# **Интеллектуальный модуль рекомендаций для платформы Исследователи и разработчики в сети**

# **Введение**

В рамках научной деятельности НИУ МЭИ существует большое количество различных задач, требующих научно-исследовательских, опытно-конструкторских и технологических компетенций, причём задачи могут быть из самых различных областей, которыми занимаются институты и кафедры МЭИ.

В настоящее время существуют большие сложности в подборе руководителей и исполнителей, так как сам поиск подходящих кандидатов и организация их взаимодействия требует много времени и не охватывает все возможные варианты. Более того, самые подходящие кандидаты, наилучшим образом удовлетворяющие требованиям задачи по квалификации, могут вовсе принадлежать кафедре другого профиля, не соответствующей области задачи, что очевидно затрудняет поиск. Традиционный метод поиска руководителей и исполнителей не может охватить всех возможных претендентов. Также, со стороны исполнителей существует подобная проблема: в настоящее время не существует единой базы со всеми предлагаемыми задачами. Потенциальный исполнитель может и вовсе не знать о возможности принять участие в каком-либо конкурсе, научном проекте и т.д.

Для решения данной проблемы была предложена платформа «Исследователи и разработчики в сети», призванная своевременно и максимально эффективно удовлетворять потребности запросов внешней среды (заказчиков) высококвалифицированными специалистами для выполнения задач.

Платформа позволяет размещать и хранить профили исполнителей. После размещения заказчиком задачи и настройки её критериев, система производит автоматическую выборку профилей подходящих исполнителей, исходя из размещенной задачи и заданных критериев для выполнения задачи. После заключения договора между заказчиком и выбранным исполнителем, происходит формирование отзыва об итогах их взаимодействия при выполнении задачи, а также осуществляется их опрос на предмет результативности взаимодействия.

Профили исполнителей частично заполняются из базы ИС «РУР ПКР» (информационной системы "Результативность и управление рисками – программа комплексного планирования"), содержащей сведения о научных публикациях, научно-исследовательских и опытно-конструкторских работах сотрудников МЭИ. Таким образом, подбор исполнителей должен осуществляться с учётом этой информации.

Таким образом, интерес представляет модуль подбора(рекомендации) исполнителей для задачи или задач для исполнителя.

# **Постановка задачи**

Для реализации описанного модуля подбора необходимо применение методов машинного обучения. Причём важным фактором здесь является отсутствие обучающих данных, так как мы не располагаем никакой информации о том, какие исполнители были отобраны для каких задач и насколько эффективно и оправдано это было сделано. Таким образом, становится понятно, что необходимо применять методы машинного обучения без учителя.

Указанные выше сведения о НИОКТР, с помощью которых заполняется профиль исполнителей, представляют из себя записи, полученные из базы данных РУР ПКР. Записи о научных публикациях содержат поля с полным ФИО сотрудника и полем описания статьи, включающем полный список всех авторов статьи, название и библиографическое описание статьи. Усреднённый формат содержимого записи приведён ниже.

|  |  |
| --- | --- |
| Автор | Описание |
| Фамилия Имя Отчество (главного автора статьи) | И.О. Фамилия, И.О. Фамилия Полное название статьи \\ библиографическое описание |

При этом важно заметить, что формат описания статьи может значительно отличаться у разных записей. Так, фамилии авторов в поле описания статьи могут содержат сначала первые буквы имени и отчества, а затем фамилию, или наоборот, сначала фамилию, затем первые буквы. Также фамилия и вовсе может быть указана полностью. Таких различий в формате написания большое количество, начиная от простановки запятых и пробелов, и заканчивая регистром. Поэтому необходимо разработать алгоритм обработки текста описания статьи для исключения ненужной информации и приведении к виду, необходимому для реализации последующего алгоритма.

Также необходимо разработать сам модуль подбора исполнителя к задаче. Подбор следует осуществлять на основе слов, используемых в описании задачи, и слов, употребляемых автором в публикациях, научных работах и т.п. Также следует учитывать классификатор задач, представляющий из себя чётко определённую древовидную структуру классов для отнесения той или иной задачи к определённой профессиональной области (некий фильтр).

## **База данных**

Для функционирования модуля необходимы следующие таблицы:

1. **main\_table**. Таблица содержит данные о публикациях и НИОКТР и их авторах.
2. **authors.** Таблица содержит данные авторов.
3. **task\_table.** Таблица содержит список задач (запросов на подбор исполнителя)
4. **dict**. Словарь. Таблица содержит все уникальные слова из описаний статей
5. **words\_step2**. Таблица содержит связки слов из словаря и авторов, которые их применяли, а также связки “слово – задача”.
6. **jobs.** Таблица содержит задачи или статьи предназначенный для обработки (разделения на слова, очистки и т.д.)

1. **used\_tables.** Таблица содержит некоторые используемые таблицы с их ObjectID в базе данных
2. **RegularExpr.** Таблица содержит все применяемые регулярные выражения для очистки записи
3. **Numbers.** Вспомогательная таблица для разделения записи на слова. Содержит последовательно числа от 1 до 50
4. **postfixes.** Таблица содержит все возможные окончания в русском яхыке
5. **prefixes.** Таблица содержит все возможные приставки в русском языке
6. **trimmed\_t.** Таблица содержит слова, у которых обрезаны приставки и окончания
7. **relation\_t.** Таблица обеспечивает связь между таблицами dict и trimmed\_t многие ко многим
8. **author\_classifier.** Таблица содержит данные об авторах и их принадлежности к классам из классификатора
9. **classifier.** Таблица содержит классификатор. Используется древовидная структура со сквозным id.

|  |  |  |
| --- | --- | --- |
| Название поля | Тип | Назначение |
| id | Int | Уникальный идентификатор |
| word | Nvarchar(400) | Содержание уровня классификатора |
| Tree\_id | Int | Id из этой же таблицы с указателем на верхний элемент (у наивысшего элемента -1) |
| \_level | Int | Уровень классификатора |
| koef | float | Коэффициент определяющий влияние классификатора на подбор |

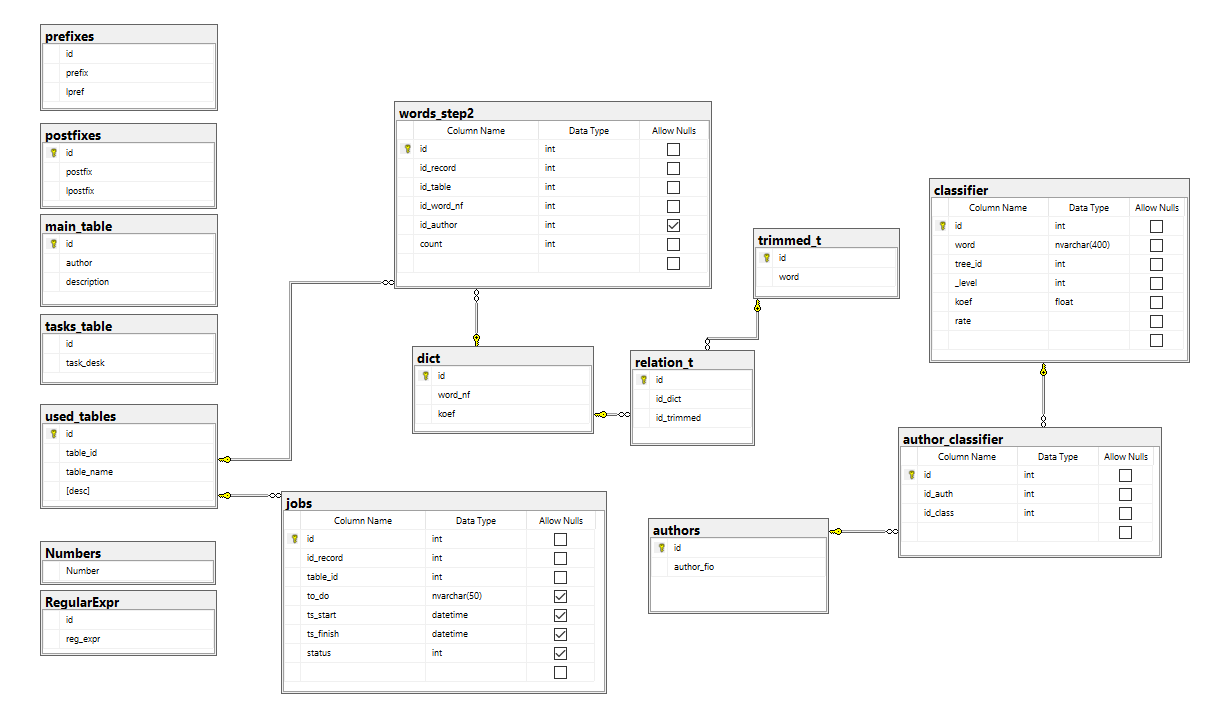


Рисунок 1. Диаграмма базы данных модуля подбора ПИРС

Набор необходимых функций, процедур и триггеров Базы данных для работы модуля:

|  |  |
| --- | --- |
| **Название** | **Описание** |
| **Функции** | |
| get\_fios2 | Используется для получения всех возможных вариантов написания ФИО авторов для их отчистки из описаний статей |
| uf\_inline\_split\_me\_pirs | Используется для разделения строки на отдельные слова по заданному разделителю |
| clear\_with\_reg\_expr | Очистка заданной строки с помощью регулярных выражений из таблицы RegularExpr |
| delete\_after\_sleshes | Удаление строки после двойных слешей в заданной строке если двойные слеши находятся во второй половине строки |
| **Процедуры** | |
| change\_koef | Изменение коэффициента слова в таблице dict (используется администратором модуля для настройки коэффициента учёта слова в рейтинге) |
| clean\_fio\_str | Очистка строки от ФИО по различным возможным шаблонам (применяя функцию get\_fios2) |
| get\_recommend | Получение списка рекомендованных авторов с процентным совпадением и рейтингом для конкретной задачи. Используется прямое сравнение слов |
| get\_recommend\_cognates | Получение списка рекомендованных авторов с процентным совпадением и рейтингом для конкретной задачи. Используется сравнение с учётом однокоренных слов |
| get\_words\_and\_count | Получить все слова и количество их применений для конкретного автора |
| **Триггеры** | |
| tr\_main\_table | После добавления новой записи в main\_table, новые данные заносятся в таблицу jobs для дальнейшей обработки |
| tr\_on\_words\_step2 | Перед добавлением в таблицу проверяется наличие слова с таким id у автора. Если такое слово уже есть, увеличивается count у этого слова |
| dict\_insert | При добавлении нового слова в словарь оно обрезается с помощью таблиц с приставками и окончаниями, добавляются новые слова в таблицу trimmed\_t, образуются новые связи в таблице relation\_t |
| **CLR функции** | |
| RegExReplace | Используется в процедуре clear\_with\_reg\_expr, заменяет совпадение по паттерну из строки на указанный символ |
| GROUP\_CONCAT\_D | Используется в get\_recommend и get\_recommend\_cognates, oобъединяет слова в одну строку |

**get\_recommend\_cognates\_classifier**. Основная функция побора исполнителей с учётом однокоренных слов и классификатора. В результате процедура заносит выборку авторов с подсчитанными рейтингами (см. ниже) в таблицу recommend\_results. Выходного значения нет.

**Общий алгоритм работы модуля**

Для обработки статей авторов используется таблица **main\_table.** В ней содержаться все научные работы авторов. Для интегрирования модуля необходимо иметь доступ к подобной таблице. Также необходимо иметь доступ к таблице с уникальными авторами.

При появлении новой записи в main\_table срабатывает триггер, заносящий данные о новой статье в таблицу jobs. Эта таблица содержит номер записи, автора и статус обработки записи. Для каждой записи из этой таблицы со статусом 0 (не обработано) будет запускаться функция обработки с помощью **заданий SQL Server Agent**. При обработке запись очищается от ненужных символов и разбивается на отдельные слова. Эти слова, если они новые, заносятся в таблицу **dict** (при добавлении слово обрезается с помощью приставок из таблиц posfixes и prefixes. Обрезанные слова заносятся в таблицу trimmed\_t). По итогу обработки статьи, у автора появляются новые слова (или увеличивается количество использований) в таблице words\_step2.

В модуль уже внедрён классификатор. В нём содержится 4 уровня, связи осуществляются с помощью древовидной структуры.

Процесс подбора рекомендаций осуществляется с помощью процедуры get\_recommend\_cognates\_classifier. Процедура получает на вход текст задачи и цепочку из классификатора. Задача очищается и разбивается на слова. Затем происходит поиск среди авторов по словам и их формам. Так же осуществляется проверка на соответствие классификатору с помощью таблицы author\_classifier. Совпадение на каждом уровне даёт определённое количество рейтинга. Наибольшее количество рейтинга прибавляет совпадение на последнем уровне. Результат задачи представляется в виде выборки со списком авторов, процентом совпадений слов (прямое совпадение), рейтинг (учитываются однокоренные, полностью совпавшие слова и коэффициенты классификатора) и рейтинг по классификатору.

**Пример подсчёта рейтинга**

Результат подбора формируется в скрипте **script\_recommend\_cognates\_classifier.sql**

**Формулировка задачи:** "Разработать систему очистки сточных вод"

**В классификаторе задачи отмечено:**

* Системы очистки бытовых и промышленных сточных вод (*level=1*)
  + Системы химической очистки (*level=2*)
    - Система озонирования сточных вод (*level=3*)
    - Система хлорирования сточных вод  (*level=3*)

**Классификатор автора 1 (ID = 2):**

* Системы очистки бытовых и промышленных сточных вод (*level=1*)
  + Системы химической очистки (*level=2*)
    - Система озонирования сточных вод (*level=3*)
    - Система хлорирования сточных вод (*level=3*)

**Классификатор автора 2 (ID = 532):**

* Системы очистки бытовых и промышленных сточных вод (*level=1*)
  + Системы химической очистки (*level=2*)
    - Система хлорирования сточных вод (*level=3*)

Таблица classifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | word | tree\_id | \_level | koef | rate (вычислимый столбец) |
| 284 | 37.00 Сбор и обработка сточных вод | -1 | 0 | 10 | 0 |
| 285 | Системы очистки бытовых и промышленных сточных вод | 284 | 1 | 10 | 10 |
| 294 | Системы химической очистки | 285 | 2 | 10 | 20 |
| 295 | Система озонирования сточных вод | 294 | 3 | 10 | 30 |
| 296 | Система хлорирования сточных вод | 294 | 3 | 10 | 30 |

Установленный администратором системы коэффициент учёта классификатора:*koef=10*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID автора | ФИО автора | **Общий рейтинг** | По анализу слов в задаче | | | **Рейтинг по классификатору** |
| Процент совпадений, % | **Рейтинг по словам** | Совпавшие слова |
| 2 | ФИО автора 1 | 90 | 0 | 0 | NULL | 90 |
| 532 | ФИО автора 2 | 70 | 12 | 10 | сточных | 60 |
| 53 | ФИО автора 3 | 19 | 31 | 19 | сточных, очистка | 0 |
| 1002 | ФИО автора 4 | 14 | 12 | 14 | сточных | 0 |

. . .

**Рейтинг по классификатору** (rate в таблице выше) высчитывается как сумма level\*koef по всем позициям классификатора автора, совпавшим с классификатором задачи

**Процент совпадений** считается как кол-во слов, совпавших у автора и формулировки задачи (с учётом  "однокоренных" слов), делённое на кол-во слов в задаче так же с учётом однокоренных. Полученное отношение переводится в проценты.

**Рейтинг по словам** высчитывается как сумма числа употреблений слов из формулировки задачи, встречающихся у автора (включая "однокоренные"), умноженных на коэффициент учёта слов.

Таблица dict

|  |  |  |
| --- | --- | --- |
| ID | word\_nf | koef |
| 1 | надёжность | 1 |
| 2 | систем | 0,1 |
| 3 | энергоблок | 2 |

<- коэффициент учёта слов

**Совпавшие слова** — это слова упомянутые в задаче и имеющиеся у автора в словаре (только напрямую совпавшие слова, без "однокоренных")

**Общий рейтинг** — это сумма Рейтинга по классификатору и Рейтинга по словам

Сортировка сначала идёт по**Общему рейтингу**, затем по **Рейтингу по классификатору** и затем по **Рейтингу по словам**

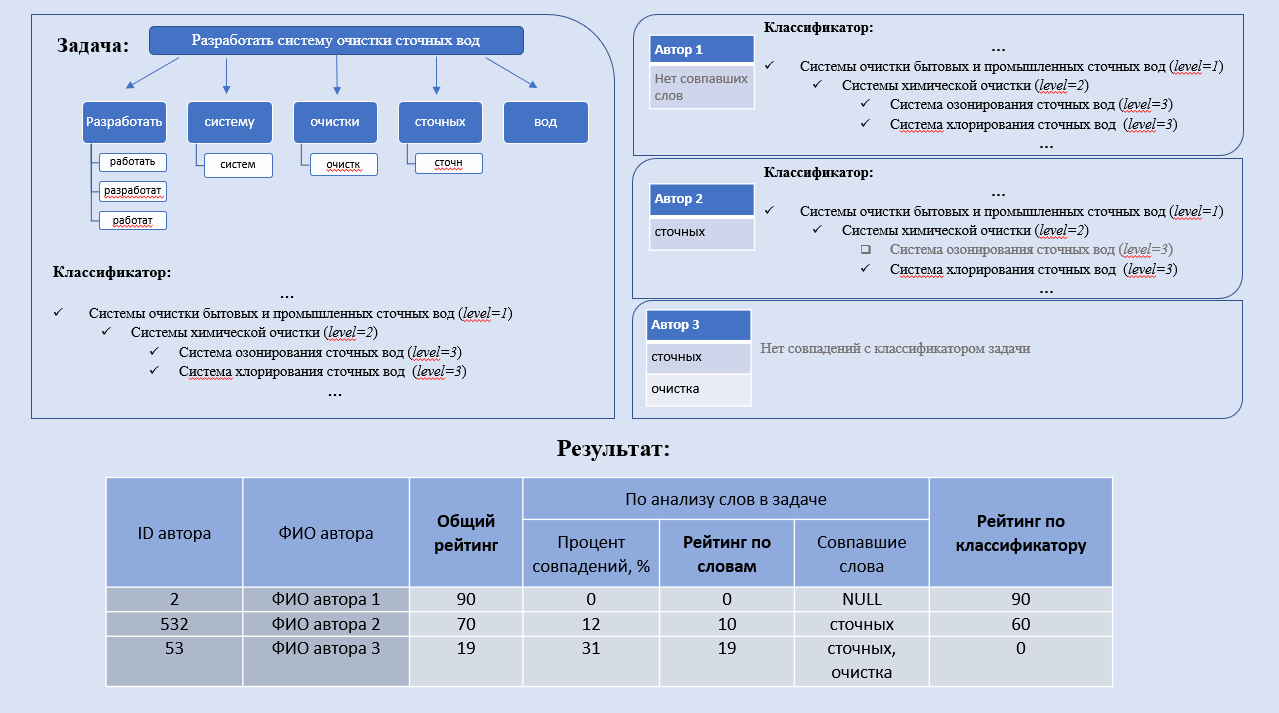


Рисунок 1. Схема подсчёта рейтинга

