доцент, к. ф.-м. н. _____ С. В. Миронов

Саратов 2026

ВВЕДЕНИЕ

Компании сегодня тратят много ресурсов на поиск и оценку кандидатов. Резюме приходится просматривать вручную, и это не только занимает много времени, но и зависит от восприятия конкретного рекрутера. Один и тот же опыт разные специалисты могут оценить по-разному. Отсюда растёт интерес к автоматизации: модели машинного обучения и методы обработки естественного языка позволяют быстрее и точнее анализировать данные о кандидатах.

Работа выполнена в рамках проекта «HR-Интеллект» по программе Стартап как диплом. Его задача — собрать программный комплекс, который умеет разбирать резюме, выделять компетенции и сопоставлять их с требованиями конкретных позиций. Система ориентирована на этап первичного скрининга: она помогает отсеять нерелевантные отклики и выделить подходящих кандидатов ещё до ручного просмотра. Это снижает нагрузку на HR-отдел, убирает часть субъективности и делает процесс подбора более предсказуемым за счёт опоры на данные.

Отдельное внимание уделено качеству обработки текста: резюме часто пишутся в свободной форме, с разным уровнем детализации и различной структурой. Модель должна уметь работать с такими данными — извлекать навыки, учитывать контекст и корректно интерпретировать формулировки. Для этого необходимо использовать современные подходы в NLP, позволяющие учитывать не только отдельные слова, но и их смысловые связи внутри текста.

При активном найме система должна за минуты обработать сотни резюме и выдать список кандидатов с наибольшим соответствием требованиям. Это не заменяет рекрутера, а даёт ему удобный инструмент для принятия решений.

Команда проекта состоит из студента 451 группы Кудряшова Александра, студентов 411 группы Прохорова Максима и Пшеничникова Стемира.

Цель дипломной работы:

Дообучение модели Sentence-BERT для задачи автоматизированного сопоставления текстов резюме с матрицей компетенций IT-профессий.

В соответствии с поставленной целью выделены следующие задачи:

1. Изучить современные подходы и технологии в области автоматического анализа текстовой информации, обработки естественного языка (NLP) и применения предобученных языковых моделей, в частности, архитектуры

BERT и ее модификации SBERT.

2. Исследовать методы автоматического выявления и классификации профессиональных компетенций в текстах резюме.
3. Разработать архитектуру и спроектировать программные модули для предобработки данных, обучения модели и инференса.
4. Подготовить данные для обучения модели, включая формирование матриц компетенций для выбранных ИТ-профессий.
5. Реализовать модель на основе SBERT для семантического ранжирования компетенций в тексте резюме.

Структура и объем работы. Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и трёх приложений. Общий объём работы составляет 65 страниц. Датасет для обучения моделей размещён в открытом репозитории GitHub, список использованных источников информации включает 22 наименования.

Первый раздел «Системный анализ и теоретические аспекты семантического моделирования профессиональных компетенций» посвящён исследованию существующих технологий и методов, которые могут быть использованы для решения поставленных задач. В рамках данного раздела рассмотрены концепции и функциональная структура системы «HR-Интеллект» (рисунок 1).

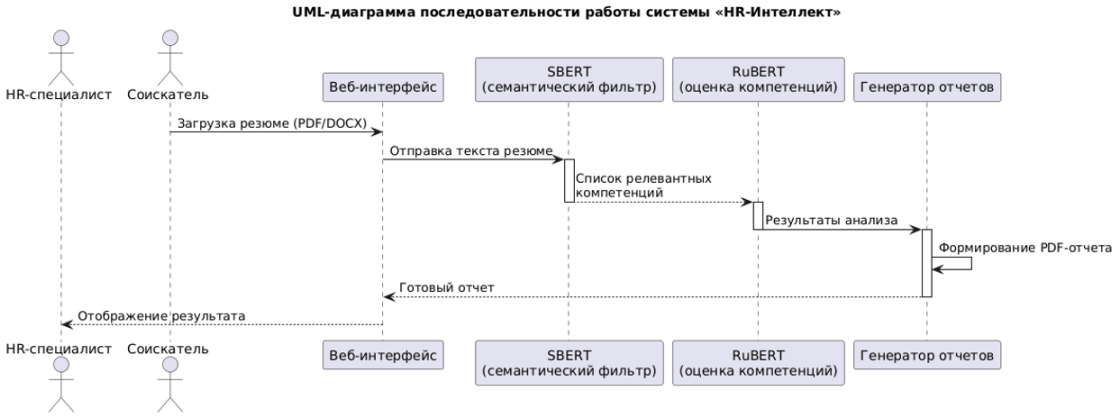

Рисунок 1 – UML-диаграмма последовательности работы системы «HR-Интеллект»

Данная система представляет собой интеллектуальный конвейер, включающий последовательную обработку резюме: загрузку документа в формате PDF или DOCX, извлечение текстового содержимого, предварительную очистку от шумов, семантический поиск компетенций с помощью дообученной модели Sentence-BERT, затем контекстную оценку уровня владения каждой компетенцией с помощью модели RuBERT, и наконец формирование структурированного отчёта с рекомендациями для HR-специалиста.

Цель данной работы заключалась в дообучении модели SBERT с целью дальнейшего сопоставления текстов из резюме с матрицами компетенций. Это позволило определять отсутствующие компетенции и передавать найденные в модель RuBERT для их последующей оценки.

Проведён сравнительный анализ традиционных методов поиска и современных семантических подходов. Методы поиска по ключевым словам (Keyword Search) и булевы алгоритмы (Boolean Search) не учитывают контекст и синонимию, что даёт низкую точность. Статистический метод TF-IDF позволяет оценить важность слов в документе, однако также не обладает семантическим пониманием и плохо работает с длинными текстами. Методы, основанные на эмбеддингах (Embedding Search), преобразуют текст в многомерный вектор и

вычисляют косинусное сходство, что уже даёт учёт семантической близости, но требует качественных векторных представлений. Наиболее совершенными являются методы семантического поиска на базе трансформерных архитектур, которые обеспечивают очень высокую гибкость и точность, поскольку учитывают контекст каждого слова во всём предложении.

В разделе подробно изложены теоретические основы обработки естественного языка. Рассмотрены статические эмбеддинги слов (Word2Vec, GloVe, FastText) и их ограничение — проблема омонимии и полисемии, когда одно слово получает фиксированный вектор независимо от контекста. Затем описана архитектура Transformer, предложенная в работе «Attention Is All You Need». Её ключевой элемент — механизм self-attention, который позволяет модели напрямую устанавливать зависимости между любыми элементами последовательности независимо от расстояния между ними. Также рассмотрены многоголовое внимание (multi-head attention) и позиционное кодирование.

Далее представлена модель BERT (Bidirectional Encoder Representations from Transformers), которая обучается на двух задачах: маскирование языка (MLM) и предсказание следующего предложения (NSP). Благодаря двунаправленности BERT учитывает как левый, так и правый контекст, что позволяет получать качественные контекстуальные эмбеддинги. Для русского языка существует адаптация RuBERT. Однако классический BERT не оптимизирован для задачи семантического поиска, поскольку попарное сравнение предложений требует больших вычислительных затрат. Поэтому была разработана модель Sentence-BERT (SBERT), которая модифицирует BERT для получения фиксированных эмбеддингов целых предложений. Эти эмбеддинги можно сравнивать с помощью простых метрик, например косинусного сходства. В работе использована русскоязычная версия sbert_large_nlu_ru от компании ai-forever с размерностью эмбеддинга 1024. Для оценки качества модели информационного поиска определены метрики Hit@k (доля запросов, для которых правильный ответ оказался среди первых k) и MRR (среднее обратное ранжирование).

Второй раздел «Процесс обучения модели» содержит описание используемых технологий, программных библиотек и фреймворков, а также ход экспериментов. Для реализации модели машинного обучения выбран язык Python и библиотека PyTorch. Использованы фреймворк Hugging Face Transformers для работы с предобученными моделями и библиотека sentence-transformers для

дообучения SBERT.

Для обучения и оценки моделей был сформирован специализированный датасет, содержащий вручную размеченные примеры соответствия фрагментов резюме профессиональным компетенциям. Каждая компетенция описывает отдельную область знаний или навыков, например: «Методы машинного обучения», «SQL базы данных», «Основы глубокого обучения». Уровень владения принимает значения от 1 до 3, где 1 соответствует базовому знакомству с областью, 2 — практическому опыту применения, 3 — уверенному владению и способности решать нетривиальные задачи. Нулевой уровень (полное отсутствие компетенции) в датасете не представлен, поскольку модель SBERT обучается только на положительных парах «резюме — компетенция». На этапе инференса, если для фрагмента резюме косинусное сходство со всеми эмбеддингами компетенций оказывается ниже порога (в данной работе берется порог 0.6), система считает, что ни одна компетенция не обнаружена, что соответствует нулевому уровню владения.

Данные хранятся в формате CSV с разделителем «;» и кодировкой UTF-8. Структура файла представлена в таблице 1.

Таблица 1 – Структура датасета

Поле	**Тип**	**Описание**
competency	строка	Наименование компетенции из матрицы
resume_text	строка	Фрагмент текста резюме
level	целое (1–3)	Уровень владения компетенцией

Для исследования зависимости качества моделей от объёма обучающей выборки были подготовлены четыре варианта датасета: 500, 1000, 1500 и 2000 записей. Они стали одним из ключевых практических результатов данной работы. Для обеспечения воспроизводимости экспериментов и возможности их использования другими исследователями все датасеты опубликованы в открытом доступе. Это позволяет независимо проверять полученные результаты, сравнивать разные модели на одинаковых данных и использовать подготовленную выборку в качестве бенчмарка для задач классификации компетенций. Пример данных можно увидеть на рисунке 2.

```
competency;resume_text;level
информационный поиск;слышал про поиск по векторным представлениям.;1
языки программирования и библиотеки (python, c++);слышал про виртуальное окружение venv в python.;1
качество и предобработка данных, подходы и инструменты;применял pipeline для объединения шагов предобработки.;2
sql базы данных (greenplum, postgres, oracle);оптимизировал распределение данных в greenplum через ключи партиционирования.;3
массово параллельные вычисления для ускорения машинного обучения (gpu);реализовывал распределенное обучение на нескольких gpu через ddp.;3
информационный поиск;применял поиск с обучением ранжированию (learning to rank) в elasticsearch.;3
nosql базы данных (cassandra, mongodb, elasticsearch, neo4j, hbase);настраивал шардирование в mongodb для горизонтального масштабирования.;3
оценка качества работы методов ии;применял многомерную калибровку через логистическую регрессию на вероятностях.;3
методы машинного обучения;использовал оптимизацию гиперпараметров через hyperopt.;2
определения, история развития и главные тренды ии;знаю, как эволюционировали методы обучения: от обратного распространения до rlhf.;2
методы машинного обучения;реализовывал gradient boosting с нуля на numpy.;3
потоковая обработка данных (data streaming, event processing);настраивал consumer group для масштабирования.;2
языки программирования и библиотеки (python, c++);разработал обёртку для cuda кода на python с помощью pybind11.;3
информационный поиск;использовал поиск с фильтрацией по диапазонам.;2
массово параллельные вычисления для ускорения машинного обучения (gpu);использовал torch.jit для ускорения инференса на gpu.;2
основы глубокого обучения;реализовывал кастомную функцию потерь на pytorch.;3
```

Рисунок 2 – Пример данных из датасета

В качестве функции потерь выбрана MultipleNegativesRankingLoss (MNRL), которая особенно эффективна при ограниченном объёме размеченных данных. Её ключевое преимущество — механизм in-batch негативов. Модель кодирует все тексты резюме и все названия компетенций, получая эмбеддинги. Затем вычисляется матрица косинусных сходств между каждым резюме и каждой компетенцией. Диагональные элементы соответствуют правильным парам, а все остальные элементы строки автоматически становятся негативными примерами. Таким образом, для каждого резюме модель учится выбирать правильную компетенцию среди всех компетенций текущего батча. Функция потерь минимизирует отрицательный логарифм вероятности правильной пары. Такой подход позволяет обходиться без ручной генерации негативных примеров и эффективно использовать вычислительные ресурсы. Размер батча был установлен равным 32 для увеличения количества негативных примеров внутри одной итерации.

Обучение проводилось в два этапа. На первом этапе был реализован базовый эксперимент с фиксированным числом эпох (5) без улучшений. Модель загружалась из предобученного чекапоинта, создавался загрузчик данных и объект функции потерь. Обучение выполнялось на датасетах различного размера, после каждой эпохи модель не сохранялась автоматически.

Таблица 2 – Результаты обучения SBERT на датасетах различного размера в первом эксперименте

Размер	**Hit@1**	**Hit@3**	**MRR**
500	0.34	0.6	0.526
1000	0.51	0.73	0.644
1500	0.587	0.84	0.721
2000	0.575	0.81	0.71

Результаты базового эксперимента (таблица 2) показали закономерность: метрика Hit@1 стабильно росла с увеличением объёма обучающей выборки от 500 до 1500 строк, достигнув значения 0,587, однако при переходе к 2000 строкам наблюдалось незначительное снижение до 0,575. Baseline — предобученная модель без дообучения — показала Hit@1 = 0,145, что более чем в три раза выше случайного выбора. Это подтверждает, что даже без тонкой настройки SBERT обладает общими семантическими знаниями. Однако снижение качества на 2000 строках свидетельствовало о переобучении из-за отсутствия механизма ранней остановки.

На втором этапе в процесс обучения были внесены улучшения, направленные на повышение качества и устойчивости модели при ограниченном объёме данных. Во-первых, применена аугментация данных методом случайного удаления слов (word dropout) с вероятностью p = 0,1. Для каждого текста резюме создаётся дополнительная версия, в которой каждое слово удаляется с указанной вероятностью, причём тексты короче трёх слов не изменяются. Это имитирует реальные вариации формулировок в резюме. Аугментация фактически удваивает количество обучающих пар без ручной разметки: например, при датасете из 1500 строк после разбиения на обучающую и валидационную части (90/10) получается 1350 примеров, а после аугментации — 2700 пар. Во-вторых, внедрён механизм ранней остановки (early stopping) с параметром patience = 2. Обучение теперь выполняется по одной эпохе за итерацию, после каждой эпохи вычисляются метрики на валидационной выборке. Если значение Hit@1 не улучшается в течение двух последовательных эпох, обучение прекращается. Это предотвращает переобучение и сохраняет модель в наилучшем состоянии. В-третьих, использован прогрев learning rate (warmup) на первой эпохе, когда скорость обучения линейно возрастает от нуля до номинального значения в течение первых 10% шагов. Это позволяет избежать резкого изменения предобученных весов. В-четвёртых, ограничена максимальная длина входной последовательности 128 токенами, что снижает потребление видеопамяти примерно в 16 раз (по сравнению со стандартными 512 токенами) за счёт квадратичной сложности self-attention. В-пятых, применена смешанная точность вычислений (mixed precision) и явная очистка видеопамяти GPU после каждого эксперимента, чтобы избежать ошибок нехватки памяти при последовательном запуске на четырёх датасетах.

Таблица 3 – Результаты обучения SBERT на датасетах различного размера во втором эксперименте

Размер	**Hit@1**	**Hit@3**	**MRR**
500	0.46	0.66	0.60
1000	0.51	0.72	0.64
1500	0.64	0.84	0.75
2000	0.61	0.81	0.72

Как видно из таблицы 3, наилучшие значения достигнуты на датасете из 1500 строк: Hit@1 = 0,64, Hit@3 = 0,84, MRR = 0,75. Это означает, что в 64% случаев модель ставит правильную компетенцию на первое место, а в 84% — в первую тройку. Среднее обратное ранжирование 0,75 соответствует тому, что правильный ответ в среднем находится на позиции 1,33 (то есть чаще всего на первом или втором месте). По сравнению с базовым экспериментом улучшения дали прирост Hit@1 на 5-6 процентных пунктов. Дообученная модель на 1500 строках превосходит baseline более чем в четыре раза. Снижение метрик при переходе к 2000 строкам объясняется несколькими факторами: неоднородностью дополнительных данных, появлением более сложных и неоднозначных формулировок, а также фиксированным размером батча, который ограничивает количество in-batch негативов.

Для наглядной демонстрации возможностей разработанной модели в составе системы «HR-Интеллект» приведены результаты анализа реального резюме (рисунки 3, 4, 5).

```
"Data Scientist": {
    "Определения, история развития и главные тренды ИИ": 2,
    "Процесс, стадии и методологии разработки решений на основе ИИ (Docker, Linux/Bash, Git)": 2,
    "Статистические методы и первичный анализ данных": 3,
    "Оценка качества работы методов ИИ": 2,
    "Языки программирования и библиотеки (Python, C++)": 3,
    "SQL базы данных (GreenPLum, Postgres, Oracle)": 2,
    "NoSQL базы данных (Cassandra, MongoDB, ElasticSearch , Neo4J, Hbase)": 1,
    "Hadoop, SPARK, Hive": 1,
    "Качество и предобработка данных, подходы и инструменты": 3,
    "Работа с распределенной кластерной системой": 1,
    "Методы машинного обучения": 3,
    "Рекомендательные системы": 2,
    "Методы оптимизации": 2,
    "Основы глубокого обучения": 3,
    "Анализ изображений и видео": 1,
    "Машинное обучение на больших данных": 2,
    "Глубокое обучение для анализа естественного языка": 3,
    "Обучение с подкреплением и глубокое обучение с подкреплением": 1,
    "Глубокое обучение для анализа и генерации изображений, видео": 1,
    "Анализ естественного языка": 3,
    "Информационный поиск": 2,
    "Массово параллельные вычисления для ускорения машинного обучения (GPU)": 2,
    "Потоковая обработка данных (data streaming, event processing)": 1,
    "Массово параллельная обработка и анализ данных": 2
}
```


Рисунок 3 – Результат анализа резюме


```
"profession": "Data Engineer",
"match_percent": 22.5,
"final_levels": {
    "SQL базы данных (GreenPLum, Postgres, Oracle)": 3,
    "Оценка качества работы методов ИИ": 0,
    ...
    "Качество и предобработка данных, подходы и инструменты": 0,
    "Рекомендательные системы": 0
},
"missing_skills": [
    {
    "name": "Определения, история развития и главные тренды ИИ",
    "required_level": 1,
    "candidate_level": 0
    },
    ...
    {
    "name": "Массово параллельная обработка и анализ данных",
    "required_level": 3,
    "candidate_level": 0
    }
],
"recommendation": {
    "is_target_best": true,
    "message": "Позиция: 'Data Engineer' (совпадение 22.5%)"
}
```


Рисунок 4 – Результат анализа резюме

```
"profession": "Data Scientist",
"match_percent": 14.583333333333334,
"final_levels": {
    ...
},
"missing_skills": [
    ...
],
"recommendation": {
    "is_target_best": false,
    "better_profession": "Data Engineer",
    "alternative_professions": [
    {
    "profession": "Data Engineer",
    "match_percent": 22.5
    },
    {
    "profession": "Manager in AI",
    "match_percent": 16.666666666666666
    },
    {
    "profession": "Technical analyst in AI",
    "match_percent": 10.638297872340425
    }
    ],
    "message": "Вы хорошо подходите на позицию 'Data Engineer' (совпадение 22.5%)"
}
```


Рисунок 5 – Результат анализа резюме

На вход системе подаётся текст резюме кандидата. Модель Sentence-BERT преобразует фрагменты резюме и все компетенции из матрицы в эмбеддинги, затем вычисляет косинусное сходство и определяет, какие компетенции присутствуют в резюме. Далее модель RuBERT оценивает уровень владения каждой компетенцией. Система формирует структурированный отчёт: для профессии Data Scientist вычисляется процент соответствия, в примере он составил 14,6%. Поскольку это значение оказалось низким, система автоматически проанализировала другие профессии и выдала рекомендацию, что кандидат лучше подходит на позицию Data Engineer с процентом соответствия 22,5%. Отчёт также содержит перечень отсутствующих компетенций с указанием требуемого и фактического уровней. Таким образом, система не только оценивает соответствие целевой позиции, но и предлагает альтернативы, давая HR-специалисту прозрачную информацию для принятия решений.

ЗАКЛЮЧЕНИЕ

В заключение необходимо отметить, что цель данной работы, заключающаяся в разработке модели машинного обучения для интеллектуального анализа резюме и выявления наличия профессиональных компетенций кандидатов, была успешно достигнута. В рамках работы была спроектирована и реализована система «HR-Интеллект», предназначенная для автоматизации процессов первичного отбора персонала с использованием современных методов обработки естественного языка и технологий машинного обучения.

Во время работы были изучены подходы к семантическому анализу текстов, архитектуры трансформеров и методы построения систем оценки компетенций. Модели RuBERT и Sentence-BERT последовательно подходят к анализу резюме. Практические результаты показали, что совместное использование нескольких моделей увеличивает точность оценки и делает систему более устойчивой к ошибкам отдельных алгоритмов.

Разработанное решение представляет собой полноценный программный комплекс, включающий обработку документов, анализ текстов резюме, расчет степени соответствия кандидатов требованиям вакансии и формирование итоговых отчетов. Архитектура системы дает возможность интегрировать её в существующие HR-процессы компаний без существенных изменений инфраструктуры. При необходимости модель может быть адаптирована под внутренние матрицы компетенций конкретной организации. Решение может использоваться как в IT-компаниях, так и в крупных корпоративных структурах, где требуется обработка большого количества откликов.

Важно подчеркнуть, что данный проект реализован в рамках программы «Стартап как диплом», что показывает его прикладной характер и возможность дальнейшего развития в качестве коммерческого продукта. Подтверждающий документ приложен к выпускной квалификационной работе.

В дальнейшем систему можно расширить за счёт новых моделей обработки текста, увеличения обучающей выборки и поддержки дополнительных источников данных. Например, помимо резюме можно анализировать сопроводительные письма, результаты тестовых заданий или профили кандидатов из профессиональных платформ. Это позволит сделать оценку компетенций более полной и повысить качество рекомендаций для HR-специалистов.