ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ЛАБОРАТОРИЯ ИНФОРМАЦИОННЫХ ИССЛЕДОВАНИЙ»

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТ

по Договору № У-20/120 от 17 декабря 2020 г.

по теме:

Разработка методов понятийно-тематического анализа содержания учебных курсов и созданию модулей к программному продукту «АЛОТ», включающих в себя алгоритмы, реализующие разработанные методы

От Заказчика:	От Исполнителя:
Исполнительный директор АНО «Университет Национальной технологической инициативы 2035»	Генеральный директор ООО «Лаборатория информационных исследований»
_____ А.В. Бугаенко «_____» _____ 2021 г.	_____ Б.В. Добров «_____»_____ 2021 г.

Москва
2021

РЕФЕРАТ

Документ представляет собой отчет по работам по разработке методов понятийно-тематического анализа содержания учебных курсов и созданию модулей к программному продукту «АЛОТ», включающих в себя алгоритмы, реализующие разработанные методы, выполненных по Договору № У-20/120 от 17 декабря 2020 г.

Целью выполнения работ является выполнение пользовательских сценариев «Сравнения двух учебных курсов» и «Сравнения учебного курса и рефлексии обучающегося» путем создания методов понятийно-тематического анализа содержания учебных курсов, по которым собирается цифровой след в виде рефлексий в рамках деятельности Университета 2035,.

В качестве типового технического решения используется программный продукт Автоматизированная Лингвистическая Обработка Текстов (ПП АЛОТ), позволяющий построить модель тематического представления содержания текста на основе понятий большой лингвистической онтологии (лицензия на ПП АЛОТ была ранее приобретена Заказчиком у Исполнителя работ).

В документе описаны работы, в которых решались следующие основные задачи:

- Разработка методов понятийно-тематического анализа содержания учебных курсов, включающая в себя разработку методов и алгоритмов сравнения материалов учебных курсов, оценка “похожести” курсов, выявление ключевых образовательных результатов, которые дает курс (знания, умения, компетенции, инструменты);
- Разработка методов и алгоритмов сравнения содержания учебно-методических материалов учебных курсов с рефлексией обучающихся;
- Разработка методов и алгоритмов автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся;
- Разработка методов и алгоритмов индексирования текстов учебных курсов и текстов рефлексии обучаемых (приписывания текстам соответствующих атрибутов) по тематической таксономии АНО «Университет 2035» (Таксономии 2035);
- Разработка web-сервиса, предоставляющего интерфейс программного продукта АЛОТ с включенными в него модулями.

Документ включает 8 разделов, заключение и три приложения, выполнен на 166 листах.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Наименование работы

Выполнение работ по разработке методов понятийно-тематического анализа содержания учебных курсов и созданию модулей к программному продукту «АЛОТ», включающих в себя алгоритмы, реализующие разработанные методы.

1.2. Заказчик

Автономная некоммерческая организация «Университет Национальной технологической инициативы 2035» (далее - Университет 2035).

1.3. Исполнитель

Общество с ограниченной ответственностью «Лаборатория информационных исследований» (далее - ООО «Лаборатория информационных исследований») (включено в Единый государственный реестр юридических лиц за № 1087746774642 на основании свидетельства серии 77 № 011437611 от 24.06.2008 г.).

1.4. Обоснование

Работа выполняется по Договору № У-20/120 от 17 декабря 2020 г. (далее – Договор) , заключенному между Заказчиком и Исполнителем, в соответствии с Техническим заданием (далее – ТЗ) (Приложение № 3 к Договору).

1.5. Период оказания услуг

Начало: 17 декабря 2020 г.

Окончание: 26 февраля 2021 г.

2. ЦЕЛИ И ЗАДАЧИ ВЫПОЛНЕНИЯ РАБОТ

2.1. Целью выполнения работ является выполнение пользовательских сценариев:

- «Сравнение двух учебных курсов» - Пользователь загружает в систему учебно-методические материалы (программа, учебный план, описание) по первому курсу и по второму курсу, запускает обработку и получает отчет о результате сравнения - степень “похожести” этих курсов;
- «Сравнение учебного курса и рефлексии обучающегося» - Пользователь загружает в систему учебно-методические материалы (программа, учебный план, описание) курса, а затем наборы данных с рефлексией обучающихся. Запускает обработку и получает отчет, содержащий информацию об “усвоении” обучающимся содержания курса и выявленные сведения о ключевых образовательных результатах: знаниях, умениях, компетенциях и инструментах.

путем создания методов понятийно-тематического анализа содержания учебных курсов, по которым собирается цифровой след в виде рефлексий в рамках деятельности Университета 2035, представленных совокупностью учебно-методических материалов (программа, учебный план, описание), для решения следующих задач:

- определение совпадения и различий смыслового содержания разных учебных курсов;
- определение несовпадения смыслового содержания того, чему планирует обучить преподаватель с тем, что «усваивается» слушателем.

2.2. Для достижения указанной цели работ требуется решить следующие задачи:

- разработать методы понятийно-тематического анализа содержания учебных курсов (программа, учебный план, описание), позволяющих:
    - оценить «похожесть» курсов;
    - выявить ключевые образовательные результаты, которые дает курс (знания, умения, компетенции, инструменты);
- разработать методы сравнения содержания учебно-методических материалов учебных курсов с рефлексией обучающихся;
- создать модули к программному продукту АЛОТ (компьютерные программы) в количестве, указанном в разделе 6.3 технического задания, реализующие разработанные методы.

3. СТРУКТУРА ОТЧЕТА О ВЫПОЛНЕНИИ РАБОТЫ

Структура отчета о выполненной работе отображена в Таблице 1.

Таблица 1 – Соответствие между требованиями ТЗ и соответствующими разделами отчета

№№	Требования ТЗ	Пункты ТЗ	Раздел отчета
1	Разработка методов понятийно-тематического анализа содержания учебных курсов, включающая в себя разработку методов и алгоритмов сравнения материалов учебных курсов, оценка “похожести” курсов, выявление ключевых образовательных результатов, которые дает курс (знания, умения, компетенции, инструменты).	4.1	Раздел 4
1.1	Разработка методов и алгоритмов анализа учебно-методических материалов с точки зрения потребностей цифровой экономики, общего пространства научно-технического знания, предполагаемого объема знаний обучающихся, в том числе: - разработка методов и алгоритмов анализа отдельных учебно-методических материалов; разработка методов и алгоритмов анализа коллекции связанных учебно-методических материалов разных типов	4.1.1	Подраздел 4.1
1.1.1	Разработка методов и алгоритмов задания классификаторов для структурированного представления пространства знаний и навыков, обсуждаемых в коллекции документов, а также индексирования по заданным классификаторам.	4.1.1.1	Подраздел 4.1.1
1.1.2	Разработка методов и алгоритма представления расширенного понятийно-	4.1.1.2	Подраздел 4.1.2

№№	Требования ТЗ	Пункты ТЗ	Раздел отчета
	тематического пространства учебного курса в общем понятийно-тематическом пространстве с использованием вероятностных тематических моделей.		
1.1.3	Разработка методов и алгоритмов различного представления понятийно-тематического пространства учебного курса с использованием объектов/сущностей различной природы	4.1.1.3	Подраздел 4.1.3
1.1.4	Разработка методов и алгоритмов формирования отчетных документов для визуализации представления понятийно-тематического пространств учебных курсов.	4.1.1.4	Подраздел 4.1.4
1.2	Разработка методов и алгоритмов сравнения понятийно-тематических пространств различных учебных курсов.	4.1.2	Подраздел 4.2
1.2.1	Разработка методов и алгоритмов сравнения выявленных семантических структур содержания учебных курсов.	4.1.2.1	Подраздел 4.2.1
1.2.2	Разработка метрик, отражающих «похожесть» курсов	4.1.2.2	Подраздел 4.2.2
1.2.3	Разработка методов и алгоритмов формирования отчетных документов для визуализации в табличной форме и на графах, представлений сравнения понятийно-тематического пространств учебных курсов.	4.1.2.3	Подраздел 4.2.3
2.	Разработка методов и алгоритмов сравнения содержания учебно-методических материалов учебных курсов с рефлексией обучающихся.	4.2	Раздел 5
2.1	Разработка рекомендаций по формированию вопросников обучающихся по материалам прослушанных курсов для оптимизации	4.2.2	Подраздел 5.1

№№	Требования ТЗ	Пункты ТЗ	Раздел отчета
	процедуры автоматизации оценки степени усвоения материалов курса обучающимся.		
2.2	Разработка методов и алгоритмов представления научно-технического пространства материалов рефлексии обучаемых, в том числе расширенного, в общем пространстве знаний и навыков.	4.2.3	Подраздел 5.2
2.3	Разработка метрик, отражающих степень усвоения материалов курса обучающимся.	4.2.1	Подраздел 5.3
2.4	Разработка методов и алгоритмов сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса, в том числе расширенного, в общем пространстве знаний и навыков.	4.2.4	Подраздел 5.4
2.5	Разработка методов и алгоритмов формирования отчетных документов для визуализации в табличной форме и на графах, представлений оценки степени усвоения материалов курса обучающимся.	4.2.5	Подраздел 5.5
3	Разработка методов и алгоритмов автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся.	4.3	Раздел 6.
3.1	Разработка методов и алгоритмов сравнительного анализа лексики и терминологии: - содержимого транскриптов текстов, относящихся к тематике учебных курсов, - общего предметного поля по тематике учебного курса.	4.3.1	Подраздел 6.1
3.2	Разработка методов и алгоритмов составления ранжированного словаря	4.3.2	Подраздел 6.2

№№	Требования ТЗ	Пункты ТЗ	Раздел отчета
	лексики и терминологии общего предметного поля для автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся.		
4	Разработка методов и алгоритмов индексирования текстов учебных курсов и текстов рефлексии обучаемых (приписывания текстам соответствующих атрибутов) по тематической таксономии АНО «Университет 2035» (далее – Таксономии 2035).	4.4	Раздел 7
4.1	Разработка методов и алгоритмов интеграции структур данных Таксономии 2035 с функционалом лингвистическим обеспечением программного продукта АЛОТ.	4.4.1	Подраздел 7.1
4.2	Разработка методов и алгоритмов автоматического индексирования текстов учебных курсов по классам Таксономии 2035.	4.4.2	Подраздел 7.2
4.5	Разработка web-сервиса, предоставляющего интерфейс программного продукта АЛОТ с включенными в него модулями.	4.5	Раздел 8

4. РАЗРАБОТКА МЕТОДОВ ПОНЯТИЙНО-ТЕМАТИЧЕСКОГО АНАЛИЗА СОДЕРЖАНИЯ УЧЕБНЫХ КУРСОВ, ВКЛЮЧАЮЩАЯ В СЕБЯ РАЗРАБОТКУ МЕТОДОВ И АЛГОРИТМОВ СРАВНЕНИЯ МАТЕРИАЛОВ УЧЕБНЫХ КУРСОВ, ОЦЕНКА “ПОХОЖЕСТИ” КУРСОВ, ВЫЯВЛЕНИЕ КЛЮЧЕВЫХ ОБРАЗОВАТЕЛЬНЫХ РЕЗУЛЬТАТОВ, КОТОРЫЕ ДАЕТ КУРС (ЗНАНИЯ, УМЕНИЯ, КОМПЕТЕНЦИИ, ИНСТРУМЕНТЫ)

Общее техническое решение формируется на базе типовых научно-технических решений, разработанных ранее Исполнителем, и поставленных на основе неисключительных решений Заказчику в соответствии с Договором № У-19/118 от 23.10.2019 г.

Схема общего типового технического решения структурирования коллекции текстов предметной области с использованием лингвистических онтологий представлена на Рисунке 1.

Типовое техническое решение включает:

- Лингвистическое обеспечение в составе:
    - Комплекса лингвистических онтологий (далее также ЛО) РуТез. Разные онтологии покрывают различные совокупности предметных областей. Онтологии создаются по единым принципам и могут быть объединены в различных конфигурациях, в зависимости от предметной области решаемых задач. Лингвистические онтологии РуТез представляют собой множество понятий, связанных парными отношениями в направленную сеть без циклов. К понятиям приписываются текстовые входы. Пара (понятие, текстовый вход) называется термином онтологии. Текстовые входы,

приписанные к одному понятию, являются между собой синонимами;

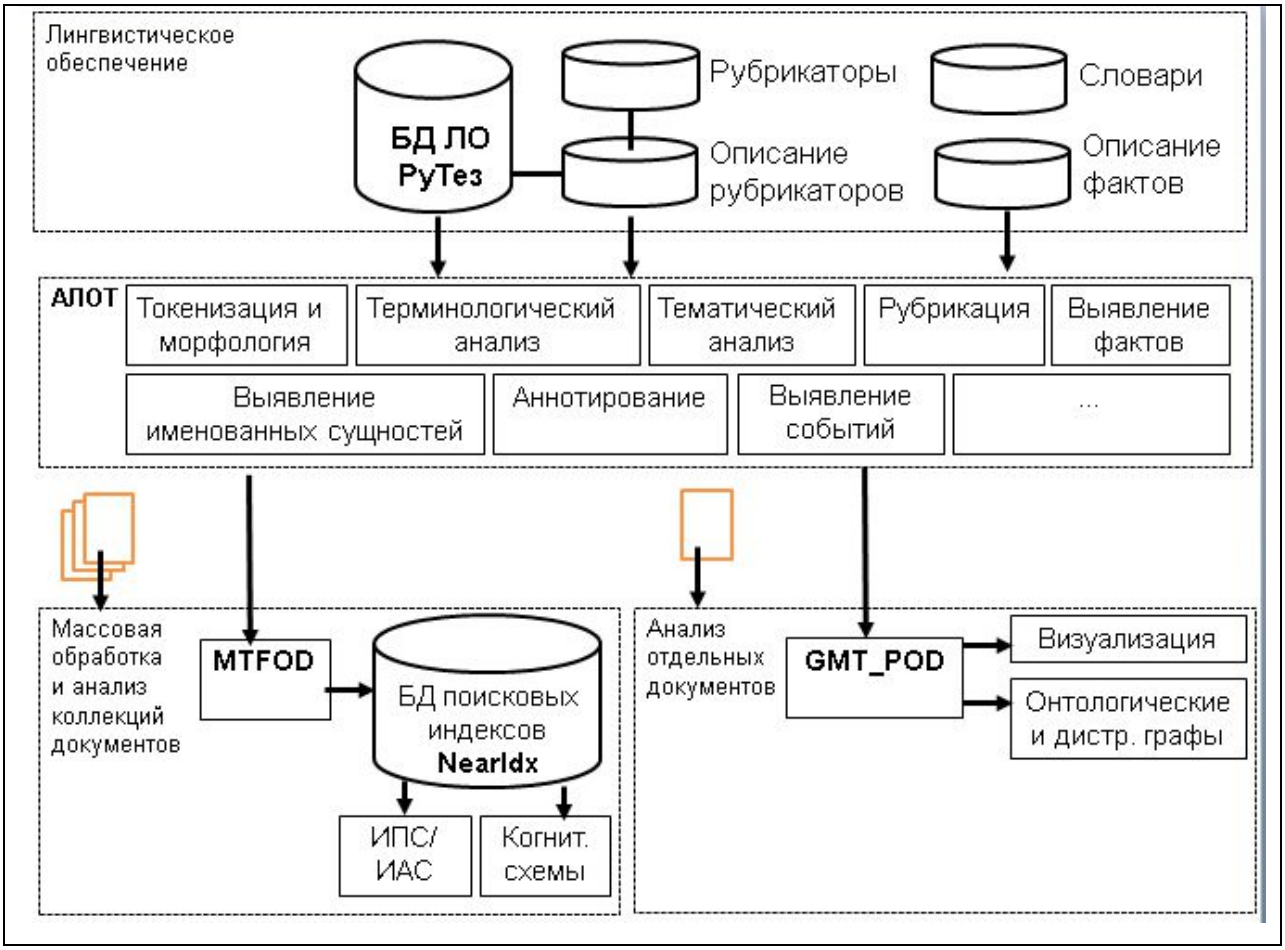


Рисунок 1 - Схема общего типового технического решения структурирования коллекции текстов предметной области с использованием лингвистических онтологи

- Комплекса рубрикаторов, предназначенных для классификации документов предметной области. Обычно совокупность рубрикаторов описывает границы и требуемую степень детализации описания предметной области;
- Комплекса описаний шаблонов фактов, на основе которых происходит выявления фактов и событий в анализируемых текстах;

  - Иных, требуемых при автоматической обработке текстов словарей;
- Программного продукта АЛОТ (Автоматизированная Лингвистическая Обработка Текстов), который осуществляет различные виды анализа текстов:
  - Токенизацию (графематический анализ), разбивающую текст на элементарные единицы – словоформы, ссылки, предложения, абзацы и т.п.;
  - Морфологический анализ – осуществляет приведение словоформ к нормальной (словарной) форме;
  - Терминологический анализ – производит выявление в тексте терминов онтологий, включая при необходимости разрешение многозначности значений, а также, при необходимости, выявление терминоподобных слов и словосочетаний;
  - Тематический анализ – формирует тематическое представление содержания текста путем выявления основных и локальных тем, все остальные понятия считаются «упоминавшимися». В результате в зависимости от места в тематическом представлении текста кааждое понятие получает оценку значимости относительно содержания текста;
  - На основе тематического представления содержания текста производится классификация и аннотирование текста;
  - Также осуществляется выделение именованных сущностей, фактов, событий и т.п. информации;

Реализовано две технологии использования результатов АЛОТ:

  - Визуальный анализ отдельных документов – приложение GMT_POD, предназначенное для детального анализа результатов обработки конкретного документа. Приложение реализует отображение результатов обработки в веб-

интерфейсе с избирательной, управляемой пользователем, подсветкой тех или иных результатов обработки документа. Также порождает графовую форму представления результатов - .gexf файл, который потом может быть визуализирован с использованием специально приложения визуализации и анализа графов GraphView. Реализовано в виде веб-сервиса;

- Массовая обработка и анализ (больших) коллекций документов – приложения mtfod и NearIdx. Приложение mtfod преднозначено для обработки коллекций докмуентов и формирования результатов обработки в виде записей (.nld файлы), единообразных для разных типов изслекаемой при анализе информации. Файлы .nld загружаются в поисковый индекс noSQL базы данных и поисковой машины NearIdx, которая предоставляет API в виде веб-сервиса для поиска информации в накапливаемой базе данных. Обычно предусматривается разработка специализированного пользовательского веб-интерфейса для доступа к информации путем выполнения запросов к NearIdx.

Общая схема технического решения выполнения работы представлена на Рисунке 2, где представлена общая схема модификации функционала ПП АЛОТ, которая касается следующих функций:

- интеграция в программную оболочку ведения онтологий ПП АЛОТ описания рубрик классификатора «Таксономия 2035» (программные модули № 18 и № 19);
- разработка дополнительных модулей обработки содержания текстов, включая:
  - разработка дополнительных модулей тематической классификации (модули №№ 01-05);

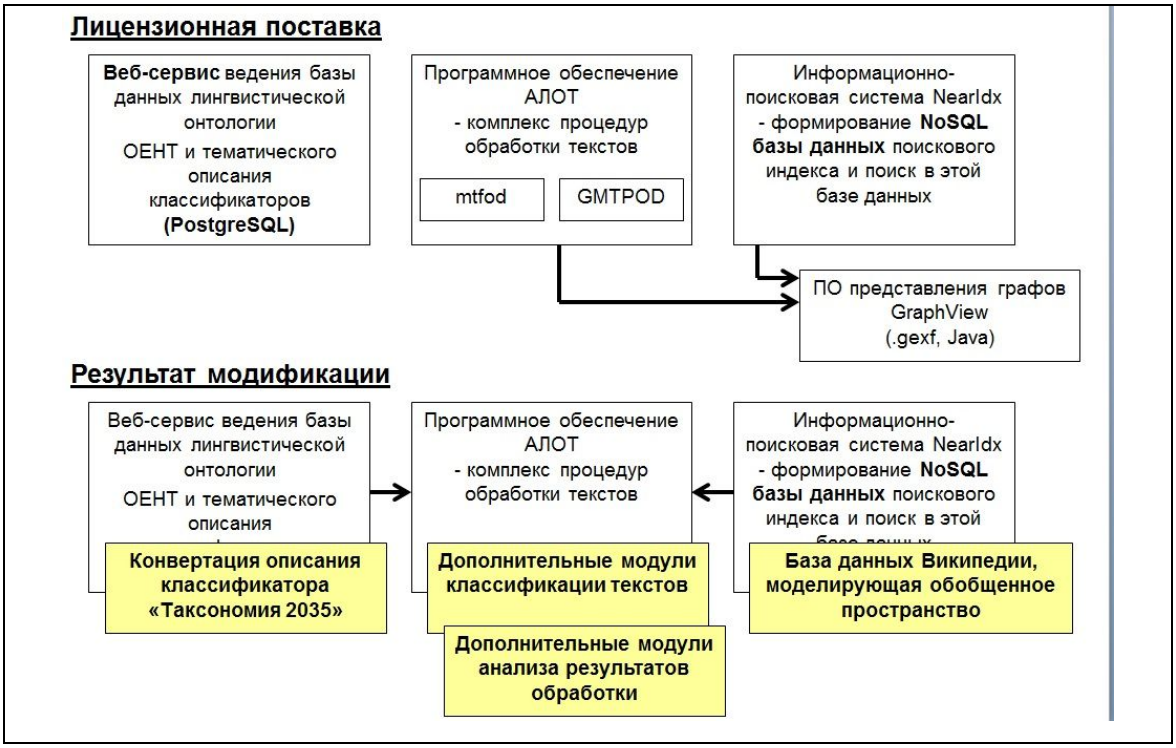


Рисунок 2 - Общая логика работы АЛОТ и его модификации

- дополнительных модулей анализа содержимого отдельных текстов и сравнения результатов обработки разных текстов (разных учебных курсов, учебного курса и рефлексии обучающихся), включая анализ содержимого с использованием знаний, извлекаемых из обобщенного пространства, моделируемого текстовой коллекцией Википедии (модули №№ 06-10, 11, 13-15), включая модули для улучшения фиксации рефлексии (модули №№ 12, 16-17).

4.1. Разработка методов и алгоритмов анализа учебно-методических материалов с точки зрения потребностей цифровой экономики, общего пространства научно-технического знания, предполагаемого объема знаний обучающихся, в том числе: разработка методов и алгоритмов анализа отдельных учебно-методических материалов; разработка методов и алгоритмов анализа коллекции связанных учебно-методических материалов разных типов

ПП АЛОТ при анализе текста формирует «тематическое представление содержания текста».

Описанные в онтологии иерархии понятий позволяют сгруппировать выявленную в тексте сеть понятий онтологии в *тематические линии* (Рисунок 3).

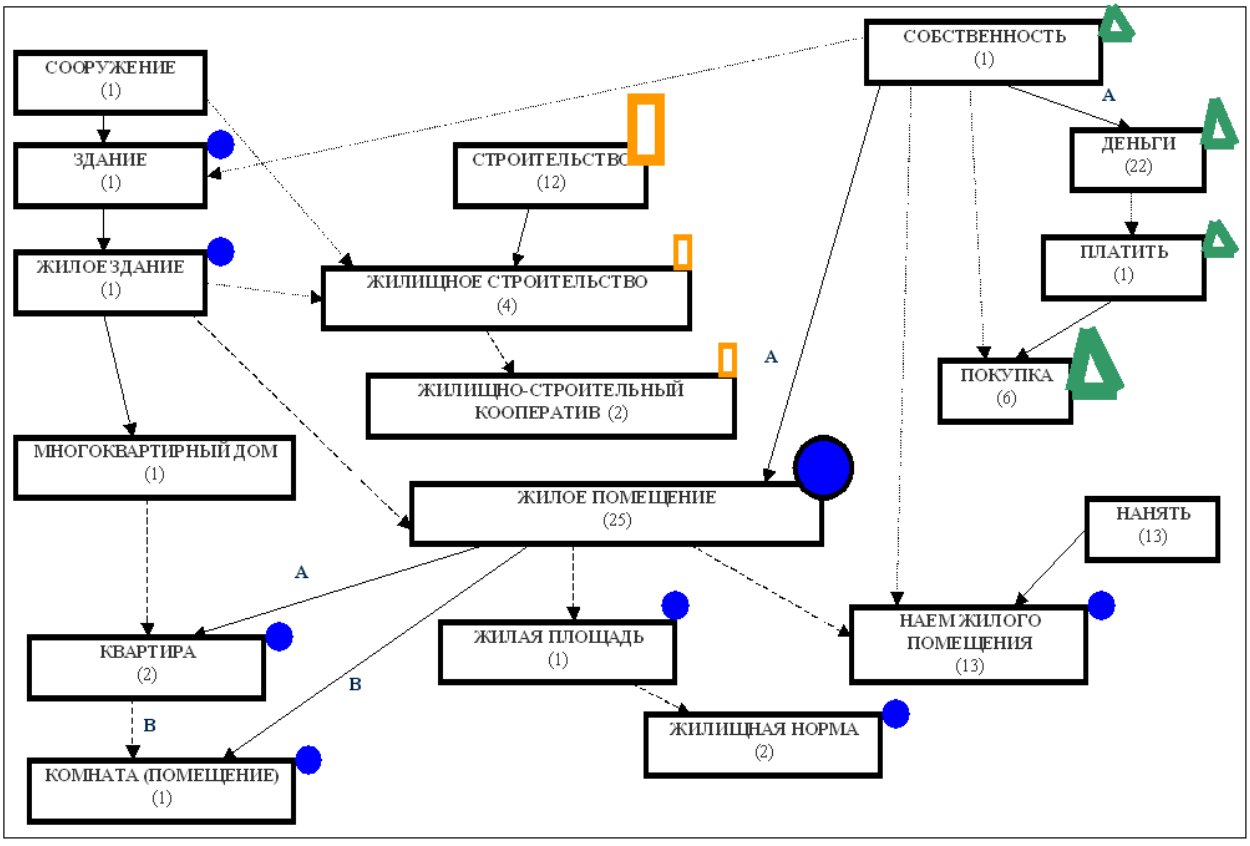


Рисунок 3 – Пример фрагмента тематических линий текста

Термин, который наиболее точно характеризует развиваемую в тексте тему и который соответственно может стать тематическим центром одного из тематических узлов текста, обычно некоторым образом выделяется в пространстве всех тематически близких терминов, а именно: такой термин может быть употреблен в заголовке и/или в начале текста, иметь максимальную частотность среди других тематически близких терминов. Тематическим центром может стать любой термин онтологии, независимо от уровня его общности/специфичности.

В процессе обработки текста создание тематического узла начинается с выбора главного понятия тематического узла. Сначала тематические узлы собираются вокруг понятий заголовка и первого предложения текста. Затем тематические узлы собираются для остальных понятий, начиная с самых частотных. Те понятия, которые уже попали в тематический узел некоторого понятия, свой тематический узел не образуют.

Предполагается, что понятия основных тематических узлов постоянно встречаются рядом друг с другом (связаны по тексту).

Чтобы оценить связанность между понятиями в тексте, вводится понятие “текстовая связь’: данное понятие считается связанным по тексту с теми понятиями, которые находятся на расстоянии не более n понятий от очередного вхождения данного понятия безотносительно к порядку следования понятий в тексте. В настоящее время n=2 и, таким образом, каждое вхождение понятия рассматривается в последовательности понятий, имеющем длину 7, тем самым мы предполагаем, что в тексте чаще всего подряд встречается не более семи не связанных между собой по Тезаурусу понятий. Связанность между понятиями уменьшается, если между их вхождениями находятся знаки абзацев, подзаголовки и т.п. В результате для каждого понятия текста получается совокупность текстовых связей.

После того как созданы тематические узлы, текстовые связи понятий каждого тематического узла суммируются и определяются текстовые связи между тематическими узлами.

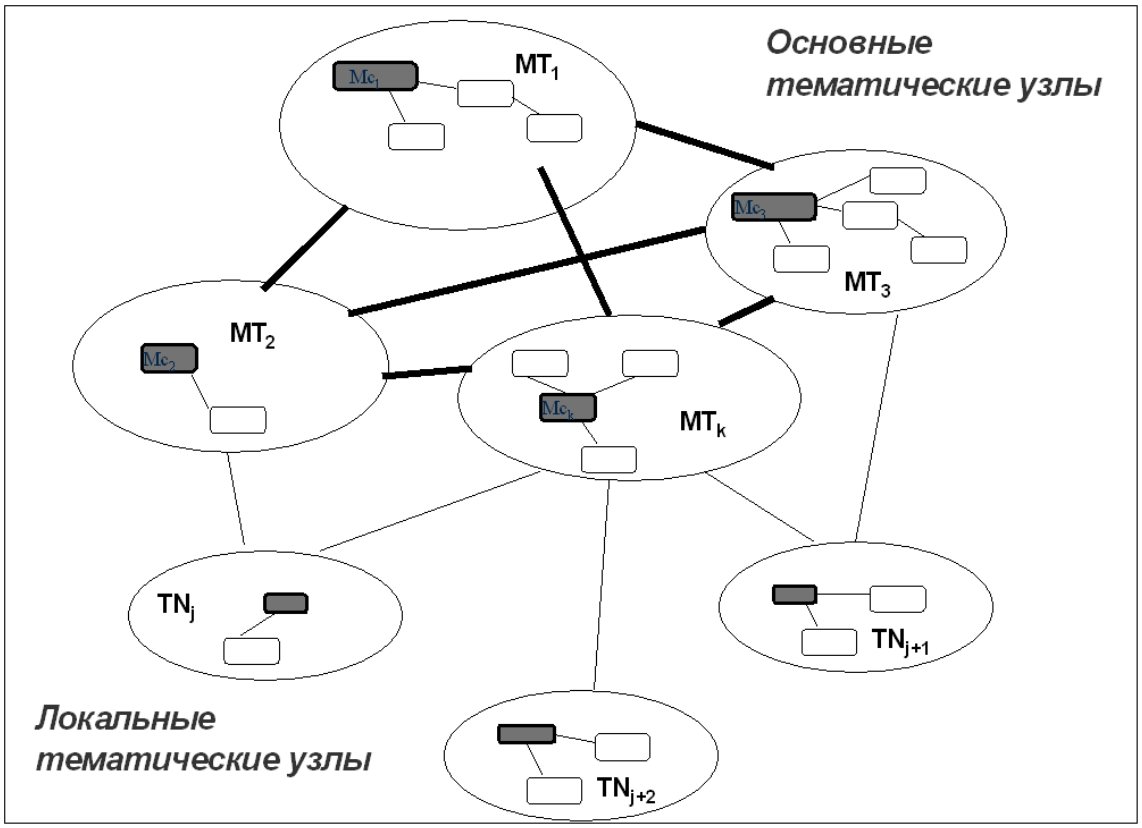


Рисунок 4 - Общая структура тематического представления

В соответствии с моделью предполагается (Рисунок 4), что основными тематическими узлами в первую очередь являются такие тематические узлы, которые:

- все связаны между собой текстовыми связями;
- сумма частот текстовых связей между ними максимальна.

Локальные тематические узлы представляют собой некоторые важные характеристики основных тематических узлов. Тематический узел считается локальным, если этот узел имеет текстовую связь с частотностью большей единицы с одним из основных тематических узлов. Понятия, не вошедшие в

состав основных и локальных тематических узлов, объявляются "упоминавшимися" в тексте.

В информационных системах часто бывает удобно представить смысл содержания документа через ограниченное множество рубрик тематического классификатора.

В ПП АЛОТ при создании образа рубрики каждая рубрика C описывается дизъюнкцией альтернатив, каждый дизъюнкт D_i представляет собой конъюнкцию:

$$R = \bigcup_i D_i = \bigcup_i \left[\bigcap_j K_{ij} \right] = \bigcup_i \left[\bigcap_j \left(\bigcup_k d_{ijk} \right) \right] \qquad (*)$$

Конъюнкты $K_{i,j}$ в свою очередь описываются экспертами с помощью так называемых «опорных» понятий онтологии $d_{i,j,k}$. Для каждого опорного понятия задается правило его расширения $f(\cdot)$, определяющее, каким образом вместе с опорным понятием учитывать подчиненные ему по иерархии понятия. Выделяются несколько способов расширения: без расширения, полное расширение по дереву иерархии онтологии, расширение только по родовидовым связям, расширение по всем связям по иерархии вниз на один шаг.

Опорный концепт может быть как «положительным», который добавляет нижерасположенные понятия в описание конъюнкта, так и «отрицательным», который вырезает свои подчиненные понятия. Последовательность учета положительных и отрицательных опорных понятий регулируется заданием специального атрибута. Результатом применения расширения опорных понятий является совокупность понятий онтологии, полностью описывающая конъюнкт.

Следует подчеркнуть, что в данной методологии достаточно хранить только опорные понятия, полное же описание рубрики может быть каждый раз пересчитано заново при изменении состава онтологии.

Типичные цифры о параметрах описания: на одну рубрику рубрикатора в среднем приходится 1-2 дизъюнкта, 2-3 конъюнкта, 10-20 опорных понятия («положительных» и «отрицательных»), 200-400 понятий полного описания, то есть 400-800 текстовых входов.

Типовое технологическое решение с использованием ПП АЛОТ подразумевает некоторую настройку на тексты предметной области путем пополнения и уточнения понятий и отношений используемой онтологии. При этом основным критерием является покрытие наиболее частотных понятий в текстовом корпусе предметной области, что минимизирует среднюю ошибку решаемых задач обработки текстов.

В случае необходимости обрабатывать тексты разных предметных областей, а также в случае появления необходимости оперативного ввода новых рубрик, следует использовать гибридные методы представления содержания анализируемых текстов:

- с использованием не тематической лексики, которая не включается в состав онтологии, путем описания смысла рубрик в виде запросов (на языке запросов к поисковой машине);
- с использованием методов машинного обучения, задавая множество положительных и отрицательных примеров;
- с использованием вероятностного тематического моделирования.

4.1.1. Разработка методов и алгоритмов задания классификаторов для структурированного представления пространства знаний и навыков, обсуждаемых в коллекции документов, а также индексирования по заданным классификаторам

4.1.1.1. Методы представления понятийно-тематического пространства учебного курса с использованием объектов/сущностей различной природы

4.1.1.1a. Типовое техническое решение в рамках комплекса ПП АЛОТ включает информационно-поисковую систему NearIdx.

Для всех обрабатываемых текстов порождаются (с использованием программного обеспечения mtfod) файлы типа .nld, которые содержат всю извлекаемую информацию в табличной форме:

- тип объекта (наименование классификатора);
- наименование объекта;
- (опционально) идентификатор объекта, если используется закнутый классификатор;
- частотность объекта в тексте;
- вес (оценка значимости объекта для содержания текста);
- привязка объекта к абзацам, предложениям и словопозициям анализируемого текста.

Также включается таблица соответствия словопозиций байтовым сдвигам с начала текста.

Объекты, отнесенные ко всему тексту, ассоциируются с нулевой словопозицией.

В информационно-поисковой системе эти данные хранятся в виде поисковых индексов:

- обратного – от объекта к документу;
- прямого – от документа к объекту.

Поисковые запросы (язык поисковых запросов NearIdx приведен в Приложении А) применяются к обратному поисковому индексу.

В рамках текущих работ была выделена модификация к ПП АЛОТ в виде реализации исполнения запроса в формате поисковой машины NearIdx непосредственно к структурам данным в формате .nld файлов.

Это позволяет, в частности, организовать рубрицирование анализируемых документов по описанию рубрик классификаторв «Таксономия 2035» в виде запросов, которые преобразованы в формат ИПС NearIdx.

4.1.1.1б. Альтерантивным способом является задание описания динамических классификаторов (формируемых на основе содержания коллекции документов) при задании положительных и отрицательных примеров отнесения к рубрикам, а также метод и алгоритм индексирования текстов документов учебного курса по заданным классификаторам в качестве входных данных.

Этот способ может использоваться в случае, когда экспертам предметной области трудно сформулировать явные правила, объясняющие отнесение текста документа (фрагмента документа) к той или иной рубрике. Но эксперт может оценить документ «в целом», отнеся или не отнеся его к той или иной рубрике.

В этой постановке задача является классической задачей машинного обучения – классификации текстов «с учителем».

При этом текст документа рассматривается в виде вектора составляющих его объектов, при этом требуется построить отделяющую поверхность, разделяющую положительные и отрицательные примеры.

В рамках текущей работы в качестве таких методов были выбраны методы логистической регрессии и бустинга, как хорошо зарекомендовавшие себя в классе линейных статистических методов машинного обучения для решения задачи классификации.

В основе алгоритма лежит использование множества классификаторов для предсказания вероятности наличия каждой рубрики в документе. Из исходных данных извлекаются все леммы и термины и рассматриваются как отдельные признаки с tf-idf весами. В случае наличия новых слов или терминов в тестовых данных, они будут проигнорированы, так как отсутствовали в тестовом множестве.

В качестве классификаторов могут выступать:

- Логистическая регрессия --model_type logreg
- “Бустинг” над деревьями решений --model_type xgboost

Логистическая регрессия применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков. Математически логистическая регрессия представляет собой скалярное произведение обученных весов с вектором входных признаков с последующим применением логистической функции над результатом скалярного произведения. Во время обучения, методом максимизации правдоподобия происходит подбор вектора весов для признаков таким образом, чтобы минимизировать функцию потерь на обучающей выборке.

Бустинг – это способ последовательного построения ансамбля моделей. Деревья решений – алгоритм машинного обучения, который представляет решающие правила в иерархической структуре, состоящей из элементов двух типов — узлов и листьев. В узлах находятся решающие правила, и производится проверка соответствия примеров этому правилу по какому-

либо признаку обучающего множества. Лист представляет собой конечный узел дерева, на основании которого выбирается метка класса.

Особенностью текущей работы является использование в качестве элементов просторанства не только слов (что является стандартным), но и выявленных ПП АЛОТ понятий онтологии.

4.1.1.2. Алгоритмы, реализующие методы представления понятийно-тематического пространства учебного курса с использованием объектов/сущностей различной природы

4.1.1.2а. Алгоритм, реализующий метод классификации по статистическим классификаторам включает следующие этапы:

- На вход полается текстовый файл с материалами учебного курса или рефлексии слушателей;
- Также на вход подается текстовый файл rubr_nld_nix.lst содержащий описания рубрик для поисковика (основной сценарий использования – классификация по модифицированным описаниям рубрик классификатора «Таксономия 2035» в виде поисковых запросов, формируемый последовательным применением модулей №№ 18 и 19);
- Результатом является список рубрик с указанием их «веса» - оценки релевантности, которая формируется как оценка релевантности поискового запроса в информационно-поисковой системе, содержащей один (входной) документ. Результат оформляется в требуемом формате.

4.1.1.2б. Алгоритм, реализующий методы машинного обучения классификации по динамически задаваемым классификаторам включает следующие этапы:

- Этап обучения (реализуется программным модулем № 02):

- Создание для каждой рубрики поддиректории на диске, в которой содержатся 2 директории \plus и \minus, а также файл rublic.txt, содержащий в первой строке имя рубрики;
- Наполнение поддиректорий \plus и \minus, соответственно положительными (релевантными смыслу рубрике) и отрицательными (нерелевантными) примерами результатов обработки текстовых файлов в формате файлом .nld;
- Запуск процедуры обучения (модуль №02), результатом которой является обученная модель машинного обучения, включая оценку качества машинного обучения. При этом может быть выбран конкретный методо машинного обучения – логистическая регрессия или бустинг;

- Этап классификации (реализуется модулем № 03):
  - Для выбранной модели метода классификации на вход подается результаты обработки текстового файла (материалов учебного курса или рефлексии);
  - Результатом работы является файл результата (оформляемый в требуемом формате), содержащий список рубрик с указанием «веса» - оценки степени релевантности содержимого текстового файла смыслу рубрики.

4.1.1.3. Программный модуль, реализующий алгоритм индексирования текста по статистическому классификатору, рубрики которого описаны в формате запросов к ИПС NearIdx (ПП АЛОТ) (Модуль №01)

Назначение: Программный модуль, реализующий алгоритм индексирования текста по статистическому классификатору, рубрики которого описаны в формате запросов к ИПС NearIdx (ПП АЛОТ).

Входные данные. Внутри модуля находится функция rubr_nld_nix, в нее параметром передается текст NLD файла, функция берет рядом лежащий с модулем файл rubr_nld_nix.lst содержащий описания рубрик для поисковика.

Выходные данные. Функция возвращает объект в формате json, содержащий текстовый блок, а так же разобранный список найденных рубрик с весами.

Имя файла исходным кодом: m01_rubr_nld_nix.py

Проверка функционирования

Пакетный файл: __test_mod01.bat

Директория с тестовыми данными __test01

API: Вызов функции **rubr_nld_nix:**

```
C:\Python38\python.exe
     -u m01_rubr_nld_nix.py
     --nld __test01\nld.nld
     --out  __test01\res.res
```

где

--nld - имя файла nld

--out – имя файла с результатом

4.1.1.4. Программный модуль, реализующий алгоритм обучения модели машинного обучения динамических классификаторов, формируемых на основе содержания коллекции документов при задании положительных и отрицательных примеров отнесения к рубрикам (Модуль №02)

Назначение: Программный модуль, реализующий алгоритм обучения модели машинного обучения динамических классификаторов, формируемых

на основе содержания коллекции документов при задании положительных и отрицательных примеров отнесения к рубрикам.

Входные данные: Входными данными для модуля являются результаты обработки ПП АЛОТ текстовых данных в формате .nld, помещенные в две директории – положительных (релевантных рубрике) и отрицательных (не релевантных) примеров. Каждая рубрика представляется директорией, в которой содержатся 2 директории plus и minus, а также файл rublic.txt, содержащий в первой строке имя рубрики. Примеры входных данных лежат в директории rubr_2035_train/data.

Выходные данные: Выходными данными для модуля является обученная модель машинного обучения, включая оценка качества машинного обучения.

Примеры выходных данных (моделей), обученных над **rubr_2035_train/data**, лежат в **rubr_2035_train/text_classification**:

- model_lr.bin – модель на основе логистической регрессии.
- model_xgb.bin – модель на основе “бустинга” над деревьями решений.

API: Вызов исполняемого файл train.py.

```
train.py [-h] --input_dir INPUT_DIR
              --output_dir OUTPUT_DIR
              --model_name_or_path
                MODEL_NAME_OR_PATH
              [--logger_path LOGGER_PATH]
              [--stop_words_path STOP_WORDS_PATH]
              [--model_type MODEL_TYPE]
              [--test_size TEST_SIZE]
```

Необязательные аргументы:

- --input_dir Директория с входными данными, представленными набором директорий

- --output_dir Директория, в которую будет сохранена модель
- --model_name_or_path Имя модели
- --logger_path Путь до файла для логгирования (по умолчанию в log.txt в директорию модуля)
- --stop_words_path Путь до файла со списком стоп слов (каждая строка отдельное слово), требуется кодировка utf-8. Опционально.
- --model_type Имя модели: logreg или xgboost
- --test_size Размер валидационного датасета (создается как test_size от тренировочного), значение по умолчанию 0.2.

Содержание директорий:

- rubr_2035_train: корневая директория
  - data: директория с примерами входных данных
  - text_classification: директория с логикой модуля
  - train.py: скрипт реализующий api модуля
  - classificator.py: файл содержащий функционал, основные классы и методы модуля
  - stopwords_wiki.txt: пример вспомогательного файла со стоп словами
  - requirements.txt: файл описывающие python зависимости
  - model_lr.bin: пример обученного классификатора на основе логистической регрессии
  - model_xgb.bin: пример обученного классификатора на основе xgboost

Требования для использования модуля: Python 3.7

Способ развертывания модуля:

1) Скопировать модуль на диск

2) Установить зависимости из файла requirements.txt вызовом pip install –r requirements.txt

4.1.1.5. Программный модуль, реализующий алгоритм индексирование текста с использованием разработанных моделей машинного обучения (Модуль №03)

Назначение: Программный модуль, реализующий алгоритм индексирования текста с использованием разработанных моделей машинного обучения.

Входные данные. Входными данными для модуля являются:

- материалы учебного курса, обработанные ПП АЛОТ в формате .nld;
- обученная модель статистического рубрикатора (Модуль №2).

Примеры входных данных лежат в директории **rubr_2035_server/data** в виде трех nld файлов.

Выходные данные. Выходными данными для модуля является перечень рубрик, найденных в учебном курсе обученной моделью статистического рубрикатора.

Выходные данные возвращаются как dict объект с ключом «rbr» и значением в виде массива. Элементами массива являются рубрики в виде dict объекта, где под ключом «label» находятся имена рубрик. Под ключом «top_words» находится вектор из наиболее весомых для данной рубрики слов в исходном документе. Все это кодируется в json объект.

Примеры выходных данных лежать в директории **rubr_2035_server/data**. Они представляют собой обработанные nld файлы моделями и имею разрешения .logreg.json и .xgboost.json. Файлы имеют кодировку utf-8.

API: Модуль представляет собой сервис, взаимодействие с которым происходит через POST запрос. Запуск сервиса производится скриптом app_flask.py:

```
app_flask.py [-h] [--input_dir INPUT_DIR] [--output_dir OUTPUT_DIR]
               [--model_size MODEL_SIZE] [--min_df MIN_DF]
               [--result_model_name RESULT_MODEL_NAME]
```

```
[--wabbit_path WABBIT_PATH] [--zip_count ZIP_COUNT]
[--topic_modeling_config TOPIC_MODELING_CONFIG]
```

Необязательные аргументы:

- --input_dir Директория с входными данными, представленными набором zip архивов
- --output_dir Директория в которой будет сгенерированная модель и иные промежуточные данные
- --model_size Размер векторов модели
- --min_df Параметр отсекания слов по документной частоте снизу
- --result_model_name Итоговое имя модели
- --wabbit_path Параметр контроллируешь имя промежуточного файла, по умолчанию auto (автоматически будет выбрано имя на основе имени входной директории)
- --zip_count Количество zip файлов для обработки (выбираются случайным образом). Параметр оптионален, без указания значения будут обработаны все
- --topic_modeling_config путь до конфигурационного файла для обучения, по умолчанию config.json

4.1.2. Разработка методов и алгоритма представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве с использованием вероятностных тематических моделей

4.1.2.1. Методы представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве с использованием вероятностных тематических моделей

Вероятностное тематические моделирование относится к методам классификации текстов, в которых сам состав рубрик, так называемых, «тем» («тематик», «топиков») формируется автоматически.

В основе лежит порождающая модель, предполагающая, что каждый документ порождается некоторым количеством заранее неизвестных тематик, так что каждое слово (более общо – текстовый объект) в документе порождается на основе одной тематики.

Метод вероятностного тематического моделирования заключается в том, что по заданной текстовой коллекции и заданному количеству тематик производится решение обратной задачи представления оптимальным образом состава документов коллекции, минимизируя «перплексию» - меру «удивления» (отклонения от предсказанного) порождения текстовых объектов по сравнению с фактически наблюдаемым.

Тематическое моделирование можно представить как алгоритм мягкой кластеризации. По тренировочному датасету алгоритм пытается построить указанное количество тематик таким образом, чтобы более похожие документы группировались вместе. Математическая суть подхода в разложении матрицы «терм-документ» на две матрицы «слово-топик» и «топик-документ». Для этого применяется итерационный EM-алгоритм,

При обучении происходит подбор таких векторных представлений слов и тематик, чтобы максимизировать функцию максимизации правдоподобия на тренировочных данных. При предсказании, используя обученные матрицы «слово-топик» и «топик-документ» рассчитывается тематическое представление документа в виде вектора тем.

Вероятностные тематики могут использоваться для оценки ассоциативного сходства по достаточно далеко лексически отличающихся текстов.

В рамках текущей работы в качестве текстовой коллекции (не имея представительной коллекции материалов учебных курсов) были выбраны коллекции текстов русской Википедии, моделирующей научно-техническое содержание учебных курсов, а также коллекция документов типа «вакансии» и «резюме», моделирующая тематики восстребованности компетенций рынком.

Количество тописков может варьироваться, в проведенных вычислительных экспериментах рассматривались типовые значения 100 или 500 топиков.

В процессе подбора оптимальных параметров важным оказались следующие отличия от стандартных схем реализации вероятносного тематического моделирования:

- Использование помимо слов также словосочетаний и наименований понятий онтологии;
- Автоматическая фильтрация тематик путем удаления часто встречающихся объектов, что улучшает избирательность тематик;
- Использование специальных словарей для очистки автоматически формируемых тематик для уменьшения эффекта «шума» из-за паразитных элементов оформления, общих слов, характерных для коллекции того или иного жанра.

Отдельной задачей является выбор названия для автоматически формируемого топика, в качестве которого использовалась строка, составленная из нескольких наиболее значимых для содержания топика объектов.

Рассмотрим пример материала учебного курса:

> *Введение: Определение сферы инфорамционного поиска, задачи информационного поиска, Информационно-поисковы системы различной направленности. Архитектура информационно-поисковых систем.*
>
> *Модели информационного поиска. Оценка качества информационного поиска.*
>
> *Методы расширения информационно-поисковых запросов. Вопросно-ответные системы. Диалоговые системы.*
>
> ===
>
> *Диалоговые системы.*
>
> *Учет различных факторов. Анализ ссылок, логи запросов, анализ кликов, персонализация выдачи, поисковые сессии, словосочетания и близость расположения, тематические вероятностные модели.*
>
> *Комбинирование факторов. Модели Learning to-rank.*
>
> *Автоматическая классификация и кластеризация текстов. Типы классификации. Тематическая классификация и анализ тональности. Методы классификации. Особенности методов классификации. Методы автоматической кластеризации.*
>
> *Автоматическое аннотирование.*
>
> *Учет различных факторов. Анализ ссылок, логи запросов, анализ кликов, персонализация выдачи, поисковые сессии, словосочетания и близость расположения, тематические вероятностные модели.*
>
> *Комбинирование факторов. Модели Learning to-rank.*
>
> *Автоматическая классификация и кластеризация текстов. Типы классификации. Тематическая классификация и анализ тональности. Методы классификации. Особенности методов классификации. Методы автоматической кластеризации….*

Результаты тематического моделирования по коллекции текстов Википедии приведены на Рисунке 5.

Топики	Наименование топика	Текстовые объекты в анализируемом тексте	вес
TOPIC 0 (topic_3)	**система задача качество оценка различный модель определение сфера функция_-_член_класса запрос поиск_информации расширение_для_браузера**	система управление разработка процесс решение использование техническое_устройство технология информация задача создание стандарт	0.25
TOPIC 1 (topic_436)	**задача различный определение модель качество оценка поиск_информации функция_-_член_класса введение запрос система информационно-поисковая_система**	математический_граф ref задача являться theory множество_(понятие_математики) модель_(схема_описания) алгоритм аралия искусственный_интеллект computer mathematical	0.24
TOPIC 2 (topic_83)	**информационно-поисковая_система различный качество система сфера запрос расширение_для_браузера поиск_информации модель оценка задача определение**	услуга предпринимательство компания сеть рекламная_деятельность рынок_товаров_или_услуг ref предоставлять маркетинг продажа технология покупка_(деятельность)	0.16
TOPIC 3 (topic_312)	**определение различный оценка качество функция_-_член_класса система модель сфера задача направленность поиск_информации запрос**	термин метод использовать определение являться понятие значение определенный определять пример называть различный	0.13
TOPIC 4 (topic_301)	**запрос система различный качество функция_-_член_класса архитектура_(принцип_устройства) расширение_для_браузера определение задача поиск_информации информационно-поисковая_система модель**	компьютер коммуникационная_инфраструктура использовать сервер_(компьютер) сетевой_протокол пользователь техническое_устройство данные_(сведения) 25d0 информация допуск,_право_доступа приложение_клиент/сервер	0.08
TOPIC 5 (topic_78)	**модель архитектура_(принцип_устройства) расширение_для_браузера система качество различный задача определение направленность сфера оценка введение**	процессор_компьютера техническое_устройство ref микросхема мегагерц компьютер apple запоминающее_устройство программное_обеспечение_интел класс_(тип_данных) оперативная_память контроллер	0.04

Рисунок 5 – Пример результатов вероятностного тематического соделирования

В первом столбце приведены номер топика по значимости и отсылка на номер топика по всей коллекции. Далее наименование топика по наиболее характерным элементам. В третьем столбце соответствующие текстовые объекты в анализируемом тексте, в последнем столбце «вес» - оценка значимости топика для содержания текста.

4.1.2.2. Алгоритм формирования вероятностных тематической моделей по коллекции текстов в виде совокупности векторов тематик, в том числе с использованием лингвистических онтологий ПП АЛОТ

В основе алгоритма лежит аппарат тематического моделирования, представленный реализацией широкоизвестной библиотеки BigARTM, подготовка данных для применения которой модифицирована для использования помимо отдельных слов, также текстовых объектов лингвистической онтологии.

Этапы алгоритма формирования вероятностных тематической моделей по коллекции текстов:

- Формирование текстовой коллекции и обработка с использованием ПП АЛОТ, формирование коллекции результатов обработки в формате .nld;
- Задание словарей стоп-слов (точнее текстовых объектов), которые будут игнорироваться;
- Задание параметров тематического моделирования.

4.1.2.3. Программный модуль, реализующий алгоритм построения вероятностных тематических моделей (в том числе с использованием лингвистических онтологий ПП АЛОТ) по коллекции текстов (Модуль №04)

Назначение. Программный модуль, реализующий алгоритм обучения вероятностной тематической модели по корпусу текстов.

Входные данные. Входными данными для модуля являются текстовые материалы внешнего источника информации (например, Википедии), обработанные ПП АЛОТ в формате .nld в виде наборов zip файлов.

Пример входных данных расположен в директории topic_modeling_train/topic_modeling_2035/data .

Выходные данные. Выходными данными для модуля являются:

- структуры данных вероятностных тематических моделей;
- перечень лексических единиц модели в виде словаря с документными частотами для каждого элемента.

Пример выходных данных (модель), обученный по представленным входным данным лежит в topic_modeling_train/topic_modeling_2035/module/test_output .

Моделью являются файлы с разрешениями .model и .dict.

Остальные файлы в директории – результат промежуточных стадий обучения.

API: Модуль представляет собой сервис, взаимодействие с которым происходит через POST запрос. Запуск сервиса производится скриптом train.py.

```
train.py
                [-h]
                [--input_dir INPUT_DIR]
                [--output_dir OUTPUT_DIR]
                [--model_size MODEL_SIZE]
                [--min_df MIN_DF]
                [--result_model_name RESULT_MODEL_NAME]
                [--wabbit_path WABBIT_PATH]
                [--zip_count ZIP_COUNT]
                [--topic_modeling_config TOPIC_MODELING_CONFIG]
```

Необязательные аргументы:

- --input_dir Директория с входными данными, представленными набором zip архивов

- --output_dir Директория в которой будет сгенерированная модель и иные промежуточные данные
- --model_size Размер векторов модели
- --min_df Параметр отсекания слов по документной частоте снизу
- --result_model_name Итоговое имя модели
- --wabbit_path Параметр контроллируешь имя промежуточного файла, по умолчанию auto (автоматически будет выбрано имя на основе имени входной директории)
- --zip_count Количество zip файлов для обработки (выбираются случайным образом). Параметр оптионален, без указания значения будут обработаны все
- --topic_modeling_config путь до конфигурационного файла для обучения, по умолчанию config.json

Конфигурационный файл имеет json формат и содержит в себе dict с полями:

- ruthes_path: путь до файла содержащего тезаурус
- artm_dir: путь до bigartm модуля
- topic_modeling_dir: путь до директории, содержащей модуль topic_modeling

Содержание директорий:

- topic_modeling_train: корневая директория
  - BigARTM: установленный модуль bigartm для Windows
  - data: директория с примерами входных данных
  - topic_modeling: директория с логикой модуля
  - train.py: скрипт реализующий api модуля
  - topic_modeling: директория, содержащая функционал для работы с тематическими моделями
  - stopwords_wiki.txt: вспомогательный файл со стоп словами
  - stopwords_vacancy.txt: вспомогательный файл со стоп словами

- ru_stopwords.txt : вспомогательный файл со стоп словами
- StopWords.L : вспомогательный файл со стоп словами
- requirements.txt: файл описывающие python зависимости
- config.json: конфигурационный файл
- test_output: директория с примерами выходных файлов

Требования для использования модуля: Python 3.7

Способ развертывания модуля:

1) Скопировать модуль на диск
2) Установить зависимости из файла requirements.txt вызовом pip install –r requirements.txt
3) Добавить в системную переменную PATH путь до bin директории внутри BigARTM директории
4) Добавить в системную переменную PYTHONPATH путь до Python директории внутри BigARTM директории
5) Добавить системную переменную ARTM_SHARED_LIBRARY с путем до файла artm.dll внутри директории bin модуля BigARTM

4.1.2.4. Алгоритм представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве с использованием вероятностных тематических моделей

Алгоритм классификации текстов с использованием вероятностного тематического моделирования реализуется в виде сервиса:

- При загрузке сервиса загружается и модель, лежащая в директории проекта.
- По обращению с post запросом модель обрабатывает входные данные и предсказывает содержащиеся топики и их вес.

- После этого для каждой рубрики (топика) рассчитываются наиболее весомые слова из данного документа на основании частоты в документе и веса слов в самих темах.
- Результат оформляется в заданном формате.

4.1.2.5. Программный модуль, реализующий алгоритм представления модели содержания текста учебного курса в виде вектора вероятностных тематик (Модуль №05)

Назначение: Программный модуль, реализующий алгоритм построения вероятностных тематических моделей (в том числе с использованием лингвистических онтологий ПП АЛОТ) по коллекции текстов.

Входные данные. Входными данными для модуля являются данные материалов учебного курса, обработанные ПП АЛОТ в формате .nld

Примеры выходных данных расположены в topic_modeling_server/topic_modeling_2035/data в виде трех nld файлов.

Выходные данные. Выходными данными для модуля являются:

- вектор весов тематик вероятностных тематических моделей, полученных в Модуле № 04;
- упорядоченный перечень лексики и понятий текста материалов учебного курса для каждой из тематик.

Примеры выходных данных расположены в topic_modeling_server/topic_modeling_2035/data, где для каждого nld файла построен соответствующий ему .topic.json файл.

API: Модуль представляет собой сервис, взаимодействие с которым происходит через POST запрос. Запуск сервиса производится скриптом run.bat, который вызывает python скрипт topic_modeling_http_server.py

```
topic_modeling_http_server.py
```

```
[-h]
[-l LISTEN]
[-p PORT]
[--topic_modeling_config TOPIC_MODELING_CONFIG]
```

Необязательные аргументы:

- -l LISTEN (или --listen LISTEN) - Адрес на котором будет работать сервис (например, 0.0.0.0)
- -p PORT (или --port PORT) - Порт на котором будет работать сервис
- --topic_modeling_config - Путь до конфигурационного файла, по умолчанию config.json

Конфигурационный файл имеет json формат и содержит в себе dict с полями:

- artm_models: содержанием которого является dict, ключами для которого служат имена моделей, а значением является список из двух элементов:
  - Путь до модели
  - Размерность модели
- ruthes_path: путь до файла содержащего тезаурус
  - artm_dir: путь до bigartm модуля
  - topic_modeling_dir: путь до директории, содержащей модуль topic_modeling

Для запроса к модулю необходимо послать POST запрос с параметром «nld» и значением эквивалентным содержанию nld файла.

Содержание директорий:

- topic_modeling_train: корневая директория
  - BigARTM: установленный модуль bigartm для Windows
  - data: директория с примерами входных и выходных данных
  - topic_modeling: директория с логикой модуля
  - app_flask.py: скрипт реализующий работу сервиса

- topic_modeling: директория, содержащая функционал для работы с тематическими моделями
- requirements.txt: файл описывающие python зависимости
- config.json: конфигурационный файл
- trained_models: директория с примерами обученных моделей по википедии и вакансиям

Требования для использования модуля: Python 3.7

Способ развертывания модуля:

1) Скопировать модуль на диск
2) Установить зависимости из файла requirements.txt вызовом pip install –r requirements.txt
3) Добавить в системную переменную PATH путь до bin директории внутри BigARTM директории
4) Добавить в системную переменную PYTHONPATH путь до Python директории внутри BigARTM директории
5) Добавить системную переменную ARTM_SHARED_LIBRARY с путем до файла artm.dll внутри директории bin модуля BigARTM

4.1.3. Разработка методов и алгоритмов различного представления понятийно-тематического пространства учебного курса с использованием объектов/сущностей различной природы

4.1.3.1. Методы различного представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве

Типовые материалы учебного курса представляют собой документы достаточно сложной структуры (Рисунки 6, 7).

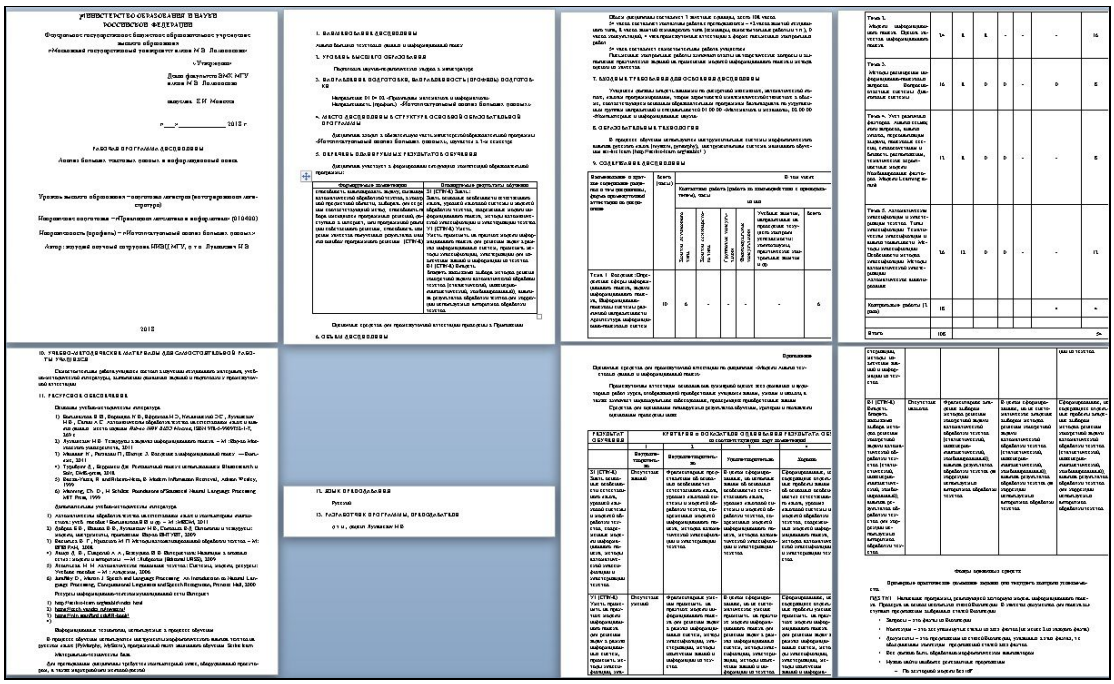
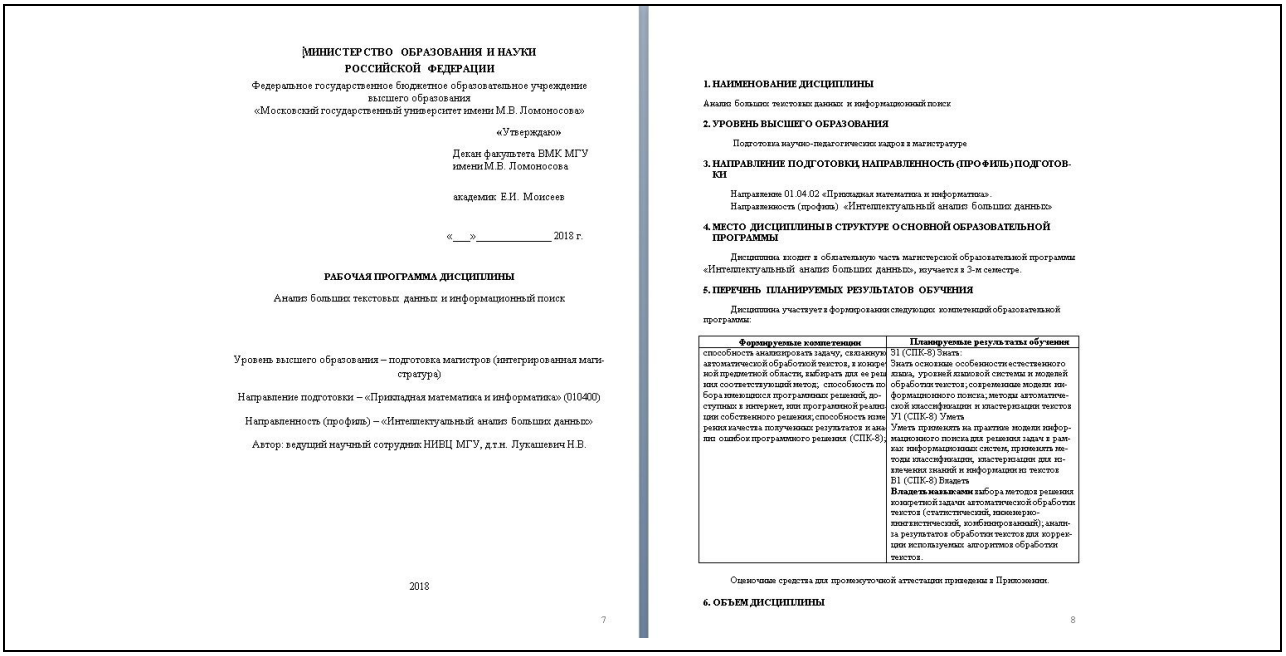


Рисунок 6 – Пример структуры основного типового документа описания учебного курса – программы обучения



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный университет имени М.В. Ломоносова»

«Утверждаю»

Декан факультета ВМК МГУ
имени М.В. Ломоносова

академик Е.И. Моисеев

«___»_____ 2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Анализ больших текстовых данных и информационный поиск

Уровень высшего образования – подготовка магистров (интегрированная магистратура)

Направление подготовки – «Прикладная математика и информатика» (010400)

Направленность (профиль) – «Интеллектуальный анализ больших данных»

Автор: ведущий научный сотрудник НИВЦ МГУ, д.т.н. Лукашевич Н.В.

2018

7

1. НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ

Анализ больших текстовых данных и информационный поиск

2. УРОВЕНЬ ВЫСШЕГО ОБРАЗОВАНИЯ

Подготовка научно-педагогических кадров в магистратуре

3. НАПРАВЛЕНИЕ ПОДГОТОВКИ, НАПРАВЛЕННОСТЬ (ПРОФИЛЬ) ПОДГОТОВКИ

Направление 01.04.02 «Прикладная математика и информатика».
Направленность (профиль) «Интеллектуальный анализ больших данных»

4. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина входит в обязательную часть магистерской образовательной программы «Интеллектуальный анализ больших данных», изучается в 3-м семестре.

5. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Дисциплина участвует в формировании следующих компетенций образовательной программы:

Формируемые компетенции	Планируемые результаты обучения
способность анализировать задачу, связанную с автоматической обработкой текстов, в конкретной предметной области, выбирать для ее решения соответствующий метод; способность подбирающихся программных решений, доступных в интернет, или программной реализации собственного решения; способность измерения качества полученных результатов и анализ ошибок программного решения (СПК-8).	З1 (СПК-8) Знать Знать основные особенности естественного языка, уровней языковой системы и моделей обработки текстов, современные модели информационного поиска, методы автоматической классификации и кластеризации текстов У1 (СПК-8) Уметь Уметь применять на практике модели информационного поиска для решения задач в рамках информационных систем, применять методы классификации, кластеризации для извлечения знаний и информации из текстов В1 (СПК-8) Владеть **Владеть навыками** выбора методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов обработки текстов для коррекции используемых алгоритмов обработки текстов.

Оценочные средства для промежуточной аттестации приведены в Приложении.

6. ОБЪЕМ ДИСЦИПЛИНЫ

8

Рисунок 7 – Пример первых двух страниц типовой программы обучения

Документы имеют большое количество элементов оформления (см.также пример типовой программы обучения, приведенный в разделе Б.1 Приложения Б), которые можно ошибочно трактовать как значимые

элементы курса, особенно для учебных курсов, посвященным гуппанитарным вопросам.

Поэтому для автоматического анализа учебного курса требуется первоначально произвести очистку текста от «паразитного» (то есть сопутствующего, не содержащего полезной информации) оформления.

Для этого используется словарь очистки, содержащий подстроки, которые игнорируются при содержательном анализе материалов учебного курса (фрагмент такого словаря приведен в разделе Б.2 Приложения Б).

Как уже указывалось при описании типового технического решения на основе ПП АЛОТ, результатом Автоматизированной Лингвистической Обработки Текстов являются текстовые объекты, которые выделены из текста, включая оценки их значимости для содержания текста.

Также для анализируемого текста выводятся тематические рубрики:

- С использованием стандартной процедуры ПП АЛОТ на основе явного описания смысла рубрик через логические формулы над понятиями лингвистической онтологии;
- С использованием новых программных модулей:
    - С использованием явного описания рубрик на языке запросов информационно-поисковой системы NearIdx (Модуль № 01);
    - С использованием неявного описания смысла задаваемых рубрик на основе машинного обучения по размеченныи примерам (Модуль № 02, Модуль № 03);
    - С использованием явного, автоматически формируемого на основе вероятностного тематического моделирования, описания смысла рубрик путем «мягкой» кластеризации (нежесткого группирования) похожих тематически документов (Модуль № 04 и Модуль № 05).

Использование тематических классификаторов позволяет сравнивать документы различных учебных курсов в обобщенном смысле – по сходству

приписанных тематик, при том что лексическая похожесть может отсутствовать.

К проблемам использования тематических классификаторов именно для анализа материалов учебных курсов можно отнести потенциально возможный сильный «авторский» компонент конкретного учебного курса, материалы которого излагаются автором с использованием не стандартной, то есть мало распространенной лексики и терминологии.

В этом сулчае ранее перечисленные методы – лингвистичсекие онтологии, вероятностные тематические модели – больше ориентированы на описание текстовых объектов, обеспечивающих максимальное «среднее» покрытие лексики и терминологии текстовых коллекций предметной области, и могут не обеспечивать достаточное покрытие содержания конкретного учебного курса.

В качестве авторитетного информационного ресурса, который предлагается использовать для экспертизы содержания учебного курса рассматривается русская Википедия, которая описывает более 1,5 оригинальных статей (еще примерно столько же синонимичных объектов, отсылающих на оригинальные статьи).

Предполагается, что если курс посвящен сколько-нибудь значимому вопросу, то он должен иметь пересечения с какими-то статьями Википедии. Если нет – то скорее всего, курс является «слишком» оригинальным и требует уже именно экспертной процедуры оценивания.

Следует учитывать, что используемая в АЛОТ лингвистическая онтология примерно в 10 раз меньше по составу Википедии, но имеет при этом выверенные иерархические связи, позволяющие осуществлять уверенный логический вывод. С использованием Википедии можно установить лишь ассоциативные связи.

Разработанный в рамках настоящей работы метод заключается в следующем:

- Для оперирования содержимым Википедии было осуществлено:
- Скачивание русской Википедии;
- Конвертация страниц Википедии в формат файлов .htm и .hdr для загрузки в поисковую машину NearIdx;
- При этом оригинальные статьи Википедии были модифицированы – синонимы из ссылочных статнй, осуществляющих редирект на оригинальные статьи были добавлены к оригинальным статьям, при этом ссылочные статьи после не рассматривались.

Ассоциативная связь – прежде всего для не входящих в состав лингвистической онтологии терминоподобных слов и словосочетаний – определялась на основании установления связи с известными объектами лингвистической онтологии, которые входят в начало текста статьи Википедии. То есть для каждого терминоподобного текстового объекта производится поиск по первым 50 словопозициям текстов статей Википедии, и производится подсчет близости с известными объектами по онтологии, отдавая предпочтение понятиям онтологии уже найденным в анализируемом тексте. Это позволяет связать неизвестные объекты с известными. Что далее позволяет связывать между собой несвязанные напрямую объекты разных курсов и т.д.

4.1.3.2. Алгоритм различного представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве

Алгоритм расширенного представления понятийно-тематического пространства учебного курса включает в себя:

- Предварительную очистку текста материалов учебного курса от элементов оформления с использванием словаря очистки;

- Запуск АЛОТ для получения .nld файла, содержащего все выделяемые объекты, включая понятия онтологии, рубрики тематических классификаторов;
- Запуск модуля № 01 для получения тематических рубрик описываемых на языке запросов к поисковой машине;
- Запуск модуля № 03 (использующего результаты обучения в модуле № 02) для получения тематических рубрик на основе применения методов машинного обучения по размеченным примерам;
- Запуск модуля № 05 (использующего результаты обучения в модуле № 04) для получения тематических рубрик (топиков) на основе вероятностного тематического моделирования;
- Поиск ассоциативных связей, прежде всего, для терминоподобных слов и словосочетаний с использованием информационно-поисковой машины NearIdx по русской Википедии, модеоирующей расширенное понятийное пространство.

Совокупнсть полученных объектов, ассоциированных с анализируемым документом, представляет расширенное понятийно-тематического пространства учебного курса.

4.1.3.3. Программный модуль, реализующий алгоритм различного представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве (Модуль № 06)

Назначение: Программный модуль, реализующий алгоритм различного представления расширенного понятийно-тематического пространства учебного курса в общем понятийно-тематическом пространстве.

Технологическое описание: модуль для создания полного индекса, обращается к функциям mtfod ПП АЛОТ и получает NLD файл, а также вызывает модули № 01, № 03, № 05.

Входные данные. Модуль содержит функцию make_nld_full, в нее параметром передается текст файла материалов учебного курса.

Выходные данные. Функция возвращает объект содержащий данные индекса найденных сущностей. Результирующий файл сохраняется в исходное наименование с добавлением расширения «.res_mod06».

Имя файла: m06_make_nld_full.py

Проверка функционирования

Пакетный файл: __test_mod06.bat

Директория с тестовыми данными __test06

API: Вызов функции make_nld_full:

```
C:\Python38\python.exe
    -u m06_make_nld_full.py
    --infile __test06\infile.txt
```

Здесь:

--infile – имя входного файла

4.1.4. Разработка методов и алгоритмов формирования отчетных документов для визуализации представления понятийно-тематического пространств учебных курсов

4.1.4.1. Методы представления понятийно-тематического пространств учебных курсов в результате индексирования материалов учебных курсов

Основная цель отчетных документов, визуализирующих представления понытино-тематического пространства учебного курса – контроль

результативности анализа материалов учебного курса, которые планируется использовать для решения основных задач работы – сравнения содержания разных учебных курсов, сравнения содержания учебного курса и рефлексии слушателей.

Основной документ, содержащий информацию о выделенных ПП АЛОТ текстовых объектах – файл формата .nld . В файле .nld приводится информация об исходном тексте, и все выделенные объекты с привязкой к абзацам, предложениям и словопозициям:

```
<?xml version="1.0" encoding="UTF-8"?>
<doc id="\">
<zone>
<text>-  [Мои курсы](https://cat.2035.university/rall/my/)
       ...
       Проектирование в системе AutoCAD
       ...
       Программа курса
       Модуль 1.
       Основы AutoCad
       Тема 1.1 Интерфейс графической среды AutoCad.
       Тема 1.2. Средства пространственной ориентации.
       Тема 1.3. Пользовательский интерфейс AutoCAD
       Тема 1.4. Системы координат
       Тема 1.5. Работа с примитивами.
       Тема 1.6. Свойства примитивов
       Тема 1.7. Управление экраном
       Тема 1.8. Методы построения углов.

       <index>
SENT   ABZ          99     1|1|1
...
LEM    2D-ПРИМИТИВЫ        25     520|1|30
...
LEM    AUTOCAD             86     555|1|32
...
TERM   COURSERA     275924 61     16|1|1
...
RUBR   Digital Economy Competences/I305000000 Некомпьютерное электронное оборудование
       400062 56     24|1|1
RUBR   Sciences/T800900000000 КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ      706608 91     24|1|1
RUBR   Sciences/S700060000000 ВЫЧИСЛИТЕЛЬНЫЕ И КОММУНИКАЦИОННЫЕ РЕСУРСЫ  706578 91
       24|1|1
RUBR   Sciences/S700000000000  ИНФОРМАЦИОННЫЕ  ТЕХНОЛОГИИ  И  ВЫЧИСЛИТЕЛЬНЫЕ  СИСТЕМЫ
       706576 91     24|1|1
RUBR   Sciences/T800000000000 ТЕХНОЛОГИИ      706494 78     24|1|1
RUBR   Sciences/T800800000000 ПРОМЫШЛЕННЫЕ ТЕХНОЛОГИИ      706532 61     24|1|1
RUBR   Sciences/S100011000000 МАТЕМАТИКА      701100 61     24|1|1
...
TERMTREE     ПРОГРАММНЫЙ ОБЪЕКТ  275164 77     38|1|1
TERMTREE     ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT 277039 62     38|1|1
TERMTREELOW  ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT 277039 62     38|1|1
...
RUBR   Digital Economy Competences/F106000000 Языки программирования     500003 62
       38|1|1
RUBR   Digital Economy Competences/D101000000 Создание IT-продукта       211000 62
       38|1|1
RUBR   Digital Economy Competences/I305000000 Некомпьютерное электронное оборудование
```

```
      400062 56     38|1|1
RUBR  Sciences/T800900000000 КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ      706608 91     38|1|1
```

Для учета информации об объектах, которые могут быть полезны в рамках работы, в том числе об объектах выделенных в результате работы модулей №№ 01, 03, 05, формируется файл .dat, содержащий, прежде всего, в табличном виде агрегированную информацию про выделенные текстовые объекты (тип, наименование, частотность в документе, вес – оценка значимости для содержания, идентификатор, если имеется):

```
term count:  119
    TERM     COURSERA     2      61     275924
    TERM     PDF-ДОКУМЕНТ 1      61     273863
    TERM     АВТОМАТИЗАЦИЯ1      61     924
...
    TERM     ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT 2      62     277039
    TERM     ЯКУТИЯ 5      55     103250

tfm count:  200
    TFM      АДДИТИВНОЕ ПРОИЗВОДСТВО    2      9      -1
    TFM      БАЗА ДАННЫХ  1      4      -1
    TFM      ВОЗМОЖНОСТЬ ПРОГРАММЫ      1      4      -1
    TFM      ВСТАВКА БЛОКОВ      1      4      -1
    TFM      ВЫВОД ЧЕРТЕЖА1      4      -1
...
    TFM      ЭТАП РАЗРАБОТКИ ПРОМЫШЛЕННОГО ДИЗАЙНА  1      4      -1
    TFM      ЯЗЫК ГРАФИЧЕСКОГО ПРОГРАММИРОВАНИЯ     1      4      -1

lem count:  23
    LEM      ОСНОВНОЙ     3      66     -1
    LEM      СКРЫТЫЙ      1      25     -1
    LEM      ФОРМАТ 3     50     -1
    LEM      НЕРЮНГРИ     1      25     -1
...
    RUB3     digit  1     99     -1
    RUB3     tech   1     99     -1
...
    RUB5     КОМПЬЮТЕРНАЯ_ПРОГРАММА    АВТОР_РАЗРАБОТКИ    ИСХОДНЫЙ_КОД    CODE
             ОПЕРАЦИОННАЯ_СИСТЕМА_WINDOWS     1      33     -1
    RUB5     АДМИНИСТРИРОВАНИЕ   ДОКУМЕНТАЛЬНЫЙ_МАТЕРИАЛ   ИНФОРМАЦИЯ   ДОКУМЕНТ
             ДАННЫЕ_(СВЕДЕНИЯ)   1      10     -1
    RUB5     ИННОВАЦИЯ,_ВНЕДРЕННОЕ_НОВШЕСТВО                          ТЕХНОЛОГИЯ
             ИННОВАЦИОННАЯ_ТЕХНОЛОГИЯ НАУЧНО-ТЕХНИЧЕСКАЯ_СФЕРА НАУКА    1      9
                    -1
    RUB5     УНИВЕРСИТЕТ  ГОСУДАРСТВЕННЫЙ_УНИВЕРСИТЕТ  ВЫСШЕЕ_УЧЕБНОЕ_ЗАВЕДЕНИЕ
             УЧЕБНЫЙ_ИНСТИТУТ ПЕДАГОГИЧЕСКАЯ_ДЕЯТЕЛЬНОСТЬ  1      8      -1
    RUB5     ОБЩЕОБРАЗОВАТЕЛЬНАЯ_ШКОЛА ОБУЧАЮЩИЙ_ПРОЦЕСС ШКОЛА ШКОЛЬНЫЙ_УЧИТЕЛЬ
             ОБРАЗОВАТЕЛЬНАЯ_СИСТЕМА   1      6      -1
    RUB5     ТОРГОВЛЯ ЭКСПОРТИРОВАНИЕ СЕЛЬСКОЕ_ХОЗЯЙСТВО ДОБЫВАЮЩЕЕ_ПРЕДПРИЯТИЕ
             ТЕКСТИЛЬНОЕ_ПРЕДПРИЯТИЕ   1      6      -1
    RUB5     ОПУБЛИКОВАНИЕ  ПРОИЗВЕДЕНИЕ_(ПРОДУКТ_ТРУДА)  СОАВТОР  ИЗДАТЕЛЬСКО-
             ПОЛИГРАФИЧЕСКАЯ_ДЕЯТЕЛЬНОСТЬ ИЗДАТЕЛЬ  1      5      -1
    RUB5     АРХИВНЫЙ_ФАЙЛ PDF-ДОКУМЕНТ ХОККЕЙ_НА_ТРАВЕ ПАКИСТАН ИНТЕРНЕТ-АДРЕС
                    1      3      -1
    RUB5     ОФИЦИАЛЬНЫЙ_САЙТ ИНТЕРНЕТ-ПОРТАЛ ИНТЕРНЕТ-САЙТ БЛОГ ИНТЕРНЕТ-ФОРУМ
                    1      3      -1
    RUB5     КОММУНИКАЦИОННАЯ_ИНФРАСТРУКТУРА                          ЛИНИЯ_СВЯЗИ
             ДАННЫЕ_В_ЭЛЕКТРОННОМ_ВИДЕ СЕТЕВОЙ_ПРОТОКОЛ СЕТЕВОЕ_СОЕДИНЕНИЕ    1
                    2      -1
...
```

```
    RUBX      3D моделирование    1      27     -1
    RUBX      3D печать    1      24     -1
    RUBX      AutoCAD      1      86     -1
    RUBX      AutoLISP[-]  1      25     -1
...
    RUBR      3D моделирование    1      27     -1
    RUBR      3D печать    1      24     -1
    RUBR      AutoCAD      1      86     -1
    RUBR      AutoLISP[-]  1      25     -1
...
```

Также файл .dat содержит информацию о значимых взаимосвязях объектов, полученных с использованием обобщенного пространства (моделируется Википедией):

```
...

TFM :: АДДИТИВНОЕ ПРОИЗВОДСТВО
#REQ: ВНАЧАЛЕ50(ВОКНЕ2(АДДИТИВНОЕ ПРОИЗВОДСТВО))
#wiki doc count:  3  use:  3
    TERM     АВТОМАТИЧЕСКИЙ РЕЖИМ      1      120021
             WRC:    Селективное лазерное спекание
    TERM     АДДИТИВНОСТЬ 9      143244
             WRC:    Селективное лазерное спекание
             WRC:    Проволочное электронно-лучевое аддитивное производство
             WRC:    Аддитивные технологии
...
    TFM      ПРОИЗВОДСТВО ДЕТАЛЕЙ      1      -1
             WRC:    Аддитивные технологии
    TFM      ПРОИЗВОДСТВО ИЗДЕЛИЙ      2      -1
             WRC:    Проволочное электронно-лучевое аддитивное производство
             WRC:    Аддитивные технологии
    TFM      ПРОИЗВОДСТВО ИЗДЕЛИЙ ПРОИЗВОЛЬНОЙ ФОРМЫ1      -1
             WRC:    Проволочное электронно-лучевое аддитивное производство
...

TFM :: БАЗА ДАННЫХ
#REQ: ВНАЧАЛЕ50(ВОКНЕ2(БАЗА ДАННЫХ))
#wiki doc count:  782  use:  10
...
    TERM     АССОЦИАТИВНЫЙ МАССИВ      2      272224
             WRC:    База данных «ключ-значение»
    TERM     БАЗА ДАННЫХ  37     2194
             WRC:    UniProt
             WRC:    Модель базы данных
             WRC:    Онлайн-база данных
             WRC:    Маркетинг на основе баз данных
             WRC:    Резидентная база данных
             WRC:    База данных «ключ-значение»
             WRC:    Базы данных дистанционного зондирования Земли
             WRC:    Упреждающая журнализация
             WRC:    CiNii
             WRC:    Дедуктивная база данных
...
    TERM     ДАННЫЕ (СВЕДЕНИЯ)   6      107581
             WRC:    База данных «ключ-значение»
             WRC:    Маркетинг на основе баз данных
             WRC:    Базы данных дистанционного зондирования Земли
...
    TERM     ЗАПИСЬ (ТО,ЧТО ЗАПИСАНО)  5      259205
             WRC:    Упреждающая журнализация
...

TFM :: ГЕОМЕТРИЧЕСКИЙ ОБЪЕКТ
```

```
#REQ: ВНАЧАЛЕ50(ВОКНЕ2(ГЕОМЕТРИЧЕСКИЙ ОБЪЕКТ))
#wiki doc count:  19  use:  10
    TERM     АЛГЕБРА      8      7846
             WRC:    Геометрическая теория групп
             WRC:    Геометрическая алгебра (значения)
             WRC:    Кривизна
...
    TERM     ГЕОМЕТРИЯ    27     30
             WRC:    Геометрическая алгебра (значения)
             WRC:    Множество Радона — Никодима
             WRC:    Геометрическая теория групп
             WRC:    Кривизна
             WRC:    Геометрический решатель САПР
             WRC:    Олоид
             WRC:    Срединная ось
             WRC:    Пучок (геометрия)
             WRC:    AABB
             WRC:    Геометрическое квантование
...
    TERM     КООРДИНАТНАЯ ОСЬ    4      151855
             WRC:    Срединная ось
             WRC:    AABB
    TERM     КРИВАЯ (НЕПРЯМАЯ ЛИНИЯ)    6      119810
             WRC:    Пучок (геометрия)
             WRC:    Кривизна
             WRC:    Олоид
...
    TERM     МАТЕРИАЛЬНЫЙ ОБЪЕКТ 7      259801
             WRC:    Геометрическая алгебра (значения)
             WRC:    Множество Радона — Никодима
             WRC:    Кривизна
             WRC:    Олоид
             WRC:    Срединная ось
             WRC:    Пучок (геометрия)
...

TERM :: ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT
#REQ: ВНАЧАЛЕ50(/TERM="ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT")
#wiki doc count:  417  use:  10
....
    TERM     БРАУЗЕР      7      122005
             WRC:    JSFiddle
             WRC:    Букмарклет
             WRC:    Ненавязчивый JavaScript
             WRC:    Движок JavaScript
...
    TERM     ВЕБ-РАЗРАБОТКА      3      272014
             WRC:    JSFiddle
             WRC:    Ненавязчивый JavaScript
    TERM     ВЕРСИЯ 5      224961
             WRC:    Rhino
             WRC:    Dojo
             WRC:    Underscore
             WRC:    Nashorn (движок JavaScript)
```

Также формируется файл .rld, содержащий информацию о связи между рубриками и текстовыми объектами:

```
RUB3    digit   -1
        RITEM   TERM    сапр_autocad    -1      93
        RITEM   LEM     autocad -1      83
        RITEM   LEM     программа       -1      60
        RITEM   LEM     чертеж  -1      35
        RITEM   LEM     разработка      -1      19
        RITEM   TERM    чертеж  -1      15
        RITEM   LEM     графический     -1      14
```

```
        RITEM   LEM     электронный     -1      13
        RITEM   LEM     технология      -1      13
        RITEM   LEM     система -1      13
RUB3    tech    -1
        RITEM   LEM     программа       -1      69
        RITEM   TERM    сапр_autocad    -1      53
        RITEM   LEM     autocad -1      51
        RITEM   LEM     тема    -1      37
        RITEM   LEM     технология      -1      19
        RITEM   TERM    программный_объект      -1      19
        RITEM   TERM    проектировать,_создавать_проект -1      15
        RITEM   LEM     графический     -1      14
        RITEM   LEM     электронный     -1      14
        RITEM   LEM     система -1      13
RUB5    КОМПЬЮТЕРНАЯ_ПРОГРАММА АВТОР_РАЗРАБОТКИ ИСХОДНЫЙ_КОД CODE
                        ОПЕРАЦИОННАЯ_СИСТЕМА_WINDOWS  -1
        RITEM   TERM    КОМПЬЮТЕРНАЯ_ПРОГРАММА  -1      94
        RITEM   TERM    РАЗРАБОТКА_ПРОГРАММНОГО_ОБЕСПЕЧЕНИЯ     -1      14
        RITEM   TERM    ОПЕРАТОР_ПРОГРАММЫ      -1      13
        RITEM   TERM    ПРОГРАММНЫЙ_МОДУЛЬ      -1      6
        RITEM   TERM    ПОЛЬЗОВАТЕЛЬСКИЙ_ИНТЕРФЕЙС      -1      6
        RITEM   LEM     ПРОГРАММА       -1      5
        RITEM   TERM    СРЕДА_РАЗРАБОТКИ_ПРОГРАММЫ      -1      5
        RITEM   TERM    ИНТЕРНЕТ-САЙТ   -1      3
        RITEM   TERM    ПРОГРАММНЫЙ_БЛОК        -1      3
        RITEM   TERM    ЯЗЫК_(СИСТЕМА_ЗНАКОВ)   -1      2
RUB5    АДМИНИСТРИРОВАНИЕ ДОКУМЕНТАЛЬНЫЙ_МАТЕРИАЛ ИНФОРМАЦИЯ ДОКУМЕНТ
                        ДАННЫЕ_(СВЕДЕНИЯ) -1
        RITEM   TERM    АДМИНИСТРИРОВАНИЕ       -1      158
        RITEM   TERM    ДОКУМЕНТ        -1      5
        RITEM   TERM    РАЗРАБОТКА_ПРОГРАММНОГО_ОБЕСПЕЧЕНИЯ     -1      4
        RITEM   TERM    СУДОПРОИЗВОДСТВО        -1      4
        RITEM   TERM    СТАНДАРТ        -1      3
        RITEM   TERM    ИНФОРМАЦИЯ      -1      3
        RITEM   LEM     УПРАВЛЕНИЕ      -1      2
        RITEM   LEM     ОСНОВНОЙ        -1      2
        RITEM   LEM     СИСТЕМА -1      2
        RITEM   TERM    ТРУД    -1      2
...
```

В совокупности указанных таблиц достаточно для оценки похожести различных учебных курсов, сравнения понятийно-тематических пространств материалов учебного курса и рефлексии слушателей.

В качестве графового представления учебного курса формируется xml файл .gexf, который может быть отображен с укладке «пять лучей» (Рисунок 8) - когда задается пять лучей, исходящих от центра графа, центральную часть занимает дистрибутивный граф, а на концах лучей располагаются объекты «ЗНАНИЯ»--«ИНСТРУМЕНТЫ»--«НАВЫКИ»--«ПОЗИЦИИ»--«ПРОЕКТЫ».

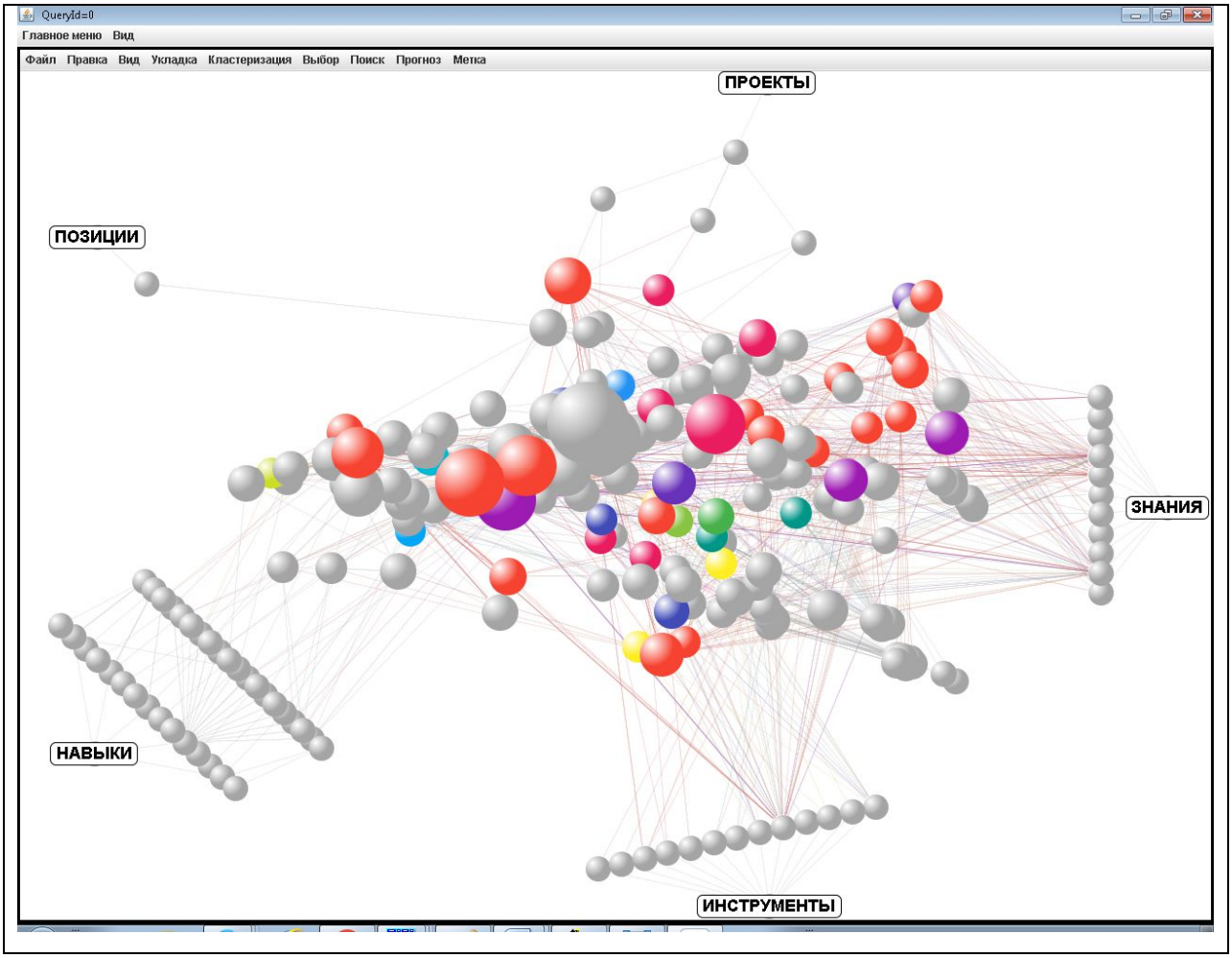


Рисунок 8 – Схема укладки графов «пять лучей» - с фиксированными объектами «ЗНАНИЯ»--«ИНСТРУМЕНТЫ»--«НАВЫКИ»--«ПОЗИЦИИ»--«ПРОЕКТЫ»

4.1.4.2. Алгоритм представления понятийно-тематического пространств учебных курсов в результате индексирования материалов учебных курсов в табличной форме и на графах

Алгоритм представления понятийно-тематического пространств учебных курсов в результате индексирования материалов учебных курсов в табличной форме и на графах заключается в формировании описанных в предыдущем пункте структур на основе полученных в модуле № 06 данных.

4.1.4.3. Программный модуль, реализующий алгоритм представления понятийно-тематического пространств учебных курсов в результате индексирования материалов учебных курсов в табличной форме и на графах (Модуль № 07)

Назначение: Программный модуль, реализующий алгоритм представления понятийно-тематического пространств учебных курсов в результате индексирования материалов учебных курсов в табличной форме и на графах.

Технологическое описание: модуль для создания файлов графов, обращается к GMTPOD (ПП АЛОТ), передает текст и получает файл графа, а также сохраняет файл связей

Входные данные. На вход подается текст материалов учебного курса в виде текстового файла. Внутри модуля находится функция make_tbl_one_k, в нее параметром передается текст файла,

Выходные данные. Результат сохраняется на диск в файлы с тем же именем, дополненным *.res_mod07.gexf, *.res_mod07.rld .

Имя файла с исходным кодом: m07_make_tbl_one_k.py

Проверка функционирования

Пакетный файл: __test_mod07.bat

Директория с тестовыми данными __test07

API: Вызов функции make_tbl_one_k:

```
C:\Python38\python.exe
    -u m07_make_tbl_one_k.py
    --infile __test07\infile.txt.res_mod06
```

Здесь: --infile – параметр указывает имя файла с результатом модуля № 06.

4.2. Разработка методов и алгоритмов сравнения понятийно-тематических пространств различных учебных курсов.

4.2.1. Разработка методов и алгоритмов сравнения выявленных семантических структур содержания учебных курсов

4.2.1.1. Методы сравнения выявленных семантических структур содержания учебных курсов, выявления области пересечения и области несовпадения

После получения в модуле № 06 полной информации об объектах анализируемого текста и фиксации этой информации с использованием модуля № 07 возникает возможность сравнить два понятийно-тематических представления.

Простейшим видом сравнения является сопоставление объектов соответствующих типов, которые определяют зону совпадающих («общих») объектов (так называемая зона EQUAL или «**общее между курсами**»).

После чего в двух сравниваемых представлениях остаются объекты, не имеющие прямого сопоставления в другом представлении.

Разработан метод сравнения понятийно-тематических пространств учебных курсов.

При этом первый сравниваемый курс имеет служебное название BASE (базовый или «**Курс 1**»), второй – SAMPLE (пример или «**Курс 2**»).

Выделяются следующие зоны при сравнении:

B_DET – область детализации для объектов из BASE, то есть объект из BASE «детализируется» в рамках другого курса («**детализируется в К2**»), найдется не менее одного объекта, которые расположен ниже в иерархии лингвистической онтологии.

B_GEN – область обобщения для объектов из BASE (не входящих в B_DET), то есть объект из BASE является «**пререквизитом для К2**».

B_ASC – область ассоциативных связей – «**слабая связь с К2**» (для объектов, не вошедших в B_DET, B_GEN), когда определяется связь через общие статьи в Википедии, в начале текста которых (что моделирует попадание в определение статьи, включая список синонимов) используются оба объекта.

B_LOST – область «**уникального в К1**» - объектов, для которых не нашлось совпадающих или похожих объектов.

Аналогично определяются зоны обобщения для SAMPLE: S_DET, S_GEN, S_ASC, S_LOST.

Например, если объект из BASE «АЛГОРИТМ», а в SAMPLE есть объект «МЕТОД ОПТИМИЗАЦИИ», которые не связаны с другими объектами, то «АЛГОРИТМ» попадает в B_DET («детализируется в К2»), а «МЕТОД ОПТИМИЗАЦИИ» попадает в область S_GEN.

4.2.1.2. Алгоритм сравнения выявленных семантических структур содержания учебных курсов, выявления области пересечения и области несовпадения

Алгоритм сравнения выявленных семантических структур содержания учебных курсов состоит в том, что:

Для каждого курса независимо производится агрегирование имеющихся связей по иерархии лингвистической онтологии и/или по установленным ассоциациям по Википедии. Формируются файлы _01_02_comp.dat и _02_01_comp.dat независимого сравнения двух курсов между собой (поиск в Википедии ограничен выбором количества статей, поэтому результаты сравнения могут отличаться), где для каждого объекта одного курса выводятся связи с объектами другого курса (символ «=» используется для обозначения совпадения):

```
..
TERM :: АВТОМАТИЗАЦИЯ   f: 1    r: 61   eq: 0.4 sp: 0.36
```

```
    40  TERM    ИНТЕРНЕТ-САЙТ   f2:1    r2:61
    40  TERM    САПР AUTOCAD    f2:22   r2:74
    70  TERM    ТРУД    f2:25   r2:84
...
TERM :: БАЗА ДАННЫХ     f: 1    r: 47   eq: 0.4 sp: 1.0
    70  TERM    АЛГОРИТМ        f2:1    r2:61
    70  TERM    ДАННЫЕ (СВЕДЕНИЯ)       f2:2    r2:51
    10  TERM    ИНФОРМАЦИЯ      f2:3    r2:62
    20  TERM    КОМПЬЮТЕРНАЯ ПРОГРАММА  f2:1    r2:61
    10  TERM    МАТЕРИАЛЬНЫЙ ОБЪЕКТ     f2:1    r2:15
    10  TERM    ПРОДУКТ ПРОИЗВОДСТВА    f2:1    r2:61
    10  TERM    ТЕХНИЧЕСКОЕ УСТРОЙСТВО  f2:1    r2:50
    70  TERM    ТРУД    f2:25   r2:84
...
TERM :: ГРАФИЧЕСКАЯ ПРОГРАММА   f: 3    r: 52   eq: 0.4 sp: 1.0
    70  TERM    АЛГОРИТМ        f2:1    r2:61
    10  TERM    ИНФОРМАЦИЯ      f2:3    r2:62
    10  TERM    КОМПЬЮТЕРНАЯ ПРОГРАММА  f2:1    r2:61
    10  TERM    ПРОДУКТ ПРОИЗВОДСТВА    f2:1    r2:61

TERM :: ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС  f: 3    r: 47   eq: 0.4 sp: 1.0
    70  TERM    АЛГОРИТМ        f2:1    r2:61
    10  TERM    ИНТЕРФЕЙС       f2:1    r2:50
    10  TERM    ИНФОРМАЦИЯ      f2:3    r2:62
    70  TERM    КОМПЬЮТЕРНАЯ ПРОГРАММА  f2:1    r2:61
    70  TERM    ПОЛЬЗОВАТЕЛЬ    f2:1    r2:50
    20  TERM    ТЕХНИЧЕСКОЕ УСТРОЙСТВО  f2:1    r2:50

TERM :: ГРАФИЧЕСКИЙ РЕДАКТОР    f: 1    r: 47   eq: 1.0 sp: 1.0
    ==  TERM    ГРАФИЧЕСКИЙ РЕДАКТОР    f2:1    r2:50
    70  TERM    АЛГОРИТМ        f2:1    r2:61
    10  TERM    ИНФОРМАЦИЯ      f2:3    r2:62
    10  TERM    КОМПЬЮТЕРНАЯ ПРОГРАММА  f2:1    r2:61
    20  TERM    ПРОВЕРИТЬ, УДОСТОВЕРИТЬСЯ В ПРАВИЛЬНОСТИ        f2:1    r2:15
    10  TERM    ПРОДУКТ ПРОИЗВОДСТВА    f2:1    r2:61
    70  TERM    ТЕКСТ   f2:1    r2:47
...
TERM :: КОМПЬЮТЕРНАЯ ПРОГРАММА  f: 1    r: 61   eq: 1.0 sp: 1.0
    ==  TERM    КОМПЬЮТЕРНАЯ ПРОГРАММА  f2:1    r2:61
    70  TERM    АВТОР РАЗРАБОТКИ        f2:1    r2:50
    50  TERM    ВИЗУАЛИЗАЦИЯ    f2:1    r2:47
    30  TERM    ГРАФИЧЕСКИЙ РЕДАКТОР    f2:1    r2:50
    70  TERM    ДАННЫЕ ПРОГРАММЫ        f2:1    r2:50
    30  TERM    ИНТЕРНЕТ-САЙТ   f2:1    r2:61
    3B  TERM    ИНТЕРФЕЙС       f2:1    r2:50
    30  TERM    ОБУЧАЮЩАЯ ПРОГРАММА     f2:1    r2:50
    40  TERM    ОПЕРАТОР ПРОГРАММЫ      f2:3    r2:51
    30  TERM    РАСШИРЕНИЕ ДЛЯ БРАУЗЕРА f2:1    r2:15
    30  TERM    САПР AUTOCAD    f2:22   r2:74
    70  TERM    ФУНКЦИИ ПРОГРАММЫ       f2:3    r2:72
    70  TERM    АЛГОРИТМ        f2:1    r2:61
    10  TERM    ИНФОРМАЦИЯ      f2:3    r2:62
    1A  TERM    ПРОДУКТ ПРОИЗВОДСТВА    f2:1    r2:61
...
RUB3 :: digit   f: 1    r: 99   eq: 1.0 sp: 1.0
    ==  RUB3    digit   f2:1    r2:99

RUB3 :: tech    f: 1    r: 99   eq: 1.0 sp: 1.0
    ==  RUB3    tech    f2:1    r2:99

RUB5    ::    КОМПЬЮТЕРНАЯ_ПРОГРАММА    АВТОР_РАЗРАБОТКИ    ИСХОДНЫЙ_КОД    CODE
            ОПЕРАЦИОННАЯ_СИСТЕМА_WINDOWS  f: 1    r: 33   eq: 1.0 sp: 1.0
    ==    RUB5       КОМПЬЮТЕРНАЯ_ПРОГРАММА  АВТОР_РАЗРАБОТКИ  ИСХОДНЫЙ_КОД  CODE
            ОПЕРАЦИОННАЯ_СИСТЕМА_WINDOWS  f2:1    r2:24

RUB5    ::    АДМИНИСТРИРОВАНИЕ    ДОКУМЕНТАЛЬНЫЙ_МАТЕРИАЛ    ИНФОРМАЦИЯ    ДОКУМЕНТ
            ДАННЫЕ_(СВЕДЕНИЯ) f: 1    r: 10   eq: 1.0 sp: 1.0
```

```
    ==   RUB5      АДМИНИСТРИРОВАНИЕ ДОКУМЕНТАЛЬНЫЙ_МАТЕРИАЛ ИНФОРМАЦИЯ ДОКУМЕНТ
           ДАННЫЕ_(СВЕДЕНИЯ) f2:1    r2:2

RUB5  ::  ИННОВАЦИЯ,_ВНЕДРЕННОЕ_НОВШЕСТВО  ТЕХНОЛОГИЯ  ИННОВАЦИОННАЯ_ТЕХНОЛОГИЯ
           НАУЧНО-ТЕХНИЧЕСКАЯ_СФЕРА НАУКА      f: 1    r: 9    eq: 0.4 sp: 0

RUB5   ::   УНИВЕРСИТЕТ   ГОСУДАРСТВЕННЫЙ_УНИВЕРСИТЕТ   ВЫСШЕЕ_УЧЕБНОЕ_ЗАВЕДЕНИЕ
           УЧЕБНЫЙ_ИНСТИТУТ ПЕДАГОГИЧЕСКАЯ_ДЕЯТЕЛЬНОСТЬ    f: 1     r: 8     eq:
           0.4 sp: 0

RUB5  ::  ОБЩЕОБРАЗОВАТЕЛЬНАЯ_ШКОЛА  ОБУЧАЮЩИЙ_ПРОЦЕСС  ШКОЛА  ШКОЛЬНЫЙ_УЧИТЕЛЬ
           ОБРАЗОВАТЕЛЬНАЯ_СИСТЕМА      f: 1    r: 6    eq: 1.0 sp: 1.0
    ==      RUB5           ОБЩЕОБРАЗОВАТЕЛЬНАЯ_ШКОЛА   ОБУЧАЮЩИЙ_ПРОЦЕСС   ШКОЛА
           ШКОЛЬНЫЙ_УЧИТЕЛЬ ОБРАЗОВАТЕЛЬНАЯ_СИСТЕМА      f2:1    r2:20
...
```

Затем данные файлы последовательно обрабатываются, исключая зоны: EQUAL, B_DET, B_GEN, B_ASC, в результате формируется B_LOST, затем аналогично формируются S_DET, S_GEN, S_ASC, S_LOST/

4.2.1.3. Программный модуль, реализующий алгоритм сравнения выявленных семантических структур содержания учебных курсов, выявления области пересечения и области несовпадения (Модуль № 08)

Назначение: Программный модуль, реализующий алгоритм сравнения выявленных семантических структур содержания учебных курсов, выявления области пересечения и области несовпадения.

Технологическое описание: модуль для сравнения данных двух курсов, для которых ранее с помощью других модулей были получены данные в виде различных индексов

Входные данные. На вход подается данные индексов .nld

Внутри модуля находится функция compare_2_log1, она сравнивает индексы двух файлов, в том числе по дереву

Использует файлы:

_fulltree.txt

ru_stopwords.txt

StopWords.L

Выходные данные. Функция сохраняет в файлы результат сравнения двух курсов. Результат – файлы, содержащие данные сравнения .res_mod08 в формате json.

Имя файла с исходным кодом: m08_comp_d1d2.py

Проверка функционирования

Пакетный файл: __test_mod08.bat

Директория с тестовыми данными __test08

API: Вызов функции compare_2_log1:

```
C:\Python38\python.exe
      -u m08_comp_d1d2.py
      --infile1 __test08\infile.txt.res_mod06
      --infile2 __test08\infile2.txt.res_mod06
      --usewiki 0
```

Здесь:

--infile1 – результат модуля 6 для первого курса

--infile2 – результат модуля 6 для второго курса

--usewiki – использование википедии (1) или не использование (0).

4.2.2. Разработка метрик, отражающих «похожесть» курсов

4.2.2.1. Методы формирования метрик, отражающих «похожесть» курсов

Общий подход заключается в том, чтобы оценить вклад различгых зон в степень похожести курсов, вклад разных объектов в зонах, обеспечить монотонность значения метрики при модификации сравниваемых текстов.

Принципы, закладываемые в функционал расчета метрики:

Наибольший позитивный вклад вносит совпадение в зоне EQUAL – совпадающих объектов;

Затем идет зона _DET – если имеются объекты, детализирующие объекты курса, то это считается более позитивным, чем наличие объектов в

зоне _GEN – в другом курсе используются более «обобщенные» объекты, связь имеется но она может быть слабой (идти по другой ветви иерархии).

Еще более слабый вклад вносят объекты из зоны _ASC – кроме того, здесь возрастает вероятность ошибки в определении силы ассоциативной связи.

Объекты из зоны _LOST входят в формулы расчета метрик как нормирующие – чем больше таких объектов, тем значение метрики связности будет меньше.

4.2.2.2. Алгоритм формирования метрик, отражающих «похожесть» курсов

Расчет метрик связности между двумя учебными курсами реализуется следующим образом:

$$S_2(T_1, T_2) = \alpha_{12} * S_0(T_1, T_2) + \alpha_{21} * S_0(T_2, T_1)$$

Схожесть текстов – взвешенная сумма похожести 1го на 2ой и, наоборот.

$$\alpha_{12} = \alpha_{21} = 0.5$$

$$S_0(T_1, T_2) = \beta_{TERM} * \Phi_{TERM}(T_1, T_2) + \beta_{TFM} * \Phi_{TFM}(T_1, T_2) + \beta_{LEMM} * \Phi_{LEMM}(T_1, T_2)$$

Взвешенная сумма схожести по терминам, словосочетаниям и отдельным словам.

$$\beta_{TERM} = 0.6 , \quad \beta_{TFM} = 0.2 , \quad \beta_{LEMM} = 0.2,$$

$$\Phi_{TERM}(T_1, T_2) =$$

$$= \frac{\sum_{TERM(i) \in T1} \omega_{ALOT}(TERM(i), T_1) * \chi_{equal}(TERM_i, T_2) * \varphi_{support}(TERM_i, T_2)}{\sum_{TERM(i) \in T1} \omega_{ALOT}(TERM_i, T_1))}$$

Схожесть по терминам – отношение сумм весов, штрафуемых в числителе за отклонение.

$\Phi_{TFM}(T_1, T_2)$ - вычисляется аналогично

Веса терминоподобных слов и словосочетаний определяем как *tf* * *idf*, где последний считается по Википедии

$$\chi_{equal}(TERM_i, T_2) = 1.0 - (1.0 - \gamma) * (abs(F_1-F_2) / (F_1+F_2))$$

Штрафует (но не до нуля, а только до $\gamma = 0.4$), если частотность термина в сравниваемых текстах отличается.

$$F_1 = freq(TERM_i, T_1) , \qquad F_2 = freq(TERM_i, T_2)$$

{ если $F_1 = F_2$, то $\chi_{equal}(TERM_i, T_2) = 1$;

если $F_1 \neq 0$, $F_2 = 0$, то $\chi_{equal}(TERM_i, T_2) = \gamma$ }

$$\varphi_{support}(TERM_i, T_2) = max \{ \theta_{equal}(TERM_i, T_2) , \theta_{OEHT\_LOW}(TERM_i, T_2) , \theta_{OEHT\_HIGH}(TERM_i, T_2) , \theta_{WIKI}(TERM_i, T_2) \}$$

здесь:

$\theta_{equal}(TERM_i, T_2) = 1$, если $TERM_i \in T_2$,

то есть $TERM_i$ *входит и в* T_1 *и в* T_2 ,

то есть термин из T_1 просто поддерживается таким же термином из T_2

$$\theta_{OEHT\_LOW}(TERM_i, T_2) = min\ (1.0, \sum_{TERM(j) \in T2} \delta_{LOW})$$

где $\delta_{LOW} = 0.3$ – то есть для поддержки отношениями (АСЦ = 50, НИЖЕ=30, ЧАСТЬ=40, АСЦ2=72) надо для 1.0 набрать не менее 3-4 (при $\delta_{LOW} = 0.3$ – четыре поддерживающих отношения)

$$\theta_{OEHT\_HIGH}(TERM_i, T_2) = min\ (1.0, \sum_{TERM(j) \in T2} \delta_{HIGH})$$

где $\delta_{HIGH} = 0.18$ – то есть для поддержки отношениями (ВЫШЕ=10, ЦЕЛОЕ=20, АСЦ1 =71) надо для 1.0 набрать не менее 5-6 (при $\delta_{HIGH} = 0.18$ – шесть поддерживающих отношений)

$$\theta_{WIKI}(TERM_i, T_2) = min\ (1.0, \sum_{TERM(j) \in T2} \delta_{WIKI})$$

где $\delta_{WIKI} = 0.12$ – то есть для поддержки отношениями (асц(WIKI)) надо для 1.0 набрать не менее 8-10 (при $\delta_{WIKI} = 0.12$ – девять поддерживающих отношений)

4.2.2.3. Программный модуль, реализующий алгоритм формирования метрик, отражающих «похожесть» курсов (Модуль № 09)

Назначение: Программный модуль, реализующий алгоритм формирования метрик, отражающих «похожесть» курсов.

Технологическое описание: модуль вычисления двух метрик. Внутри модуля находится функция calc_rang_d1_d2, она вычисляет метрики и итоговые ранги

Входные данные: На вход подается данные сравнения двух курсов в виде объектов содержащих связи индексных сущностей двух курсов (результат модуля № 08 для двух курсов).

Выходные данные: Функция возвращает рассчитанные ранги, файл с расширением .res_mod09, кладутся в папку указанную параметром --resdir

Имя файла с исходным кодом: m09_calc_metr.py

Проверка функционирования

Пакетный файл: __test_mod09.bat

Директория с тестовыми данными __test09

API: Вызов функции calc_rang_d1_d2:

```
C:\Python38\python.exe
      -u m09_calc_metr.py
      --infile1 __test09\infile.txt.res_mod06.res_mod08
      --infile2 __test09\infile2.txt.res_mod06.res_mod08
      --resdir __test09\
```

Здесь:

--infile1 – результат модуля 8 для первого курса

--infile2 – результат модуля 8 для второго курса

--resdir – папка для сохранения файлов результатов

4.2.3. Разработка методов и алгоритмов формирования отчетных документов для визуализации в табличной форме и на графах, представлений сравнения понятийно-тематического пространств учебных курсов

4.2.3.1. Методы представления результатов сравнения понятийно-тематических пространств учебных курсов в результате индексирования материалов учебных курсов

Результатом работы модуля № 08 является разбиение понятийно-тематического пространства сравниваемых курсов по зонам связности объектов. Имеется одна зона общих объектов, а также «парные» зоны B_DET и S_DET, B_GEN и S_GEN, B_ASC и S_ASC, B_LOST и S_LOST,

Это определяет естественную табличную структуру сравнения двух курсов:

EQUAL	
B_DET	S_DET
B_GEN	S_GEN
B_ASC	S_ASC
B_LOST	S_LOST

Опыт эксплуатации показал, что наибольший интерес представляет сравнение наиболее значимых объектов. Поэтому интерфейсно добавочно решается вопрос с выбором ограниченного количества объектов, чтобы улучшить визуальное восприятие.

Другой проблемой, которая решается при визуализации является фильтрация ошибочных вариантов, которые возниают по следующим

причинам фундаментальным (сложно устранимым автоматическим способом):

- Прошедшие фильтрацию паразитные элементы обрамления;
- Многозначные текстовые объекты, при выборе правильного значения которых были допущены ошибки;
- Слишком общие объекты, которые не представляют интереса при сравнении содержания конкретных курсов.

Предложено простое решение поддержки в базе данных фильтров объектов, включая «глобального фильтра» - одного для всех пользователей и сравнения курсов, а также курсов и рефлексий, и «локального» - только в рамках текущего задания сравнения.

Пользователь может убрать в фильтр любой элемент, который после этого не будет появляться в табличных и/или графовых визуализациях сравнения.

Представляется, что при наличии общих характеристик в коллекции курсов и рефлексий «глобальный фильтр» достатчоно быстро будет сформирован, что обеспечит улучшение качества визуализации сравнения учебных курсов.

Дополнительно к табличному представлению разработано представление результатов на гистограммах.

Для графового представления результатов сравнения двух курсов предложен следующий подход:

На графе предусматриваются узлы-«агрегаторы»:

- BASE («Курс 1») и SAMPLE («Курс 2»);
- EQUAL, причем объекты зоны связываются как с узлом EQUAL, так и с узлами BASE и SAMPLE;

- B_DET, B_GEN, B_ASC, B_LOST, причем объекты соответствующих зон, связываются как с узлом-агрегатором, так и с узлом BASE;
- Аналогично для S_DET, S_GEN, S_ASC, S_LOST/

4.2.3.2. Алгоритм представления результатов сравнения понятийно-тематических пространств учебных курсов в результате индексирования материалов учебных курсов в табличной форме и на графах

Первоначально табличное представление формируется автоматически в виде HTML файла, с дублированием всей информации в виде .json файла.

Дальнейшая работа по корректировке табличного представления, которое является базовым производится в интерфейсе путем задания фильтров и количества отображаемых объектов.

В результате пользователь получает результат в табличной форме, форме гистограмм или графовой форме с учетом наложенным фильтров.

В графовом представлении для улучшения визуального восприятия объекты зоны располагаются на диагонали, проходящей через точку на отрезке, соединяющем соотвествующие узля-агрегаторы в отношении 2:1 от узла-«курса».

Диагональ наклонена по отношении к вертикали на 20 градусов. Направление отклонения от вертикали определяется знаком произведения разницы координат соединяемых узлов-агрегаторов: (x1-x2)*(y1-y2). Расстояние между узлами по вертикали определяется как размер экрана по вертикали, деленный на количество объектов в зоне, умноженном на четыре (чтобы помещалось три зоны и хватило метса на промежутки).

4.2.3.3. Программный модуль, реализующий алгоритм представления результатов сравнения понятийно-тематических пространств учебных курсов в результате индексирования материалов учебных курсов в табличной форме и на графах (Модуль № 10)

Технологическое описание: модуль создает итоговые таблицы результата сравнения двух курсов

Входные данные: На вход подается данные сравнения двух курсов в виде объектов содержащих связи индексных сущностей двух курсов

Выходные данные: Функция сохраняет в файлы с таблицами результатов сравнения двух курсов: res1.htm, res_rub5.htm, res_rub3.htm, res_rubx.htm, res_rubr.htm, res2.htm, res2.json.

Папка для сохранения передается параметром

Имя файла с исходным кодом: m10_make_res.py

Проверка функционирования

Пакетный файл: __test_mod10.bat

Директория с тестовыми данными __test10

API: Вызов функции make_compare_d1_res_file:

```
C:\Python38\python.exe
      -u m10_make_res.py
      --infile1 __test10\infile.txt.res_mod06.res_mod08.res_mod09
      --infile2 __test10\infile2.txt.res_mod06.res_mod08.res_mod09
      --resdir __test10\
```

Здесь:

--infile1 – результат модуля 9 для первого курса

--infile2 – результат модуля 9 для второго курса

--resdir - папка для сохранения результатов

5. РАЗРАБОТКА МЕТОДОВ И АЛГОРИТМОВ СРАВНЕНИЯ СОДЕРЖАНИЯ УЧЕБНО-МЕТОДИЧЕСКИХ МАТЕРИАЛОВ УЧЕБНЫХ КУРСОВ С РЕФЛЕКСИЕЙ ОБУЧАЮЩИХСЯ

В разделе Б.3 Приложения Б приведен типичный пример данных рефлексии обучаемых, предоставленный при проведении работы.

Анализ данных свидетельствует:

- В текущих условиях обучаемые сообщают часто весьма общую информацию по предмету курса;
- Часто их замечания касаются формы, а не содержания курса.

Здесь существует несколько проблем:

- Проблема получить от обучаемых адекватную реакцию по содержанию курса при задании только «общих» вопросов на адекватные ответы на которые у обучаемых может не хватать компетенции;
- Проблема трудоемкости сбора данных рефлексии, что было бы желательно делаит с использованием средств распознавания речи, которые к сожалению имеют недостаточно высокие характеристики при распознавании речи, содержащей специальные термины.

5.1. Разработка метрик, отражающих степень усвоения материалов курса обучающимся

5.1.1. Методы формирования метрик, отражающих степень усвоения материалов курса обучающимся

В целом, подход сравнения содержания курса и рефлексии аналогичен расчету сравнения содержания двух курсов, но имеет свои особенности, так как рефлексия является связанным контентом от содержания курса.

Отличие - специальная модель обработки S_LOST – «оригинального» для рефлексии. Если текстовый объект в S_LOST повторяется три (вообще говоря, параметр, но можно зафиксировать) и более раза – тогда мы считаем, что на самом деле такой объект был в курсе, но мы его «не видим» по тексту. Поэтому все такие объекты волевым образом переносятся S_ASC.

5.1.2. Алгоритм формирования метрик, отражающих степень усвоения материалов курса обучающимся

Расчет схожести курса и рефлексии аналогичен расчету схожести между двумя курсами.

Отличия заключаются в следующем:

- При расчете сходства от курса к рефлексии $S_0(T_1, T_2)$

  $\delta_{LOW} = 0.50$ *(для курсов было* $\delta_{LOW} = 0.3$*)*

  $\delta_{HIGH} = 0.10$ *(для курсов* $\delta_{HIGH} = 0.18$*)*

- При расчете сходства от рефлексии к курсу $S_0(T_2, T_1)$

  $\delta_{LOW} = 0.10$ *(для курсов было* $\delta_{LOW} = 0.3$*)*

  $\delta_{HIGH} = 0.50$ *(для курсов было* $\delta_{HIGH} = 0.18$*)*

Значения «перевернуты», так как для пары курс-рефлексия имеется зависимость между документами – рефлекси является производной от курса, поэтому можно считать, что B_DET сопряжен с S_GEN, а B_GEN с S_DET.

5.1.3. Программный модуль, реализующий алгоритм формирования метрик, отражающих степень усвоения материалов курса обучающимся (Модуль № 11)

Назначение: Программный модуль, реализующий алгоритм формирования метрик, отражающих степень усвоения материалов курса обучающимся.

Технологическое описание: модуль вычисления метрик для курса и рефлексии. Внутри модуля находится функция calc_rang_d1_d2, она вычисляет метрики и итоговые ранги.

Входные данные: На вход подается данные сравнения курса и рефлексии в виде объектов содержащих связи индексных сущностей (результат модуля № 14 для курса и рефлексии).

Выходные данные: Функция возвращает рассчитанные ранги

Имя файла с исходным кодом: m11_calc_metr.py

Проверка функционирования

Пакетный файл: __test_mod11.bat

Директория с тестовыми данными __test11

API: Вызов функции calc_rang_d1_d2:

```
C:\Python38\python.exe
      -u m11_calc_metr.py
      --infile1 __test11\infile.txt.res_mod06.res_mod14
      --infile2 __test11\infile2.txt.res_mod13.res_mod14
      --resdir __test11\
```

Здесь:

--infile1 – результат модуля № 14 для курса

--infile2 – результат модуля « 14 для рефлексии

--resdir – папка для сохранения файлов результатов

5.2. Разработка рекомендаций по формированию вопросников обучающихся по материалам прослушанных курсов для оптимизации процедуры автоматизации оценки степени усвоения материалов курса обучающимся

Требуется разработать формы вопросников (возможно, собираемых в различное время – до, во время или по окончанию курса), включая типы вопросов, подразумевающих текстовые ответы, а также методы и алгоритмы автоматизированного формирования таких вопросников.

Также разработать методы и алгоритмы, позволяющие автоматизировано оценить по результатам заполнения вопросников:

- уровень знаний и навыков, имевшихся у обучаемых до прослушивания курса;
- возможно, ожиданий, имевшихся у обучаемых от курса;
- знаний, которые получил обучаемый в результате курса, в том числе с градацией на хорошо или недостаточно усвоенные;
- навыков, которые улучшил обучаемый в результате прослушивания курса, в том числе с градацией на хорошо или недостаточно усвоенные;
- замечаний по дидактическим и методологическим вопросам представления материалов курса, формам и методам преподавания.

5.2.1. Методы формирования вопросников обучающихся по материалам прослушанных курсов для оценки знаний и навыков обучаемых

Предполагается, что по курсу имеется список терминов, упомянутых в текстах, соответствующих курсу. Эти термины взяты из ОЕНТ или Википедии. Термины упорядочены по значимости в курсе, например, по тематическому представлению и/или по частотности. Например, термины, которые не входят в тематическое представление, могут иметь в качестве

веса нормализованную частотность (частотность термина по отношению к максимальной частотности термина в материалах курса).

Предполагается, что с использованием автоматического анализа можно оценить значимость конкретных терминов, входящих в курс, а также оценить значимость взаимосвязи между парой теминов.

Предлагается формулировать для слушателя в рамках одного опроса три задания следующих типов:

- Задание 1 – назвать 10-12 наиболее значимых терминов курса. Список таких терминов по сути представлвет собой определенный подвид рефлексии и может быть оценен как текст относительно содержания курса.

  Сначала составить список всех «терминов слушателей» с их частотностями.

  Результат – список «терминов слушателей» с частотками, упорядоченный по убыванию частотки и по возрастанию алфавита.

  Далее трактуем список «терминов слушателей» как текст рефлексии и сравниваем меру сходства курса и рефлексии.

- Задание 2 – система предлагает пользователю список терминов, которые набираются из разных «корзин» - наиболее значимых терминов, менее знаимых, а также случайных. Представляется, что слушатели курса должны уверено определять значимые и случайные термины.

- Задание 3 – аналогично заданию два, но пользователю предъявляются не термины, но пары терминов.

Результаты опросов накапливаются в базе данных и могут быть использованы для оценки степени усвоения материалов учебного курса.

5.2.2. Алгоритм формирования вопросников обучающихся по материалам прослушанных курсов для оценки знаний и навыков обучаемых

5.2.2.1. Алгоритм формирования вопросников

Формируется список значимых терминов – далее «термины курса».

- Задание 1. Назвать основные термины курса (3-5).

  Цель – получить альтернативный способ формирования рефлексии о курсе в виде «аннотации списком терминов».

  Здесь одна функция- суммарная оценка ответов слушателей (возможно, потом понадобится еще оценка отдельных слушателей).

  Входные данные – текстовый файл, содержащий список пар

  UserID <разделитель> строка «термин слушателя»

- Задание 2. Упорядочить предложенные термины по их значимости для курса.

  Пусть имеется упорядоченный по убыванию значимости списки текстовых объектов курса TERM.

  Цель вопросника определить, насколько адекватно слушатели поняли основное в курсе и контрольно определить степень доверия к слушателям.

  Предполагается, что испытуемые должны адекватно определить основные термины курса и уверено определить нерелевантные термины.

  Здесь две функции:

  - формирование вопросника
  - оценка результатов ответа на вопросник

2_1. Формирование вопросника «второго типа»

Формируется список «терминов курсов», содержащий

- N1 объектов из верхней части списков (группа G1),
- N2 объектов из средней части списков (группа G2),
- N3 объектов из нижней части списков (группа G3),
- N4 объектов не из списков (группа G4).

Списки делятся следующим образом:

- 12 первых – верхняя часть списка (возможно, разные квоты для разных типов объектов)
- от 13 до 30 – средняя часть списка
- далее 30 – нижняя часть списка

На вход поступает также параметр о количестве создаваемых вопросников.

N1 = 4-6, N2 = 4-6, N3 = 4-6, N4 = остаток до 20

N1 = 4-6 – означает, что количество объектов – случайно от 4 до 6.

(для простоты сначала можно взять N1 = 5, N2 = 5, N3 = 5, N4 = 5)

Результат – список упорядоченных по алфавиту объектов, у каждого указано: тип TERM, к какой части списка принадлежит (G1,…,G4).

voprID <разд.> термин <разд.> индекс по G

Здесь voprID – идентификатор вопросника.

Примечание – нерелевантные термины случайным образом выбираются из SCI_THES (лучше всего подняться от случайного TERM курса вверх на один-два шага, затем на два шага вниз, но не из терминов курса. Для простоты – поначалу просто выбрать случайный из Revry к классификатору «Таксономия 2035»).

2_2. Оценка ответов пользователей по вопроснику «второго типа»

Предполагается, что в интерфейсе слушателям предъявляется упорядоченный по алфавиту список объектов и предлагается поставить рядом оценку важности для содержания курса:

-- 3 (важный термин) (группа H1)

-- 2 (термин, релевантный теме) (группа H2)

-- 1 (упоминавшийся термин) (группа H3)

-- 0 (нерелевантный термин) (группа H4)

Поэтому входными данными для функции является текстовый файл, содержащий строчки типа:

userID <разделитель> термин <разделитель> оценка <разделитель> индекс по G

Лучше сразу преобразовать в вид

userID <разделитель> термин <разделитель> индекс по H <разделитель> индекс по G

Считается степень доверия к ответу, что определяется по верно определенным нерелевантным терминам.

Пусть

M4 – общее количество объектов, которые слушатель отнес к группе G4

K4 – количество правильных объектов, которые пользователь отнес к группе G4.

Тогда

F1 (G4) = 2.0 * P(G4) * R(G4) / (P(G4) + R(G4))

где P(G4) = K4/M4, R(G4) = K4/N4

То есть

F1 (G4) = 2.0 * K4 * / (M4 + N4)

Оцениваем правильность ответов слушателя по группе G1

F1 (G1) = 2.0 * K1 * / (M1 + N1)

Результат – усредненные значения F1(G1) и F1(G4) по всем пользователям.

Также для каждого объекта выводится его группа G и среднее значение (по всем слушателям) абсолютного отклонения в по индексу групп G и H (то есть если должен быть в G1, попал в H3 – отклонение = 2, должен быть в G4, попал в H1 – отклонение = 3).

термин <разделитель> индекс по G <разделитель> среднее отклонение <разделитель> частотка во входном файле

Сортировка – по группам G, по убыванию среднего, по убыванию частотки

Примечание - При формировании отчета будут учитываться объекты из G1, G2, которые имеют минимальное и максимальное отклонения (хорошо или плохо «усвоились»).

- Задание 3. Упорядочение пар терминов по их смысловой близости

Пусть имеется упорядоченный по убыванию значимости список терминов онтологии TERM.

Цель вопросника определить, насколько адекватно слушатели поняли взаимосвязи между терминами курса и контрольно определить степень доверия к слушателям.

Предполагается, что испытуемые должны адекватно определить связи основных терминов курса и уверено определить ошибочные связи.

Здесь две функции:

-- формирование вопросника «третьего типа»

-- оценка результатов ответа на вопросник

3_1. Формирование вопросника «третьего типа»

Сначала отбираем наиболее значимые термины курса (случайно L1 из первых 20)

Затем L2 из терминов в списке от 21 до 40

(Пусть L1 = 6, L2 =9)

Затем выписываем для них пары с отношениями, случайно выбирая :

- синонимичные отношения (левая/правая часть не должны совпадать) (группа G1)
- отношения на один уровень (может быть два уровня по дереву), где оба TERM должны быть в списке терминов курса (группа G2)
- отношения дальше чем на два уровня по дереву (группа G3)
- не связанные по дереву (лучше с переломом по дереву с длиной пути не меньше 4 – чтобы было посложнее. Для простоты можно случайно).

На вход поступает также параметр о количестве создаваемых вопросников.

Результат - список в формате

voprID <разд.> «Термин1» <разд.> «Термин2»

<разд.> индекс по L <разд.> индекс по G

Здесь

voprID – идентификатор вопросника.

«Термин1» - это TERM (то есть дескриптор SCI_THES – наименование концепта из списка L1 или L2),

«Термин2» - либо синоним, либо какой-то TERM.

3_2. Оценка результатов ответа на вопросник «третьего типа»

Предполагается, что в интерфейсе слушателям предъявляется упорядоченный по алфавиту список пар объектов, не показывая вид

отношения и предлагается выбрать одно из отношений для содержания курса:

-- 3 (это одно и тоже) (группа H1)

-- 2 (сильно связаны) (группа H2)

-- 1 (связаны) (группа H3)

-- 0 (никак не связаны) (группа H4)

Далее – аналогичные подсчеты как в 2_2.

То есть на вход поступает что-то типа

userID <разд.> «Термин1» <разд.> «Термин2» <разд.> оценка <разд.> индекс по L <разд.> индекс по G

который преобразуем в

userID <разд.> «Термин1» <разд.> «Термин2» <разд.> индекс по H <разд.> индекс по L <разд.> индекс по G

и далее аналогично.

5.2.2.2. Алгоритм интеграции вопросников в интерфейс (Модуль № 20).

Всего три основных функции и еще одна-две вспомогательные.

Общая логика:

- «слушателю» предлагается вопросник из трех заданий
- «слушатель» отвечает на вопросы заданий
- на основании ответов «слушателей» формируется оценка степени усвоения слушателями материалов курса

При этом:

- Задание № 1. Назовите 10-12 наиболее важных терминов прослушанного курса, разделяя их символом «точка с запятой»

  В качестве ответа предполагается список терминов (строка), разделенных «;».

  В базу данных вносится запись

CourseID -- UserID -- OprosID -- Z1 -- строка терминов

где OprosID – идентификатор предъявляемого впросника из трех заданий.

- Задание № 2. Оцените предложенные термины по значимости по шкале: 1 (не релевантный), 2 (упоминался), 3 (релевантный), 4 (важный).

  По умолчанию стоит 0 – как признак отсутствия ответа. При этом желательно заставить пользователя проставить ответ – нули не пропускать.

  Список терминов выбирается из базы данных на основании ранее выполненной функции формирования задания №2 («2_1»). При этом в базе проставлены предполагаемые группы важности.

  То есть в базе есть группы связанных терминов типа:

  CourseID -- Z2 -- VoprID -- ТЕРМИН -- индекс группы G

  (для Задания №2 заранее формируется множество вопросников – групп терминов с оценками, которые потом случайно выбираются для предъявления пользователю).

  (Индекс G от пользователя скрыт).

  Ответ заключается в том, что в базу данных ответов вносятся записи вида

  CourseID -- UserID -- OprosID -- Z2 -- VoprID -- ТЕРМИН

  -- индекс группы G

  -- оценка_пользователя

- Задание № 3. Оцените силу связности предложенных пар терминов по значимости для материалов курса по шкале: 1 (не релевантный), 2 (ассоциация), 3 (сильно связаны), 4 (одно и то же).

По умолчанию стоит 0 – как признак отсутствия ответа. При этом желательно заставить пользователя проставить ответ – нули не пропускать.

Список пар выбирается из базы данных на основании ранее выполненной функции формирования задания №3 («3_1»). При этом в базе проставлены предполагаемые группы важности.

То есть в базе есть группы связанных терминов типа:

CourseID -- Z3 -- VoprID -- ТЕРМИН1 -- ТЕРМИН2

-- индекс группы G

(для Задания №3 заранее формируется кучка вопросников – групп пар терминов с оценками, которые потом случайно выбираются для предъявления пользователю).

Ответ заключается в том, что в базу данных ответов вносятся записи вида

CourseID -- UserID -- OprosID -- Z2

-- VoprID -- ТЕРМИН1 -- ТЕРМИН2

-- индекс группы G -- оценка_пользователя

5.2.2.3. Требования к функционалу веб-сервиса

Основные функции:

- (Функция № 20-12-1).Сформировать вопросники

  Вызовом функций 12_2_1, 12_3_1 предварительно формируются записи типа

  CourseID -- Z2 -- VoprID -- ТЕРМИН -- индекс группы G

  CourseID -- Z3 -- VoprID -- ТЕРМИН1 -- ТЕРМИН2

  -- индекс группы G

(хранить их в одной таблице или разных – на выбор)

- (Функция № 20-12-2). Функция сопровождения опроса слушателя

  При активации - формируется OprosID и соответствующий вопросник с тремя подформами для заданий №№ 1-3. При этом вопросники по заданиям №№ 2-3 выбираются случайно (можно брать просто следующий по VoprID – т.к. они случайным образом генерятся).

  Пользователь заполняет формы. Результаты форм записываются в базу данных.

- (Функция № 20-12-3). Функция получения оценок усвоения.

  Вызов функции 12_2_2, куда передается json json со всеми записями типа:

  CourseID -- UserID -- OprosID -- Z2 -- VoprID -- ТЕРМИН

  -- индекс группы G -- оценка_пользователя

  Результат – ответный json, форма его представления будет уточнена.

  Вызов функции 12_3_2, куда передается json со всеми записями типа:

  CourseID -- UserID -- OprosID -- Z2 -- VoprID

  -- ТЕРМИН1 -- ТЕРМИН2

  -- индекс группы G -- оценка_пользователя

  Результат – ответный json, форма его представления будет уточнена.

  Вызов функции 12_1, куда передается json со всеми записями типа

  CourseID -- UserID -- OprosID -- Z1 -- строка терминов

Результат – ответный json, форма его представления будет уточнена.

5.2.3. Программный модуль, реализующий алгоритм формированию вопросников обучающихся по материалам прослушанных курсов для оценки знаний и навыков обучаемых (Модуль № 12)

Назначение: Программный модуль, реализующий алгоритм формирования вопросников для обучающихся по материалам прослушанных курсов для оценки знаний и навыков обучаемых.

Входные данные: На вход подается параметр –fn – определяющий этап построения

Возможные значения 0, 1, 21, 22, 31, 32

Так же на вход подается папка --dir, в которой в зависимости от параметра fn должны находится файлы:

- kurs.txt – текст курса
- m12_in2_2.txt – входные данные для 2.2
- m12_in3_2.txt – входные данные для 3.2
- m12_res2_1.txt – результат 2.1
- m12_res2_2.json – результат 2.2
- m12_res3_1.txt – результат 3.1
- m12_res3_2.json – результат 2.3
- opros.txt – данные терминов пользователей для 1
- kurs.txt.term_res - – входные данные для 2.1 и результат 0

Используется соединение с СУБД PostgreSQL для получения данных лингвистической онтологии.

Выходные данные: Функция сохраняет в файлы результатов.

Имя файла с исходным кодом: m12_voprosnik.py

Проверка функционирования

Пакетные файлы:

__test_mod12_0.bat

__test_mod12_1.bat

__test_mod12_21.bat

__test_mod12_22.bat

__test_mod12_31.bat

__test_mod12_32.bat

Директория с тестовыми данными __test_m12.bat

API: Вызов функции m12_voprosnik.py:

```
C:\Python38\python.exe
      -u m12_voprosnik.py
      --fn 0
      --dir __test_m12\
```

5.3. Разработка методов и алгоритмов представления научно-технического пространства материалов рефлексии обучаемых, в том числе расширенного, в общем пространстве знаний и навыков

5.3.1. Методы представления научно-технического пространства материалов рефлексии обучаемых, в том числе расширенного в общем пространстве знаний и навыков

По результатам анализа данных выяснилось, что в целом методы представления научно-технического пространства материалов рефлексии обучаемых не отличаются от методов представления представления научно-технического пространства материалов учебных курсов.

5.3.2. Алгоритм представления научно-технического пространства материалов рефлексии обучаемых, в том числе расширенного в общем пространстве знаний и навыков

Алгоритм представления научно-технического пространства материалов рефлексии обучаемых в целом не отличается от аналогичного для учебных курсов, включая вызовы модулей № 01, № 03, № 05.

Отличие заключается в том, что для очистки данных рефлексии используется другой файл игнорируемых текстовых выражений.

5.3.3. Программный модуль, реализующий алгоритм представления научно-технического пространства материалов рефлексии обучаемых, в том числе расширенного в общем пространстве знаний и навыков (Модуль № 13)

Назначение: Программный модуль, реализующий алгоритм представления научно-технического пространства материалов рефлексии обучаемых, в том числе расширенного в общем пространстве знаний и навыков.

Входные данные: На вход подается текст. Внутри модуля находится функция make_nld_full, в нее параметром передается текст файла,

Выходные данные: Функция возвращает объект содержащий данные индекса найденных сущностей. Результирующий файл сохраняется в исходное наименование + .res_mod13

Имя файла с исходным кодом: m13_make_nld_full.py

Проверка функционирования

Пакетный файл: __test_mod13.bat

Директория с тестовыми данными __test13

API: Вызов функции make_nld_full:

```
C:\Python38\python.exe
    -u m13_make_nld_full.py
    --infile __test06\infile.txt
```

Здесь: --infile – имя входного файла

5.4. Разработка методов и алгоритмов сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса, в том числе расширенного, в общем пространстве знаний и навыков

5.4.1. Методы сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса, в том числе расширенного, в общем пространстве знаний и навыков

Методы сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса в целом не отличаются от методов сравнения двух учебных курсов.

За исключением того, что содержимое рефлексии является производным текстом от текстов материалов учебного курса, что учитывается при учете повторов в зоне S_LOST (неоднократное повторение слушателями текстового объекта трактуется ка наличие такого объекта в курсе, или наличие связанного объекта).

5.4.2. Алгоритм сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса, в том числе расширенного, в общем пространстве знаний и навыков

Алгоритм сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса в целом не отличается от алгоритма сравнения двух учебных курсов.

5.4.3. Программный модуль, реализующий алгоритм сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса, в том числе расширенного, в общем пространстве знаний и навыков (Модуль № 14)

Назначение: Программный модуль, реализующий алгоритм сравнения научно-технических пространств (обобщенных пространств) материалов рефлексии обучаемых и материалов учебного курса, в том числе расширенного, в общем пространстве знаний и навыков.

Входные данные: На вход подается данные индексов - результаты модуля № 06 обработки данных материалов учебного курса и модуля № 13 обработки данных материалов рефлексии. Внутри модуля находится функция compare_2_log1_rfl, она сравнивает индексы двух файлов, в том числе по иерархии онтологии.

Использует файлы:

_fulltree.txt

ru_stopwords.txt

StopWords.L

Выходные данные: Функция сохраняет в файл результат сравнения двух курсов .res_mod14 в формате json.

Имя файла: m14_comp_d1d2.py

Проверка функционирования

Пакетный файл: __test_mod14.bat

Директория с тестовыми данными __test14

API: Вызов функции compare_2_log1_rfl:

```
C:\Python38\python.exe
    -u m14_comp_d1d2.py
    --infile1 __test14\infile.txt.res_mod06
    --infile2 __test14\infile2.txt.res_mod13
    --usewiki 0
```

Здесь:

--infile1 – результат модуля 6 для курса

--infile2 – результат модуля 13 для рефлексии

--usewiki – использование вики(1) или без него (0)

5.5. Разработка методов и алгоритмов формирования отчетных документов для визуализации в табличной форме и на графах, представлений оценки степени усвоения материалов курса обучающимся

5.5.1. Методы представления результатов оценки степени усвоения материалов курса обучающимся

Методы представления результатов сравнения содержания учебного курса и рефлексии аналогичны методам представления результатов сравнения двух курсов.

Отличие различается в наименовании зон:

- EQUAL – «Усвоено» - слушатели воспроизводят термины курса;
- B_DET – «Курс. Понятно»;

- S_GEN – «Рефлексия. Понятно»;
- B_GEN – «Курс. Нечетко»;
- S_DET – «Рефлексия. Нечетко»;
- B_ASC – «Курс. Неуверено»;
- S_ASC – «Рефлексия. Неуверено»;
- B_LOST – «Пропущено»;
- S_LOST – «Добавлено».

5.5.2. Алгоритм представления результатов оценки степени усвоения материалов курса обучающимся в табличной форме и на графах

Алгоритм аналогичен (с точностью до обозначения зон и уточненной трактовки неоднократно названных объектов в S_LOST) алгоритму сравнения двух учебных курсов.

5.5.3. Программный модуль, реализующий алгоритм представления оценки степени усвоения материалов курса обучающимся в табличной форме и на графах (Модуль № 15)

Назначение: Программный модуль, реализующий алгоритм представления оценки степени усвоения материалов курса обучающимся в табличной форме и на графах.

Технологическое описание: модуль создает итоговые таблицы результата сравнения курса и рефлексии

Входные данные: На вход подается данные сравнения курса и рефлексии в виде объектов содержащих связи индексных сущностей двух курсов

Выходные данные: Функция сохраняет в файлы с таблицами результатов сравнения двух курсов

res1.htm, res_rub5.htm, res_rub3.htm, res_rubx.htm, res_rubr.htm, res2.htm, res2.json

Папка для сохранения передается параметром

Имя файла с исходным кодом: m15_make_res.py

Проверка функционирования

Пакетный файл: __test_mod15.bat

Директория с тестовыми данными __test15

API: Вызов функции make_compare_d1_res_file:

```
C:\Python38\python.exe
      -u m15_make_res.py
      --infile1 __test15\infile.txt.res_mod06.res_mod14.res_mod11
      --infile2 __test15\infile2.txt.res_mod13.res_mod14.res_mod11
      --resdir __test15\
```

Здесь:

--infile1 – результат модуля 14 для курса

--infile2 – результат модуля 14 для рефлексии

--resdir - папка для сохранения результатов

6. РАЗРАБОТКА МЕТОДОВ И АЛГОРИТМОВ АВТОМАТИЧЕСКОГО ПОПОЛНЕНИЯ СЛОВАРЕЙ ДЛЯ СИСТЕМ РАСПОЗНАВАНИЯ РЕЧИ, ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА РАСПОЗНАВАНИЯ РЕЧИ ЛЕКТОРОВ И ОБУЧАЮЩИХСЯ

6.1. Разработка методов и алгоритмов сравнительного анализа лексики и терминологии: содержимого транскриптов текстов, относящихся к тематике учебных курсов, общего предметного поля по тематике учебного курса

6.1.1. Методы сравнительного анализа лексики и терминологии содержимого транскриптов текстов, относящихся к тематике учебных курсов, общего предметного поля по тематике учебного курса

Исследование вопроса о качестве распознавания изложения учебных материалов курса или рефлексии в речи выявил следующие особенности:

- Современные системы распознавания речи в основном основаны на нейросетевых подходах. Нейронные сети обучаются на больших корпусах текстов, но из-за ограниченности размера производят определенное огрубление, игнорируя относительно редкие слова и выражения.
- В результате при появлении на вход неизвестного нейронной сети слова, оно пропускается, не сообщая ничего пользователю.
- Выходом является возможность использования пользовательских словарей. Такую возможность предоставляет, например, сервис Amazon Transcribe (см. Приложение В), позволяя подключать словарь величиной до 50 Кбайт (примерно 2500 русскоязычных слов в кодировке UTF-8).

С учетом обычно ограниченного размера текста описания учебного курса задача может быть сформулирована следующим образом – на основе краткого описания необходимо сделать предположение о составе лексики учебного курса и/или рефлексии обучаемых для улучшения качества распознавания.

Предлагается в качестве основы взять наиболее характерные текстовые объекты, выделенные ПП АЛОТ из анализируемого текста, и затем расширить данный список за счет ассоциированных объектов, которые определяем в документах Википедии.

Примечание – Требует отдельного исследования, нужно ли включать в словарь разные словоформы одного слова. С одной стороны, это может несколько повысить качество, так как автоматически они не строятся — сказали в инфинитиве, значит будем везде писать инфинитив. Обычно, сервис считает формы близкими к инфинитиву, оно помечает такое слово «желтым» или даже «зеленым» («уверен»), иногда прикрепляя мелкое слово «красного» цвета после инфинитива, созвучное с окончанием. То есть если добавить, то качество повысится, но не очень сильно.

С другой стороны, размер словаря может таким образом увеличиваться в 10+ раз сокращая разнообразие лексики словаря.

Примечание - Ложные срабатывания могут быть на близкие слова, а также на всякие короткие слова типа «оэс» из старого текста, которое могло иногда появляться на пустом месте или пытаться встраиваться в разницу инфинитива и словоформы для слов из словаря. Поэтому короткие слова желательно не включать в словарь.

6.1.2. Алгоритмы сравнительного анализа лексики и терминологии содержимого транскриптов текстов, относящихся к тематике учебных курсов, общего предметного поля по тематике учебного курса

Для поданного на вход текста :

(1) выделить группы наиболее значимых объектов (TERM, TFM, LEMM) — по весу или количеству:

- вес терминов — по АЛОТ
- вес словосочетаний и лемм — по tf*idf(news) по текстовой коллекции Википедии.

Здесь:

tf = freq / (1. + freq), idf = lg10 (N0/ df)

N0 — количество документов в коллекции

df — количество документов, содержащих объект.

Если df < 10, то df :=10.

(2) Образовать предварительный список по тексту — по заданному количеству .

(3) Зафиксировать некоторые параметры:

- количество объектов (возможно, по типам) (другое чем в предварительном списке)
- количество запросов == 1 на каждый тип - леммы, термины, TFM
- количество объектов в запросе == все из соответвующего списка
- количество документов в выдаче ==200

(4) Выполняем для Википедии зафиксированное количество запросов вида:

object(i) OR object(m) OR object(n) OR ….

где object'ы выбираются случайно из зафиксированного множества.

Отбираем фиксированное количество документов в выдаче. Не обращаем внимание на повторы документов — с повторами пусть будет повышен вес.

(5) Выделяем объекты из отобранных документов.

6.1.3. Программный модуль, реализующий алгоритмы сравнительного анализа лексики и терминологии содержимого транскриптов текстов, относящихся к тематике учебных курсов, общего предметного поля по тематике учебного курса (Модуль № 16)

Назначение: Программный модуль, реализующий алгоритмы сравнительного анализа лексики и терминологии содержимого транскриптов текстов, относящихся к тематике учебных курсов, общего предметного поля по тематике учебного курса.

Технологическое описание: модуль создает списки слов – лемм терминов и словоформ на основе вики-поиковика и mtfod

Входные данные: На вход подается имя файла с текстом – материалы учебного курса и/или рефлексии.

Использует файлы:

ru_stopwords.txt

StopWords.L

Выходные данные: Функция сохраняет в файлы найденные списки дописывая к имени файла: *.res16

Имя файла с исходным кодом: m16_makelist.py

Проверка функционирования

Пакетный файл: __test_mod16.bat

Директория с тестовыми данными __test_m16m17

API: Вызов функции m16_makelist.py:

```
C:\Python38\python.exe
    -u m16_makelist.py
    --fn __test_m16m17\kurs.txt
```

Здесь: --fn – параметр имени файла

6.2. Разработка методов и алгоритмов составления ранжированного словаря лексики и терминологии общего предметного поля для автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся

6.2.1. Методы составления ранжированного словаря лексики и терминологии общего предметного поля для автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся

На предыдущем этапе в модуле № 16 были получены списки разных типов объектов, которые могут быть релевантны анализируемому тексту – отдельные слова, словосочетания, понятия онтологиии.

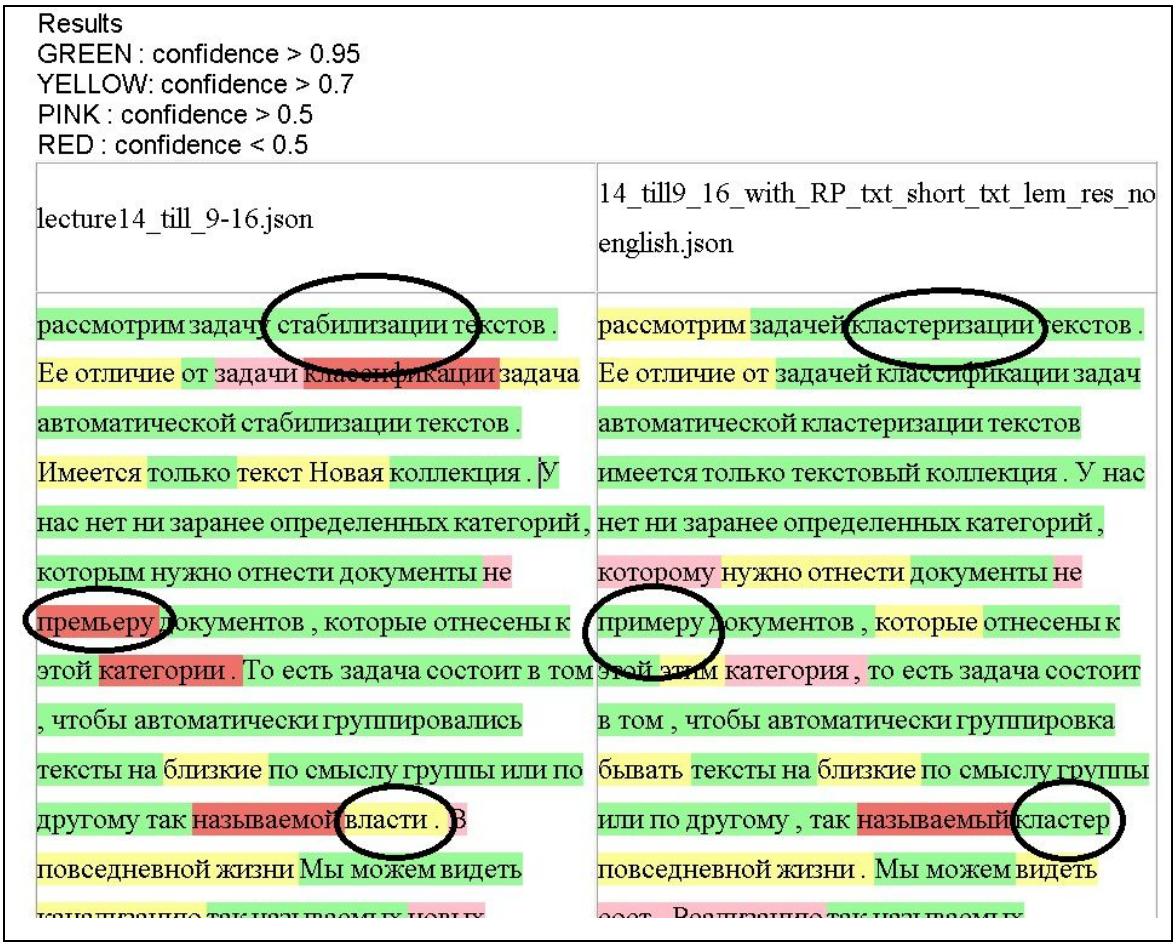
Требуется объединить данные списки, образовав единый список лексики.

Для этого текстовые объекты – словосочетания разбиваются на отдельные слова с сохранением оценки значимости, списки разных типов совмещаются, при необходимости, с домножением на соответствующий коэффициент по типам объектов, убираются дубликаты.

Для удаления из получаемого словаря слов, которые система распознавания и так знает, предлагается сначала обработать текст без специального словаря, затем удалить из специального словаря найденные системой слова, что позволит увеличить в словаре долю неизвестных системе распознавания слов.

Затем обработать текст со специальным словарем.

На Рисунке 9 приведены результаты обработки записи речи лектора лекции о кластеризации текстов (в соответствии с технологией, описанной в Приложении В). Добавление специального словаря существенно улучшает качество распознавания специальной лексики.


Results
GREEN : confidence > 0.95
YELLOW: confidence > 0.7
PINK : confidence > 0.5
RED : confidence < 0.5

lecture14_till_9-16.json	14_till9_16_with_RP_txt_short_txt_lem_res_no english.json
рассмотрим задачу стабилизации текстов . Ее отличие от задачи классификации задача автоматической стабилизации текстов . Имеется только текст Новая коллекция . У нас нет ни заранее определенных категорий , которым нужно отнести документы не премьеру документов , которые отнесены к этой категории . То есть задача состоит в том , чтобы автоматически группировались тексты на близкие по смыслу группы или по другому так называемой власти . В повседневной жизни Мы можем видеть	рассмотрим задачей кластеризации текстов . Ее отличие от задачей классификации задач автоматической кластеризации текстов имеется только текстовый коллекция . У нас нет ни заранее определенных категорий , которому нужно отнести документы не примеру документов , которые отнесены к этой этим категория , то есть задача состоит в том , чтобы автоматически группировка бывать тексты на близкие по смыслу группы или по другому , так называемый кластер повседневной жизни . Мы можем видеть

Рисунок 9 – Результат распознавания текста лекции с использованием специального словаря и без использования

6.2.2. Алгоритм составления ранжированного словаря лексики и терминологии общего предметного поля для автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся

В результате применения модуля № 17 формируются файлы

*.txt.lem_res

*.txt.term_res

*.txt.tfm_res

(1) Скорректировать и пополнить *.txt.term_res: для концепта онтологии надо взять все синонимы, у них взять LemEntryStr и вписать с теми же весами

То есть если есть строка в *.txt.term_res

327…….КЛАСТЕРНЫЙ АНАЛИЗ…….-

то она преобразуется в

327…….КЛАСТЕРИЗАЦИЯ…….-

327…….КЛАСТЕРИЗАЦИЯ ДАННЫЙ…….-

327…….КЛАСТЕРИЗОВАТЬ…….-

327…….КЛАСТЕРНЫЙ АНАЛИЗ…….-

(2) Затем надо пребразовать

*.txt.term_res

*.txt.tfm_res

в списки только лемм с теми же весами

327…….КЛАСТЕРИЗАЦИЯ…….-

327…….КЛАСТЕРИЗАЦИЯ ДАННЫЙ…….-

327…….ДАННЫЙ…….-

327…….КЛАСТЕРИЗОВАТЬ…….-

327…….КЛАСТЕРНЫЙ…….-

327…….АНАЛИЗ…….-

(3) после этого надо объединить по 2500 первых элементов всех преобразованных списков, но модифицируя веса.

Для преобразования весов использовать следующие коэффициенты:

*.txt.lem_res X 1,0

*.txt.term_res X 3.0

*.txt.tfm_res X 12.0

(4) отсортировать сначала по алфавиту и убыванию веса, и избавиться от дублей с меньшим весом

(5) затем оставшийся список отсортировать по убыванию веса — это и есть требуемый результат!

6.2.3. Программный модуль, реализующий алгоритм составления ранжированного словаря лексики и терминологии общего предметного поля для автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся (Модуль № 17)

Назначение: Программный модуль, реализующий алгоритм составления ранжированного словаря лексики и терминологии общего предметного поля для автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся.

Технологическое описание: модуль формирует словарь на основе результатов модуля № 16.

Входные данные: На вход подается имя файла *.res16.

Выходные данные: Функция сохраняет в словарь файл *.res17

Имя файла с исходным кодом: m17_makelist_res.py

Проверка функционирования

Пакетный файл: __test_mod17.bat

Директория с тестовыми данными __test_m16m17

API: Вызов функции m17_makelist_res.py:

```
C:\Python38\python.exe
    -u m17_makelist_res.py
    --fn __test_m16m17\kurs.txt.res16
```

Здесь: --fn – параметр имени файла

7. РАЗРАБОТКА МЕТОДОВ И АЛГОРИТМОВ ИНДЕКСИРОВАНИЯ ТЕКСТОВ УЧЕБНЫХ КУРСОВ И ТЕКСТОВ РЕФЛЕКСИИ ОБУЧАЕМЫХ (ПРИПИСЫВАНИЯ ТЕКСТАМ СООТВЕТСТВУЮЩИХ АТРИБУТОВ) ПО ТЕМАТИЧЕСКОЙ ТАКСОНОМИИ АНО «УНИВЕРСИТЕТ 2035»

7.1. Разработка методов и алгоритмов интеграции структур данных Таксономии 2035 с функционалом лингвистическим обеспечением программного продукта АЛОТ.

7.1.1. Методы интеграции структур данных Таксономии 2035 с функционалом лингвистического обеспечением ПП АЛОТ

Входными данными является файл в формате MS Excel, содержащий структурированный по уровням список категорий классификатора «Таксономия 2035».

Требуется преобразовать структуры входных данных в структуры программного обеспечения ведения лингвистических онтологий ПП АЛОТ.

Используется два метода:

- Для объектов, которые могут быть сопоставлены с терминами (текстовыми входами понятий онтологии) сосравляются стандартные для описания рубрик в ПП АЛОТ логические выражения, с целью дальнейшего применения стандартных функций классификации ПП АЛОТ;
- Также просхводится транформация запросов из формат входных данных в формат поисковых запросов информационно-поисковой системы NearIdx, с целью обеспечить применимость разработанного модуля № 01.

7.1.2. Алгоритм работы программной оболочки интерфейса пользователя поддержки интегрирования описания классов Таксономии 2035

Этапы алгоритма:

(1) Заполнение следующих полей:

RubricatorId = 2035 (идентификатор классификатора)

RubricId = присвоить индентификатор

Snumb = внешний код, формируется из первой буквы — «Y» и групп по три символа по файлу .csv с шагом 10 по полям Level_1 — Level_7 (то есть строчка из 22 символов) (перед началом процедуры стоит отсортировать файл по этим полям), то есть если максимум по уровню (после сортировки!) был N, и значение уровня сменилось, N+10: ...010… ...020…до правого конца кода дополняем нулями

Если вдруг пропущена рубрика уровня, то есть, например,

...040030020000

...040040010000

...040040020000

то надо добавить рубрику

...040030020000

...040040000000 — дать ей имя по уровню

...040040010000

...040040020000

Nlevel = одновременно заполняется уровень, начиная с 0

HostRubricId = указывает на RubricId предыдущего уровня (для верхнего уровня = 0)

RubricStr = значение последнего заполненного уровня

(3) Заполнить, где можно таблицы

Disjunct

Conjunct

Rubdes

Для простоты пока считаем, что у нас ровно 1 дизъюнкт и 1 конъюнкт, а все остальное - варианты.

Главное — помочь заполнить таблицу RubDes — по значению поля Expression

Конвертировать псевдо-термины, по возможности, факты-запросы и выдать протокол.

(3а) Типовой Expression может выглядеть по-простому (таких много)

= Гидродинамика

или

= "Механика твёрдого тела"

Здесь: кавычки игнорируем. Ищем по наименованию текстового входа («ё» переводим в «е»).

Если нашли, то находим первый концепт, соответствующий текстовому входу и его пишем в Conceptid

(расширение = «E»)

И пишем результаты сопоставления в Log-файл для рубрики (Snumb и RubricStr) (как положительные — по такому нашли то-то, так и отрицательные — по такому не нашли)

(3б) Если по-сложнее Expression

"Механика сплошных сред" ### or Гидродинамика or Акустика ### or "Механика твёрдого тела"

(помним, что ### — искусственно вставленные нами признаки перевода строки)

"Механика сплошных сред" or Гидродинамика or Акустика or "Механика твёрдого тела"

Это, в случае, нахождения всех текстовых входов должно превратиться в четыре записи в Rubdes

Если при этом для разных входов .csv получаем один концепт — не дублируем, конечно

(3в) Иногда может втсретиться что-то типа

or "аналитик","большие данные"

Вообще говоря, это подразумевает два конъюнкта — пока пропускаем, сделаем вручную

(3г) Если Expression сложный;

```
"Анализ больших данных"  ###  ### or phrase(4,"анализ" or
"анализировать" or "аналитика" or "аналитик","большие данные"  ### or
"BigData" or "Big Data" or "датасет" or phrase("большой объем" or
"массив"  ### or "количество","данные"))  ### or "data-аналитик" or
"data аналитик" or "дата аналитик" or "data analyst"  ### or "data-
analyst" or phrase(2,"analyse","vast","amounts","of","data")  ### or
phrase(2,"analyze","BigData" or "Big Data")  ### or
phrase(2,"analysis","of","large","data","volumes")  ### or
```

```
phrase(2,"analysis",of,"data set" or "datasets")  ### or
phrase(2,"analysis",of,"massive-scale","data")  ### or phrase(2,"set" or
"body","of","data")  ### or phrase(2,"analysis" or "analyse" or
"analytics" or "analyze","BigData" or "Big Data")  ### or
phrase(2,"BigData" or "Big Data","analysis")  ### or phrase(2,"analysis"
or "analyse" or "analytics" or "analyze","large" or "high", ### "amount"
or "volume","of","data")  ### or phrase(2,"screening","datasets" or
"data set")
```

или

```
Анализ больших данных
or phrase(4,"анализ" or "анализировать" or "аналитика"
or "аналитик","большие данные"
or "BigData"
or "Big Data"
or "датасет"
or phrase("большой объем" or "массив"  ### or "количество","данные"))
or "data-аналитик"
or "data аналитик"
or "дата аналитик"
or "data analyst"
or "data-analyst"
or phrase(2,"analyse","vast","amounts","of","data")
or phrase(2,"analyze","BigData" or "Big Data")
or phrase(2,"analysis","of","large","data","volumes")
or phrase(2,"analysis",of,"data set" or "datasets")
or phrase(2,"analysis",of,"massive-scale","data")
or phrase(2,"set" or "body","of","data")
or phrase(2,"analysis" or "analyse" or "analytics"
or "analyze","BigData" or "Big Data")
or phrase(2,"BigData" or "Big Data","analysis")
or phrase(2,"analysis" or "analyse" or "analytics" or "analyze","large" or
"high", "amount" or "volume","of","data")  or phrase(2,"screening","datasets" or
"data set")
```

То есть тут есть концепты и запросы.

Концепты отрабатываем, запросы пока разбить на отдельный и выдать в протокол

Разбор функций запросов:

1) snear(), phrase(), sentence() - список слов рядом, трактуются как последовательности

2) singleroot() - однокоренные слова. В данный момент игнорируется, т.е. singleroot("поддержка", "сохранение", "соблюдение") превращаем в "поддержка" OR "сохранение" OR"соблюдение"

3) possible() - аналогично singleroot, превращаем в список через OR

4) term(оценка|оценивать|формулировать|формулировка) - аналогично singleroot, превращаем в список через OR

5) case([C]|[C]) - аналогично singleroot, превращаем в список через OR

6) optional(ломать|разбивать) - аналогично singleroot, превращаем в список через OR

7) form(ROS, URDF, xacro), paragraph, lemma, negate - аналогично singleroot, превращаем в список через OR

8) Непонятная функция orn

Выглядит как список, но из контекста непонятно, как она обрабатывается

```
or "Кластеризация текстов" or " Text clustering" or "Лингвистический анализ" or orn(лингвистика, лингвистический, семантика, семантический) or "linguistic analysis" or "Natural Language Processing" or "NLP" or "Онтологический анализ"
```

Игнорируется.

Фрагмент протокола конвертации данных:

```
phrase(3,оценка or singleroot(анализ) or singleroot(исследование) or singleroot(изучение),singleroot(рынок) or среда) ### or market analysis and environmental assessment ### or phrase(2, market, analysis, enviromental, assessment) or phrase(3,анализ or оценка, конкурентного окружения) ### or phrase(2,анализ or изучение or исследование,конкурентов) ### or competitive environment analysis ### or phrase(2,Конкурентный or бизнес or деловая or аналитическая ### or экономическая or маркетинговая or коммерческая, разведка) ### or бизнес-разведка or Анализ конкурентных цен or Анализ конкурентных цен or Матрица Портера or phrase(3,оценка or singleroot(анализ) or singleroot(исследование) or ###
```

singleroot(изучение),singleroot(рынок) or среда) ### or market analysis and environmental assessment ### or phrase(2, market, analysis, enviromental, assessment) or Анализ целевой аудитории or целевая аудитория or phrase(2,изучение or исследование,пользователь) ### or phrase(2,user,research) ### or Cust dev or CustDev or CustDev or customer development or развитие клиента or A/B-тестирование or A/B testing or A/B тестирование or A/B-testing ### or A/B тест or A/B-тест ### or A/Б-тестирование or A/Б тестирование ### or A/Б тест or A/Б-тест or phrase(2,изучение or исследование,пользователь) ### or phrase(2,user,research) ### or Cust dev or CustDev or CustDev or customer development ### or развитие клиента or Проблемное интервью or phrase(2, маркетинговые, исследования) ### or phrase(2, исследования, рынка) ### or orn(анализ рынка, оценка среды, Cust dev, анализ конкурентного окружения, ### конкурентная разведка) ### or phrase(2, market or marketing, research) or market volume calculation ### or phrase(2,расчет or расчёт, рынок) ### or phrase(2,расчет or расчёт, объем or емкость or ёмкость or объём , рынок) or TAM or общий объём целевого рынка or Total Addressable Market or market volume calculation ### or phrase(2,расчет or расчёт, рынок) ### or phrase(2,расчет or расчёт, объем or емкость or ёмкость or объём , рынок) or Емкость рынка

Разобранный запрос:

("MARKET" AND "ANALYSIS" AND "ENVIROMENTAL" AND "ASSESSMENT")
("ИССЛЕДОВАНИЯ" AND "РЫНКА")
("МАРКЕТИНГОВЫЕ" AND "ИССЛЕДОВАНИЯ")
("A/B TESTING")
("A/B ТЕСТ")
("A/B ТЕСТИРОВАНИЕ")
("A/B-TESTING")
("A/B-ТЕСТ")
("A/B-ТЕСТИРОВАНИЕ")
("COMPETITIVE ENVIRONMENT ANALYSIS")
("CUST DEV")
("CUSTDEV")
("CUSTOMER DEVELOPMENT")
…

Текстовые входы которые искали:

ОЦЕНКА РЫНОК
РЫНОК ОЦЕНКА
ОЦЕНКА СРЕДА
СРЕДА ОЦЕНКА
АНАЛИЗ РЫНОК
РЫНОК АНАЛИЗ
АНАЛИЗ СРЕДА
СРЕДА АНАЛИЗ
ИССЛЕДОВАНИЕ РЫНОК
РЫНОК ИССЛЕДОВАНИЕ

```
ИССЛЕДОВАНИЕ СРЕДА
СРЕДА ИССЛЕДОВАНИЕ
ИЗУЧЕНИЕ РЫНОК
РЫНОК ИЗУЧЕНИЕ
ИЗУЧЕНИЕ СРЕДА
СРЕДА ИЗУЧЕНИЕ
MARKET ANALYSIS ENVIROMENTAL ASSESSMENT
ANALYSIS MARKET ENVIROMENTAL ASSESSMENT
...
МАРКЕТИНГОВЫЕ ИССЛЕДОВАНИЯ
ИССЛЕДОВАНИЯ МАРКЕТИНГОВЫЕ
ИССЛЕДОВАНИЯ РЫНКА
РЫНКА ИССЛЕДОВАНИЯ
MARKET RESEARCH
RESEARCH MARKET
MARKETING RESEARCH
RESEARCH MARKETING
РАСЧЕТ РЫНОК
РЫНОК РАСЧЕТ
...
```

7.1.3. Программный модуль, реализующий алгоритм интеграции структур данных Таксономии 2035 с функционалом лингвистического обеспечением ПП АЛОТ в виде веб-интерфейса пользователя поддержки интегрирования описания классов Таксономии 2035 с выгрузкой результатов описания в формате словарей ПП АЛОТ (Модуль № 18)

Назначение: Программный модуль, реализующий алгоритм интеграции структур данных Таксономии 2035 с функционалом лингвистического обеспечением ПП АЛОТ в виде веб-интерфейса пользователя поддержки интегрирования описания классов Таксономии 2035 с выгрузкой результатов описания в формате словарей ПП АЛОТ.

Технологическое описание: Скрипт производит разбор исходного файла с таксономиями, их иерархиями и описаниями на языке запросов

Таксономии 2035. На основании этих данных происходит создание описания в формате словарей ПП АЛОТ, и запись данных в соответствующую БД.

Входные данные: Файл в формате MS Excel 2007+, расположенный в папке со скриптом.

Выходные данные: Выходными данными для модуля являются данные поддержки интеграции, записанные в базу данных PostgreSQL.

API: Скрипт для интерпретатора PHP.

```
taxonomy_prepare.php <имя файла>
```

7.2. Разработка методов и алгоритмов автоматического индексирования текстов учебных курсов по классам Таксономии 2035

7.2.1. Методы автоматического индексирования текстов учебных курсов по классам Таксономии 2035

Входные данные описания Таксономии 2035 в виде запросов в модуле № 18 преобразуются в два типа структур:

- Стандартные для ПП АЛОТ структуры описания смысла рубрик в виде логических формул (дизъюнкты конъюнктов над опорными концептами онтологии, которые потом расширяются по иерархии онтологии);
- Структуры в виде запросов к информационо-поисковой машине NearIdx, которые могут быть использованы в качестве входных данных для классификации текстов с использованием разработанного модуля № 01.

Для индексирования текстов по сущностям Таксономии 2035 необходимо:

- Осуществить выгрузку сформированных в базе данных ведения лингвистической онтологии структур данных на диск в формате применения ПП АЛОТ и/или модуля №01;

- Запустить обработку, аналогичную применению модуля № 06, без использования модуле1 №№ 03, 05.

7.2.2. Алгоритм автоматического индексирования текстов учебных курсов по классам Таксономии 2035

Алгоритм включает три функции:

- Выгрузка структур данных для применения стандартного ПП АЛОТ – для описания рубрик Таксономии 2035 через логические формулы;
- Выгрузка структур данных для применения модуля № 01 – для описаний рубрик Таксономии 2035 в виде запросов к поисковой машине NearIdx;
- Запуск обработки текстового файла.

7.2.3. Программный модуль, реализующий алгоритм автоматического индексирования текстов учебных курсов по классам Таксономии 2035 (Модуль № 19)

Назначение: Программный модуль, реализующий алгоритм автоматического индексирования текстов учебных курсов по классам Таксономии 2035.

Технологическое описание: модуль представляет собой – в зависимости от заданного параметра (--func): функцию формирования словарей для поиска рубрик в тексте на основе описания рубрик классификатора «Таксономия 2035»:

(--func 1) формирование словарей для ПП АЛОТ в виде описаний рубрик «как факты»;

(--func 2) формирование словарей для ПП АЛОТ в виде описаний рубрик как логические формулы над понятиями онтологии;

либо

(--func 3) формирование словарей в виде описаний рубрик как запросов к NLD файлам и автоматическое индексирование текстов по классам Таксономии 2035:

- по описанию классов логическими формулами - стандартным механизмом с использованием результатов работы Модуля № 18 в Модулях № 06 и № 13;
- по описанию классов в формате запросов к информационно-поисковой системе в Модуле № 01.

Входные данные: На вход подается номер функции (--func) и имя папки в которую сохранить словари либо имя файла для обработки.

Выходные данные:

--func 1: Файл rubr_reqs.fct, содержащий описание рубрик

--func 2: theman.lst.zip – словари для «стандартного» ПП АЛОТ, содержащие в том числе данные описания рубрик классификатора «Таксономия 2035».

--func 3: Файлы обработки .nld; .rbr3; .rbr5; .rbrreq .

Имя файла с исходным кодом: module19.php

Проверка функционирования

Пакетные файлы: __test_mod19_1.bat, __test_mod19_2.bat, __test_mod19_3.bat.

Директории с тестовыми данными: __test19_1_2, __test19_3 .

API: Варианты вызова функции m19_thesvoc.py:

```
C:\Python38\python.exe
      -u m19_thesvoc.py
      --func 1
      --inout __test19_1_2\

C:\Python38\python.exe
      -u m19_thesvoc.py
      --func 2
```

```
      --inout __test19_1_2\

C:\Python38\python.exe
      -u m19_thesvoc.py
      --func 3
      --inout __test19_3\kurs.txt
```

8. РАЗРАБОТКА WEB-СЕРВИСА, ПРЕДОСТАВЛЯЮЩЕГО ИНТЕРФЕЙС ПРОГРАММНОГО ПРОДУКТА АЛОТ С ВКЛЮЧЕННЫМИ В НЕГО МОДУЛЯМИ

8.1. Описание web-сервиса, предоставляющего интерфейс программного продукта АЛОТ с включенными в него модулями

Работа с веб-сервисом начинается с авторизации (Рисунок 10), логины и пароли были переданы Заказчику.

Рисунок 10 – Форма авторизации пользователей

Основные разделы веб-сервиса:

- Ведение перечня своих курсов, включая ввод данных рефлексии;
- Запуск сравнения курсов и сравнения курса и рефлексии;
- Контроль выполнения сформированных заданий;
- Формирование отчетных форм.

На Рисунке 11 приведена экранная форма перечня курсов пользователя, для которых указываются учетные данные, количество введенных документов.

Запись можно удалить или скорректировать, используя кнопки, расположенные в правой части каждой записи.

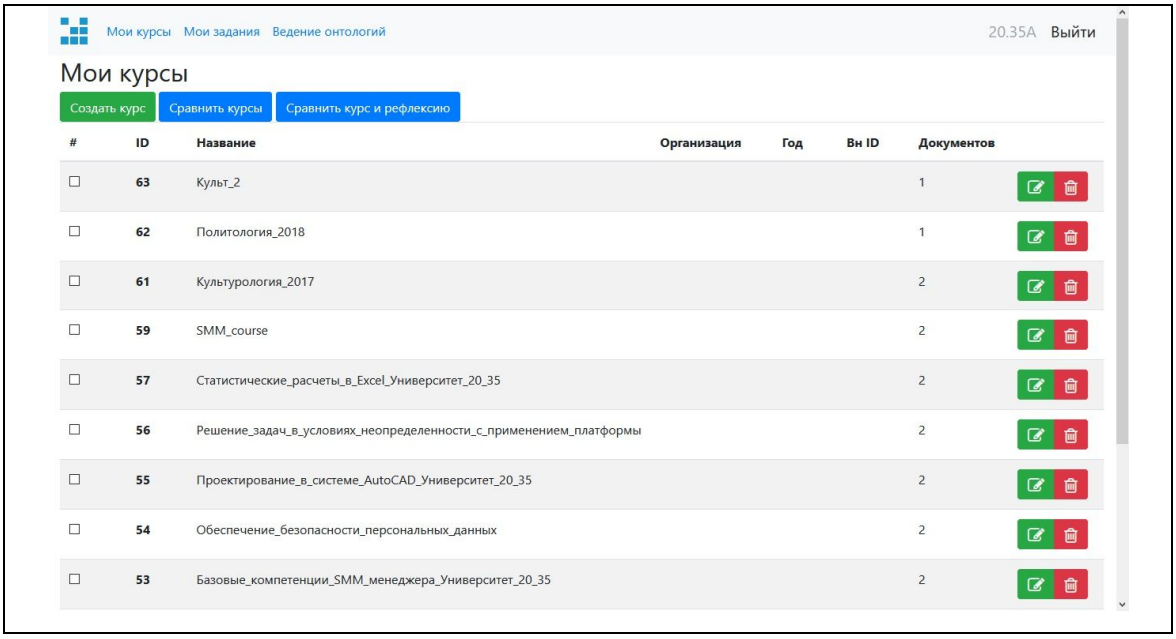


Рисунок 11 – Перечень курсов

Форма корректировки описания учебного курса представлена на Рисунке 12. Имеются возможности ввода нового документа, корректировки метаданных о курсе (Рисунок 13) и об отдельном документе.

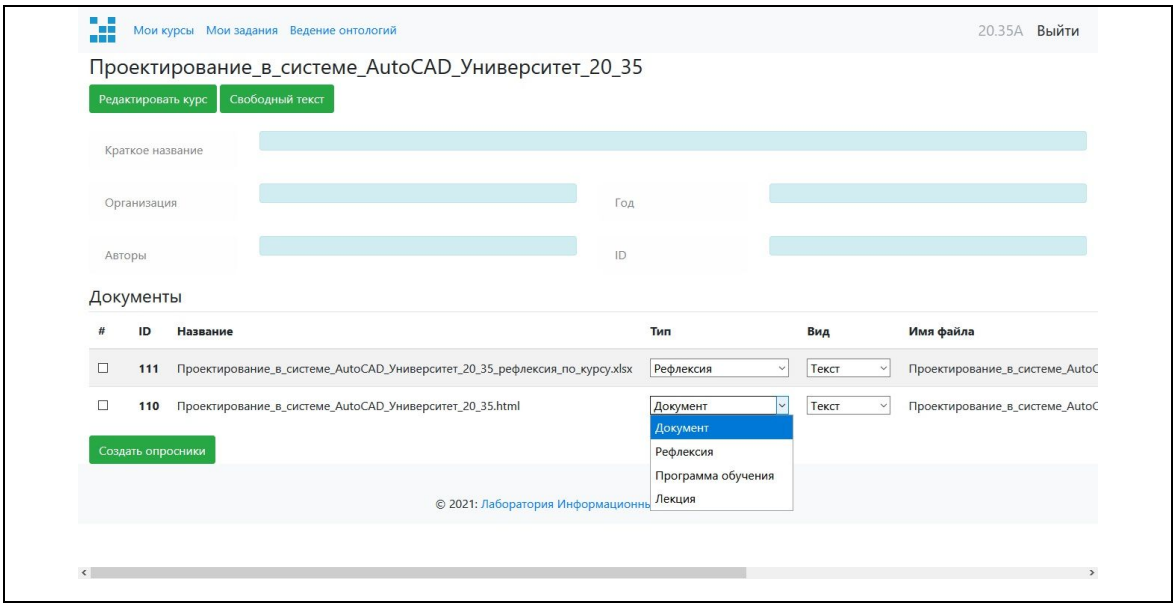


Рисунок 12 – Форма просмотра информации о материалах курса

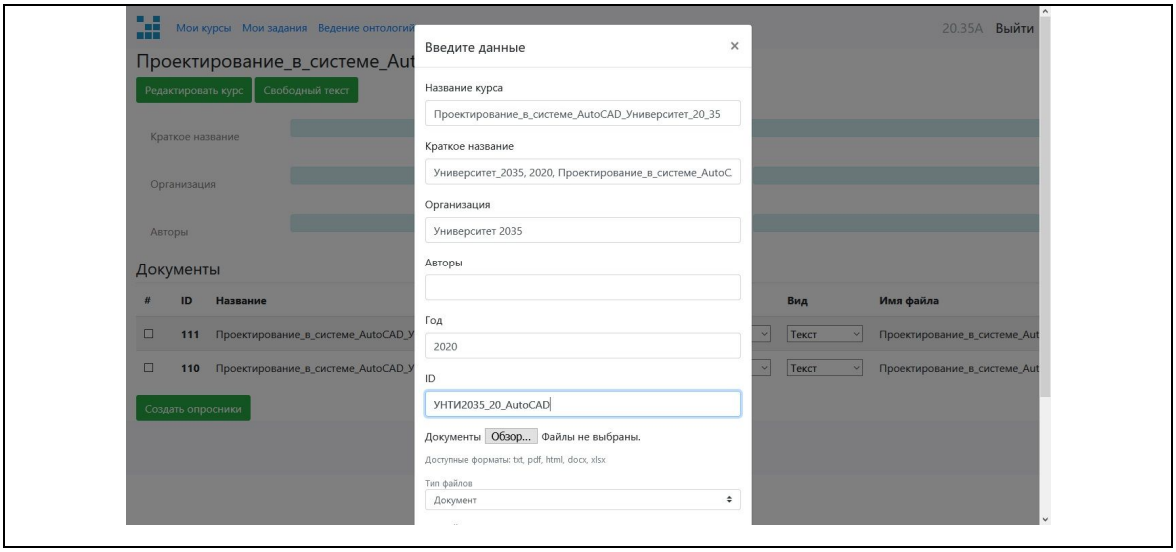


Рисунок 13 – Редактирование метаданных о курсе

На Рисунке 14 приведена форма просмотра «текстового» слоя материалов курса, которая будет автоматически анализироваться в дальнейшем.

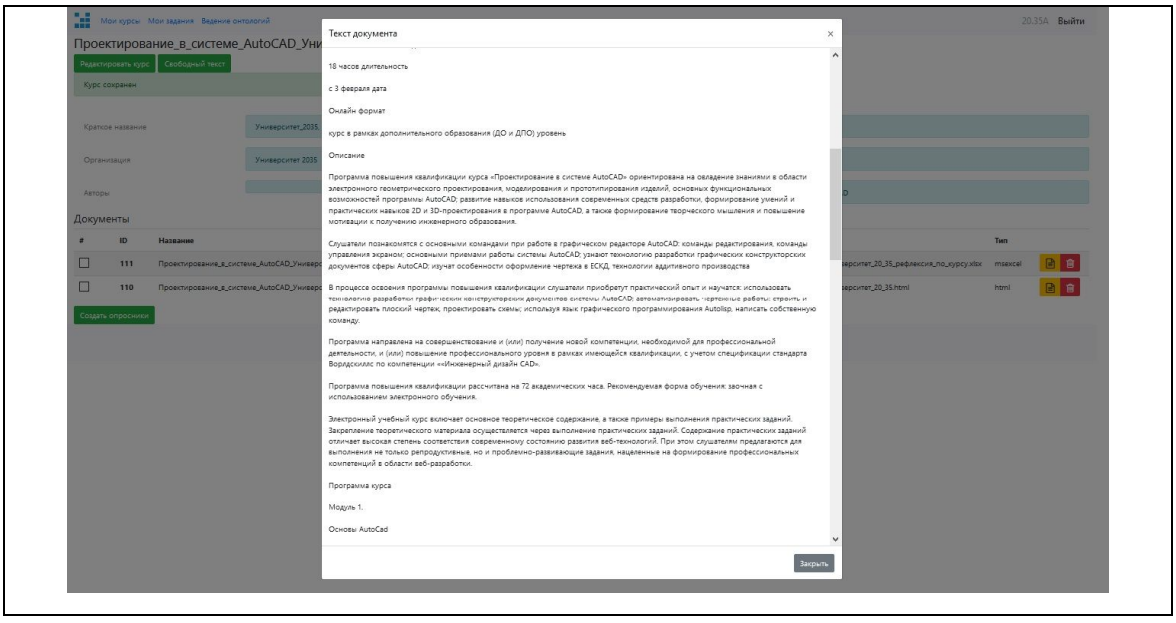


Рисунок 14 – Текст документа материала учебного курса

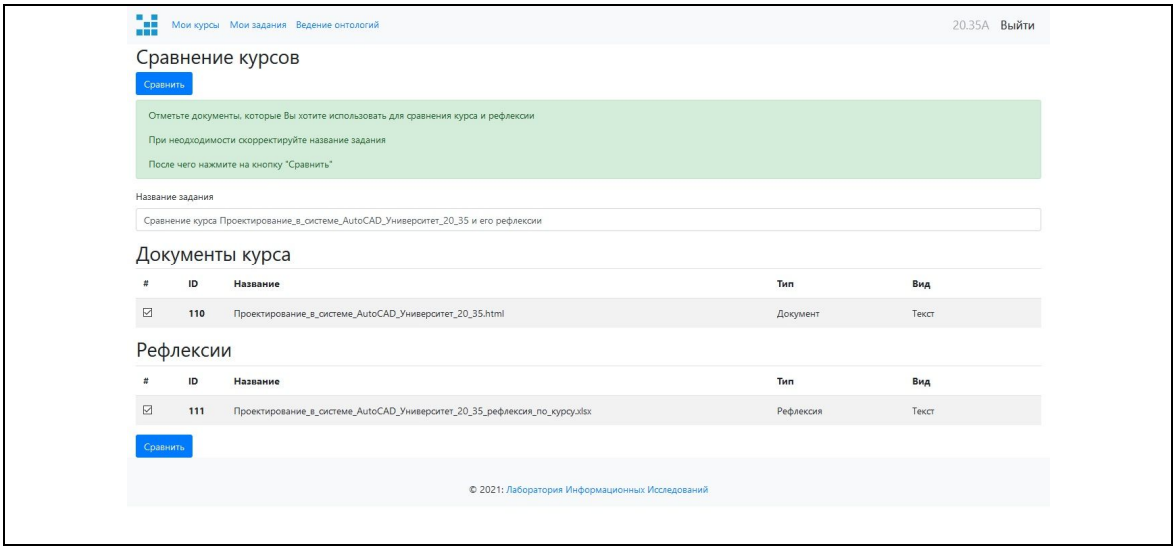


Рисунок 15 – Задание параметров сравнения учебных курсов

Одна из основных вункций веб-сервиса – сравнение двух курсов, для этого в перечне курсов следует выбрать ровно два курса и активировать кнопку «Сравнение курсов». Появится экранная форма щадания параметров сравнения курсов (Рисунок 15). Требуется выбрать документы курсов, которые будут участвовать в сравнении (предполагается, что курс может включать различные документы), и активизировать кнопку «Сравнить».

После чего следует перейти в форму «Мои задания» (Рисунок 16), визуализирующей статус выполнения задания, предоставляющей возможность прямого доступа к наиболее важным оезультатам выполнения задания после его завершения.

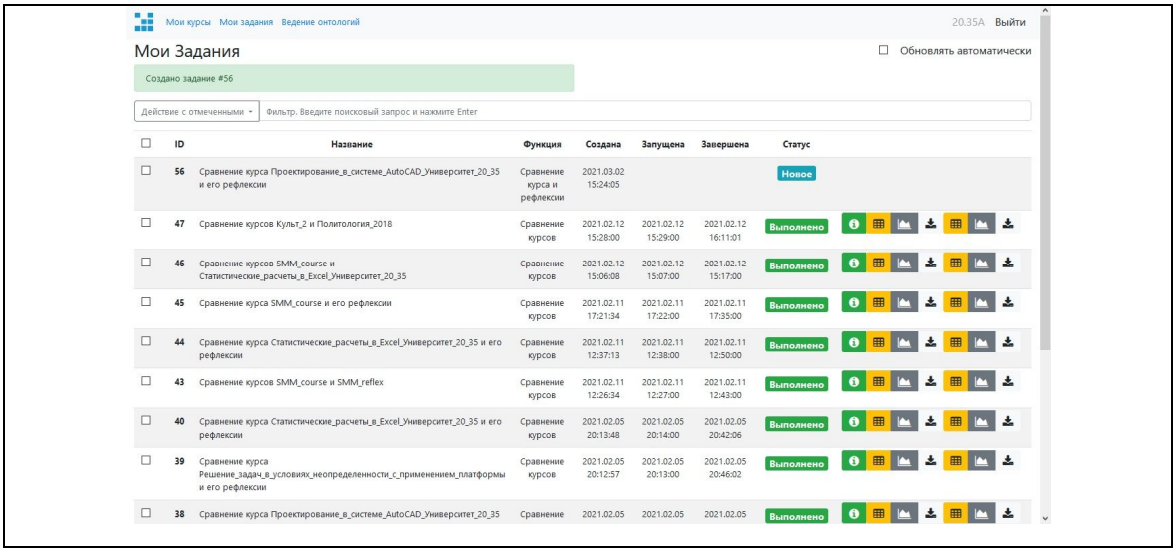


Рисунок 16 – Мониторинг заданий

Ключевой формой работы с результатами сравнения кусров является форма редактирования результатов (Рисунок 17), которая позволяет добавлять в фильтр нежелательные для отображения элементы (которые после этого начинают отображаться бледным цветом).

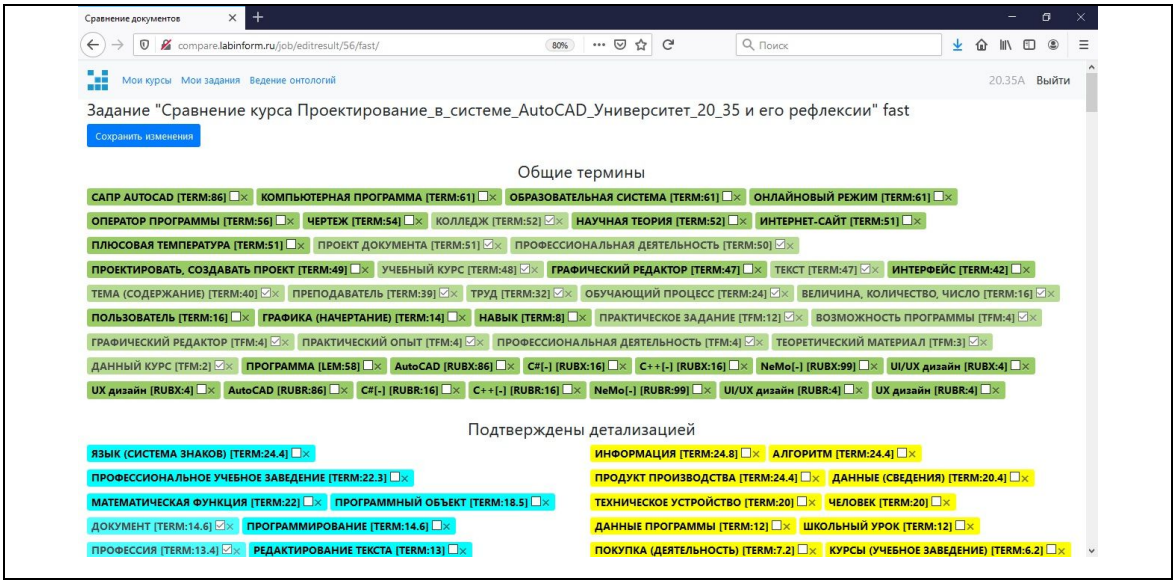


Рисунок 17 – Редактирование состава результатов, формирование фильтра

После редактирования можно перейти в одну из форм представления результатов: табличную, гистограмм или графовую. При переходе в табличную форму предлагается установить параметры отображения (Рисунок 18).

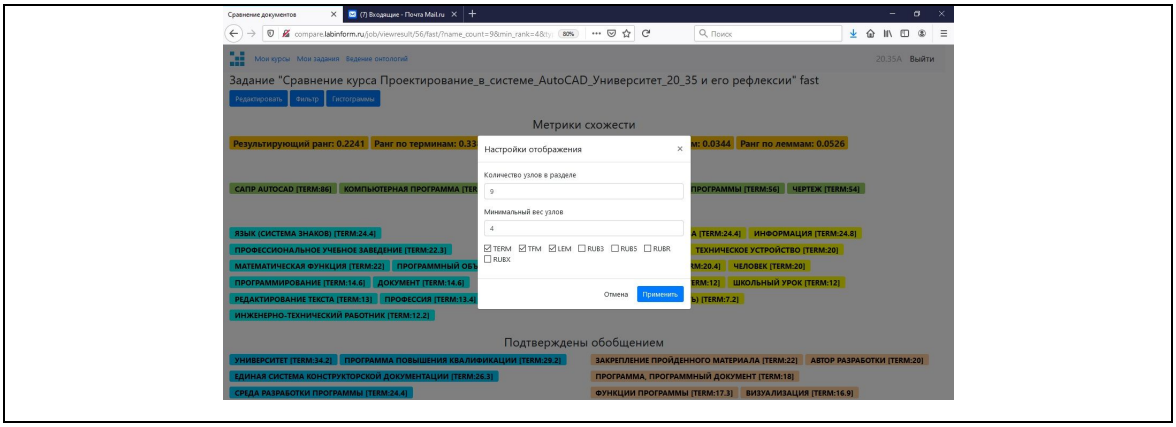


Рисунок 18 – Задание параметров табличного представления

В результате табличное представление отображает только значимые объекты, что позволяет нагляднее видеть результаты (Рисунок 19).

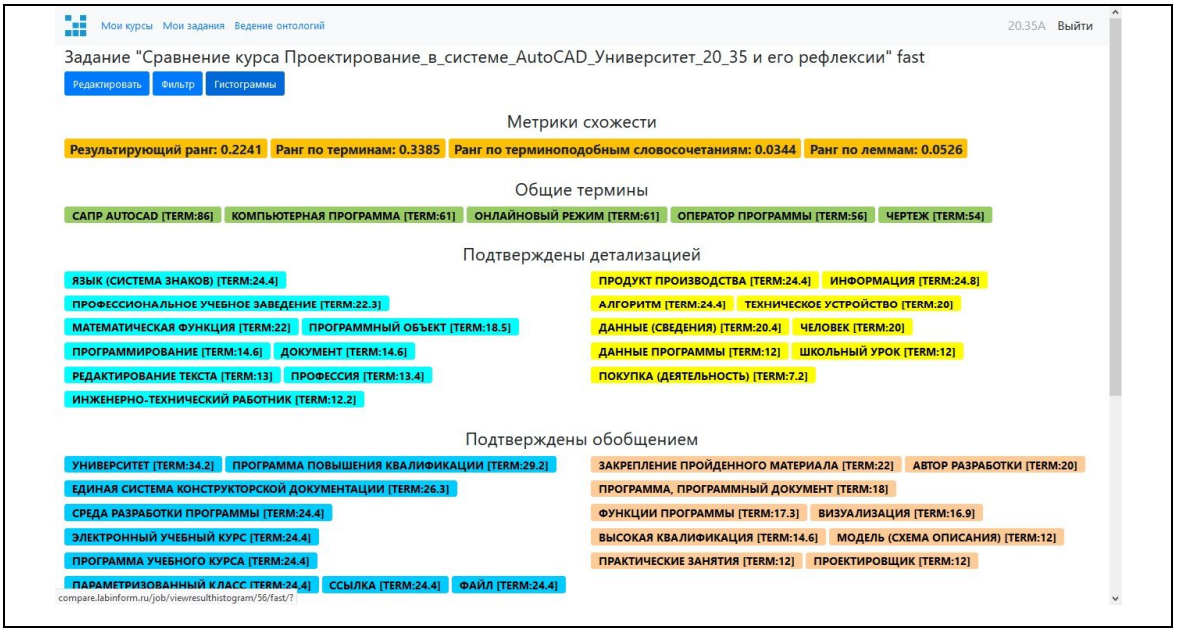


Рисунок 19 – Табличное представление результатов сравнения курсов

Аналогично формируется представление результатов на гистограммах (Рисунок 20, Рисунок 21).

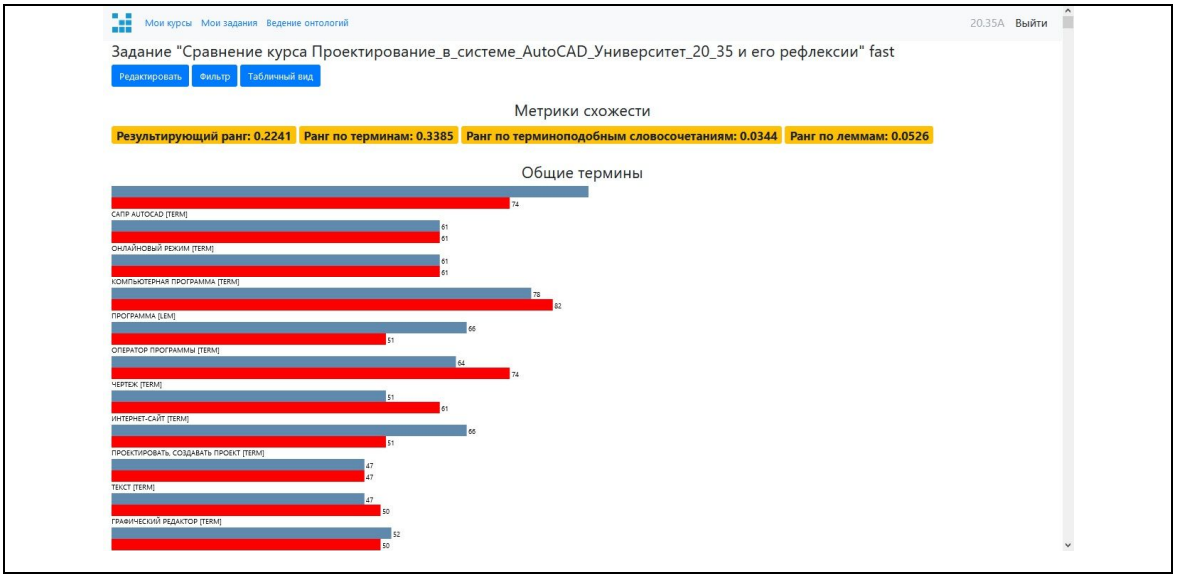


Рисунок 20 – Представление общих терминов на гистограммах

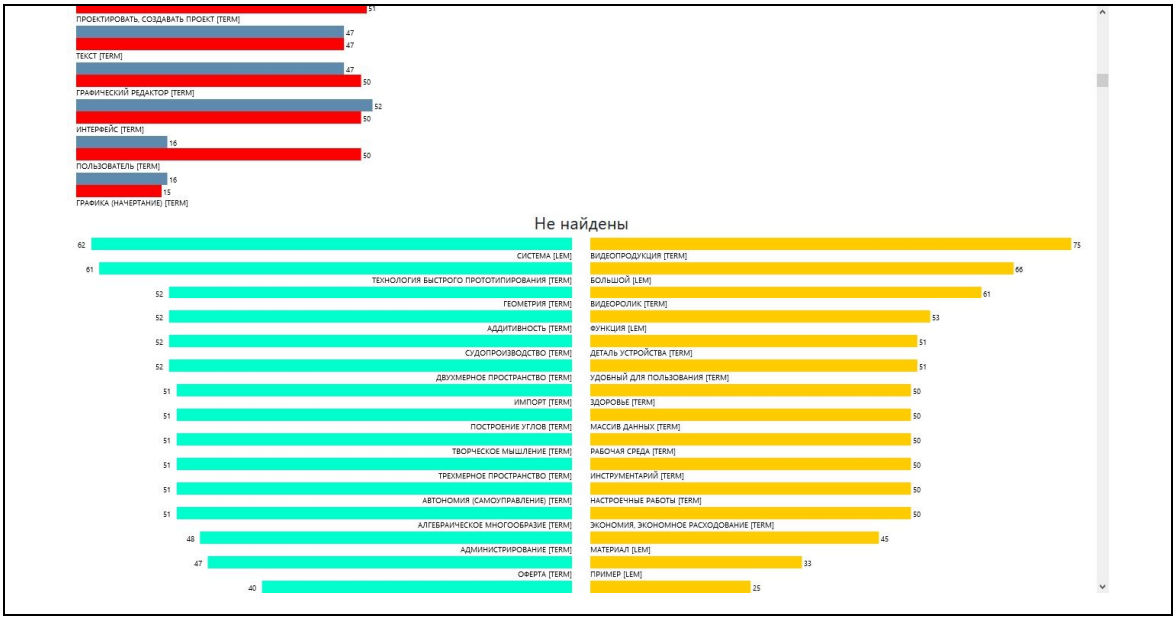


Рисунок 21 – Представление уникальных терминов для сравниваемых курсов на гистограммах

При формировании данных графового представления также предъявляется форма задания параметров отображения (Рисунок 22).

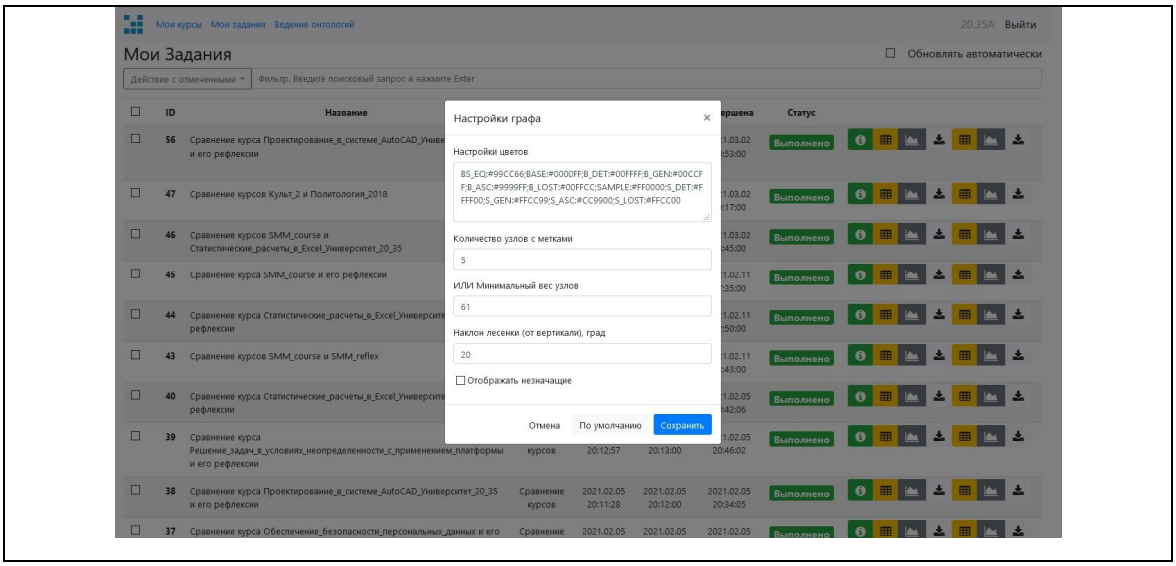


Рисунок 22 – Форма задания параметров отображения на графах

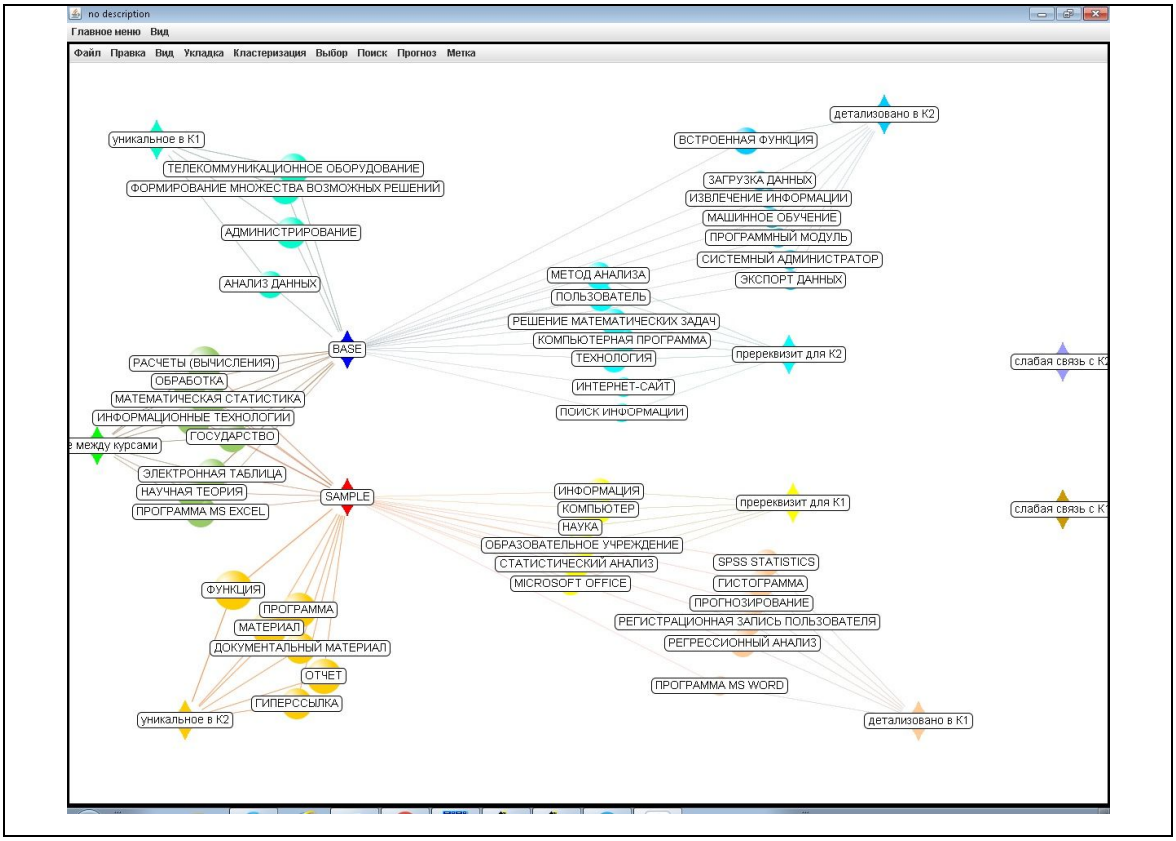


Рисунок 23 – Графовое представление результатов сравнения двух курсов

Визуализация графового представления (Рисунок 23) реализуется с использованием локального приложения GraphView, входящего в комплект поставки ПП АЛОТ.

8.2. Программный модуль, реализующий web-сервис, предоставляющего интерфейс программного продукта АЛОТ с включенными в него модулями

8.2.1. Техническое описание программного модуля web-сервиса

Назначение: Веб-сервис, предоставляющий интерфейс программного продукта АЛОТ с включенными в него модулями.

Технологическое описание: Данный модуль представляет из себя пользовательский интерфейс.

Включает служебные управляющие модули: модуль № 20-0 (сравнение курсов) и модуль № 20-00 (сравнение курса и рефлексии).

Модуль состоит из связки сервиса Apache (установлен в папку C:\Apache24), интерпретатора PHP (установлен в папку C:\PHP74) и программного кода модуля (установлен в папку C:\usr\www).

Конфигурационные файлы Apache и PHP (для варианта, когда ПО разложено по выше описанным папкам файлы расположены, и веб-сервис доступен по адресу http://compare.stand2035.ru) скопированы в папку C:\LII\Dist\Data.

Если ПО установлено в другие папки, или предполагается другой url сервиса, необходимо изменить соответствующие конфигурационные файлы.

Запуск сервиса осуществляется автоматически. Доступ к сервису – путем открытия в браузере соответствующего url-адреса.

Входные данные: Введенные пользователем описания курсов, документы, относящиеся к курсам.

Выходные данные: Выходными данными являются архивы формата zip, содержащие результаты сравнения учебных курсов , в форматах html, json, gexf

API: Взаимодействие с сервисом осуществляется через броузер, посредством стандартных GET и POST запросов по протоколу HTTP.

8.2.1.1. Управляющий модуль для сравнения двух курсов (Модуль № 20-0)

Назначение: Управляющий модуль для сравнения двух курсов, вызывает модули № 08, № 09, № 10.

Входные данные модуля № 20-0: На вход подается папка с двумя файлами курсов с расширениями *.txt.

Выходные данные модуля № 20-0: Результатом работы является архив исходной папке с насчитанными данными другими модулями

Имя файла с исходным кодом: comp2txt.py

API модуля № 20-0: Скрипт на языке python:

```
comp2txt.py –dir <имя_директории>
```

Здесь: --dir – параметр директории

8.2.1.2. Управляющий модуль для сравнения текстов курса и рефлексии (Модуль № 20-00)

Назначение модуля № 20-00: Управляющий модуль для сравнения текстов курса и рефлексии, вызывает модули № 14, № 11, № 15.

Входные данные модуля № 20-00: На вход подается папка с двумя файлами курса и рефлексии с расширениями *.txt.

Выходные данные модуля № 20-00: Результатом работы является архив исходной папке с насчитанными данными другими модулями

Имя файла с исходным кодом: comp2txt_rfx.py

API модуля № 20-00:

```
comp2txt_rfx.py --dir <имя_директории>
```

Здесь: --dir – параметр директории

8.2.2. Интеграция модулей, реализующих алгоритмы анализа и сравнения содержания учебных курсов

Схема интеграции программных модулей для сравнения материалов двух разных учебных курсов представлена на Рисунке 24:

- Анализ материалов отдельного учебного курса реализуется модулем № 06, который использует стандартные средства ПП АЛОТ, дополненные классификаторами новых модулей № 01 (классификация по запросам), № 03 (статистических классификаторов методами машинного обучения – требует предварительного обучения с использованием модуля № 02), № 05 (классификации с использованием вероятностного тематического

моделирования – требует предварительного обучения с использованием модуля № 04);

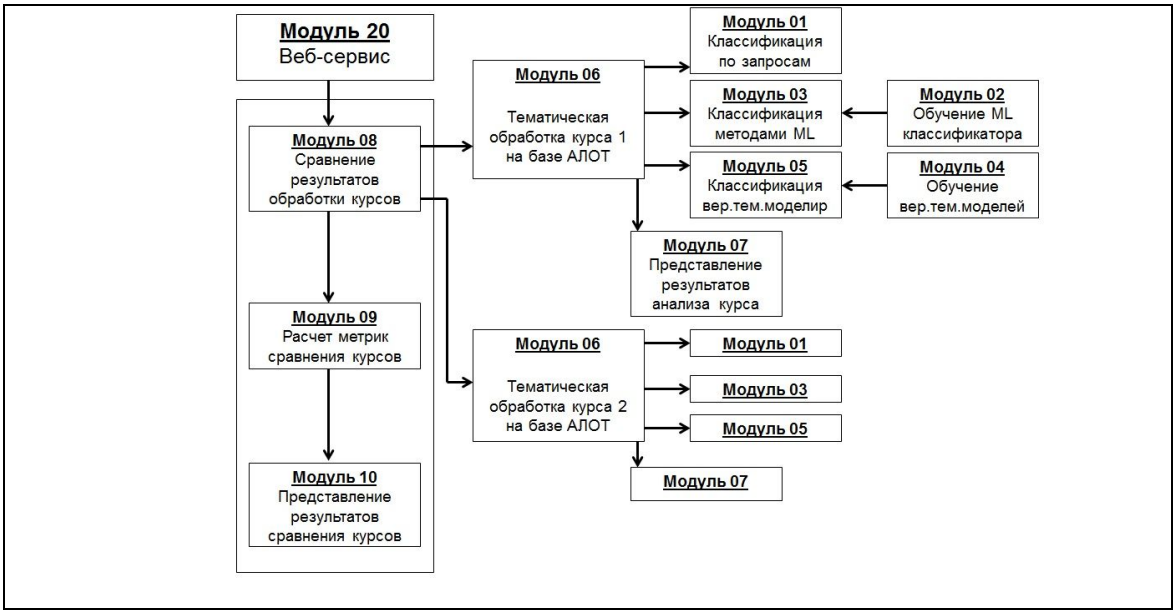


Рисунок 24 - Интеграция программных модулей при сравнении курсов

- Результаты анализа отдельного курса формируются соответствующими модулями № 07;
- Сравнение результатов анализа отдельных курсов с использованием модуля № 08;
- Расчет метрик схожести между курсами с использованием модуля № 09;
- Формирование визуального табличного и графового представления результатов сравнения с использованием модуля № 10.

8.2.3. Интеграция модулей, реализующих алгоритмы анализа и сравнения содержания учебного курса и рефлексии обучаемых

Схема интеграции программных модулей для сравнения материалов учебного курса и рефлексии обучающихся представлена на Рисунке 25:

- Формирование вопросников обучающихся в целях получения более детального представления об усвоении обучаемыми материалов курса (модуль № 12);

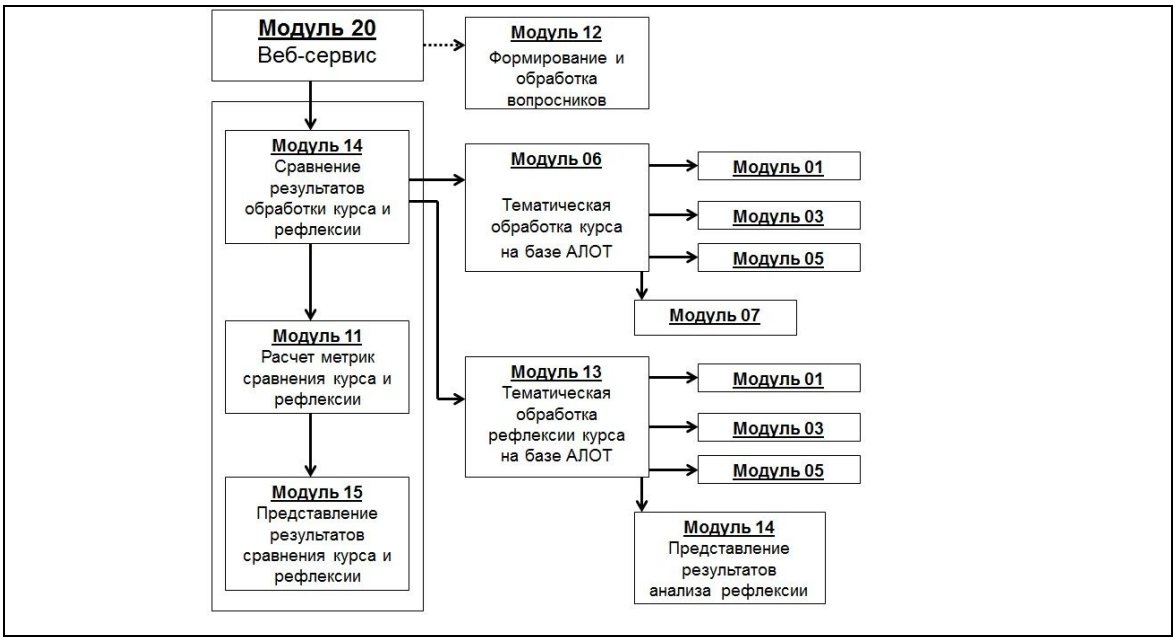


Рисунок 25 - Интеграция программных модулей при сравнении курса и его рефлексии

- Анализ материалов учебного курса реализуется модулем № 06, который использует стандартные средства ПП АЛОТ, дополненные классификаторами новых модулей № 01, № 03, № 05. Результаты анализа материалов учебного курса формируются модулем № 07;
- Анализ материалов рефлексии реализуется модулем № 13 (в целом аналогичном модулю № 06). Результаты анализа материалов учебного курса формируются модулем № 14;
- Сравнение результатов анализа курса и рефлексии производится с использованием модуля № 14;
- Расчет метрик схожести между курсом и рефлексией с использованием модуля № 11;
- Формирование визуального табличного и графового представления результатов сравнения с использованием модуля № 15.

Дополнительно, программные модули № 16 и № 17 (Рисунок 26) разработаны для улучшения результатов транскрибирования (распознавания речи) аудио-файлов рефлексии.

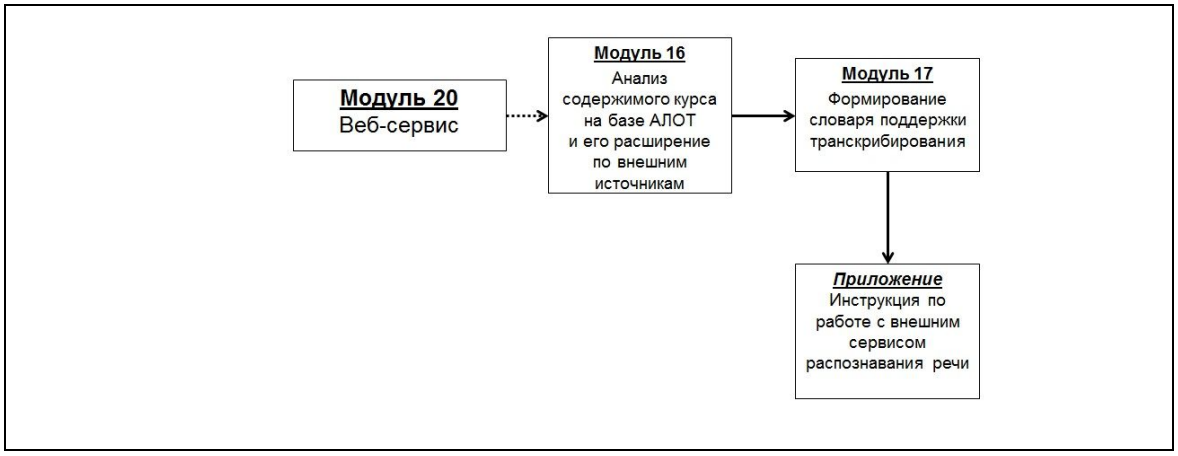


Рисунок 26 - Логика интеграции программных модулей поддержки транскрибирования

8.2.4. Интеграция программных модулей, реализующих алгоритмы индексирования текстов учебных курсов и текстов рефлексии обучаемых по тематической таксономии АНО «Университет 2035»

Схема интеграции программных модулей для индексирования текстов с использованием классификатора «Таксономия 2035» представлена на Рисунке 27:

- Модуль № 18 позволяет сконвертировать структуры данных описания классов классификатора «Таксономия 2035» в структуры данных базы данных описания лингвистических онтологий ПП АЛОТ;
- Модуль № 19 позволяет выгрузить получившиеся структуры данных в формате, который используется программными модулями ПП АЛОТ, отвечающими за индексирование текстов.

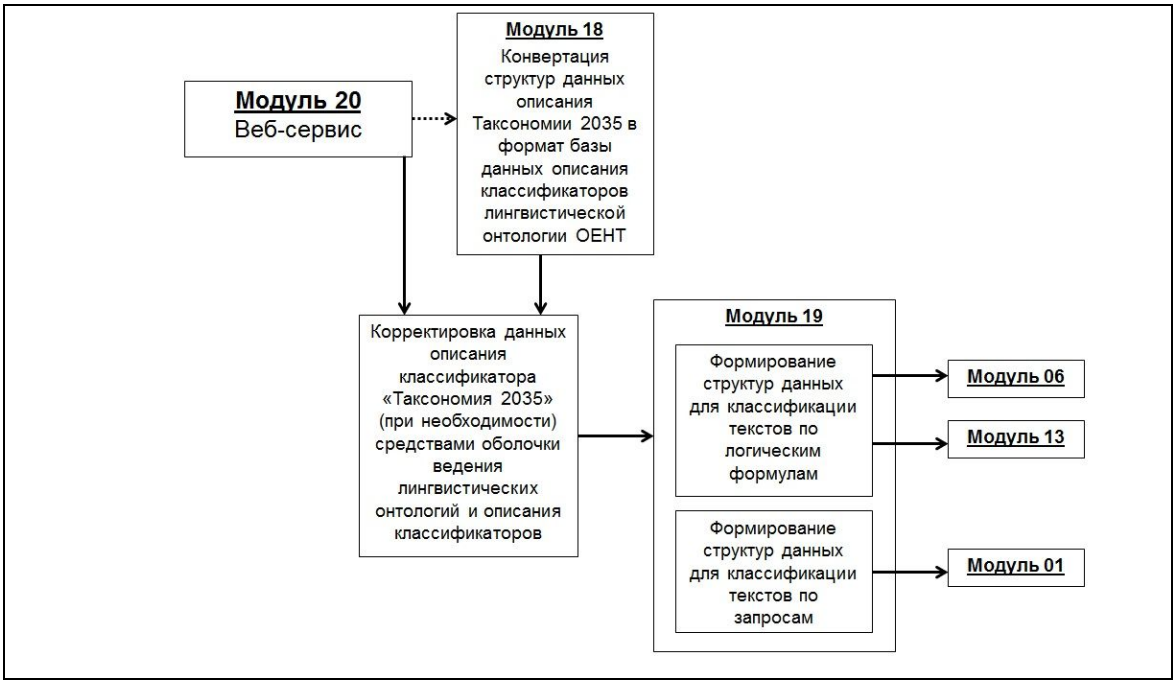


Рисунок 27 - Логика интеграции программных модулей классификации с использованием Таксономии 20.35

8.2.5. Реестр директорий программных модулей на стенде 2035

1. Рабочие папки проекта

C:\var\data\ - сохраненные данные

C:\var\scripts\ - модули, кроме 18 и 20

C:\var\www\ - модули 18 и 20

C:\Program Files\PostgreSQL\11\data\ - БД Postgresql

2. Папки установленного ПО

C:\Apache24\ - веб-сервер Apache

C:\Program Files\PostgreSQL\ - БД Postgresql

C:\ PHP74\ - интерпретатор PHP

C:\Python38 \ - Питон 3.8

C:\Util \ - вспомогательные утилиты типа curl, xpdf

3. Поставка

C:\LII\Distr\soft\ - дистрибутивы ПО из п.2

C:\LII\Distr\data\ - дампы БД, данные насчитанные на нащем стенде

C:\LII\Distr\Exe\ - поставка, исполняемые модули

C:\LII\Distr\Src\ - поставка, исходные коды

Примечание – при описании модулей №№ 01, 06, 07, 9, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20:

Используются директории (на стенде Заказчика на диске C) в /var/scripts/:

- comp2txt – модули №№ 01, 06 – 16;
- gmtpod5 – gmtpod (ПП АЛОТ), использующийся для получения графов (модуль № 07);
- mt_fod – использующийся для получения NLD (ПП АЛОТ);
- ruwiki – индекс и поисковик NearIdx (ПП АЛОТ).

Все модули на языке Python подключают модуль config.py, в котором находятся настройки https-сервисов и настройки соединения с СУБД PostgreSQL.

ЗАКЛЮЧЕНИЕ

Целью выполнения работ является выполнение пользовательских сценариев «Сравнения двух учебных курсов» и «Сравнения учебного курса и рефлексии обучающегося» путем создания методов понятийно-тематического анализа содержания учебных курсов, по которым собирается цифровой след в виде рефлексий в рамках деятельности Университета 2035,.

В качестве типового технического решения используется программный продукт Автоматизированная Лингвистическая Обработка Текстов (ПП АЛОТ), позволяющий построить модель тематического представления содержания текста на основе понятий большой лингвистической онтологии (лицензия на ПП АЛОТ была ранее приобретена Заказчиком у Исполнителя работ).

Возможная оригинальность и широта охвата материалов учебных курсов потенциально требует привлечения знаний, не описанных в текущей веерии используемых лингвистических онтологий ПП АЛОТ. Для расширения возможностей анализа содержания материалов учебных курсов и рефлексии обучающихся в рамках работ произведена модификация функционала программного продукта путем разработки дополнительных программных модулей, интегрированных с ПП АЛОТ.

В рамках работы решались следующие задачи:

- Разработка методов понятийно-тематического анализа содержания учебных курсов, включающая в себя разработку методов и алгоритмов сравнения материалов учебных курсов, оценка “похожести” курсов, выявление ключевых образовательных результатов, которые дает курс (знания, умения, компетенции, инструменты);

- Разработка методов и алгоритмов сравнения содержания учебно-методических материалов учебных курсов с рефлексией обучающихся;
- Разработка методов и алгоритмов автоматического пополнения словарей для систем распознавания речи, для повышения качества распознавания речи лекторов и обучающихся;
- Разработка методов и алгоритмов индексирования текстов учебных курсов и текстов рефлексии обучаемых (приписывания текстам соответствующих атрибутов) по тематической таксономии АНО «Университет 2035» (Таксономии 2035);
- Разработка web-сервиса, предоставляющего интерфейс программного продукта АЛОТ с включенными в него модулями.

В результате работ разработаны все предусмотренные методы и алгоритмы понятийно-тематического анализа содержания учебных курсов, создано 20 основных программных модулей к программному продукту «АЛОТ».

Таким образом все задачи работ по Договору № У-20/120 от 17 декабря 2020 г. выполнены.

ПРИЛОЖЕНИЕ А – ЯЗЫК ЗАПРОСОВ ИНФОРМАЦИОННО-ПОИСКОВОЙ СИСТЕМЫ NEARIDX

Простой поисковый запрос предполагает ввод в строку поиска условия поиска. Пользователь может ввести в поисковую форму одно слово или последовательность слов.

Информационно-поисковая система позволяет обрабатывать достаточно сложные запросы с использованием логических операторов и вложенности условий запроса (задаются скобками).

Условия запроса записываются как совокупность «операндов» (подзапросов). Выделяются два основных типа операндов:

- запрос по контексту, который записывается на естественном языке, система сама осуществляет перевод слов запроса в нормальную «словарную» форму. Если требуется точная форма слова (слов), то необходимо заключить слова такого запроса в кавычки;
- запрос по элементам поисковых индексов, которые задаются следующим образом:

  /AAAAA=”BBBBB”

  здесь **/AAAAA** – тип условия (предваряется символом «слэш») – название классификатора, **BBBBB** – параметр условия – значение элемента классификатора.

Некоторые запросы по элементам поисковых индексов относятся ко всей публикации (например, автор или год издания), некоторые имеют привязку к контексту (например, понятия тезауруса).

Подзапросы соединяются в запросе с помощью логических операторов и скобок, задающих вложенность и порядок обработки запроса:

- И (в любом регистре, можно также писать AND). Пробел между операндами или перевод строки также трактуются как условие И. В

запросе **A and B** релевантный документ должен удовлетворять условиям обоих подзапросов,

- ИЛИ (OR). В запросе **A or B** релевантный документ должен удовлетворять условиям хотя бы одного подзапроса,
- НЕ (NOT) – отрицание. В запросе **not A** релевантный документ не должен удовлетворять условиям подзапроса A.

Так что запрос может выглядеть одним из следующим образом (прописные буквы обозначают операнды):

```
A
A B
A and B
A and not B
A not B
A or B
A (B or C or D) and not (E or (F and G))
```

и т.д.

В строке поиска запрос записывается в текстовом виде и может быть отредактирован непосредственно пользователем.

При вычислении контекстно-зависимых запросов более релевантными публикациями являются такие, что содержащиеся в них элементы запроса более частотны, более характерны для данной публикации (по сравнению со средним уровнем по коллекции), расположенные ближе к началу публикации.

Допустимы следующие типы запросов:

Тип запроса	Примеры запросов	Семантика
простой контекстный запрос	турбинный цех	поиск документов, содержащих слова запроса, в том числе в других грамматических

Тип запроса	Примеры запросов	Семантика
		формах
контекстный запрос с условиями	(турбинный или генераторный) (цех или отделение)	дополнительно производится учет скобок и логических условий
точные формы слов	«турбинного цеха» (равносильно запросу: «турбинного» «цеха»)	будут искаться слова именно в той же форме
/Термин	/Термин=”РЕМОНТ”	поиск публикаций, где упоминалось именно это понятие
/Термин_расш	/Термин_расш=”РЕМОНТ”	поиск публикаций, где упоминалось именно это или любое из подчиненных понятий
/Теги	/Теги="ВОЕННЫЙ ВРЕМЯ"	поиск по ключевым словам

Специальные виды запросов

МИНУС (MINUS, NOT, [символ -]) - оператор вычитания выборки документов.

Пример запроса:

```
долг -чистый
долг NOT (чистый)
```

ИЛИ (OR, UNION) - оператор ИЛИ

Пример запроса:

```
рога ИЛИ копыта
```

И (AND, [пробел]) - оператор И (по сути пробел в тексте запрос всегда считается оператором И) .

Пример запроса:

```
цк кпсс
```

_РЯДОМ - поиск рядом стоящих

Пример запроса:

```
денег _РЯДОМ нет
```

ВНАЧАЛЕ - поиск сущности или запроса в начале текста (100 словопозий) .

Пример запроса:

```
ВНАЧАЛЕ( долг Казахстана )
```

ВКОНЦЕ - поиск в конце текста (100 словопозий)

Пример запроса:

```
ВКОНЦЕ( долг Казахстана )
```

ВНАЧАЛЕ20( ) - поиск сущности или запроса в первых 20 (N - любое) словопозициях .

Пример запроса:

```
ВНАЧАЛЕ10(долг Казахстана)
```

ВПРЕДЛ - поиск в одном предложении.

Пример запроса:

```
ВПРЕДЛ(россия америка)
```

ВПРЕДЛ3 ( ) - поиск в N предложениях текста

Пример запроса:

```
ВПРЕДЛ2(россия америка)
```

_ВНУТРИ - поиск сущности внутри другой сущности, например фамилию внутри прямой речи

Пример запроса:

```
/Пр_речь="@" _ВНУТРИ /Должность="@" //документы, где должность
                внутри прямой речи
```

НЕТВНУТРИ - поиск документов в которых одна сущность не находиться внутри другой .

Пример запроса:

```
/Пр_речь="@" _НЕТВНУТРИ /Должность="@" //документы, где есть
                прямая речь без должности внутри
```

ПРИЛОЖЕНИЕ Б – ПРИМЕРЫ ТИПОВОГО ОФОРМЛЕНИЯ МАТЕРИАЛОВ УЧЕБНОГО КУРСА И РЕЗУЛЬТАТОВ СБОРА РЕФЛЕКСИИ СЛУШАТЕЛЕЙ

Б.1. Пример текста типовой учебной программы

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный университет имени М.В. Ломоносова»

«Утверждаю»

Декан факультета ВМК МГУ

имени М.В. Ломоносова

академик Е.И. Моисеев

«___»_____ 2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Анализ больших текстовых данных и информационный поиск

Уровень высшего образования – подготовка магистров (интегрированная магистратура)

Направление подготовки – «Прикладная математика и информатика» (010400)

Направленность (профиль) – «Интеллектуальный анализ больших данных»

Автор: ведущий научный сотрудник НИВЦ МГУ, д.т.н. Лукашевич Н.В.

2018

1. НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ

Анализ больших текстовых данных и информационный поиск

2. УРОВЕНЬ ВЫСШЕГО ОБРАЗОВАНИЯ

Подготовка научно-педагогических кадров в магистратуре

3. НАПРАВЛЕНИЕ ПОДГОТОВКИ, НАПРАВЛЕННОСТЬ (ПРОФИЛЬ) ПОДГОТОВКИ

Направление 01.04.02 «Прикладная математика и информатика».

Направленность (профиль) «Интеллектуальный анализ больших данных»

4. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина входит в обязательную часть магистерской образовательной программы «Интеллектуальный анализ больших данных», изучается в 3-м семестре.

5. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Дисциплина участвует в формировании следующих компетенций образовательной программы:

Формируемые компетенции	Планируемые результаты обучени
способность анализировать задач связанную с автоматической обработк текстов, в конкретной предметной област выбирать для ее решения соответствующ метод; способность подбора имеющих программных решений, доступных в интерн или программной реализации собственно решения; способность измерения качест полученных результатов и анализ ошиб программного решения (СПК-8);	З1 (СПК-8) Знать: Знать основные особенности естественного языка, уровней языковой системы и моделей обработки текстов; современные модели информационного поиска; методы автоматической классификации и кластеризации текстов У1 (СПК-8) Уметь Уметь применять на практике модели информационного поиска для решения задач в рамках информационных систем, применять методы классификации, кластеризации для извлечения знаний и информации из текстов В1 (СПК-8) Владеть **Владеть навыками** выбора методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов обработки текстов для коррекции используемых алгоритмов обработки текстов.

Оценочные средства для промежуточной аттестации приведены в Приложении.

6. ОБЪЕМ ДИСЦИПЛИНЫ

Объем дисциплины составляет 3 зачетные единицы, всего 108 часов.

54 часов составляет контактная работа с преподавателем – 42 часов занятий лекционного типа, 8 часов занятий семинарского типа (семинары, самостоятельные работы и т.п.), 0 часов консультаций, 4 часа промежуточные аттестации в форме письменных контрольных работ.

54 часа составляет самостоятельная работа учащегося.

Письменные контрольные работы включают ответы на теоретические вопросы и выполнение практических заданий на применение моделей информационного поиска и методов оценки их качества .

7. ВХОДНЫЕ ТРЕБОВАНИЯ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Учащиеся должны владеть знаниями по дискретной математике, математической логике, языкам программирования, теория вероятностей и математической статистике в объеме, соответствующем основным образовательным программам бакалавриата по укрупненным группам направлений и специальностей 01.00.00 «Математика и механика», 02.00.00 «Компьютерные и информационные науки».

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В процессе обучения используются инструментальные системы морфологического анализа русского языка (mystem, pymorphy), инструментальная система машинного обучения sci-kit learn (http://scikit-learn.org/stable/).

9. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Наименование и краткое содержание разделов и тем дисциплины, форма промежуточной аттестации по дисциплине	**Всего (часы)**	В том числе								
		Контактная работа (работа во взаимодействии с преподавателем), часы из них						**Самостоятельная работа учащегося, часы** из них		
		Занятия лекционного типа	Занятия семинарского типа	Групповые консультации	Индивидуальные консультации	Учебные занятия, направленные на проведение текущего контроля успеваемости: коллоквиумы, практические контрольные занятия и др.	**Всего**	Выполнение домашних заданий	Подготовка рефератов и т.п..	**Всего**
Тема 1. Введение: Определение сферы информационного поиска, задачи информационного поиска, Информационно-поисковы системы различной направленности. Архитектура информационно-поисковых систем	**10**	6	-	-	-	-	**6**	**4**	**-**	**4**
Тема 2. .Модели информационного поиска. Оценка качества информационного поиска .	**24**	8	8	-	-	-	**16**	**8**	-	**8**
Тема 3. .Методы расширения информационно-поисковых запросов. Вопросно-ответные системы. Диалоговые системы.	**16**	8	0	0	-	0	**8**	8	-	**8**
Тема 4. Учет	**12**	8	0	0	-	-	**8**	4	-	**4**

различных факторов. Анализ ссылок, логи запросов, анализ кликов, персонализация выдачи, поисковые сессии, словосочетания и близость расположения, тематические вероятностные модели. Комбинирование факторов. Модели Learning to-rank.										
Тема 5. Автоматическая классификация и кластеризация текстов. Типы классификации. Тематическая классификация и анализ тональности. Методы классификации. Особенности методов классификации. Методы автоматической кластеризации. Автоматическое аннотирование..	**26**	12	0	0	-	-	**12**	10	-	**14**
Контрольные работы (2 раза)	**18**					4	**4**	14		**14**
Итого	**108**	**54**						**54**		

10. Учебно-методические материалы для самостоятельной работы учащихся

Самостоятельная работа учащихся состоит в изучении лекционного материала, учебно-методической литературы, выполнении домашних заданий и подготовки к промежуточной аттестации.

11. РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ

Основная учебно-методическая литература

1) Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных. место издания *Изд-во НИУ ВШЭ Москва*, ISBN 978-5-9909752-1-7, 269 с.

2) Лукашевич Н.В. Тезаурусы в задачах информационного поиска. – М.: Изд-во Московского университета, 2011.

3) Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. — Вильямс, 2011.

4) Турнбулл Д., Берримен Дж. Релевантный поиск с использованием Elasticsearch и Solr, DMK-press, 2018.

5) Baeza-Yates, R. and Ribeiro-Neto, B. Modern Information Retrieval, Adison Wesley, 1999.

6) Manning, Ch. D., H. Schütze. Foundations of Statistical Natural Language Processing. MIT Press, 1999.

Дополнительная учебно-методическая литература

1) Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И. и др. – М.: МИЭМ, 2011.

2) Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения. Изд-во ИНТУИТ, 2009.

3) Васильев В. Г., Кривенко М. П. Методы автоматизированной обработки текстов. – М.: ИПИ РАН, 2008.

4) Ландэ Д. В., Снарский А. А., Безсуднов И. В. Интернетика: Навигация в сложных сетях: модели и алгоритмы. — М.: Либроком (Editorial URSS), 2009.

5) Леонтьева Н. Н. Автоматическое понимание текстов: Системы, модели, ресурсы: Учебное пособие – М.: Академия, 2006.

6) Jurafsky D., Martin J. Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, 2000..

Ресурсы информационно-телекоммуникационной сети Интернет

1) http://scikit-learn.org/stable/index.html

2) https://tech.yandex.ru/mystem/

3) https://nlp.stanford.edu/IR-book/

4)

Информационные технологии, используемые в процессе обучения

В процессе обучения используются инструменты морфологического анализа текстов на русском языке (PyMorphy, MyStem), программный пакет машинного обучения Scikit learn.

Материально-техническая база

Для преподавания дисциплины требуется компьютерный класс, оборудованный проектором, а также маркерной или меловой доской.

12. ЯЗЫК ПРЕПОДАВАНИЯ

Русский

13. РАЗРАБОТЧИК ПРОГРАММЫ, ПРЕПОДАВАТЕЛИ

д.т.н., доцент Лукашевич Н.В.

Приложение

Оценочные средства для промежуточной аттестации по дисциплине «Модели Анализ текстовых данных и информационный поиск»

Промежуточная аттестация основана она суммарной оценке всех домашних и аудиторных работ курса, отображающей приобретенные учащимся знания, умения и навыки, а также включает индивидуальное собеседование, проверяющее приобретенные знания.

Средства для оценивания планируемых результатов обучения, критерии и показатели оценивания приведены ниже.

РЕЗУЛЬТАТ ОБУЧЕНИЯ	**КРИТЕРИИ и ПОКАЗАТЕЛИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ из соответствующих карт компетенций**					**ОЦЕНОЧНЫЕ СРЕДСТВА**
	1	2	3	4	5	
	Неудовлетворительно	**Неудовлетворительно**	**Удовлетворительно**	**Хорошо**	**Отлично**	
З1 (СПК-8) Знать основные особенности естественного языка, уровней языковой системы и моделей обработки текстов; современные модели информационного поиска; методы автоматической классификации и кластеризации текстов	Отсутствие знаний	Фрагментарные представления об основных особенностях естественного языка, уровнях языковой системы и моделей обработки текстов; современных моделей информационного поиска; методов автоматической классификации и кластеризации текстов	В целом сформированные, но неполные знания об основных особенностях естественного языка, уровнях языковой системы и моделей обработки текстов; современных моделей информационного поиска; методов автоматической классификации и кластеризации текстов	Сформированные, но содержащие отдельные пробелы знания об основных особенностях естественного языка, уровнях языковой системы и моделей обработки текстов; современных моделей информационного поиска; методов автоматической классификации и кластеризации текстов	Сформированные систематические знания об основных особенностях естественного языка, уровнях языковой системы и моделей обработки текстов; современных моделей информационного поиска; методов автоматической классификации и кластеризации текстов	Контрольные работы, письменный экзамен
У1 (СПК-8) Уметь применять на практике модели	Отсутствие умений	Фрагментарные умения применять на практике модели информационного	В целом сформированное, но не систематич	Сформированное, но содержащее	Сформированное систематическое умение	практическое контрольное задание

информационного поиска для решения задач в рамках информационных систем, применять методы классификации, кластеризации, методы извлечения знаний и информации из текстов		поиска для решения задач в рамках информационных систем, методы классификации, кластеризации, методы извлечения знаний и информации из текстов	ческое умение применять на практике модели информационного поиска для решения задач в рамках информационных систем, методы классификации, кластеризации, методы извлечения знаний и информации из текстов	отдельные пробелы умение применять на практике модели информационного поиска для решения задач в рамках информационных систем, методы классификации, кластеризации, методы извлечения знаний и информации из текстов	применять на практике модели информационного поиска для решения задач в рамках информационных систем, методы классификации, кластеризации, методы извлечения знаний и информации из текстов	
В1 (СПК-8) Владеть **Владеть навыками** выбора методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов обработки текстов для коррекции используемых алгоритмов обработки текстов	Отсутствие навыков	Фрагментарное владение выбором методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов обработки текстов для коррекции используемых алгоритмов обработки текстов	В целом сформированное, но не систематическое владение выбором методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов обработки текстов	Сформированное, но содержащее отдельные пробелы владение выбором методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов	Сформированное систематическое владение выбором методов решения конкретной задачи автоматической обработки текстов (статистический, инженерно-лингвистический, комбинированный); анализа результатов обработки текстов для коррекции используемых алгоритмов	практическое контрольное задание

			для коррекции используемых алгоритмов обработки текстов	обработки и текстов для коррекции используемых алгоритмов обработки текстов	обработки текстов	

Фонды оценочных средств

Примерные практические домашние задания для текущего контроля успеваемости.

ПДЗ ТК1. Написание программы, реализующей векторную модель информационного поиска. Проверка на основе нескольких статей Википедии. В качестве документов для поиска выступают предложения выбранных статей Википедии

- Запросы – это факты из Википедии
- Коллекция – это все упомянутые статьи из всех фактов (не менее 2 из каждого факта)
- Документы – это предложения из статей Википедии, указанных в этих фактах, т.е. объединенная коллекция предложений статей всех фактов
- Все должно быть обработано морфологическим анализатором
- Нужно найти наиболее релевантные предложения
  - По векторной модели без idf
  - По tf.idf (idf в данном случае – это количество предложений, в которых встречалось слово)
  - Нормализация запроса и предложения
  - Т.е. выстроить все предложения из статей по мере сходства с запросом по векторной модели.
  - В отчете должны быть показаны веса выдаваемых предложений

ПДЗ ТК5. Реализация системы классификации цитат из новостных статей по тональности

- Выдаются данные для задачи анализа тональности: цитаты
  - Обучающая коллекция с ответами
  - Тестовая коллекция с ответами
  - Нужно обучиться на обучающей коллекции, затем провериться на тестовой
  - Классификация на три класса (позитивный, негативный, нейтральный)
  - Нужно выбрать метод из пакета (http://scikit-learn.org/stable/) и проверить качество классификации на разных типах векторизации: булевские вектора, частоты, tf.idf

- Встроенный механизм векторизации:
- http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

– Попробовать методы: наивный Байес, SVM, разные параметры, что-то еще
– Оценить результаты (F-мера), представить отчет.

Список вопросов для письменного ответа на промежуточной аттестации.

1. Основные понятия информационного поиска
2. Виды поисковых систем по охвату и направленности. Особенности разных типов поисковых систем
3. Особенности научного поиска
4. Основные этапы обработки текстов в поисковой машине
5. Основные этапы обработки запроса в поисковой машине
6. Что такое графематический анализ? Что такое лемматизация
7. Как работает словарный морфологический анализ?
8. Как морфологические анализаторы обрабатывают слова, отсутствующие в словаре
9. Что такое постморфологический анализ. Основные методы.
10. Булевская модель информационного поиска. Преимущества и недостатки булевской модели поиска
11. Как измеряется качество булевского поиска
12. Алгоритм сопоставления запроса с документами (Алгоритм Merge)
13. Что такое векторная модель информационного поиска?
14. Поясните смысл показателей *idf* и *tf.idf*. Способы вычисления tf.
15. Классическая процедура оценки качества информационно поиска
16. Что такое РОМИП, какие задачи в нем решаются?
17. Что такое кривая полнота-точность?
18. Что такое 11-точечный график TREC?
19. Что такое пулинг в информационном поиске? Сложности, связанные с пулингом
20. Оценка качества в поисковых машинах
21. Шкалы оценок. Мера NDCG
22. Что такое информационно-поисковые тезаурусы? Зачем они нужны? Где применяются сейчас
23. Назовите методы расширения запросов пользователей при информационном поиске.
24. Что означает термин *relevance feedback*? Поясните основные принципы работы
25. Алгоритм Роккио для *relevance feedback*

26. Порождение и применение автоматического тезауруса для расширения запросов
27. Вероятностная модель информационного поиска: основная идея, различие с векторной моделью
28. Что такое языковые статистические модели?
29. Языковая модель информационного поиска
30. Вопросно-ответные системы: постановка задачи. основные компоненты, особенности тестирования.
31. Классификация вопросов в вопросо-ответных системах. Типы вопросов и типы ответов
32. Особенности обработки фактоидных вопросов
33. Особенности обработки нефактоидных вопросов.
34. Что такое PageRank? Зачем нужен, как вычисляется
35. Алгоритм HITS
36. Особенности использования кликов пользователя в качестве фидбека от пользователя
37. Классификация запросов по цели. Зачем нужна. Особенности обработки разных типов запросов
38. Поисковые сессии в интернет. Как выделять, зачем.
39. Особенности использования кликов пользователя в качестве фидбека от пользователя. Каскадная модель при обработке кликов.

Список вопросов для письменного ответа на второй аттестации .

1. Укажите основные методы автоматической рубрикации (классификации) текстов.
2. Что такое инженерный метод классификации текстов? Плюсы и минусы инженерных методов классификации
3. Укажите плюсы и минусы ручного рубрицирования.
4. Метод Байеса для автоматической классификации текстов
5. Метод Роккио для автоматической классификации текстов
6. Метод Knn для автоматической классификации текстов
7. Поясните основной принцип метода SVM для автоматической рубрикации текстов
8. Плюсы и минусы методов машинного обучения для рубрикации текстов
9. Особенности применения методов машинного обучения при классификации текстов в зависимости от размера обучающей коллекции
10. Что такое кластеризация текстов? Чем она отличается от классификации (рубрикации) текстов?
11. Метод K-means для кластеризации текстов
12. Аггломеративная кластеризация – основной принцип и подвиды
13. Методы тестирования автоматической кластеризации
14. Особенности кластеризации потока новостей в реальном времени
15. Анализ тональности. Проблемы. Виды задач.
16. Анализ тональности как задача классификации: Методы, Признаки, Меры качества
17. Автоматическое аннотирование. Виды автоматических аннотаций.

18. Методы и признаки для отбора предложений в экстрактивном методе автоматического аннотирования
19. Метод MMR автоматического аннотирования
20. Метрика Rouge для тестирования автоматических аннотаций
21. Метод пирамид для тестирования автоматических аннотаций
22. Исправление несловарных ошибок на основе применения правила Байеса
23. Исправление ошибок перехода в другое словарное слова на основе применения правила Байса
24. Учет контекста при исправлении ошибок написания
25. Методы приблизительного вычисления сходства документов в реальных поисковых системах
26. Обработка фразовых запросов и запросов с указанием близости слов в поисковых системах
27. Позиционный индекс в поисковой системе. Зачем нужен, как обрабатывается
28. Какие факторы помимо веса tf.idf учитываются в поисковых моделях, как создаются многофакторные модели в информационно поиске
29. Метод шинглов для определения дубликатов документов
30. Краулинг в Интернет
31. Обучение ранжированию. Зачем нужно? Основные подходы
32. Понятие перестановок в задаче обучения ранжированию
33. Латентный семантический анализ (LSA). Зачем нужен? Основная идея. Преимущества и недостатки
34. Что такое тематические модели. Виды тематических моделей. Основной подход к моделированию
35. Оценки качества тематических моделей
36. Традиционные диалоговые системы: архитектура, основные этапы работы
37. Чат-боты: классификация, основные методы

Методические материалы для проведения процедур оценивания результатов обучения

Совокупное выполнение домашних заданий оценивается по шкале до пяти баллов. Производится усреднение оценок за контрольные работы и оценки за выполнение домашних заданий. Если оценка больше или равна 4, то она и считается оценкой за курс. Если оценка меньше, то проводится дополнительная письменная работа по вопросам контрольных работ и заданных домашних заданий, в результате чего выставляется оценка за курс.

Б2. Фрагмент словаря для очистки материалов учебного курса

ADISON WESLEY
EDITORIAL URSS
ISBN
MIT PRESS
АВТОР
В ОБЪЕМЕ, СООТВЕТСТВУЮЩЕМ ОСНОВНЫМ ОБРАЗОВАТЕЛЬНЫМ ПРОГРАММАМ БАКАЛАВРИАТА ПО УКРУПНЕННЫМ ГРУППАМ НАПРАВЛЕНИЙ И СПЕЦИАЛЬНОСТЕЙ
В ОБЯЗАТЕЛЬНУЮ ЧАСТЬ
В ПРОЦЕССЕ ОБУЧЕНИЯ ИСПОЛЬЗУЕТСЯ
В ПРОЦЕССЕ ОБУЧЕНИЯ ИСПОЛЬЗУЮТСЯ
В ЦЕЛОМ СФОРМИРОВАННОЕ, НО НЕ СИСТЕМАТИЧЕСКОЕ ВЛАДЕНИЕ

Б3. Пример типового результата сбора рефлексии слушателей

Чему вы научились в рамках курса?	Есть ли у вас иные комментарии к пройденному материалу?	Будут ли знания и навыки, приобретенные на курсе, полезны в вашей профессиональной деятельности (скорее нет/некоторые элементы будут полезны/однозначно будут полезны)
Базовой работе с программой AutoCad	хотелось бы больше видеоматериалов и конкретных примеров	однозначно будут полезны
базовой работе с программой AutoCad	предлагаю дать больше времени на изучение каждой темы, и конкретное видео по теме, специально для даннного курса. рассказать побольше где используется каждая функция	однозначно будут полезны
базовой работе с программой AutoCad	Необходимо добавить небольшие видео ролики, т.к. для учеников, которые впервые видят интерфейс AutoCad сложно перенести данные из лекции в AutoCad (для меня такой сложности не было).	некоторые элементы будут полезны
Курс помог овладеть начальными знаниями.	Пожелание: заменить обучающие видео. Добавить больше видео на построение сложных чертежей.	некоторые элементы будут полезны
Большое спасибо вам за вше обучение, я смогла овладеть базовыми навыками работы в AutoCad.	Хочу выразить огромную благодарность преподавателям, которые мне помогли овладеть базовыми знаниями программы.	однозначно будут полезны
данный курс позволил овладеть базовыми навыками.	Предложения сделать продвинутый курс по программе AutoCad.	однозначно будут полезны
Базовыми знаниями с программой ознакомилась, было сложно, но интересно!	Мало времени на обучение отводится, надо больше.	однозначно будут полезны
базовой работе с	Думаю, можно сделать больше	однозначно будут полезны

Чему вы научились в рамках курса?	Есть ли у вас иные комментарии к пройденному материалу?	Будут ли знания и навыки, приобретенные на курсе, полезны в вашей профессиональной деятельности (скорее нет/некоторые элементы будут полезны/однозначно будут полезны)
программой AutoCad	практических заданий.	
базовой работе с программой AutoCad	нет	однозначно будут полезны
Считаю, что данный курс вполне позволил мне овладеть базовыми знаниями работы с Autocad.	Обучение очень понравилось, хоть и возникали некие трудности, но думаю это к лучшему, т.к. это позволяло более глубже погружаться в обучение и овладевать знаниями Autocad.	однозначно будут полезны
овладел базовыми знаниями работы с программой AutoCad	Огромное спасибо нашим преподователям ! Надеюсь ,что наш колледж будет и дальше продвигать такие курсы и желаю чтоб было побольше таких хороших,заботливых и отзывчивых преподователей! Спасибо ВАМ за ваш труд!	однозначно будут полезны
овладел базовыми знаниями работы с программой AutoCad	нет	некоторые элементы будут полезны
овладел базовыми знаниями работы с программой AutoCad	нет	некоторые элементы будут полезны
овладел базовыми навыками	тяжело усваивается информация когда читаешь ее с экрана. В ресурсах интернет много доступных видео с более подробными и понятными уроками	однозначно будут полезны
овладел базовыми навыками	отличные курс, базовые знания получены, курс был очень полезен, спасибо.	однозначно будут полезны

Чему вы научились в рамках курса?	Есть ли у вас иные комментарии к пройденному материалу?	Будут ли знания и навыки, приобретенные на курсе, полезны в вашей профессиональной деятельности (скорее нет/некоторые элементы будут полезны/однозначно будут полезны)
овладел базовыми навыками	курс понятный, полезный, базовые знания помогут дальше развивать практический опыт использования автокад	однозначно будут полезны
Данный курс позволил мне освоить базовые функции программы AutoCad.	Лекционный материал был понятным, доступным и наглядным, достаточным по объему.	однозначно будут полезны
Данный курс позволил мне освоить базовые функции программы AutoCad.	Но стоит также добавить раздел по настройке рабочей среды и пространства. Назначение вспомогательных команд такие как привязка, ОРТО, полярное отслеживание и т.д. Принцип отображения в Листах, свойства объектов. Также хочется курсы для более продвинутой аудитории	однозначно будут полезны
курс полностью позволяет овладеть базовыми знаниями работы с программой.	нет	однозначно будут полезны
овладел базовыми навыками	нет	однозначно будут полезны
узнало много нового из программы AutoCad	Узнала много нового, несмотря на то что с программой знакома давно и работала в ней какое то время. Поняла, что нужно время от времени проходить подобное обучение, чтобы идти в ногу со временем и использовать максимум возможностей, которые предлагает этот программный продукт. Многими удобными функциями проектировщики не пользуются только потому, что лень разобраться, проще по-старинке прорисовывать каждый элемент, чем, например освоить функцию "массив" и	однозначно будут полезны

Чему вы научились в рамках курса?	Есть ли у вас иные комментарии к пройденному материалу?	Будут ли знания и навыки, приобретенные на курсе, полезны в вашей профессиональной деятельности (скорее нет/некоторые элементы будут полезны/однозначно будут полезны)
	таких примеров можно привести много. Когда проходишь обучающий курс волей-неволей приходится разобраться, вникнуть, потратить какое то время, где-то заставить себя. Но когда понимаешь, сколько впереди сэкономленного времени появляется, какие горизонты открываются, приходит осознание необходимости создания и проведения таких обучающих программ. Спасибо большое, буду пользоваться приобретенной информацией. Если будут открываться новые обучающие курсы и Вы меня пригласите - буду очень рада. Спасибо огромное Анжелике. Она очень оперативно отвечала на все возникающие вопросы.	
овладел базовыми навыками	Преподователи квалифицированные, доступно объясняют.	однозначно будут полезны
закрепил знания по работе в автокад	нет	однозначно будут полезны
данный курс позволил овладеть примитивными з наниями работы с программой AutoCad.	Считаю, для новичков очень хорошая программа, все понятно и доступно. Единственное пожелание - побольше сопровождающего видеоматериала, либо здорово было бы, данный курс переложить на видео, так сказать визуализировать.	однозначно будут полезны
данный курс позволил закрепить знания.	Что стоит отметить - первые задания очень простые, а начиная с задания "Сопряжения" - сразу достаточно сложные, т.е. человеку без опыта будет крайне сложно. Некоторые из применяемых команд рассматриваются в последующих темах, некоторые не рассматриваются в курсе.	однозначно будут полезны
овладел базовыми	Огромное спасибо преподавателям! Все было супер!	однозначно будут полезны

Чему вы научились в рамках курса?	Есть ли у вас иные комментарии к пройденному материалу?	Будут ли знания и навыки, приобретенные на курсе, полезны в вашей профессиональной деятельности (скорее нет/некоторые элементы будут полезны/однозначно будут полезны)
навыками		
овладел базовыми навыками	Можно было лишь добавить информацию по работе с моделью и листами. Удачи в дальнейшей работе	однозначно будут полезны
овладел базовыми навыками	нет	некоторые элементы будут полезны
овладел базовыми навыками	Мое предложение- давать больше практических заданий, с алгоритмом выполнения, для лучшего понимания возможностей программы.	некоторые элементы будут полезны
овладел базовыми навыками	Здравствуйте. Большое спасибо,новые знания получены! Все доступно и понятно.	некоторые элементы будут полезны
овладел базовыми навыками	хотелось бы побольше видеуроков, задания понравились. Они предполагают использование полученных знаний в совокупности.	однозначно будут полезны
овладел базовыми навыками	хотелось бы продолжение курса, у программы оказывается столько возможностей!!! Много интересного!!! Спасибо преподавателям, что было не понятно объясняли.	однозначно будут полезны
овладел базовыми навыками	Не хватает большего примера построения чертежей деталей с пошаговой инструкцией	некоторые элементы будут полезны
овладел базовыми навыками	нет	однозначно будут полезны
овладел базовыми навыками	Хотелось бы побольше заданий, рассчитанных на самостоятельное выполнение	однозначно будут полезны
овладел базовыми навыками	Задания к самостоятельной работе были доступно представлены.	однозначно будут полезны

Чему вы научились в рамках курса?	Есть ли у вас иные комментарии к пройденному материалу?	Будут ли знания и навыки, приобретенные на курсе, полезны в вашей профессиональной деятельности (скорее нет/некоторые элементы будут полезны/однозначно будут полезны)
овладел базовыми навыками	Спасибо организаторам за организацию курса. Считаю курс очень полезным, хочется пожелать увеличить колличество часов или разработать курс для дальнейшего обучения. Огромное спасибо!	однозначно будут полезны
овладел базовыми навыками	Данный курс помог узнать новые функции программы, которые были мне непонятны, более углубленно изучить возможности данной программы. Спасибо за обучение!	однозначно будут полезны
мною получены и применены базовые знания о проектировании в AutoCAD.	нет	однозначно будут полезны
курс мне позволил овладеть базовыми навыками.	Хотелось бы побольше практических занятий по построению деталей с пояснением. В целом очень понравилось. Хотелось бы еще пройти курсы в других направлениях.	однозначно будут полезны

ПРИЛОЖЕНИЕ В – ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ВЕБ-СЕРВИСА ATS ДЛЯ РАСПОЗНАВАНИЯ РЕЧИ

Amazon Transcribe (https://aws.amazon.com/ru/transcribe/faqs/) – это сервис AWS, позволяющий преобразовывать речь в текст. Благодаря технологии автоматического распознавания речи (ASR) клиенты могут использовать Amazon Transcribe для решения самых разных бизнес-задач, включая расшифровку телефонных обращений в службу поддержки, создание субтитров для аудио- и видеоконтента, а также текстового анализа аудио- и видеоконтента.

В.1. Подготовка и использование словаря при работе в ATS Transcribe

Требования и рекомендации по составлению словаря

- Словарь задается в текстовом редакторе. Он включает список термов в текстовом виде, каждый терм в новой строке. Словосочетания вводятся через дефис вместо пробела. Для задания слов на других языках, рекомендуется их заменять на примерное русское произношение (не поддерживаются символы не из русского языка). Аббревиатуры (для русского языка) сервисом также не поддерживаются.
- Заносить стоит наиболее частотные слова общетехнического характера (такие как метод, сегмент и проч.) и специфическую лексику из предметной области. По возможности стоит избегать маленьких коротких слов (например, объединяя их в характерные словосочетания).
- Полученный файл сохраняется в формате .txt и кодировке UTF-8

Существуют и другие, более продвинутые способы задания, например, с заданием транскрипции. Здесь они не рассматриваются

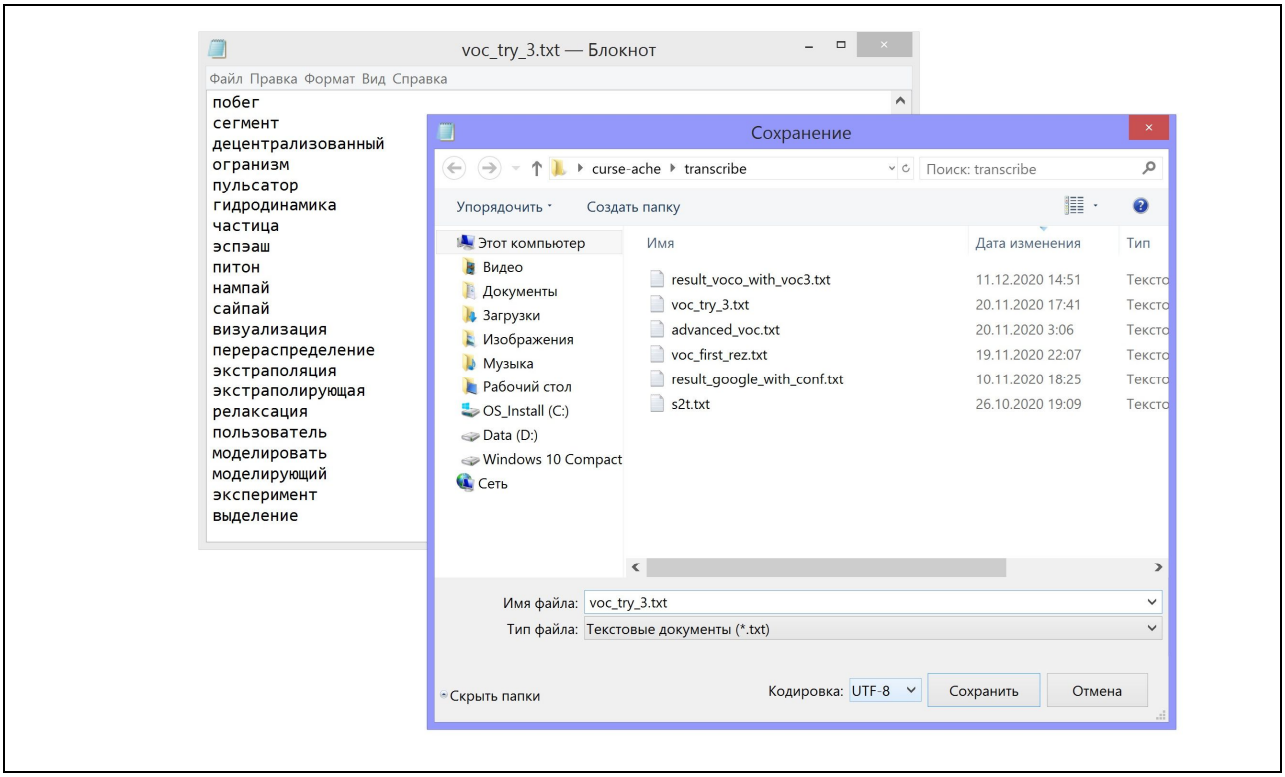


Рисунок В.1 - Пример корректно составленного словаря

В.2. Загрузка аудиофайлов для работы в ATS Transcribe

Выбор бакета в S3

- Воспользуемся сервисом Amazon S3: https://s3.console.aws.amazon.com/s3/
- Для использования обязательна регистрация
- Создаем новый бакет с помощью кнопки Create bucket или же используем один из уже созданных
- Один бакет может вмещать любое число файлов

Нужно использовать только те бакеты, регион которых будет совпадать с регионом, указанным в сервисе Transcribe.

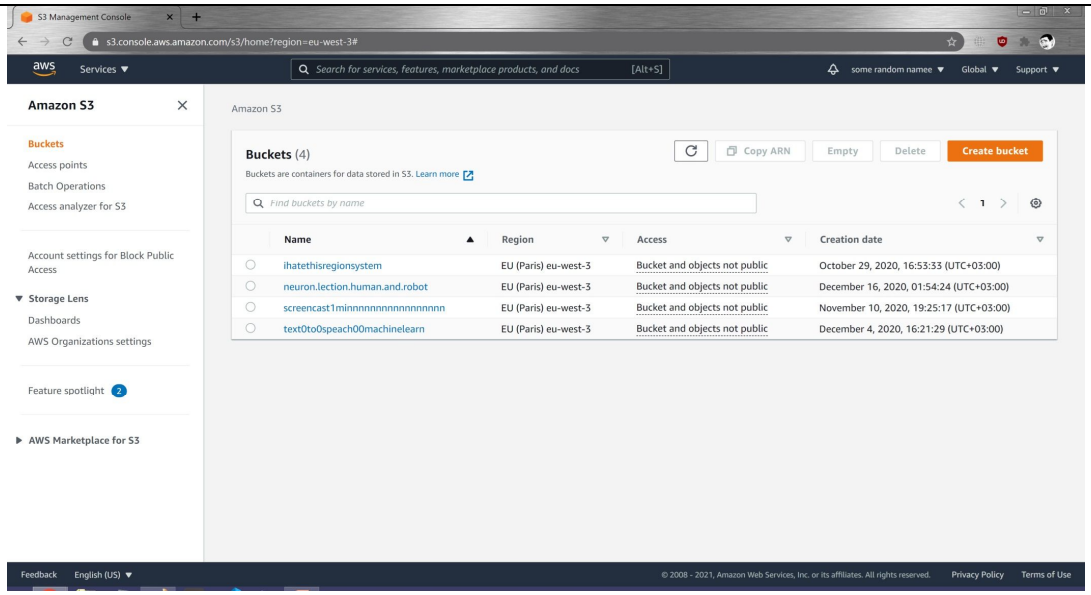


Рисунок В.2 - Примерный вид страницы сервиса S3

Создание нового бакета (Рисунки В.3а, В3б).

- Необходимо выбрать имя
- Необходимо выбрать регион, совпадающий с регионом сервиса Transcribe
- Остальные настройки можно оставить как есть
- Кнопка Create bucket внизу страницы завершает настройку

В случае успеха созданный бакет отобразится в списке доступных

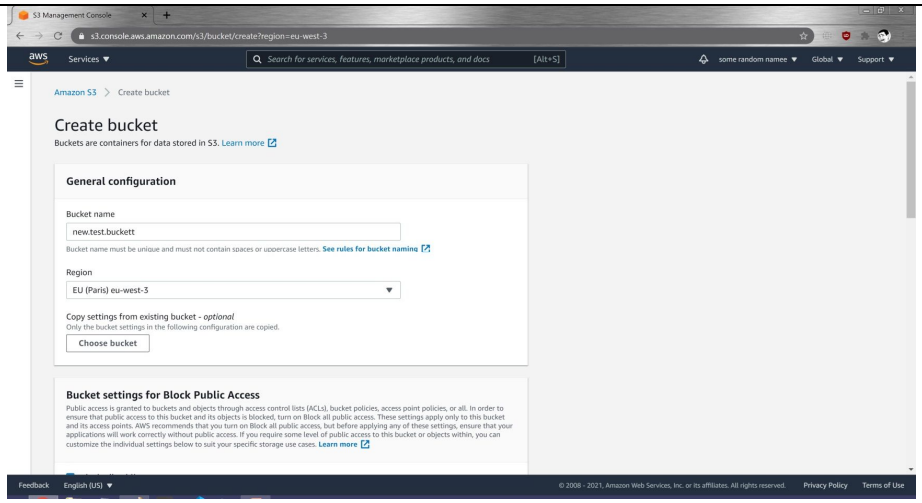


Рисунок В.3а – Создание нового бакета

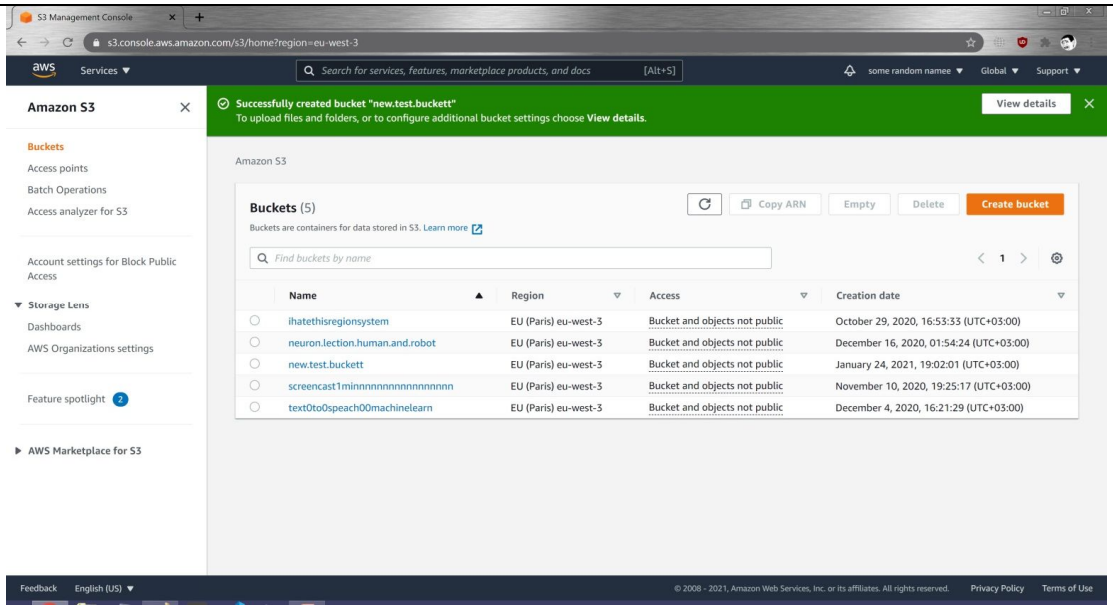


Рисунок В.3б - Создание нового бакета

В.4. Распознавание речи с помощью ATS Transcribe

Сервис и тарифы

- Сервис Amazon Transcribe: https://aws.amazon.com/ru/transcribe/
- Для использования необходим аккаунт Amazon
- Сервис условно бесплатный в течении 12 месяцев и имеет месячный лимит по суммарному времени аудио (60 минут)
- Другие лимиты для бесплатного пользования отсутствуют

Платный тариф: посекундно, 0,00040$\сек (примерно 1.5$\час)

Ценовая политика — примерно 1.5$ за час аудио. Уменьшенные тарифы начинаются где-то в пятизначных числах минут, так что по сути применимы только к реальному использованию в бизнесе.

По текущему опыту, для хорошего распознавания аудио и построения словаря по 30 минутному аудио требуется в среднем 2-4 прогонки.

Общие моменты

- Подача аудиоданных происходит с помощью сервиса S3, поэтому их надо сначала туда загрузить

- Сервис использует привязку к региону серверов. Регион можно выбрать в меню, расположенном в правом верхнем углу
- Регион должен совпадать с регионом бакета, в котором хранится аудиофайл для распознания

Для начала работы нажимаем кнопку Create job (Рисунок В.4).

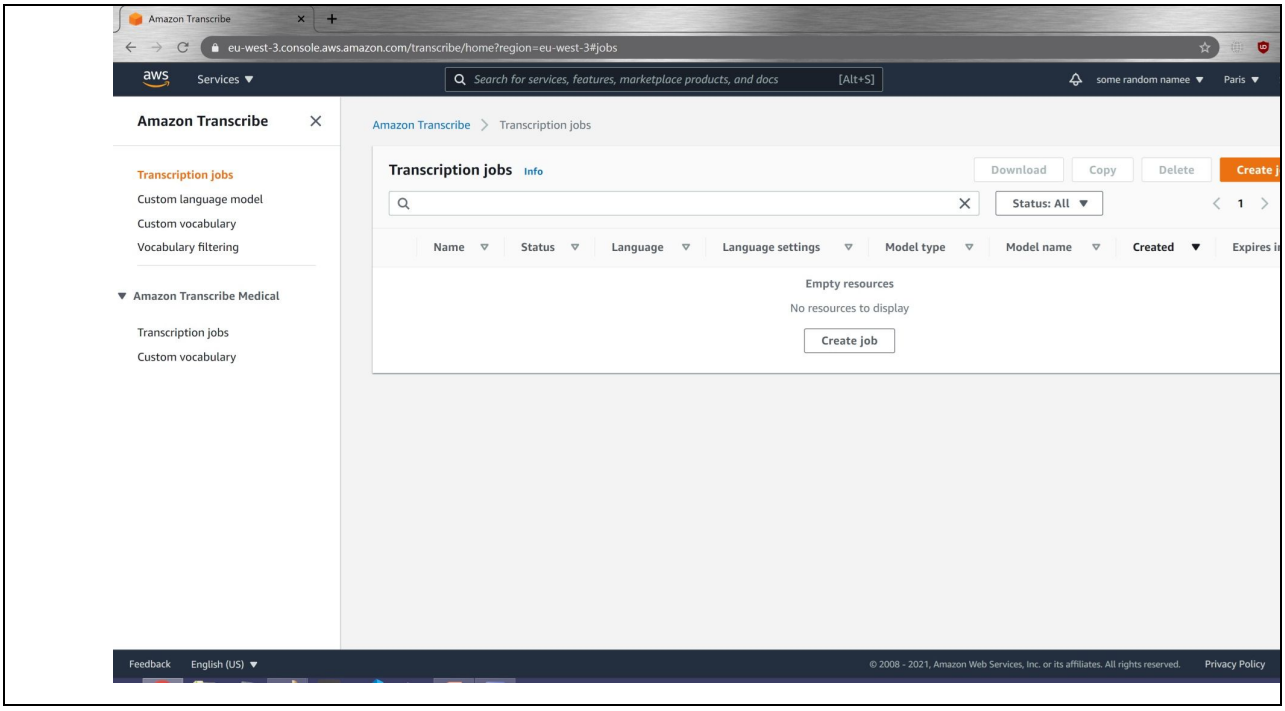


Рисунок В.4 - Примерный вид страницы сервиса Transcribe

Необходимые настройки для работы сервиса (Рисунки В.5а, В.5б).

- Необходимо выбрать имя задачи
- Выбрать Specific language, в списке выбрать русский язык
- Указать путь в S3 к исследуемому аудиофайлу
- Остальные настройки оставить как есть

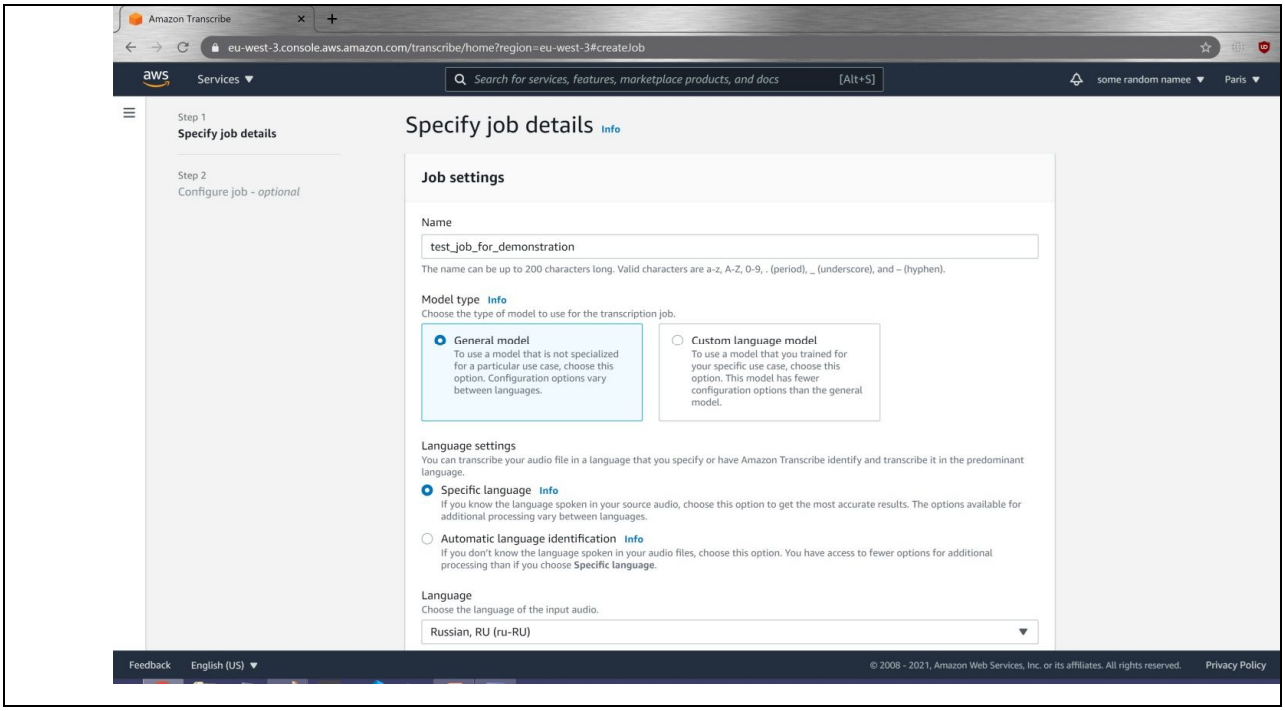


Рисунок В.5а - Необходимые настройки для работы сервиса

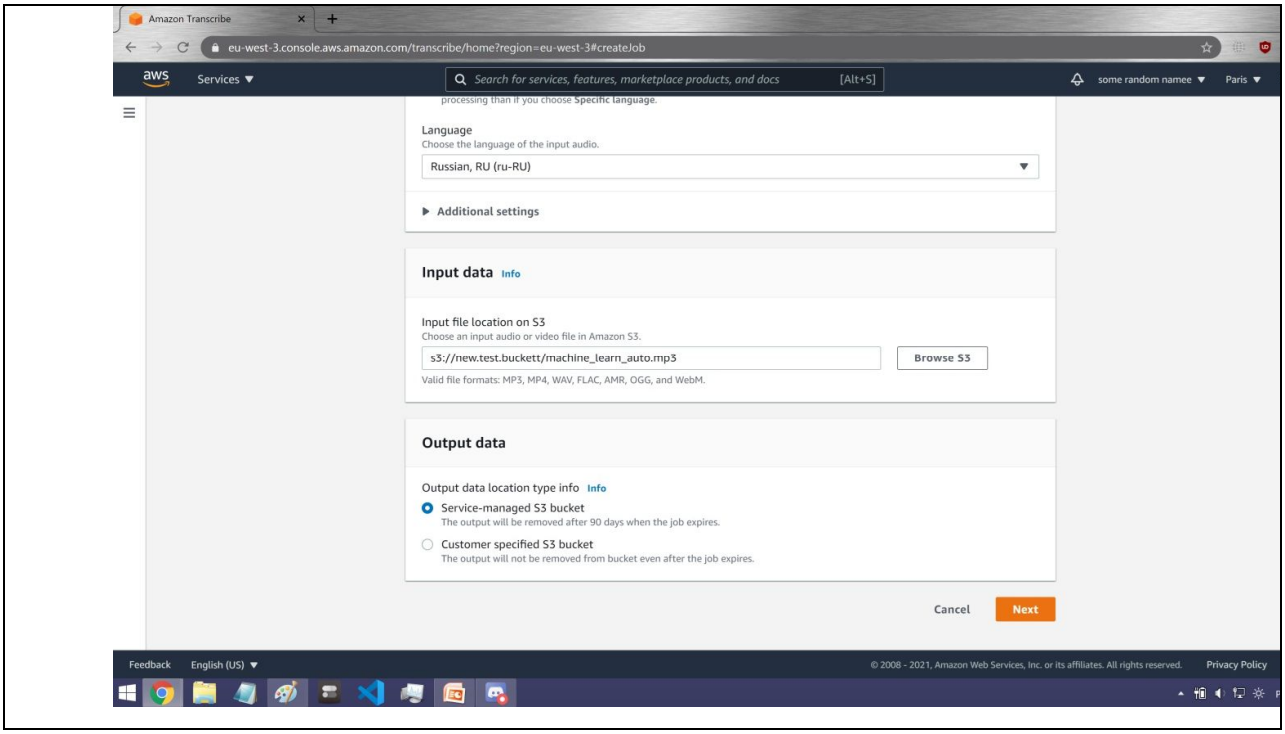


Рисунок В.5б - Для продолжения нажать кнопку Next внизу страницы

Дополнительные настройки

- Если хотим использовать словарь – включаем кнопку с Custom vocabulary и выбираем любой из ранее созданных словарей или создать новый
- Создание нового словаря происходит в отдельном окне
- Если не хотим использовать свой словарь –не трогаем настройки на этой странице

После выбора\невыбора словаря, для запуска распознавания нажимаем кнопку Create (Рисунок В.6).

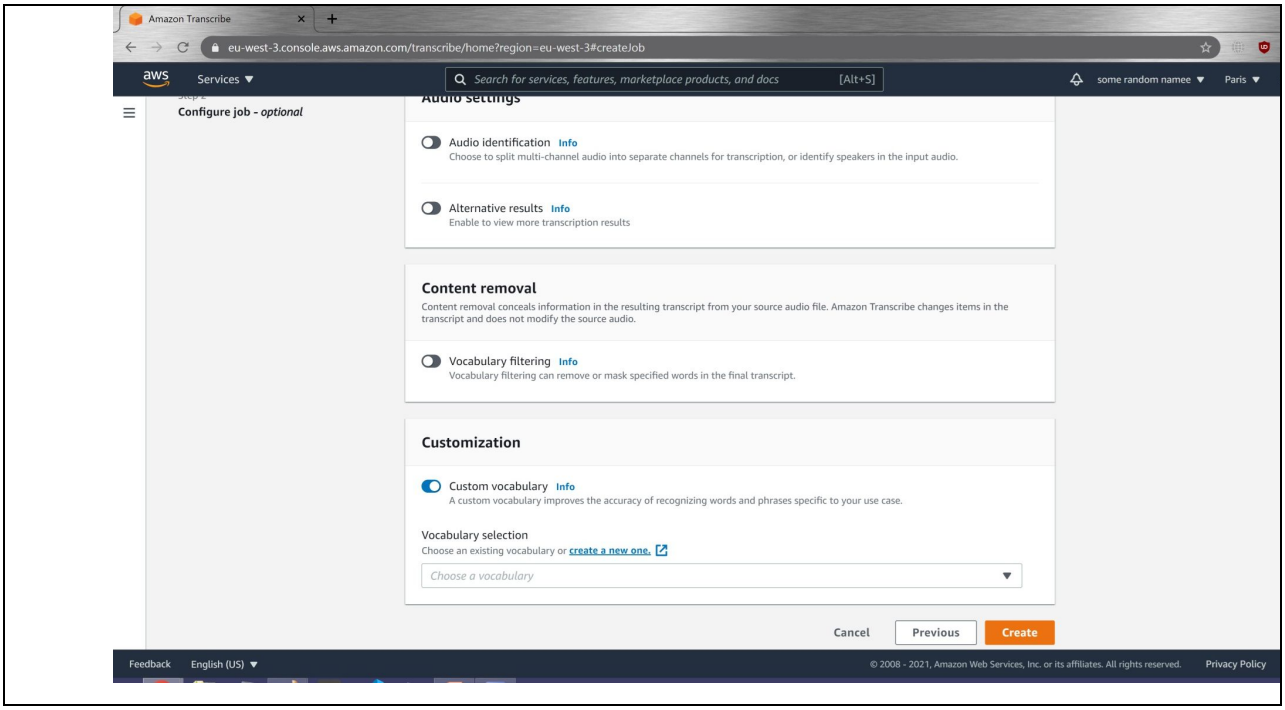


Рисунок В.6 - Окно дополнительных настроек

Добавление нового словаря (Рисунок В.7)

- Необходимо использовать уже созданный заранее словарь в формате .txt
- Необходимо ввести имя
- Необходимо выбрать русский язык
- Словарь должен быть в UTF-8, хоть это нигде и не написано – иначе работать оно не будет

- Если все сделано правильно – то словарь отобразится в списке как корректный, и его можно будет выбрать при распознавании

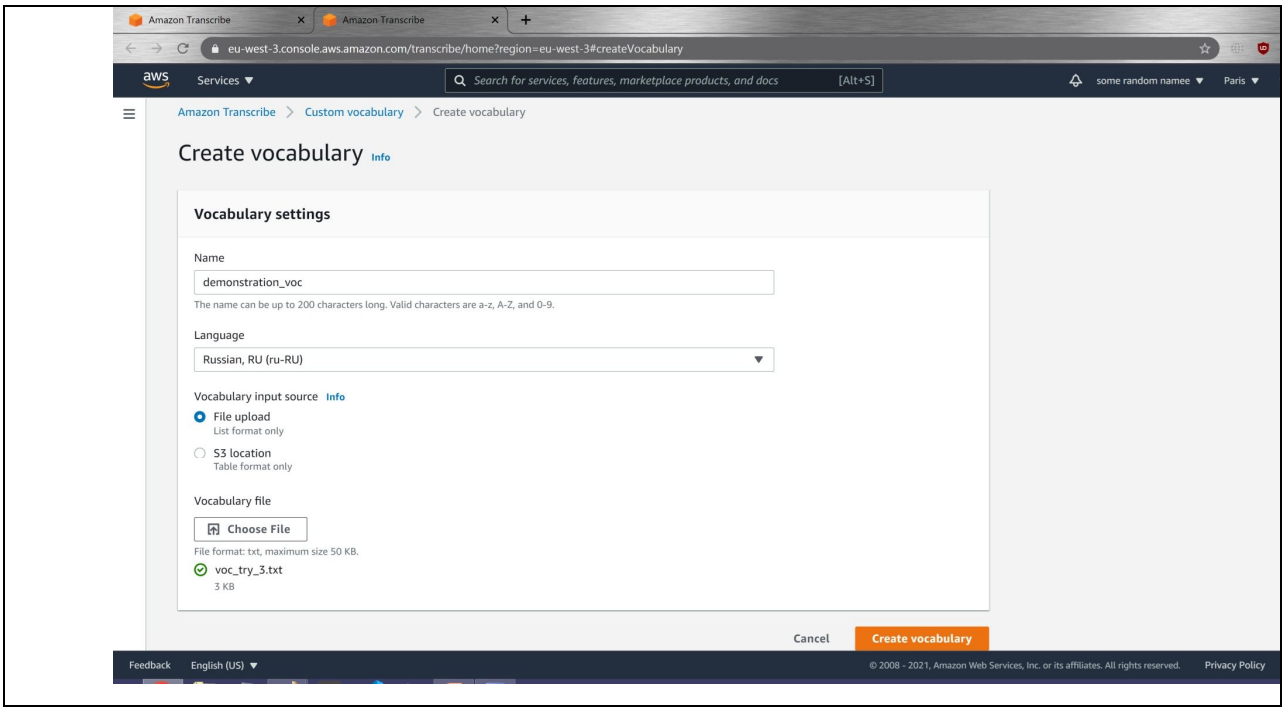


Рисунок В.7 - Окно создания словаря

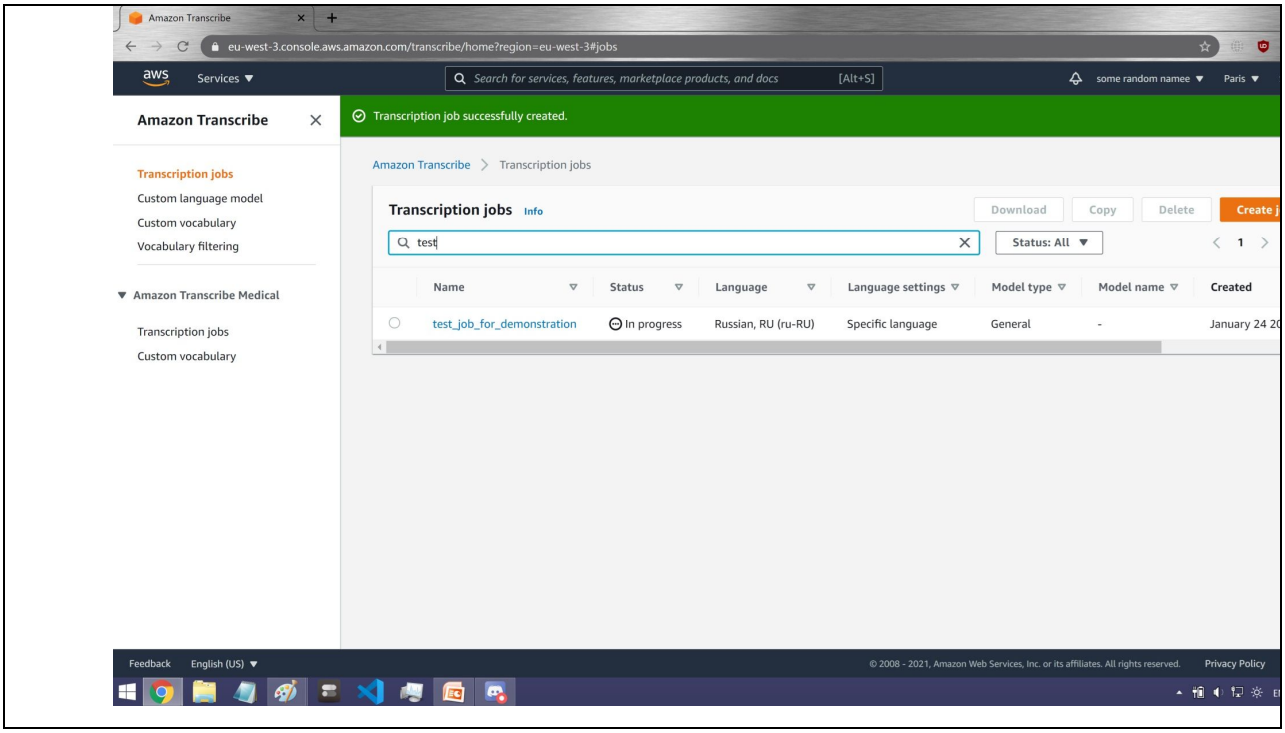


Рисунок В.8 - Задача на распознавание успешно принята

Результат распознавания (Рисунок В.9)

- Результат в формате .json можно скачать с помощью кнопки Download full transcript. В таком виде результат удобно обрабатывать на многих языках программирования
- Также можно быстро просмотреть результат в человеческом виде (например, чтобы понять, что был выбран не тот язык и распозналась какая-то чушь)
- Сервис предоставляет коэффициенты уверенности для распознанных слов
- Для удобной визуализации можно составить файл формата .html с распознанным текстом, помеченным разными цветами
- Время выполнения работы – несколько минут

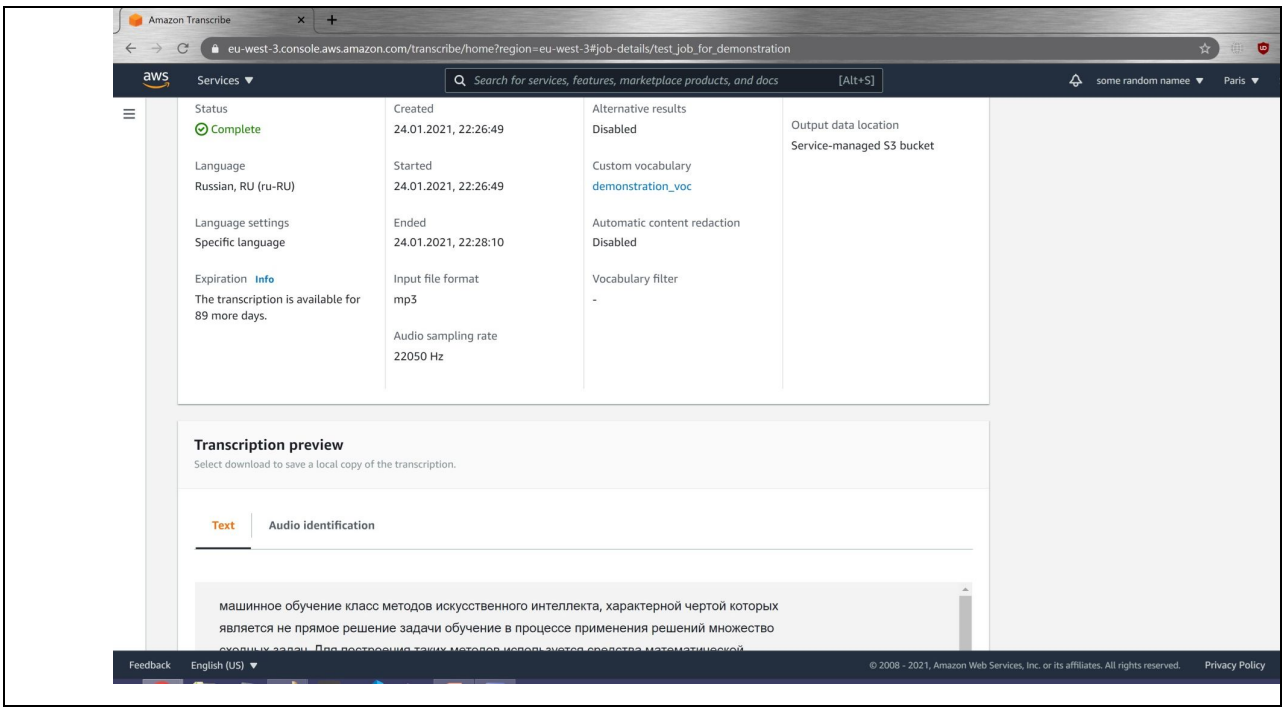


Рисунок В.9 – Доступ к результатам распознавания