Авторы:

Васильев Дмитрий Олегович

аспирант

Скоселев Денис Андреевич

магистрант

Ильин Виталий Алексеевич

магистрант

Научный руководитель:

Никольский Сергей Николаевич

д-р техн. наук, профессор

ФГБОУ ВО «Московский технологический университет»

г. Москва

ОПТИМИЗАЦИЯ СИСТЕМЫ ПОИСКА ЗАИМСТВОВАНИЙ НА ОСНОВЕ АРХИТЕКТУРЫ МНОГОАГЕНТНОЙ СИСТЕМЫ

Аннотация*: в данной работе рассмотрены проблемы, которые возникают при разработке и эксплуатации системы поиска заимствований, а также предложены методы их решения.*

Ключевые слова*: многоагентная система, система поиска заимствований, архитектура многоагентных систем.*

Развитие сети Интернет и широкое использование технологии Big Data привели к возрастанию значения проблем поиска заимствований в тексте. Для решения этой проблемы следует использовать специальные программно-аппаратные комплексы, направленные на автоматический анализ текста. В данной работе рассмотрены вопросы, которые возникают при разработке и эксплуатации системы поиска заимствований, а также предложены методы, которыми они могут быть решены.

Под многоагентной системой понимается система, взаимодействующая с несколькими интеллектуальными агентами. В информационных технологиях

интеллектуальным агентом называют самостоятельно выполняющееся задание (программа) в течение длительных промежутков времени.

В отличие от монолитных систем, многоагентные системы имеют несколько важных характеристик:

1. Автономность: агенты, хотя бы частично, независимы.

2. Ограниченность представления: ни у одного из агентов нет представления о всей системе.

3. Децентрализация: нет агентов, управляющих всей системой.

Многоагентные системы применяются в задачах, которые невозможно выполнить с помощью одного агента, или в задачах, выполнение которых в монолитной системе будет выполняться слишком долго. Задача поиска заимствований является примером второй категории таких задач.

Основными шагами автоматизированного поиска заимствований являются:

1. Прием и регистрация документа на проверку.

2. Формализация полученного документа.

3. Сравнение полученного формализованного документа с документами, имеющихся в хранилище системы поиска заимствований.

На первом шаге пользователь загружает документ в систему, который необходимо проверить и найти заимствования. Документ на данном этапе обычно загружается в форматах pdf, doc, docx, txt. Также он может содержать картинки, таблицы и другие элементы, которые не являются текстом. Поэтому, после загрузки документа в систему, необходимо привести его к виду, в котором его можно будет автоматически обработать.

На втором шаге происходит формализация документа. Для этого необходимо сделать следующие действия:

– определить формат документа;

– извлечь из документа текст, удалив из него таблицы, рисунки;

– удалить из документа техническую информацию (титульный лист, содержание, список литературы и прочее), если таковая присутствует;

– преобразовать текст в формат, с которым работают алгоритмы поиска заимствований на шаге 3.

На третьем шаге формализованный текст поступает в подсистему автоматического анализа, которая сравнивает его с документами, находящимися в хранилище, и формирует отчет о найденных заимствованиях.

Недостатки монолитных систем поиска заимствований и способы их решения

Основными недостатками монолитных систем поиска заимствований, с которыми можно столкнуться при эксплуатации монолитных систем поиска заимствований:

– низкая скорость обработки одного документа при большом количестве документов в хранилище;

– невозможность параллельной обработки нескольких документов;

– невозможность горизонтального масштабирования системы.

Обе эти проблемы приводят к возникновению больших очередей обработки документов, что негативно сказывается на работе пользователей с системой.

Для оптимизации скорости обработки документа можно искать заимствования не по всему хранилищу, а только из выборки документов, соответствующих той же тематики, что и документ, поступивший на обработку.

В работе использовался использовали алгоритм LSA на заголовки документов с целью найти схожие темы. На первом шаге требуется составить частотную матрицу индексируемых слов. В этой матрице строки соответствуют индексированным словам, а столбцы – документам. В каждой ячейке матрицы указано какое количество раз слово встречается в соответствующем документе. Затем считается косинусное расстояние между двумя векторами (в нашем случае – вектор является столбцом документа), которое находится в интервале от 0 (нет совпадений) до 1(полное совпадение).

Автоматический рубрикатор используется в разрабатываемой системе поиска заимствования. Результаты использования рубрикатора проводились на 100 проверяемых документов. Использование автоматического рубрикатора

позволило сократить время поиска в среднем в 2 с половиной раза. Приведем частный случай проверки.

До использования автоматического рубрикатора выборка текстов, в которых искались заимствования составляла 715 документов. Время поиска в документе составляло 322 253 мс, а процент оригинальности составил 72,47 процентов.

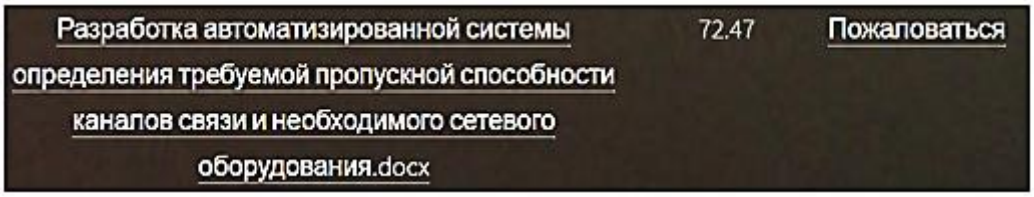

Рис. 1. Результаты работы системы поиска заимствований без применения рубрикатора

Во втором случае при использовании автоматического рубрикатора выборка составила 147 документов. Время поиска составило 94 112 мс, а процент оригинальности документа незначительно вырос.

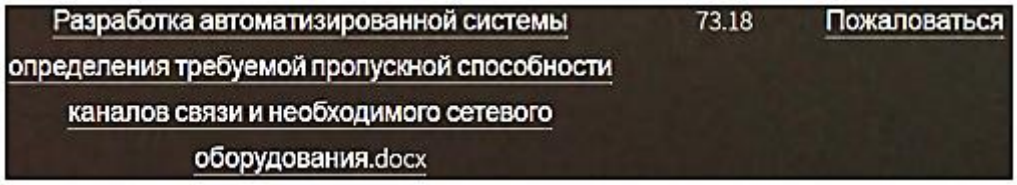

Рис. 2. Результаты работы системы поиска заимствований с применением рубрикатора

Переход от монолитной архитектуры к многоагентной

Для решения задач повышения скорости и вопросов, связанных с горизонтальным масштабированием системы в работе, предлагается следующая многоагентная структура:

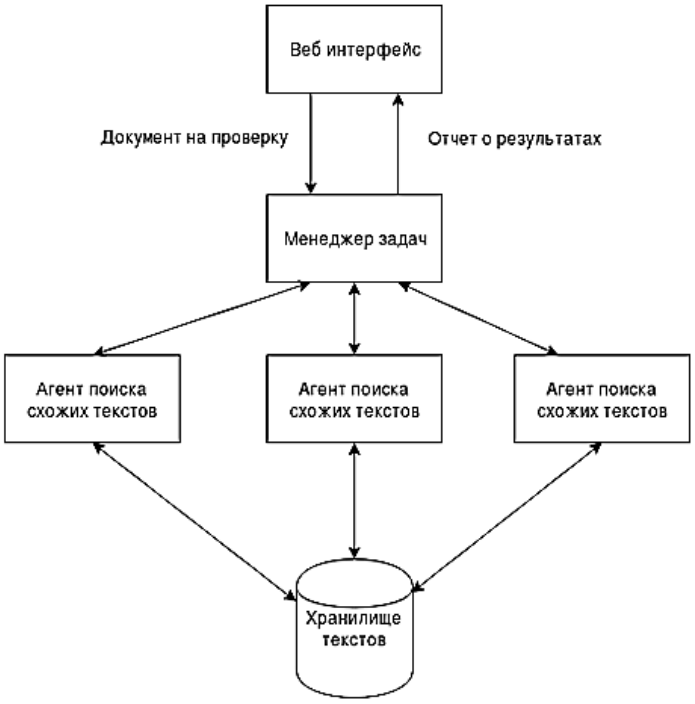

Рис. 3. Архитектура многоагентной системы поиска заимствований

Пользователь через веб-интерфейс загружает новый документ на проверку, тем самым создавая задачу в подсистеме «Менеджер задач». Данная подсистема формирует очередь обработки документов, распределяет ее между агентами, а также получает отчет о результатах проверки документа от агентов.

Агенты поиска схожих текстов выполняют задачи формализации документов, регистрации их в хранилище документов, и непосредственно, анализ документов на заимствованные фрагменты. По окончанию работ агент формирует отчет и отправляет его «Менеджеру задач», а тот в свою очередь отображает отчет пользователю через интерфейс.

Заключение

Применение автоматического рубрикатора позволило ускорить обработку документа в системе, а переход к многоагентной архитектуре дал нашей системе способность к горизонтальному масштабированию и параллельной обработке нескольких документов. Одним из способов улучшения текущей архитектуры системы – выделения нового класса агентов, выполняющих работу по формализации и регистрации документов. Данные процедуры не требуют высоких

вычислительных затрат, как анализ текста на заимствования, поэтому это могло бы ускорить обработку новых документов.

Список литературы

1. Distributed Optimization for a Class of Nonlinear Multiagent Systems With Disturbance Rejection / Y. Zhao, Y. Liu, G. Wen, G. Chen / IEEE Transactions on Automatic Control. – 2017. – С. 1655–1666.

2. Wooldridge M. An Introduction to Multiagent Systems / J.M Wooldridge // Wiley. – 2009. – 460 с.

3. Латентно-семантический анализ [Электронный ресурс]. – Режим доступа: https://habr.com/post/110078/