Задание 1

Собрать ЕЯ текстовые описания ПО.

**ЧАСТЬ 1. ТЕМАТИКА ВКР**

Для сбора ЕЯ описания ПО необходимо определиться с темой работы. Если у вас определена тема проекта ВКР, стоит использовать ее как тему курсового проекта. Если тема ВКР не определена, выберите тему, которая наиболее вероятно будет связана с ней в следующем семестре. При невозможности самостоятельного выбора темы работы, обратитесь за темой к преподавателю.

После того, как вы определились с темой работы, выберите одну или несколько ее частей, на которых будет акцентироваться курсовой проект. К примеру, разберем тему работы «веб-сервис для построения туристических маршрутов на базе генетического алгоритма». В рамках данной темы можно выделить подтемы «веб-сервис», «туристические маршруты», «генетический алгоритм». Выберите релевантные для вашей работы подтемы.

После того, как подтемы выбраны, необходимо найти научные работы, связанные с ними (в количестве 10 статей для каждой подтемы). Для этого можно использовать ресурс «ссылки на ресурсы для поиска научных работ».

A screenshot of a computer

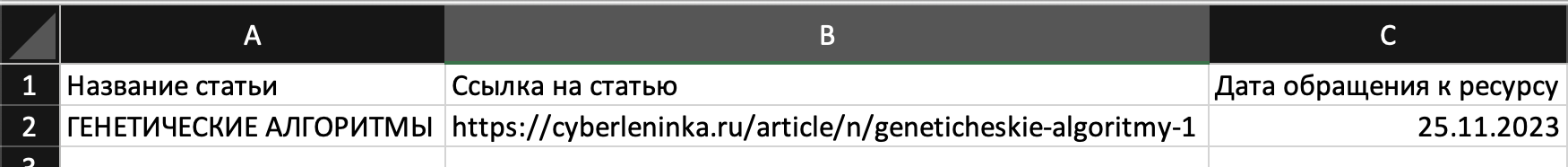
Description automatically generated

К примеру, <https://cyberleninka.ru/>. Выдача по запросу «генетический алгоритм».

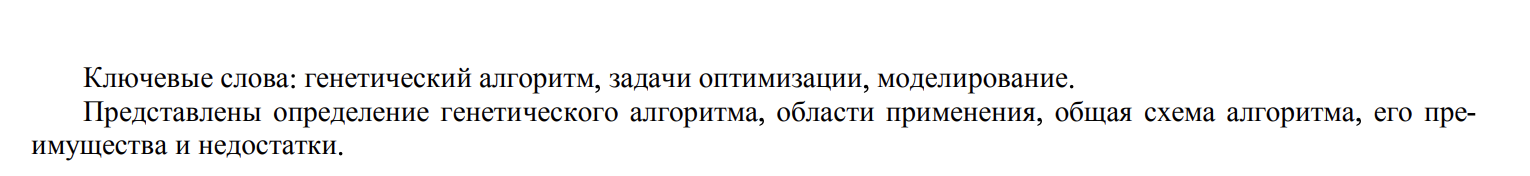
A screenshot of a computer

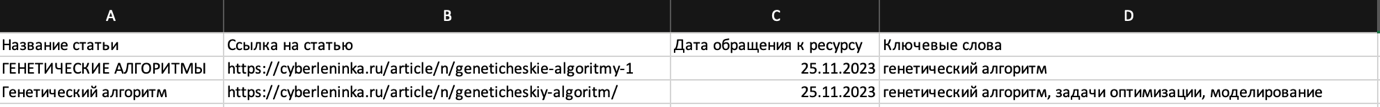
Description automatically generated

С помощью поиска, найдите достаточное число работ и сохраните их в формате «Название статьи» - «ссылка на статью» - «дата обращения к ресурсу».

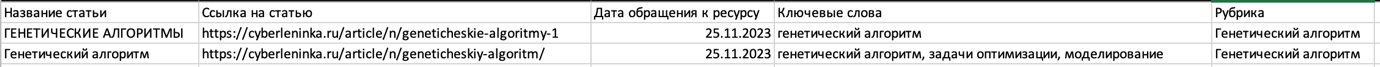


Теперь необходимо добавить к описанию ключевые слова. Для этого возьмите слова, выделенные самим автором в соответствующую категорию работы.





Теперь необходимо объединить общие темы ключевых слов для создания рубрики текстов. В данном небольшом примере этот выбор очевиден.



**ЧАСТЬ 2. МЕДИЦИНСКАЯ ТЕМАТИКА**

Для сбора данных медицинской тематике воспользуйтесь файлом распределения вариантов работы - <https://docs.google.com/spreadsheets/d/1ICcsB9c0m1vRZ-Zei5s0fLoYqdqCbTvQdW8aMykSUkA/edit?usp=sharing>. На вкладке «Распределение» указано, какой вариант относится к вашему ФИО. На вкладке «Темы» указано, какие подкатегории и статьи входят в ваш вариант. 

Для получения статей необходимо сделать следующее:

1. Получить две базовые статьи исследуемой области.

Для этого нужно найти статьи, используя указанный DOI. Самый простой способ сделать это - @SciArticleBot телеграм-бот. Помимо этого, можно просто найти статью по идентификатору в браузере.

1. Получить по 10 статей, на которые ссылается базовая статья, указанная в таблице.
   1. Если в статье достаточно ссылок, необходимо найти статьи из них по названиям и скачать.
   2. Если в статье недостаточно ссылок, необходимо перейти на второй и далее ссылочный уровень. Для этого необходимо перейти по ссылке в источники внутри этого источника.

Затем, когда работы собраны, необходимо разделить их на две категории и повторить действия, описанные в **ЧАСТЬ 1. ТЕМАТИКА ВКР.**

После завершения сбора работ необходимо скачать файлы статей для создания датасета.

Задание 2

Составить датасет для дообучения векторной модели.

Для создания датасета необходимо привести файлы статей к текстовому формату. Чтобы это сделать воспользуйтесь файлом шаблона очистки cleaner.py.

A screenshot of a computer

Description automatically generated

(приблизительная структура проекта)

В нем укажите пути к папкам с PDF-файлами и папке, куда будут выгружены результаты очистки. Для улучшения векторного представления из статей необходимо убрать метаданные. Так как имя автора и название работы занимают относительно малый объем информации, наиболее важно удалить списки источников. Предложенный код сделает это.

Архив из полученных файлов прикладывается к отчету. После того как файлы были получены, объедините их в два отдельных датасета (один для медицинской тематики и один для темы ВКР). Полученные файлы приложите к отчету. Далее они будут использоваться для обучения модели.

Задание 3.

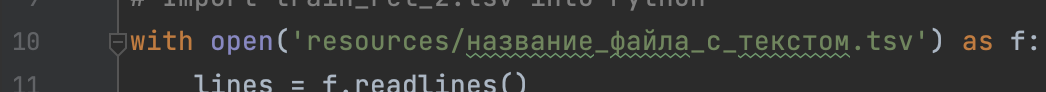
Обучение модели word2vec.

Для обучения модели необходимо скачать шаблон из соответствующего раздела курса.

A close up of a text

Description automatically generated

Далее в любой удобной среде разработке добавьте файл шаблона в проект. После этого необходимо импортировать в проект файл датасета из предыдущего шага. Сохраните его в папку “resources” в проекте. Запишите путь к файлу в параметр.



Запишите несколько слов из текстов работ в параметры методов для проверки качества работы модели. Эти слова должны содержаться в текстах датасета. Они используются для проверки того, что слова вошли в представление.

A screen shot of a computer program

Description automatically generated

И еще три слова для сравнения близости их векторных представлений (здесь из слов 2 и 3 выбирается наиболее близкое по вектору слову 1).



Например, "болезнь", ["заболевание", "инфекция"].

Укажите название файла для сохранения модели и запустите программу.



Параметры обучения модели задаются в классе World2Vec. Цель обучения состоит в том, чтобы большее число нужных для работы системы слов попали в представление. Для этого нужны проверки выше. Если слова из текстов не попадают в представление, измените значение параметра min\_count. Понизьте значение для уменьшения порога вхождения. Если в следующем пункте слова, ближайшие к векторам тем будут слишком похожи, повысьте параметр negative.

A screen shot of a computer

Description automatically generated

Задание 4.

Автоматическое рубрицирование.

Для проведения автоматического рубрицирования необходимо написать программу, реализующую алгоритм из задания.

Для того, чтобы загрузить готовую модель, используйте конструкцию «w2v\_model = Word2Vec.load("путь\_к\_файлу.model")».

Для проверки наличия слова в модели – «w2v\_model.wv.has\_index\_for(word)».

Для получения нормализованного вектора по слову – «w2v\_model.wv.get\_vector(word).sum()».

Перейдем к алгоритму из задания:

– текст разбивается на массив слов;

Используйте line.split()

– для каждого слова в массиве необходимо получить соответствующий ему нормализованный вектор из модели;

– для слов, вектор которых неизвестен, значение вектора считать равным 0;

Для этих пунктов необходимо проверить наличие слова в модели, затем получить его вектор, просуммировать и увеличить счетчик слов. Альтернативно, можно вычитать из значения длины массива, полученного в первом пункте, единицу, каждый раз, когда слово отсутствует в представлении.

– найти среднее арифметическое полученных векторов;

A computer screen shot of a program code

Description automatically generated

Таким образом каждому тексту выборки будет соответствовать числовое значение вектора. Затем необходимо произвести рубрицирование:

– отсортировать значения векторов;

– используя отсортированные значения необходимо разделить тексты на 3-5 категорий, основываясь на близости значений векторов (числовые характеристики, соответствующие темам текстов должны иметь минимальное отклонение между собой относительно всего датасета);

– определить среднее арифметическое числового значения для каждой категории (это число будет числовым значением категории);

– получить набор ближайших по вектору слов из созданной модели (эти слова будут условно-ключевыми);

Для получения блиайших слов используйте «model. wv.similar\_by\_vector(vector=np.array(your\_word\_vector), topn=5))», где второй параметр обозначает число получаемых слов.

Если слова, найденные таким образом слишком похожи, используйте увеличение значения параметра negative в предыдущем пункте. Если улучшить выдачу слов не получается, но векторы достаточно различаются, необходимо самостоятельно выделить ключевые слова и названия категорий, основываясь на текстах, входящих в категории.

Задание 5.

Сбор справочных данных.

Для сбора справочных данных необходимо произвести следующие действия для каждой из полученных в Задании 4 категории:

– выбрать описание нескольких самых важных тем, фигурирующих в каждой статье, входящей в категорию;

– собрать описания вместе и сопоставить их числовому значению категории;

На основе полученных описаний составить таблицу, содержащую поля: *числовое значение категории, справочная информация.*

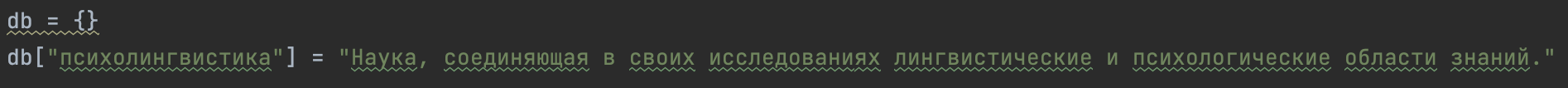
То есть, для каждой из категорий необходимо выбрать 3-5 терминов / понятий / важных для понимания групп сведений, и дать им определения. Затем составить таблицу, содержащую эти определения.

Задание 6.

Создание автоматической справочной системы.

Для создания автоматической справочной системы необходимо преобразовать таблицу, полученную в Задании 5 в базу данных, представленную в удобном для исполнителя формате (SQLite, Postgres, двумерный массив, hashMap и др.).

Например, карта реализуется подобным образом. Здесь необходимо хранить в виде ключа слово, либо вектор, а значением будет определение.



Учитывая количество записей, можно обойтись использованием такой структуры данных, дополнение проекта с целью применения баз данных производится по усмотрению исполнителя.

Необходимо реализовать программу со следующим алгоритмом работы:

1. Пользователь вводит запрос;
2. Система разбивает запрос на массив слов;
3. Для каждого слова получается нормализованный вектор из обученной модели;
4. Рассчитывается среднеарифметическое значение вектора запроса;
5. Из базы данных выбирается поле с наиболее близким числовым значением категории; В случае использования HashMap, описание получается по ключу.

Система выводит пользователю справочную информацию, соответствующую значению категории из пункта 5.

Если значения векторов достаточно близки, и система ошибается при выборе ответа, вместо обычного запроса следует подавать на вход фрагменты текстов схожих с темами категорий статей. Таким образом, система будет работать не с простыми запросами, а со значительными объемами текстовой информации.