**Данные об образовательной программе и дисциплине:**

Данные об обр. программе:

* Код и наименование направления подготовки
* Планируемые результаты освоения образовательной программы: (необязательное)
  + - Общекультурные компетенции
  + - Общепрофессиональные компетенции
  + - Профессиональные компетенции
  + - Профессионально-специализированные компетенции

Данные о дисциплинах обр. программы:

* Наименование
* Объем дисциплины (в з.е.)

**Данные об онлайн-курсах с платформы «Открытое образование» https://openedu.ru (***курсивом обозначены необязательные для заполнения поля***):**

Данные об онлайн-курсах:

* Название курса
* Ссылка для доступа к курсу
* Код и наименование направления подготовки
* Объем программы (в з.е.)
* *Компетенции, в том числе:*
  + *- Общекультурные компетенции*
  + *- Общепрофессиональные компетенции*
  + *- Профессиональные компетенции*
  + *- Профессионально-специализированные компетенции*

**Перечень необходимых работ**

1. Выполнить проектирование и разработку программного модуля (краулера) для обхода веб-страниц с целью извлечения структурированных данных из страниц онлайн курсов.
2. Сформировать дамп данных, полученных краулером в формате JSON
3. Сформировать дамп данных образовательных программ в формате JSON
4. Выполнить проектирование и разработку модуля анализа и сопоставления онлайн-курсов и образовательных программ. Модуль должен выполнять сравнение как по направлениям подготовки так и по формируемым компетенциям
5. Сформировать дамп данных по результатам сравнения
6. Выполнить проектирование и разработку модуля рекомендательного сервиса онлайн-курсов на основании данных образовательной программы.

**Описание входных данных**

Данные об образовательных программах и дисциплинах предоставляются в JSON формате:

[

{

"Directions": "01.03.02 Прикладная математика и информатика",

"Competence": [

"Компетенция 1",

"Компетенция 2"

],

"Desciplines": [

{

"История": "5"

},

{

"Философия": "3"

}

]

Данные об онлайн-курсах предоставляются в JSON формате:

[

{

"URL": "https://openedu.ru/course/ITMOUniversity/WEBDEV/",

"Title": "Веб-программирование",

"Directions": [

"09.00.00 Информатика и вычислительная техника"

],

"Credits": 4,

"Competence": [

"Компетенция 1"

"Компетенция 2"

]

},

]

**Поисковой паук (краулер) для платформы «Открытое образования»**

Необходимые данные извлечены автоматизированным способом с платформы онлайн-курсов за счет разработки поискового паука, который предоставляет необходимые данные в формате JSON, производя поиск по всем курсам платформы. Повторный запуск которого позволяет актуализировать дамп данных быстро и без затрат времени.

Разработка выполнена на языке Python c применением фреймворка Scrapy. Scrapy - это быстрый высокоуровневый механизм обхода веб-страниц и веб-сканирования, используемый для обхода веб-сайтов и извлечения структурированных данных из их страниц.

Установка осуществляется командой: pip install Scrapy.

Последующий запуск краулера, для создания дампа данных, выполняется запуском скрипта: scrapy runspider scraper.py формирующего JSON дамп.

**Модуль анализа и сопоставления курсов и образовательных стандартов**

**Поиск и сравнение вхождения направлений подготовки из образовательных программ в онлайн курсы.**

Для выявления соответствий был произведен анализ направлений подготовки, указанных в образовательных программах и в описаниях онлайн курсов. В большинстве курсов указаны коды укрупненных направлений подготовки (01.00.00 Математика и механика), тогда как в образовательных программах они детализированы (01.03.02 Прикладная математика и информатика). Это было учтено при выявлении соответствий.

**Поиск и сравнение вхождение компетенций из образовательных программ в онлайн-курсы**

Формулировки компетенций могут разнится, как в рамках различных образовательных программ, так и в рамках различных курсов. Более того, в более старых курсах встречаются формулировки компетенций, соответствующие ФГОС предыдущих поколений. В ходе анализа данных по составу компетенций образовательных программ и онлайн-курсов было выявлено не точное совпадение в написании, а также по количеству вхождений различных компетенций. Для анализа по включению онлайн-курсов на основе этих данных был внедрен компонент анализа сравнения строк. Для более точного исследования были использованы 2 метода сравнения компетенций.

**Библиотека FuzzyWuzzy** с применением расстояния Левенштейна между двумя строками в теории информации и компьютерной лингвистике. Метод основан на минимальном количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

**Метод шинглов**. На первом этапе происходит канонизация исследуемых текстов. Для этого из текста убираются предлоги, союзы, знаки препинания и т.п. в соответствии со стоп листом. Далее из сравниваемых текстов выделяются подпоследовательности слов, идущих друг за другом в соответствии с длиной шингла. Выборка происходит внахлест. Таким образом, разбивая текст на подпоследовательности, был получен набор шинглов в количестве равному количеству слов минус длина шингла плюс один. Далее происходит сравнение контрольных сумм шинглов выборок (подпоследовательностей) двух текстов между собой.

В результате работы модуля формируется выходной дамп в формате JSON, который используется для дальнейшего анализа:

[

{

"Directions": "01.03.02 Прикладная математика и информатика",

"Course\_direction\_match": [

"Курс 1",

"Курс 2",

],

"Course\_competence\_match\_shingle": [

"Курс 1",

"Курс 2"

],

"Course\_competence\_match\_fuzzy": [

"Курс 1",

"Курс 2"

]

}

]

**Описание алгоритма рекомендаций (семантика)**

Рекомендованные курсы подбираются по соответствию направлений подготовки, компетенций, оценки объема дисциплин и количества зачетных единиц. Этот список составлен в порядке важности соответствующего критерия при ранжировании курсов, соответствующих образовательным программам.

Для каждой образовательной программы существует массив рекомендованных курсов, однако с целью реализации интеллектуального алгоритма рекомендаций, курсы необходимо ранжировать в соответствии с заданными критериями.

Каждому критерию назначается вес, который влияет на итоговое число, ставящееся в соответствие курсу и влияющее на его место в списке рекомендаций. Для возможности динамического изменения весов и адаптации рекомендательного сервиса в дальнейшей работе была разработана и обучена нейронная сеть. Обучение производилось методом обратного распространения ошибки.