УДК 05.13.01

А. С. Нужный[1], *Д. И. Сорокин*[1,2]

[1]Институт проблем безопасного развития атомной энергетики
[2]Московский физико-технический институт (национальный исследовательский университет)

Создание программы интеллектуального анализа текстовой документации по вопросам захоронения РАО

Представлена программа контекстного и тематического анализа текстовой документации. Программа обрабатывает документы PDF-формата, сохраняет обратный индекс корпуса данных, а также иную служебную информацию, предоставляющую пользователю возможность поиска фрагментов текста, отвечающих введенному запросу или выбранной теме. При тематическом поиске программа ищет тексты, похожие на обучающие примеры, ранее отнесенные пользователем к выбранной теме. Тематический анализ текстового корпуса позволяет обнаруживать наличие или отсутствие в нем тех или иных типовых тем, оценивать полноту содержащейся информации.

Ключевые слова: обработка естественного языка, семантический анализ, контекстный поиск, машинное обучение.

A. S. Nuzhny[1], *D. I. Sorokin*[1,2]

[1]Nuclear safety institute of the RAS
[2]Moscow Institute of Physics and Technology (State University)

Development of a text-mining program for analysis of documentation on the disposal of radioactive waste problem

The program of contextual and thematic analysis of documents is presented. The program processes documents in PDF format, builds a reverted index of corpora, and other service information, which allows the user to search for fragments of text that meets the entered query or selected topic. In case of the topic search, the program searches for texts similar to the training examples. The thematic analysis of the text corpora allows user to detect the presence or absence of those or other typical topics, assesses the completeness of the information provided.

Key words: natural language processing, semantic analysis, contextual search, machine learning.

1. Введение

В ИБРАЭ РАН накоплен большой объем специализированной документации, относящейся к объектам захоронений радиоактивных отходов, которая состоит из научных статей, книг, нормативных актов, отчетов и разного рода технической документации. На данный момент электронное хранилище насчитывает более 6 тыс. различных документов. Постоянный рост объема текстовых данных ведет к необходимости автоматизации задачи их структурирования, анализа и поиска полезной информации.

Для классического поиска «по словам» в хранилищах текстовой документации обычно рассчитывают так называемый обратный индекс [1], который содержит информацию о том,

© Нужный А. С., Сорокин Д. И., 2020
© Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)», 2020

в каком документе сколько раз встречается то или иное слово. При получении текстового запроса от пользователя поисковая программа по заранее рассчитанному обратному индексу выявляет тексты, в которых присутствуют данные слова, и ранжирует их в соответствии с формулой, учитывающей частоту встречаемости слов запроса. Среди способов ранжирования документов наибольшую популярность получила формула, известная в литературе как BM25 [2].

Результатом использования сторонних приложений (https://www.zotero.org/, http://www.likasoft.com/ru/document-search/) являются ссылки на документы, что часто не соответствует ожиданиям пользователя, которому особенно в случае объемных документов необходимо находить страницу, главу, параграф и т.п. Ранжирование в этих программах также применяется к документам, что может приводить к некорректному результату. В частности, книги и обзорные статьи, в которых может содержаться разнородная информация, будут попадать в результат практически любого поиска. С другой стороны, книга, содержащая актуальную для пользователя главу, может быть отнесена поисковиком в конец списка, поскольку отношение содержащихся в ней слов запроса к общему числу слов в документе может быть невелико. Создаваемая программа индексирует отдельные фрагменты текста, содержащиеся в документах. Результатом выдачи программы по поисковому запросу является фрагмент текста с указанием документа, из которого он взят, и номера страницы.

Важной частью обоснования безопасности пункта захоронения радиоактивных отходов является доказательство того, что все факторы, влияющие на безопасность, должным образом учтены. Набор факторов формируется для каждого объекта на основе полного международного списка «Особенностей, событий, процессов» (ОСП) [3]. ОСП можно рассматривать как своего рода тематический базис, по которому может быть «разложена» любая текстовая информация, относящаяся к данной тематике. Перед создаваемой программой, в частности, ставилась задача автоматического выявления ОСП в документах. Анализ наличия в корпусе текстов определённой тематики, относящихся к выбранному объекту, может позволить выявить недообследованные ОСП и определить дальнейшее направление работ по данному объекту.

В работе описывается программа, позволяющая решать задачи контекстного и тематического поиска, а также рубрикации информации в документах, относящихся к ядерной энергетике. Программа позволяет оценивать полноту собранной информации по объектам захоронения радиоактивных отходов и выявлять недостаточно раскрытые темы. В работе также представлен использованный при разработке алгоритмов математический аппарат и результаты апробации программы.

2. Предобработка PDF-файлов

Анализируемые документы представлены в формате PDF. Файлы, созданные в разные годы и полученные разными путями (набранные в TeX или конвертированные из различных версий программы Microsoft Word), имеют в PDF различное описание. Это существенно усложняет задачу построения алгоритмов их обработки – разбиения текста на фрагменты (главы, абзацы и т.п.) и удаления из рассмотрения служебной информации (колонтитулов, оглавлений и т.п.). В языке Python существует несколько широко используемых пакетов для работы с файлами PDF-формата. Нами был выбран пакет Poopler (https://poppler.freedesktop.org) как наиболее подходящий для работы с большей частью файлов, имеющихся в хранилище.

Поскольку операция считывания и обработки PDF занимает много времени, было решено препроцессинг корпуса документов проводить автономно, а его результат – считанный, очищенный от служебной информации и разбитый на абзацы текст, сохранять в базу данных формата sqlite (https://www.sqlite.org). Кроме самого текста в базу данных сохраняется указание имени файла, из которого он взят, и номера страницы.

При контекстном и тематическом поиске фрагменты текста часто рассматриваются в приближении «мешка слов», когда учитывается только факт присутствия слова в документе, но не берется в рассмотрение их порядок. При анализе текстов в приближении «мешка слов» теряют смысл падежи и склонения. В этом случае целесообразно привести все слова к так называемой нормальной форме: существительные – к именительному падежу, единственному числу; прилагательные – к именительному падежу единственному числу, мужскому роду; глаголы, причастия, деепричастия – к нормальной форме глагола. Нормализация слов русского языка позволяет сжать словарь корпуса примерно в 5 раз, что в конечном итоге увеличивает точность поиска и рубрикации. Для нормализации слов в данной программе использовалась библиотека pymorphy [4].

Следующие шаги, практически всегда применяемые при анализе текстов и также призванные повысить точность поиска и рубрикации, – это удаления из рассмотрения так называемых стоп-слов (союзов, предлогов и других часто употребляемых слов) и отбрасывание редко встречающихся слов. Последние часто являются опечатками и вносят шум в данные.

Для тематического поиска часто используются методы, основанные на векторном представлении слов: когда каждому слову ставится в соответствие вектор в признаковом пространстве. Такие векторные представления стремятся создать описание, при котором близкие по смыслу слова, синонимы, имею близкие по евклидову расстоянию векторы, а далекие по смыслу слова, антонимы, – далекие. В этом случае всему тексту можно поставить в соответствие вектор, рассчитанный, как средний вектор, входящих в него слов. В случае удачного построения векторного представления векторы тематически близких текстов будут близки, далеких – далеки.

Методы отображения слов в векторное представление будут обсуждаться ниже. Программа использует заранее построенный словарь соответствия слова вектору для расчета вектора фрагмента текста. Эта процедура также ресурсоемкая, поэтому было решено расчет среднего вектора проводить автономно, а результат сохранять в базу данных.

В результате обработки корпуса в базе данных будут сохранены фрагменты текста с указанием исходного документа и номера страницы, их векторное представление и обратный индекс, рассчитанные после нормализации слов в текстах.

3. Векторное представление слов

Один из первых методов построения векторного представления текстовой информации был основан на поиске сингулярного разложения матрицы TF-IDF [5]. В результате такого разложения получаются два множества векторов – одно соответствует текстам, другое словам. Евклидовы расстояния между различными векторами слов (или векторами текстов) при этом оказываются тем меньше, чем ближе их смысловые значения.

Большее распространение получил вероятностный латентно-семантический анализ. В этом подходе решается задача автоматического выделения рубрик в корпусе документов и вероятностного отнесения документа к этим рубрикам [6]. При этом в процессе рубрикации строятся векторные представления слов на основе частот их встречаемости в тех или иных выделенных рубриках. В дальнейшем этот метод получил развитие в модели латентного размещения Дирихле (latent Dirihlet alloation – LDA) [7].

В последние годы наибольшую популярность приобрел метод отображения слов в векторное пространство, известный в литературе как word2vec [8]. Суть подхода заключается в обучении нейронной сети, состоящей из входного, скрытого, и выходного слоев предсказывать по слову (или их последовательности) в тексте следующее слово. В word2vec существует две основные архитектуры: COBOW и Skip-gram. В архитектуре COBOW нейронная сеть обучается предсказывать текущее слово, получая на вход окружающий его контекст, порядок слов контекста при этом не учитывается. В Skip-gram нейронная сеть обучается предсказывать контекст исходя из текущего слова. В результате обучения нейросети фиксируются ее синаптические коэффициенты. При этом коэффициенты ее первого

слоя в дальнейшем могут быть использованы как векторы, расстояния между которыми отражают семантическую близость слов.

Для построения отображения слов в векторное представление создаваемая программа использует библиотеку GenSim (https://radimrehurek.com/gensim/), реализующую алгоритм word2vec.

4. Тематический поиск

Для тематического поиска программа использует модель, действующую по принципу «найти похожее»: пользователь предоставляет программе фрагменты текста, относящиеся к выбранной им тематике, и помечает их как таковые. После чего в режиме поиска по темам программа ищет тексты, близкие по содержанию некоторому обобщению обучающих примеров.

Модель, обобщающую обучающие фрагменты текста, в данном случае нельзя в полной мере отнести к моделям «обучения с учителем» [9], поскольку она не использует отрицательные примеры – тексты, заведомо не принадлежащие указанной теме. Для построения такой модели было решено использовать самоорганизующиеся карты Кохонена [10]. Выбор в пользу самоорганизующихся карт был сделан в виду их хорошей обобщающей способности и вычислительной дешевизны процедуры построения, что позволит переобучать описание тем в режиме реального времени в случае внесения пользователем изменений в обучающую выборку.

Карта Кохонена – метод кластеризации данных, при котором кластеры упорядочены между собой в виде гексагональной или прямоугольной сетки. Процедуру кластеризации данных с помощью самоорганизующихся карт можно описать как подстройку двумерной сетки кластеров под рельеф данных в многомерном пространстве. В результате карта даст аппроксимацию некоторого двумерного подмножества близко лежащего к точкам обучающей выборки в многомерном признаковом пространстве. В нашем случае карта строится по множеству векторов, соответствующих обучающим текстовым примерам.

Математически карта Кохонена задается множеством центров кластеров – точек в признаковом пространстве $\{\vec{m}_k\}_{k=1}^{K}$, где K – их число. В качестве меры смысловой близости фрагмента текста, который описывается вектором $\vec{x}$, к рассматриваемой тематике берется расстояние r от $\vec{x}$ до ближайшего кластера карты:

$$r = \min_k \|\vec{x} - \vec{m}_k\| .$$

5. Описание программы

Программа написана на языке Python. Интерфейс программы разработан с помощью библиотеки PyQt.

Программа

1) выполняет предобработку корпуса PDF-документов, в которую входит построение обратного индекса и расчет векторного представления текстов;

2) выполняет контекстный и тематический поиск в корпусе или отдельном документе;

3) выявляет наличие тем (ОСП), представленных в передаваемом ей корпусе или документе;

4) имеет интерфейс, позволяющий пользователю собирать и редактировать обучающие примеры, просматривать результаты поиска и анализа документов.

На рис. 1 показан интерфейс программы. В левом верхнем окне интерфейса находится дерево ОСП. Выбрав ОСП, можно запустить тематический поиск по корпусу файлов или по отдельному документу. Результаты поиска будут приведены в нижнем окне программы.

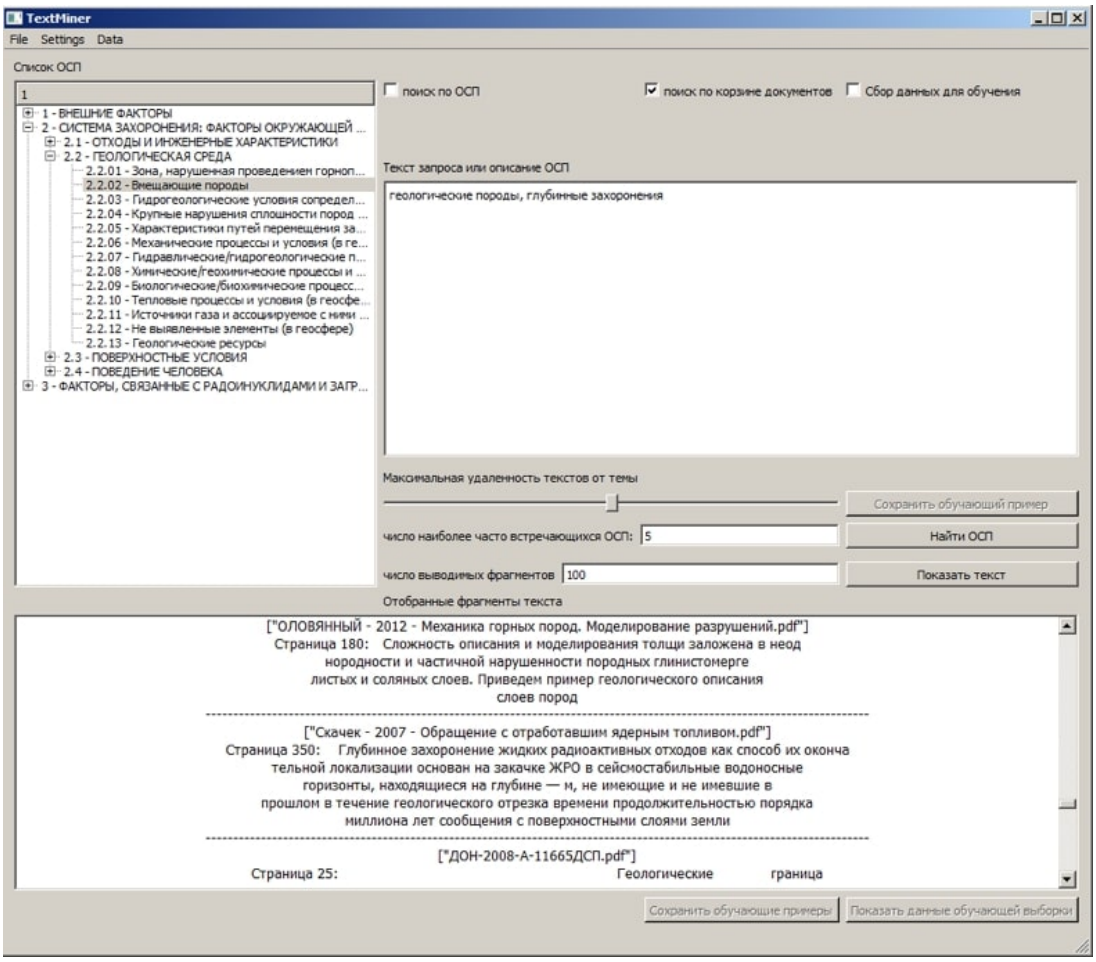

Рис. 1. Окно программы обработки текстовой документации

В правом верхнем углу интерфейса находится поле ввода поискового запроса. Результаты контекстного поиска также будут отображены в нижнем окне.

Запуск процедуры обработки корпуса, расчета для него обратного индекса и остальной служебной информации выполняется из окна программы. Для этого пользователю нужно указать путь к директории с файлами. При предобработке корпуса данных программа

1) осуществляет считывание и препроцессинг PDF-файлов, очищает тексты от служебной информации;

2) разбивает документы на фрагменты (абзацы) и индексирует их;

3) с помощью библиотеки pymorphy2 нормализует слова;

4) рассчитывает обратный индекс;

5) с помощью word2vec модели рассчитывает векторы текстов в концепции «мешка слов»;

6) сохраняет обратный индекс, векторы текстов, а также исходное содержание самих текстов в базу данных.

Создаваемая в результате обработки текстов база данных состоит из следующих таблиц:

1) таблица файлов содержит id файлов и пути к ним;

2) таблица текстов содержит id текста, id файла, из которого текст взят, номер страницы в файле, текстовое поле, хранящее содержание текста, и поле, хранящее вектор текста;

3) таблица словаря содержит список используемых слов и их id;

4) таблица обратного индекса содержит поля: id слова, id текста, в котором оно встретилось, и число включений данного слова в указанный текст.

Содержание таблицы обратного индекса позволяет быстро осуществлять поиск текстов по словам. Рассчитанные заранее векторы текстовых полей ускоряют процедуру тематического поиска.

Для сбора примеров текстов, относящихся к ОСП, пользователю необходимо активировать чекбокс «Сбор данных для обучения» и выбрать ОСП в дереве ОСП, для которого подбираются примеры. Обучающие примеры могут быть введены вручную, загружены в программу из текстового файла или выбраны из выдачи предыдущего поискового запроса.

Еще одной функцией программы является поиск ОСП, представленных в документе или корпусе документов. Эта функция позволяет проверить наличие или отсутствие в документации тех или иных типовых тем. Найденные в тексте ОСП программа выделяет в интерфейсе жирным шрифтом.

6. Результаты тестирования программы

Для оценки качества тематического поиска было проведено два типа тестов. В первом смотрелось, насколько программа хорошо отделяет относящиеся к выбранной теме фрагменты текста от остальных. Для этого было обучено несколько ОСП. Размеры обучающих множеств варьировались от 7 до 15 фрагментов текста в каждой теме. Экспертным образом были сформированы тестовые выборки. Задача программы состояла в том, чтобы распознать, как принадлежащие заданной теме фрагменты текста, относящиеся к ней по мнению эксперта, и отбросить не относящиеся. Положительные тестовые множества были сформированы для пяти тем. В каждом случае они состояли из нескольких десятков примеров. Отрицательные примеры выбирались случайным образом из корпуса данных. Размер отрицательной выборки составлял 300 примеров.

Точность описания в данном случае удобно характеризовать чувствительностью – долей правильно распознанных положительных примеров и специфичностью – долей правильно распознанных отрицательных примеров. Были получены следующие усредненные по рассмотренным пяти темам значения этих показателей:

Чувствительность = 97%.

Специфичность = 98%.

Точность распознавания получилась достаточно высокой. Однако такой результат обусловлен сравнительной простотой поставленной задачи – случайно выбранные отрицательные примеры часто относятся к совершенно другой области знания, нежели положительные и не пересекается с последними не только по каким-то ключевым словам, но и по контексту, что делает такие множества легко разделяемыми. Поэтому было решено составить другой тест, цель которого – проверить, насколько хорошо программа распознает близкие по смыслу темы. Для этого были выбраны два ОСП:

«2.2.06 – *Механические процессы и условия* (*в геосфере*)».

«2.2.08 – *Химические-геохимические процессы и условия* (*в геосфере*)».

В тестовой выборке, составленной для первого из этих ОСП, было 20 примеров. Все 20 были отнесены к данному ОСП, однако 2 из них были одновременно отнесены и ко второму ОСП. В выборке, составленной для второго ОСП, было 43 примера. Правильно распознаны были 42 из них. При этом 3 были одновременно отнесены и к первому ОСП.

Такая точность позволяет автоматически проводить первичный тематический анализ, выявляющий в документах наличие тех или иных тем, что в конечном итоге сокращает время, затраченное пользователем на анализ содержания документации.

7. Заключение

В статье описана программа, выполняющая контекстный и тематический поиск текстовой информации. Тематический поиск осуществляется по принципу «найти похожее» – пользователь подбирает примеры фрагментов текста, относящихся к выбранной тематике, а программа находит тексты, близкие по смыслу к обучающим примерам. Тестирование функции автоматического тематического поиска продемонстрировало точность, указывающую на практическую целесообразность его применения.

Дальнейшее развитие программы видится в построении диалоговой системы, которая будет формировать ответы на узкоспециальные вопросы пользователя, основываясь на информации, содержащейся в предоставленной программе специализированной документации.

Литература

1. *Bird S., Loper E., Klein E.* Natural Language Processing with Python. O'Reilly Media Inc., 2009.
2. *Robertson S.E., Walker S., Hancock-Beaulieu M.* Okapi at TREC-7. In Proceedings of the Seventh Text Retrieval Conference. Gaithersburg, USA, 1998.
3. Features, Events and Processes (FEPs) for Geologic Disposal of Radioactive Waste, NEA International FEP Database: Version 2.1. NEA OECD, 2006.
4. *Korobov M.* Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images. Social Networks and Texts. 2015. P. 320–332.
5. *Jones K.S.* A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. 1972. 28:11–21.
6. *Hofmann T.* Probabilistic latent semantic analysis // Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval. 1999.
7. *Blei D.M., Ng A.Y., Jordan M.I., Lafferty J.* Latent Dirichlet Allocation // Journal of Machine Learning Research. 3 (4–5): P. 993–1022. (January 2003).
8. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space // Proceedings of Workshop at ICLR, 2013.
9. *Нужный А.С.* Байесовская регуляризация в задаче аппроксимации функции по точкам с помощью ортогонализованного базиса // Матем. моделирование. 2011. 23:9. Р. 33–42.
10. *Kohonen T.* Self-Organizing Maps. Springer, 1997 (2-nd edition).
11. *Нужный А.С., Розанов В.Б., Степанов Р.В., Шумский С.А.* Построение полупараметрических моделей развития неустойчивости Рэлея–Тейлора с помощью нейросетевой обработки результатов численного эксперимента // Матем. моделирование. 2004. 16:7. Р. 21–30.
12. *Нужный А.С., Розанов В.Б., Степанов Р.В., Шумский С.А* Использование самообучаемых интеллектуальных моделей для прогнозирования развития НРТ индуцированного турбулентного перемешивания // Физика плазмы. 2005. Т. 31, № 4. С. 342–349.

References

1. *Bird S., Loper E., Klein E.* Natural Language Processing with Python. O'Reilly Media Inc., 2009
2. *Robertson S.E., Walker S., Hancock-Beaulieu M.* Okapi at TREC-7. In Proceedings of the Seventh Text Retrieval Conference. Gaithersburg, USA, 1998.

3. Features, Events and Processes (FEPs) for Geologic Disposal of Radioactive Waste, NEA International FEP Database: Version 2.1. NEA OECD, 2006.

4. *Korobov M.* Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts. 2015. P. 320–332.

5. *Jones K.S.* A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. 1972. 28:11–21.

6. *Hofmann T.* Probabilistic latent semantic analysis. Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval. 1999.

7. *Blei D.M., Ng A.Y., Jordan M.I., Lafferty J.* Latent Dirichlet Allocation. Journal of Machine Learning Research. 3 (4–5): P. 993–1022. (January 2003).

8. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR, 2013.

9. *Nuzhny A.S.* Bayesian regularization in the problem of point-by-point function approximation using an orthogonalized basis. Math. Model. 2011. 23:9. P. 33–42. (in Russian).

10. *Kohonen T.* Self-Organizing Maps. Springer, 1997. (2-nd edition).

11. *Nuzhny A.S., Rozanov V.B., Stepanov R.V., Shumsky S.A.* Construction of semi-parametric models for the development of Rayleigh – Taylor instability using neural network processing of the results of a numerical experiment. Math. Model. 2004. 16:7. P. 21–30. (in Russian).

12. *Nuzhny A.S., Rozanov V.B., Stepanov R.V., Shumsky S.A.* Using self-taught intelligent models to predict the development of NRT induced turbulent mixing. Plasma physics. 2005. V. 31, N 4. P. 342–349. (in Russian).

Поступила в редакцию 30.01.2019