Руководство по установке и эксплуатации программного модуля “D'Adviser”

Версия	1.3
Согласовано	Лукьяничева Е.О.
Автор	Лукьяничева Е.О.

Содержание

1. Введение

1.1. Назначение документа

"Руководство по Установке и эксплуатации" содержит описание функциональных характеристик программного обеспечения и информацию, необходимую для его установки и эксплуатации.

Материал руководства направлен на формирование у пользователя основных навыков работы с программным модулем "D'Adviser". Документ описывает ключевые моменты работы с программным модулем:

- Поиск похожих документов на множестве документов;
- Установка;
- Первоначальная настройка;
- Обслуживание программного модуля;
- Возможность интеграции с системами электронного документооборота.

1.2. Назначение программного модуля

Программный модуль предназначен для упрощения поиска похожих по содержанию документов на определенном множестве документов.

1.3. Уровень подготовки пользователей

Пользователи программного модуля "D'Adviser" должны обладать следующими навыками в зависимости от режима использования программного модуля:

Класс пользователей	Режим использования	Требования к уровню подготовки пользователей
Пользователи, использующие модуль автономно	Использование модуля автономно: • установка, • использование, • настройка, • обучение модели	• базовое понятие о нейронных сетях (о принципах их работы и обучения) • навыки программирования на языке Python.
Интеграторы/ Администраторы	Интеграция модуля в крупные системы: • установка • интеграция модуля в состав комплексных систем, • настройка модуля,	• базовое понятие о нейронных сетях (о принципах их работы и обучения)

	• обучение модели	• навыки программирования на языке Python.
Обычные пользователи систем	Использование модуля в составе комплексных систем.	Навык работы с системой, в состав которой интегрирован модуль

Таблица 1. Требования к уровню подготовки пользователей

1.4. Термины и сокращения

Термины и сокращения, используемые в документе, представлены в таблице ниже:

Термин	Значение
Нейронные сети	математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.
Естественный язык	язык, используемый для общения людей и не созданный целенаправленно(в отличие от искусственных языков).
Обработка текстов на естественном языке (*Natural Language Processing, NLP*)	общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза текстов на естественных языках. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез — генерацию грамотного текста.
Лемма	словарная форма. Например, для русского языка: • для существительных — именительный падеж, единственное число; • для прилагательных — именительный падеж, единственное число, мужской род; • для глаголов, причастий, деепричастий — глагол в инфинитиве несовершенного вида.
Мешок слов	упрощенное представление текста, которое используется в обработке естественных языков и информационном поиске. В этой модели текст (одно предложение или весь документ) представляется в виде мешка (мультимножества) его слов без какого-либо учета грамматики и порядка слов, но с сохранением информации об их количестве.
Токен	единица информации, слово

Корпус	отобранная и обработанная по определенным правилам совокупность документов, используемых в качестве базы для исследования языка.
TF-IDF	(от англ. *TF — term frequency, IDF — inverse document frequency*) — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции. Мера TF-IDF часто используется в задачах анализа текстов и информационного поиска, например, как один из критериев релевантности документа поисковому запросу, при расчёте меры близости документов при кластеризации.

Таблица 2. Список терминов, используемых в документе

2. Назначение и условия применения

2.1. Поиск похожих документов

Для старта процесса поиска похожих документов программному модулю на вход подается текстовый файл с расширением “.txt” , представляющий собой предварительно отсканированный и распознанный документ на русском языке.
Далее модуль анализирует входной документ следующим образом:

1. Документ разбивается на слова (“токены”)
2. Из полученного набора удаляются стоп-слова (предлоги, частицы, союзы, неинформативные слова из списка)
3. Проводится лемматизация
4. Определяются веса слов и на их основе определяется общая тематика и содержание документа

После завершения процесса анализа запускается процесс поиска похожих документов, информация о которых хранится в модели данных. При этом создается корпус и матрица коэффициентов tf-idf, используемые непосредственно в процедуре поиска для учета тематики и содержания документа.

Схема работы модуля представлена на рисунке ниже:

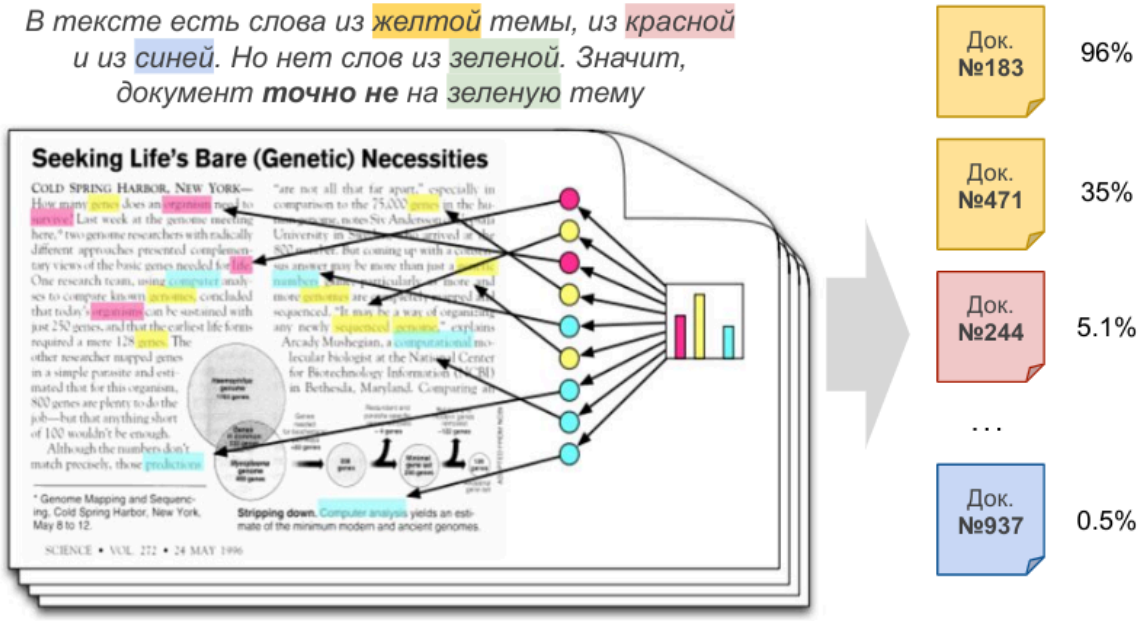

Рисунок 1. Пример работы программного модуля

2.2. Обучение модели данных

Перед использованием модуля "D'Adviser" необходимо обучить модель данных. Ниже представлен пример скрипта запуска обучения:

```
from dadviser.core import DAdviser, read_file

documents_path = "/путь/к/директории/с/документами"
adviser = DAdviser(documents_path)
```

Листинг 1. Пример для обучения модели данных для дальнейшего использования в процедуре поиска

Пример вывода во время обучения модели:

```
[nltk_data] Downloading package stopwords to /home/alex/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /home/alex/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
Recoding filenames into the index.txt file
Parallel tokenizing and lemmatazing processes...
Finished 100/500 files
Finished 200/500 files
Finished 300/500 files
Finished 400/500 files
Finished 500/500 files
Parallelization elapsed 35.67 seconds
Form the dictionary based on tokens (filenames are ordered)...
Number of words in dictionary: 24592
Forming a corpus (a list of bags of words)...
Forming a tf_idf....
Size of tf_idf: 500
Compute similarities across a collection of documents...
Save the results...
Successfully finished
```

Рисунок 2. Вывод на экран при успешном завершении обучения на 500 файлах

2.3. Интеграция с другими системами

Программный модуль "D'Adviser" может быть интегрирован в другие системы (например, в системы электронного документооборота). Ниже приведен пример запуска скрипта проверки схожести документов из программного модуля другой системы:

```
from dadviser.core import DAdviser, read_file

# создание объекта DAdviser для загрузки сохраненного обучения
documents_path = "/путь/к/директории/с/документами"
adviser = DAdviser(documents_path)

# считывание документа и его проверка
compare_with_text = read_file("/путь/к/файлу/для/проверки")
result = adviser.get_similarity(compare_with_text, toplist=10)

# как пример, вывод результата на экран
```

```
print(result)
```

Листинг 2. Пример вызова программного модуля для анализа документа

Результат анализа возвращается в виде списка словарей. Количество элементов топ-листа зависит от аргумента `top_list`. Элементы словаря:

- id (число) - номер документа в топ-листе
- name (строка) - путь к файлу
- percent (число) - процент схожести
- top_terms (кортеж) - топ список слов, которые повлияли на тематику:
    - 1-ый элемент: слово
    - 2-ой элемент: коэффициент для слова в матрице tf-idf

В данном примере корпуса подгружены из прошлого обучения, на вход программному модулю подается текст, на выходе же получен список документов, а так проценты схожести. Следует отметить, что в примере был выбран исследуемый документ из обучающей выборки, поэтому было найдено 100% совпадение с тем же самым документом.

```
[nltk_data] Downloading package stopwords to /home/alex/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /home/alex/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
Loading a corpus, dictionary... Please wait
[{'id': 1, 'name': '/home/alex/GitHub/dadviser/data/original/19536054', 'percent': '100.00', 'top_terms': [('постановление', 0.38
```

Рисунок 3. Вывод на экран результатов поиска похожих документов

Ниже представлен пример использования D'Adviser на сайте, созданном с помощью фреймворка Django.

Форма загрузки документа:

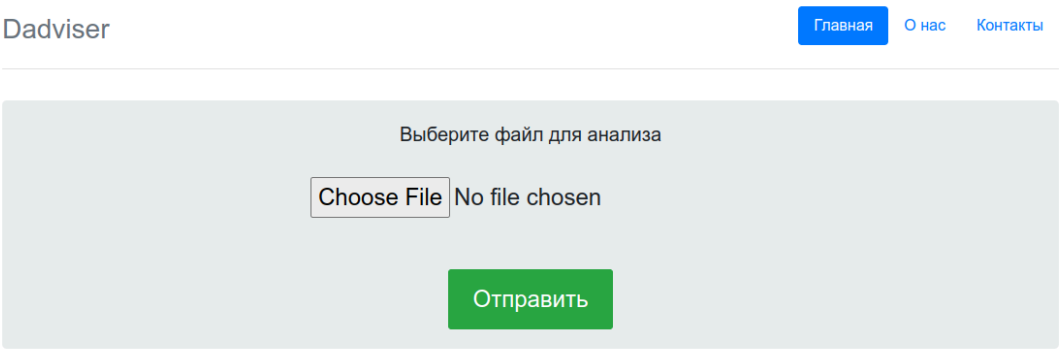

Рисунок 4. Пример пользовательской формы для загрузки документа для последующего анализа модулем "D'Adviser"

Результат поиска:

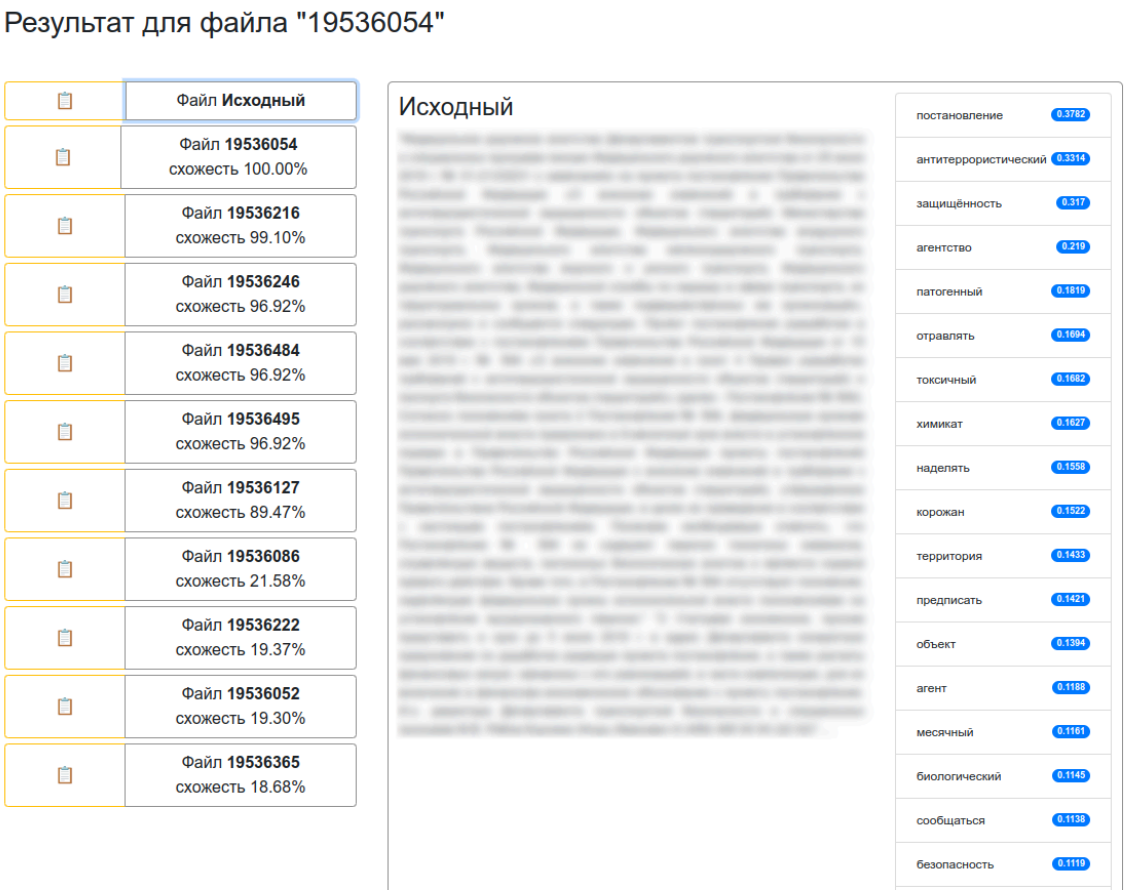

Рисунок 5. Пример вывода результатов поиска с помощью графического интерфейса

На странице результатов отображены: контент найденных документов, проценты схожести и топ-слова, влияющие на определение тематики.

Высокий процент первых топ-документов (>89%) обусловлен тем, что они являются копией исследуемого документа с незначительными изменениями в тексте для демонстрации качества поиска схожести.

3. Установка и эксплуатация

3.1. Программные и аппаратные требования к ПО

Рекомендуется устанавливать программный модуль на выделенный компьютер (сервер), отвечающий следующим техническими характеристикам:

- Процессор Core i5 и выше (от 2.4 ГГц)
- Количество ядер CPU: 6 и выше

- Размер оперативной памяти (RAM): 16 ГБ и выше (зависит от количества текста)
- Жесткий диск
    - Тип памяти SSD/HDD
    - Размер свободного места не менее 100 ГБ (зависит от количества текста)

- Поддерживаемые ОС:
    - Microsoft Windows (64-bit)
    - Fedora
    - Debian Linux
    - CentOS
    - Ubuntu

3.2. Установка

Программный модуль D'Adviser поставляется в форме лицензированной копии программного обеспечения на любом электронном носителе и пакетом электронной документации, в которое входит:

- "Руководство по установке и эксплуатации модуля D'Adviser"

Установка ПО происходит путем копирования библиотеки в целевой каталог и запуска команды для установки пакета среду Python. Требуется подключение к интернету для загрузки зависимостей.

Перед установкой библиотеки проверьте, что у вас установлена актуальная версия setuptools:

```
python3 -m pip install --upgrade setuptools
```

Листинг 3. Пример команды для проверки setuptools

С помощью следующей команды выполните установку библиотеки D'Adviser:

```
sudo python3 setup.py install
```

Листинг 4. Пример команды для установки программного модуля D'Adviser

3.3. Настройка программного обеспечения перед использованием

Перед запуском программного модуля, необходимо обучить модель данных (см. п. 2.2 "Обучение модели данных"). Для этого необходимо подготовить набор документов и запустить процедуру обучения следующим образом:

1. Создать директорию, в которой будут находиться целевые документы для анализа
2. Перенести туда эти документы в формате .txt
3. Запустить скрипт, в котором будет вызвана библиотека D'Adviser и использован её объект с методами для анализа текста. Укажите аргументом в конструкторе объекта путь к директории с текстами
4. Считать целевой документ для получения схожести в виде текста с помощью метода "`read_file`"
5. Вызвать метод "`get_similarity`", где первым аргументом является текст из п.4, который необходимо проанализировать на схожесть, а второй аргумент - количество документов в топ-листе.

3.4. Обслуживание программного обеспечения

В процессе эксплуатации возникает необходимость изменить набор документов, служащих областью поиска. Например, необходимо добавить новые документы или исключить неактуальные. Для этого необходимо удалить/добавить документы в директории, очистить директорию `data` (автоматически создаваемую скриптом для хранения информации о словаре, коэффициентах в месте запуска скрипта) и заново запустить процесс обучения модели данных.

Периодичность обучения модели данных зависит от периодичности и характера изменения области поиска.

Файлы, отвечающие за работоспособность поиска схожести (их самостоятельное редактирование может привести к некорректному поиску схожести):

1) index - список анализируемых файлов в пользовательской директории документов в формате .txt
2) corpus.mm и corpus.mm.index - обработанная информация (матрицы и мешки слов) о файлах по определенным правилам, используемых в качестве базы для анализа
3) dictionary.dict - словарь токенов и их уникальные номера (id)
4) tf_idf.tfidf_model - матрица коэффициентов tf-idf

3.5. Безопасность данных

Пользователи и лица, ответственные за безопасность хранения и утилизации данных, должны самостоятельно позаботиться о сохранности документов, использующихся для обучающей выборки и поиска. Модуль хранит данные локально и передает данные только в те системы, в которые он был специально интегрирован. (см. п. 2.3 "Интеграция с другими системами")