**УДК 004.891.2** **DOI: 10.14529/mmp260309**

# МОДЕЛИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОГО АЛГОРИТМА РАНЖИРОВАНИЯ УПОЛНОМОЧЕННЫХ ИНСТАНЦИЙ В ЗАДАЧАХ ПОДГОТОВКИ ОТВЕТОВ НА ЗАПРОСЫ

***А.А. Нефедова***[1]***, Т.В. Карпета***[1]
[1]Южно-Уральский государственный университет, г. Челябинск, Российская Федерация

Исследованию применения методов семантического сопоставления векторных представлений текста в задаче разработки интеллектуального сервиса ранжирования исполнителей при формировании ответов на запросы рассматривается в данной работы. Моделирование системы строится на примере определения исполнителей для подготовки ответов на обращения граждан в органы государственной власти. Разработанная система способна не только выявлять релевантных исполнителей, но и устанавливать понятную связь между обращением и полномочиями органов власти. Это позволяет получить объяснение результата работы алгоритма. Гибридная архитектура с дообученной моделью нейронной сети, на основе архитектуры SBERT, легла в основу системы. Результаты тестирования полученного алгоритма показывают существенное улучшение качества ранжирования. Итоговая модель достигает $MRR = 0,996$ и $0,739$ на обучающей и тестовой выборках соответственно. Такие показатели подтверждают применимость разработанного подхода для автоматизации процесса обработки обращений граждан.

*Ключевые слова: объяснимый искусственный интеллект; семантическое сопоставление; модели ранжирования; обработка естественного языка.*

## Введение

Ежедневно в органы государственной власти поступает огромный массив обращений граждан, и этот поток текстовой информации требует тщательного анализа. Ручная обработка таких объемов данных ложится тяжелым бременем на сотрудников, часто провоцируя кадровый дефицит и, как следствие, срывы установленных законом сроков реагирования. Автоматизированные решения, используемые сегодня, в основном сводятся к поверхностному поиску ключевых слов, что явно недостаточно для глубокого понимания сути проблемы и точного распределения документов по инстанциям. Более продвинутые методы машинной классификации, хотя и демонстрируют высокую точность, зачастую не дает пользователю прозрачного обоснования своих выводов. Предметом данного исследования являются обращения граждан и основные полномочия исследуемых органов власти.

В работе целью ставится создание интеллектуальной системы, способной упорядочивать исполнителей по степени их компетенции для подготовки ответов на обращения. Кроме того, система должна обеспечивать понятность своих решений, подсвечивая те части обращения, которые наиболее повлияли на выбор конечного получателя. Результатом внедрения такой системы станет оптимизация процедуры направления обращений граждан к соответствующим специалистам.

## 1. Постановка задачи

Задача ставится как объясняемое ранжирование. Необходимо, чтобы система, получив на вход текст обращения, определяла наиболее релевантных исполнителей, уполномоченных для подготовки ответа на данное обращение, а также давала объяснение такого ранжирования. Основная идея решения строится на том, чтобы каждому исполнителю сопоставить список его основных полномочий, а после сопоставлять отдельные термы из текста обращения с этими функциями. Такое решение дает возможность точно определить какой терм из текста стал опорным для выделения того или иного исполнителя [1–3].

Формальная постановка задачи. Пусть дано множество исполнителей – $D = \{d_1, d_2, ..., d_m\}$. У каждого исполнителя имеется множество полномочий – $d_i = \{q_{i1}, q_{i2}, .., q_{ik}\}$, где $q_{ij}$ – $j$-е полномочие $i$-го исполнителя. Обращение $S$ состоит из $n$ предложений – $S = \{s_1, s_2, ..., s_n\}$. Каждое полномочие и предложение кодируются в векторное представление (эмбеддинг) – $v(q_{ji}), v(s_i)$. Семантической схожестью двух фраз

называется косинусное расстояние между векторами представлений (эмбеддингами) этих фраз:

$$f(s_t, q_{ji}) = cos\_sim(v(s_t), v(q_{ji})). \tag{1}$$

Назовем $F(S, d_i)$ – итоговым баллом соответствия обращения $S$ исполнителю $d_i$:

$$F(S, d_i) = \Psi(S, d_i, f), \tag{2}$$

где $\Psi$ – функция агрегации; $f$ – функция подсчета семантического сходства между предложениями и функциями органа. Тогда задача заключается в нахождении множества исполнителей с наибольшими значением функции соответствия:

$$D^* = \text{top}_k\{F(S, d_i) \mid d_i \in D\}, \tag{3}$$

где $|D^*| = k$ – заданное количество исполнителей в топе; $\text{top}_k$ оператор выбора $k$ наибольших значений [4].

Рассмотрим формальную постановку задачи ранжирования. Первым этапом отбирается множество $p$ – пары из полномочия $q_{ji}$ и предложения $s_t$, для которых семантическая схожесть больше заданного порога $\theta$:

$$p = \{(s_t, q_{ji}) \mid f(s_t, q_{ji}) \geq \theta\}. \tag{4}$$

Далее для каждого предложения $s_j$ из отобранных пар оставляется не более $k_{func}$ полномочий с наибольшим $f(s_t, q_{ji})$, обозначим это множество $P$. Тогда итоговая функция агрегации:

$$\begin{gathered}\Psi(S, d_i, f) = 0,7\frac{1}{|P_i|}\sum_{(s,q)\in P_i} f(s, q) + 0,3|P_i| + h(d_i); \\ P_i = P \cap \{(s_t, q_{ji})|q_{ji} \in d_i\},\end{gathered} \tag{5}$$

где $P_i$ – множество отобранных пар, относящихся к исполнителю $d_i$; $|P_i|$ – количество таких пар; $h(d_i)$ – контекстный бонус исполнителя $d_i$. С использованием данной функции ранжирования находится итоговое множества исполнителей.

## 2. Архитектура алгоритма

Для выполнения поставленной задачи разработан ИИ-алгоритм, включающий несколько ключевых модулей (рис. 1).

1. Фильтрация – цель этого блока исключить из анализа речевые шаблоны и формальные фразы («Здравствуйте», «Прошу рассмотреть» и т.п.), которые не несут смысловой нагрузки для определения исполнителя. Текст обращения разбивается на предложения и из них отбирается заданное количество предложений методом экстрактивной суммаризации.
2. Векторизация – в этом блоке работает основная модель всей системы – модель векторизации. Дообученная модель преобразует обработанные предложения в эмбеддинги. Полномочия органов власти векторизуются заранее и хранятся в векторном виде.
3. Контекстный бонус – цель этого блока добавить дополнительную обработку всего текста целиком. С помощью основной модели подсчитывается эмбеддинг всего текста и сравнивается его косинусное расстояние с усредненным эмбеддингом полномочий органов.
4. Сопоставление эмбеддингов предложений из текста и полномочий из базы знаний, находятся заданное количество наибольших по косинусному расстоянию полномочий для конкретного предложения.
5. Ранжирование – для каждого органа подсчитывается итоговый балл, по которому происходит ранжирование и отбор топа самых релевантных исполнителей.

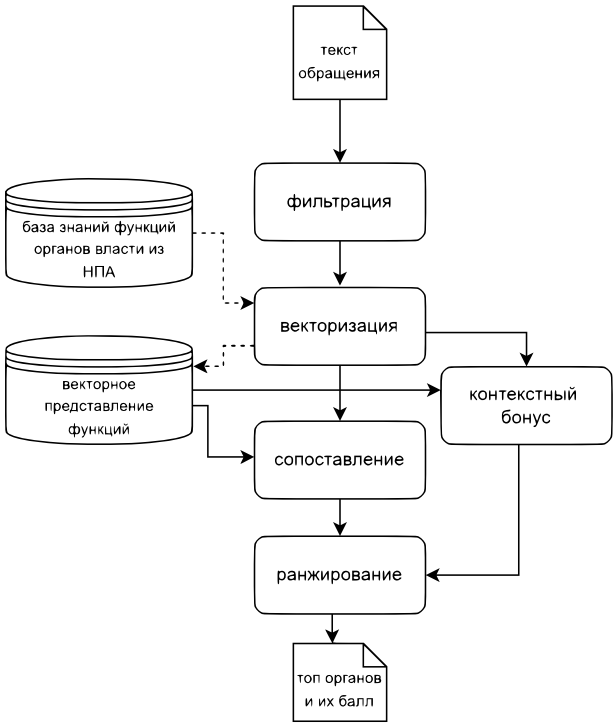


**Рис. 1**. Схема архитектуры ИИ-алгоритма

Такая архитектура обеспечивает полноценный функционал работы сервиса, совмещающий в себе нейросетевые технологии, классическое машинное обучение и ранжирование [4].

Фильтрация использует определенный набор правил для выбора значимых предложений. Сначала исходный текст обращения разбивается на термы, а после проходит через, разработанный набор правил, по которым терму присваивается определенное количество баллов. По итогу отбирается 70% от исходного количества термов, набравших наибольшее количество положительных баллов. Используется, как лексический анализ отдельных слов, так и алгоритм LSA как часть фильтрации.

После фильтрации все отобранные термы векторизуются и сопоставляются с векторным представлением полномочий из базы знаний. То есть считается косинусное расстояние между вектором терма и полномочия. А далее происходит постобработка в виде ранжирования. Математическая модель финального ранжирования служит объединяющим каркасом: она формализует, как количественные выходы всех этапов агрегируются в итоговый, интерпретируемый рейтинг органов власти. Рассмотрим поэтапно как устроена математика ранжирования.

1. При сопоставлении термов с полномочиями органов власти между их эмбеддингами подсчитывается косинусное расстояние. Отбираются пары, для которых косинусное расстояние больше установленного порога.
2. После для каждого терма оставляется не более 2 ($topKFunc = 2$) полномочий органов власти, для того чтобы оставить только самые значимые.
3. Для вычисления контекстного бонуса ($context$) сначала вычисляется косинусное расстояния между эмбеддингом всего текста и усредненным эмбеддингом полномочий всех органов. После для органа власти, у которого такое косинусное расстояние получилось самым большим, контекстный бонус берется как добавка размера 0,3, 0,2 – для второго по величине, 0,1 – для третьего, для остальных бонус не добавляется. Полученный бонус прибавляется к итоговому баллу ($total\_score$).
4. Для каждого органа власти считаем итоговый балл:

$$total\_score = \frac{\sum\limits_{i=0}^{n} sim_i}{n} \cdot 0,7 + n \cdot 0,3 + context, \tag{6}$$

где $sim$ – косинусное расстояние пар из полномочий и термов для этого органа власти; $n$ – количество таких пар; $context$ – контекстный бонус этого органа.

5. По полученной величине итогового бонуса отбирается установленное значение топ $K$ органов ($topKOrg = 5$), которые выводятся пользователю как ранжированный результат [4].

Таким образом, представленная математическая модель ранжирования обеспечивает формальное объединение результатов всех этапов интеллектуальной обработки обращений граждан.

## 3. Модель векторизации

В блоке векторизации используется обученная модель нейронной сети. Для адаптации модели к задаче сопоставления обращений граждан и полномочий органов власти была разработана архитектура, использующая в основе предобученную модель (рис. 2). Для дообучения в качестве базовой модели была выбрана русскоязычная версия Sentence-BERT от компании сбербанк, а именно подразделения sberbank AI, находящаяся в открытом доступе. Sentence-BERT – это модификация архитектуры BERT, оптимизированная под получения семантических векторных представлений (sentence embeddings), пригодных для вычисления сходства текстов [5].

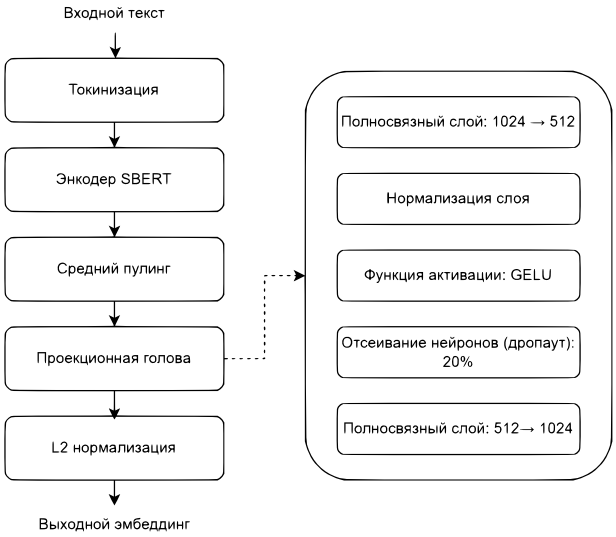


**Рис. 2**. Схема архитектуры модели векторизации

Архитектура состоит из энкодера, пулинга и проекционной головы. Энкодер – это слои базовой модели. Он формирует представления токенов, которые далее агрегируются в вектор предложения. После прохождения текста через слои энкодера получается последовательность скрытых состояний. Для получения фиксированного вектора предложения используется стратегия среднего пулинга (mean pooling) – усреднение по всем токенам скрытых состояний [6]. Далее для адаптации модели к предметной области была добавлена дополнительная проекционная голова. Она реализована в виде двух полносвязных слоев с промежуточной размерностью 512, функцией активации GELU, слоем нормализации и техникой регуляризации. Данный блок выполняет функцию адаптивного преобразования исходного эмбеддинга, позволяя модели перераспределить признаки без существенного искажения базового языкового пространства. После прохождения через эти слои выполняется L2-нормализация, что проецирует эмбеддинги на единичную гиперсферу и делает косинусное сходство эквивалентным скалярному произведению нормированных векторов. Пример последовательной обработки можно увидеть на рис. 3. Такая адаптация позволяет добавить дополнительные обучающие параметры, которые будут учится с нуля, за счет чего смогут лучше скорректировать работу всей модели под конкретную задачу [7].

```
Исходный текст: 'Это текст для демонстрации этапов работы модели'

Токенизированный текст:
   Токены: ['это', 'текст', 'для', 'демонстрации', 'этапов', 'работы', 'модели']
   размер: [1, 9]
   закодированные токены: [101, 736, 8258, 849, 22019, 29793, 2218, 6860, 102]
   декодировано: '[CLS] это текст для демонстрации этапов работы модели [SEP]'

Эмбеддинг текста после энкодера:
   размер: [1, 9, 1024]
   первые 10 значений: [0.5670368075370789, 0.20619051158428192, 0.5033450126647949,
-0.10953009873628616, 0.5615037083625793, -0.28234532475471497, 0.0141895920226717,
0.3991846442222595, 0.1663999706506793, 0.7440507411956787]

Эмбеддинг текста после пулинга:
   размер: [1, 1024]
   первые 10 значений: [0.553853452205658, -0.08569473773241043, 0.5970013737678528,
-0.07489902526140213, 0.06230998039245655, -0.5428573489189148, 0.18716730177402496,
-0.3782186210155487, 0.35551661252975464, -0.032412610948085785]

Эмбеддинг текста после проекционной головы:
   размер: [1, 1024]
   первые 10 значений: [0.10027159750461578, -0.031435512006282806, 0.10542413592338562,
0.016723938286304474, -0.028420507305145264, -0.07775447517633438, 0.11808395385742188,
-0.07402434200048447, -0.09865652024745941, 0.07027578353881836]

Эмбеддинг текста после нормализации (итоговый):
   размер: [1, 1024]
   первые 10 значений: [0.027006732299923897, -0.008466709405183792, 0.02839449606835842,
0.004504355601966381, -0.00764981284737587, -0.020942065864801407, 0.03180423751473427,
-0.019937407225370407, -0.026571735739707947, 0.018927786499261856]
```

**Рис. 3**. Пример этапов создания эмбеддинга

В процессе обучения модели была создана база данных, включающая информацию о 27 органах власти. Каждому органу власти было присвоено от 12 до 19 полномочий. Для каждого полномочия разработали 45 примеров текстов, иллюстрирующих проблемы граждан, которые могут быть решены с помощью данного полномочия. Генерация производилась с помощью больших языковых моделей, находящихся в общем доступе. Например, для органа власти «Главное управление лесами» одним из полномочий было создано «Регулирование сделок с лесными участками и лесными насаждениями», а для данного полномочия в качестве положительных примеров были сгенерированы такие предложения: «Не удается получить сведения о нарушениях при оформлении договоров лесопользования, прошу помощи», «Прошу проконтролировать законность заключенных договоров купли-продажи лесных участков», «В районе отсутствует информация о регистрации сделок с лесными участками, прошу вмешательства». После генерации предложения были отчищены от часто встречаемых в начале слов, таких как: «Хотелось бы», «Нужно», «Интересует», «Прошу», «Жалуюсь на», «Обращаюсь», «Прошу уточнить», «Не могу понять». Это сделано для того чтобы модель не концентрировалась на фразах, которые используются в обращения всех органов власти. Общий массив размеченных данных включал 17100 записей. Обучающая выборка составила 15580 примеров, оставшиеся 1520 образцов были зарезервированы для валидации (тестовый набор). При формировании тестовой подвыборки для каждого из 45 классов полномочий использовалось по 3 репрезентативных случая.

Оптимизация параметров нейросетевой модели осуществлялась посредством алгоритма обратного распространения градиента ошибки. В качестве целевого функционала минимизации применялась функция триплетных потерь (triplet loss):.

$$TripletLoss = max(0, d(a, p) - d(a, n) + margin), \tag{7}$$

где $d(x, y)$ – расстояние между объектами; $a$ – опорный объект; $p$ – объект, который должен находится близко к якорю; $n$ – объект, который должен находится далеко от якоря; $margin$ – константа, задающая минимальное допустимое различие между позитивной и негативной парами, принадлежащий множеству $[0; 1]$. В решаемой задачи в качестве якоря используется полномочие органа власти, в качестве позитивного примера текст сгенерированный для

данного полномочия, а негативный пример – это текст другого полномочия. В качестве метрики расстояния используется косинусная дистанция: $d(x, y) = 1 - cos(x, y)$ – преобразование косинусного расстояния из отрезка $[-1; 1]$ в $[0; 1]$ [7].

В процессе обучения применялась техника частичной заморозки слоев (layer freezing) энкодера. А именно замораживались нижние слои, отвечающих за базовые синтаксические и морфологические признаки. Градиенты для этих слоев не вычисляются, и параметры остаются неизменными. Заморозка нижних слоев позволяет сохранить универсальные языковые представления, одновременно адаптируя верхние слои к целевой задаче. Такой подход снижает риск забывания моделью данных и уменьшает количество обновляемых параметров.

Обучение проводилось с использованием трех различных этапов выбора негативных примеров. На первом этапе осуществляется обучение на простых негативных примерах. На этом этапе установлено значение $margin = 0,3$. Стратегия формирования негативных примеров основывается на применении циклического сдвига позитивных примеров. Основной задачей этого этапа являлась начальная стабилизация модели. Умеренно сложными негативные примерами использовались на втором этапе обучения. Значение $margin$ устанавливалось равным $0,2$ для данного этапа. Негативные примеры выбирались алгоритмом, согласно которому для каждого якорного позитивного вектора производится поиск таких негативных примеров, косинусное расстояние до которых находилось в интервале между значением расстояния в позитивной паре за вычетом $\delta = 0,1$ и самим значением расстояния в позитивной паре: $d(a, p) - \delta < d(a, n) < d(a, p)$. А третий этап посвящен работе со сложными негативными примерами. Параметр $margin$ сохраняется такими же, как на предыдущем этапе. Стратегия формирования негативных примеров основывается на алгоритме, который для каждого якорного позитивного вектора осуществляет поиск топ-10 наиболее семантически близких примеров с другими полномочиями органов власти. Для обеспечения разнообразия обучающих примеров между эпохами применяется механизм циклического выбора. Использование поэтапного обучения с постепенным переходом от простых к сложным примерам обеспечивает устойчивую сходимость процесса обучения и позволяет достичь высоких показателей точности семантического сопоставления [4,6,7]. Конкретная настройка для каждого этапа обучения представлена в табл. 1.

**Таблица 1**

Настройка обучения по этапам

| Количество эпох | Этап обучения | Количество размороженных слоев энкодера | Обучается ли проекционная голова | Скорость обучения |
|---|---|---|---|---|
| 5 | Легкие негативы | 0/24 | Да | $1,5 \cdot 10^{-5}$ |
| 7 | Легкие негативы | 4/24 | Нет | $2 \cdot 10^{-5}$ |
| 10 | Легкие негативы | 5/24 | Да | $1,5 \cdot 10^{-5}$ |
| 15 | Умеренно сложные негативы | 9/24 | Нет | $0,7 \cdot 10^{-5}$ |
| 5 | Умеренно сложные негативы | 9/24 | Да | $0,7 \cdot 10^{-5}$ |
| 10 | Сложные негативы | 12/24 | Нет | $0,7 \cdot 10^{-5}$ |

Процесс обучения модели представлен в виде схемы алгоритма на рис. 4. На первом этапе выполняется загрузка датасета и инициализация модели. Перед каждым циклом обучения задается этап обучения, количество эпох для данного этапа и количество слоев энкодера, которые будут обучатся. В начале все параметры модели включая слои базовой модели и проекционную голову замораживаются, так их веса не будут обновляться при обучение. Далее размораживается указанное для этого этапа количество слоев базового энкодера, которые будут обучатся, а также, на некоторых этапах размораживаются параметры проекционной головы. После происходит инициализация оптимизатора, который будет использоваться для обновления весов модели на протяжении всех эпох данного этапа. И

запускается цикл по эпохам, в ходе которого создаются негативные примеры и происходит обучение эпохи и валидация на ней. Такой процесс повторяется для всех заданных этапов обучения, после чего обученная модель сохраняется для последующего использования в сервисе [8].

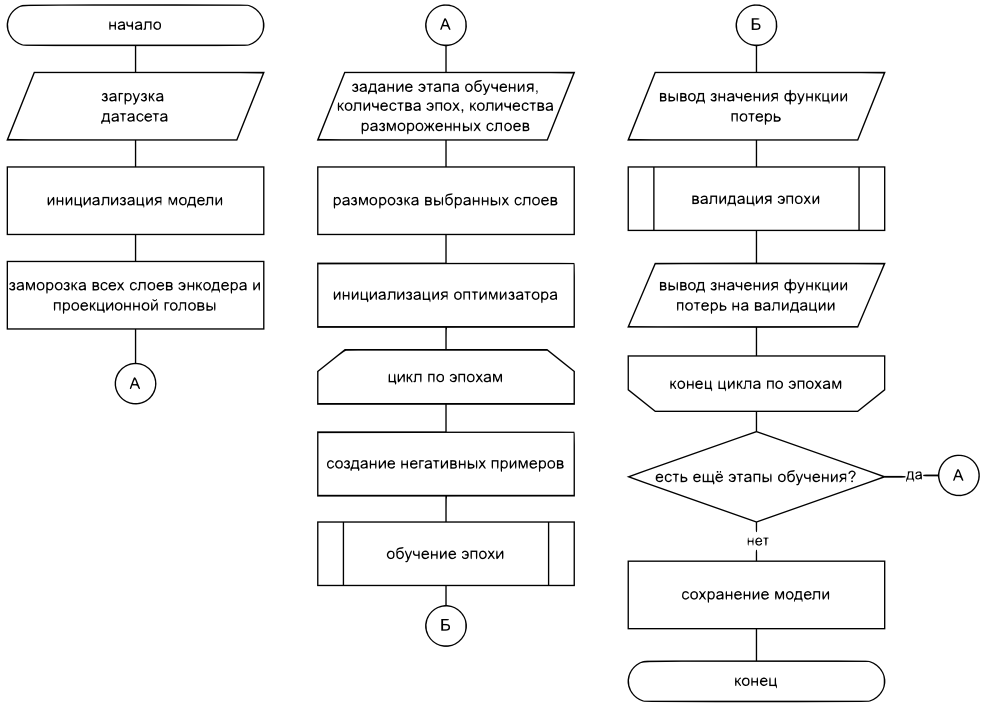


**Рис. 4**. Схема алгоритма обучения модели векторизации

## 4. Результаты

Проведенное тестирование выявило значительное усовершенствование исходной модели для решения специфической задачи. Модель стала менее склонна к выбору нерелевантных пар из функциональных предложений, что привело к более точному ранжированию. В процессе валидации отслеживались следующие показатели: recall@k – доля корректных примеров в числе k лучших; MRR – среднее значение обратных рангов правильных ответов; а также медианная и средняя позиция соответственного ответа (наилучшим считается значение, равное 1, для всех метрик). В таблице 2 представлены данные по метрикам для обучающей выборки, а для валидационной – в таблице 3. Измерения проводились до начала процесса обучения и после завершения каждого этапа. Этапы обучения использовали простых негативные примеры, средней сложности и сложные соответственно в три этапа.

**Таблица 2**

Изменение метрик на обучающей выборки

| Метрика | До | Этап 1 | Этап 2 | Этап 3 |
|---|---|---|---|---|
| Recall@1 | 0,166 | 0,555 | 0,959 | 0,993 |
| Recall@10 | 0,527 | 0,963 | 0,996 | 0,999 |
| MRR | 0,281 | 0,696 | 0,973 | 0,996 |
| MeanRang | 36,820 | 2,800 | 1,280 | 1,060 |
| MedianRang | 9,000 | 1,000 | 1,000 | 1,000 |

Основной целью дообучения было привести векторное пространство к особенностям текстов обращений граждан. На обучающей выборке метрики уверенно росли, что говорит об успешной адаптации. Некоторое снижение результатов на тестовой выборке связано с разницей в распределении данных. Тем не менее, достигнутый уровень точности достаточно высокий для работы ранжирующего алгоритма. Сравнивая метрики до и после обучения,

видно значительное улучшение качества модели. Этот результат можно считать хорошим для задачи семантического сопоставления текстов обращений граждан, учитывая сложность темы и разнообразие входных данных.

**Таблица 3**

Изменение метрик на тестовой выборки

| Метрика | До | Этап 1 | Этап 2 | Этап 3 |
|---|---|---|---|---|
| Recall@1 | 0,155 | 0,458 | 0,626 | 0,650 |
| Recall@10 | 0,484 | 0,894 | 0,882 | 0,880 |
| MRR | 0,263 | 0,607 | 0,719 | 0,739 |
| MeanRang | 38,400 | 5,960 | 10,030 | 9,800 |
| MedianRang | 12,000 | 2,000 | 1,000 | 1,000 |

В качестве примера работы всего алгоритма в целом был загружен текст: «Прошу провести проверку и принять меры по улучшению качества дорожного ремонта на улице Ленина. После проведенного ремонта на данном участке появились ямы и выбоины, что создает угрозу безопасности движения и затрудняет передвижение транспортных средств и пешеходов. Прошу организовать повторный осмотр дороги и устранить выявленные недостатки в кратчайшие сроки. В случае необходимости прошу сообщить о планируемых действиях и сроках их выполнения». На что система определила обращение в «Министерство дорожного хозяйства и транспорта», с баллом 1,502. Система сопоставила первое и третье предложения данного обращения с полномочием: «Содержание и ремонт дорог областного значения, включая трассы и межмуниципальные», со схожестью 0,867 и 0,854 соответственно.

Еще одним примером тестирования был текст содержащий сразу несколько главных проблем: «Добрый день. Обращается к вам жительница дома № 5 улицы красной г.Челябинск. С ноября 2024 рядом с нашим домом была организована незаконная парковка автомобилей, которая не огорожена и вообще ужасная. Машины занимают часть тротуара, чем мешают пешеходам. Плюс к этому во время прогрева автомобилей под окнами стоит ужасная вонь выхлопных газов, невозможно дышать. Просьба разобраться в законности парковки». Здесь первые два и последнее предложения были отброшены на этапе фильтрации, как не содержащие важной информации. А по итогом работы алгоритма четвертое предложение сопоставилось с полномочием: «Проведение мероприятий по модернизации и развитию транспортных систем», а пятое с полномочием: «Комплексное снижение выбросов загрязняющих веществ в атмосферный воздух в крупных промышленных центрах России, с целью уменьшения уровня загрязнения и улучшения качества воздуха». По этим сравнениям система поставила министерство дорожного хозяйства и транспорта на первое место, а министерство экологии на второе. Из такого примера видно, что такой алгоритм позволяет работать с текстами содержащими сразу несколько ключевых проблем.

Также алгоритм тестировалась на более абстрактных и эмоциональных текстах. Например фраза из обращения: «Управляющая компания – уютный дом называется, смешно даже – отвечает одно и то же: Нет денег в фонде, В плане на следующий год, Подождите», была сопоставлена с полномочием: «Проверка правильности начисления и оплаты услуг ЖКХ», со схожестью 0,619. Фраза «Проблема в том, что с нами никто не считается, хотя мы тоже люди, платим за квартиру исправно», сопоставлена с полномочием «Консультирование граждан по вопросам жилищного законодательства и обеспечения жилищных прав», со схожестью 0,614. А фраза «Работаю на двух работах, внучку воспитываю, а тут еще эти проблемы – сама должна крышу чинить, что ли?», с полномочием «Контроль содержания и ремонта общедомового имущества в жилых комплексах», со схожестью 0,565. Все эти полномочия относятся к главному управлению «Государственная жилищная инспекция», выделив его на первое место по ранжированию с баллом 1,52. Это показывает, что алгоритм может работать с любыми обращениями.

## Заключение

В ходе выполнения работы была достигнута поставленная цель – разработана интеллектуальная система, обеспечивающая объясняемое ранжирование исполнителей, уполномоченных за подготовку ответов на обращения граждан. Проведенное исследование подтвердило,

что применение многоэтапной гибридной обработки текста в сочетании с дообученной нейросетевой моделью позволяет обеспечить определения ответственного органа власти и предоставить в качестве объяснение соотношение фрагментов текста и полномочий выбранного органа. Такая система может стать помощником в процессе обработке обращений, помогая эксперту принимать решение о назначение исполнителя. Использовать такой алгоритм предлагается с помощью веб-приложения, в котором в правой колонке выводятся релевантные органы власти. А слева в исходном тексте выделяются опорные фразы, на которые можно навести мышкой и увидеть полномочия с которыми эта фраза была сопоставлена.

## Литература


1. Edverest. Обработка естественного языка и нейронные сети. – URL: https://edverest.ru/unit/obrabotka-estestvennogo-yazyka-i-nejronnye-seti/ (дата обращения: 31.01.2026)
2. Шалагин, Н.Д. Обзор алгоритмов семантического поиска по текстовым документам / Н.Д. Шалагин // Международный журнал открытых информационных технологий. –2024. – Т. 12, № 9. – P. 11–20
3. Китов, В.В. Латентный семантический анализ / В.В. Китов. – URL: https://deepmachinelearning.ru/docs/Neural-networks/Embeddings/Latent-semantic-analysis (дата обращения: 11.01.2026).
4. Нефедова А.А. Модель ИИ для объяснимой маршрутизации обращений граждан в органы власти / А.А. Нефедова // Материалы II Всероссийской научно-практической конференции с международным участием. – Ханты-Мансийск: Югорский государственный университет, 2026. – С. 103.
5. Reimers, N. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks / N. Reimers, I. Gurevych // arXiv. – URL: https://arxiv.org/abs/1908.10084 (дата обращения: 02.02.2026)
6. Feng, F. Language-Agnostic BERT Sentence Embedding / F. Feng, Y. Yang // arXiv. – URL: https://arxiv.org/abs/2007.01852 (дата обращения: 30.01.2026)
7. ШАД. – URL: https://education.yandex.ru/handbook/ml (дата обращения: 11.11.2025).
8. SentenceTransformers Documentation. – URL: https://sbert.net/ (дата обращения: 21.10.2025)


Анна Александровна Нефедова, студент, кафедра «Прикладная математика и программирование», Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), anna.nefedovaa@yandex.ru.

Татьяна Васильевна Карпета, кандидат физико-математических наук, доцент, кафедра «Прикладная математика и программирование», Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), karpetatv@susu.ru.

*Поступила в редакцию 15 июня 2026 г.*

**MSC 68T50** **DOI: 10.14529/mmp260309**

## MODELING OF AN INTELLIGENT ALGORITHM FOR RANKING AUTHORIZED INSTITUTIONS IN RESPONSE TO REQUESTS

***A.A. Nefedova*[1], *T.V. Karpeta*[1]**
[1]South Ural State University, Chelyabinsk, Russian Federation
E-mail: anna.nefedovaa@yandex.ru, karpetatv@susu.ru

The presented work explores the application of semantic matching methods for vector representations of text to create an intelligent service that can rank authorized entities in the task of preparing responses to requests. The proposed system is modeled using the example of identifying performers for preparing responses to citizens' appeals to government agencies. The system is designed to not only identify relevant performers but also provide an explainable connection between the content of the appeal and the authority of the government agency, making it an effective tool for professionals. A hybrid architecture is considered, which includes a pre-trained neural network model based on the SBERT architecture. The results demonstrate a significant improvement in ranking quality: the final model achieves $MRR = 0,996$ on the training set and $0,739$ on the test set, which confirms the applicability of the developed approach for automating the process of processing citizens' appeals.

*Keywords: explainable artificial intelligence; semantic matching; ranking models; natural language processing.*

## References


1. Edverest. *Obrabotka estestvennogo yazyka i neyronnye seti* [Natural Language Processing and Neural Networks]. Available at: https://edverest.ru/unit/obrabotka-estestvennogo-yazyka-i-nejronnye-seti/ (accessed on: 31.01.2026) (in Russian)
2. Shalagin N.D. [Review of Semantic Search Algorithms for Text Documents]. *Mezhdunarodnyy zhurnal otkrytykh informatsionnykh tekhnologiy* [International Journal of Open Information Technologies], 2024, vol. 12, no. 9, pp. 11–20. Available at: https://cyberleninka.ru/article/n/obzor-algoritmov-semanticheskogo-poiska-po-tekstovym-dokumentam (accessed on: 15.02.2026) (in Russian)
3. Kitov V.V. *Latentnyy semanticheskiy analiz* [Latent Semantic Analysis]. Available at: https://deepmachinelearning.ru/docs/Neural-networks/Embeddings/Latent-semantic-analysis (accessed on: 11.01.2026) (in Russian)
4. Nefedova A.A. [Model' II dlya ob"yasnimoy marshrutizatsii obrashcheniy grazhdan v organy vlasti] [AI Model for Explainable Routing of Citizens' Appeals to Government Bodies]. Iskusstvennyy intellekt: nauchnye dostizheniya i prikladnye zadachi: Materialy II Vserossiyskoy nauchno-prakticheskoy konferentsii s mezhdunarodnym uchastiem [Artificial Intelligence: Scientific Achievements and Applied Problems: Proceedings of the II All-Russian Scientific and Practical Conference with International Participation]. Khanty-Mansiysk, Yugorskii Gosudarstvennyi Universitet, 2026, p. 103. (in Russian)
5. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. *arXiv*. Available at: https://arxiv.org/abs/1908.10084 (accessed on: 02.02.2026)
6. Feng F., Yang Y., Language-Agnostic BERT Sentence Embedding. *arXiv*. Available at: https://arxiv.org/abs/2007.01852 (accessed on: 30.01.2026)
7. *ShAD*. Available at: https://education.yandex.ru/handbook/ml (accessed on: 11.11.2025) (in Russian)
8. *SentenceTransformers Documentation*. Available at: https://sbert.net/ (accessed on: 21.10.2025)


*Received June 15, 2026*