

Санкт-Петербургский государственный университет

Кафедра технологии программирования

Буланина Екатерина Дмитриевна

Предсказание исполнителя задания в системе электронного документооборота

Курсовая работа

Научный руководитель:
к. ф.-м. н., доцент Добрынин В. Ю.

Санкт-Петербург
2018

SAINT-PETERSBURG STATE UNIVERSITY

Applied Mathematics and Control Processes Faculty
Programming Technology Chair

Ekaterina Bulanina

Assignee prediction in document automation system

Graduation Thesis

Scientific supervisor:
Vladimir Dobrynin

Saint-Petersburg
2018

Оглавление

| | |
|--|----|
| Введение | 4 |
| 1. Постановка задачи | 5 |
| 2. Обзор | 6 |
| 3. Исследование предоставленных данных | 7 |
| 4. Реализация | 8 |
| 5. Эксперименты | 9 |
| Заключение | 10 |
| Список литературы | 11 |

Введение

Во многих компаниях, оперирующих с большим количеством документов, для автоматизации и ускорения работы используются системы электронного документооборота. Такие системы значительно уменьшают время таких операций с документами, как регистрация, рассылка, хранение или использование содержащейся в них информации.

Современные системы электронного документооборота (СЭД) при помощи компьютерной обработки и методов машинного обучения добавляют множество новых возможностей: распознавание и выгрузка текстов из отсканированных изображений, автоматическое определение типа документа и т.д.

Задача автоматического назначения исполнителя задания является интересной в контексте анализа данных, а её решение способствует расширению функционала СЭД. Исполнитель — это сотрудник, который будет обрабатывать документ и, после выполнения над ним необходимых операций, передавать его другому сотруднику. Для компаний с большой иерархической организацией было бы очень полезно заранее знать, какую последовательность (цепочку) исполнителей пройдет созданный в системе документ.

Одной из наиболее популярных систем документооборота в России является система Docsvision [4], представляющая из себя платформу для организации и автоматизации управления в отраслях государственного сектора, банковской сферы, оборонно-промышленного комплекса и других областей.

1. Постановка задачи

Целью данной работы является создание инструмента для предсказания цепочки исполнения документа. Для этого были поставлены следующие задачи:

- Провести анализ данных, предоставленных компанией Digital Design [3];
- Выполнить обзор существующих подходов для предсказания исполнения и выбрать подходящий;
- Реализовать выбранный алгоритм;
- Протестировать полученную реализацию.

2. Обзор

В качестве основного источника для изучения методов машинного обучения и анализа данных я использовала книгу К. Маннинга «Введение в информационный поиск» [5]. В этом учебнике рассматривается современный подход ко всем аспектам проектирования и внедрения систем сбора, индексирования и поиска документов, методы оценки систем и использование методов машинного обучения в наборах текстов. В частности, в этой книге описываются алгоритмы стемминга и лемматизации, которые используются в данной работе.

Основной алгоритм, использованный в решении задачи, — алгоритм Apriori [2]. Этот алгоритм реализует поиск ассоциативных правил [1] (т.е. зависимостей между элементами) в больших наборах данных. Использование этого алгоритма для предсказания исполнителя хорошо описано в статье «Bug Assignee Prediction Using Association Rule Mining» [8]. В ней рассматривается решение задачи предсказания разработчика, который будет работать над исправлением ошибки в проекте. Авторы статьи анализируют ассоциативные правила, найденные алгоритмом Apriori, для предсказания на примере пяти ведущих разработчиков в нескольких проектах.

Отдельного внимания требует работа Н. Чурикова [11]. В этой работе рассматривается задача рекомендации исполнителя документа и методы её решения. Моя работа представляет альтернативный подход для решения этой задачи.

При реализации решения задачи я использовала язык Python и библиотеки Pandas и NumPy, для изучения которых была полезна книга В. МакКини «Python для анализа данных» [6].

3. Исследование предоставленных данных

Во время регистрации документа в системе Docsvision заполняется форма, представленная ниже. В ней вручную или автоматически заполняются параметры (атрибуты) документа: тип, автор, категория, краткое описание и т.п. После загрузки в базу DocsVision документ и его атрибуты представляют собой метаданные, хранящиеся в формате JSON. Аналогичным образом представляются в системе задания, атрибутами которых могут служить время создания, срочность и описание поручения. Задания также хранятся в формате JSON.

Рис. 1: Создание документа в системе DocsVision

The screenshot shows a web form for creating a document in the DocsVision system. The form includes the following fields and elements:

- Тип документа (Document Type):** A dropdown menu with "Письмо" (Letter) selected.
- Краткое описание (Brief description):** A text area containing "Возможность кредитования предприятия по разработке Тюменского месторождения".
- От кого (From):** A dropdown menu with "ФГУП СКБТ 'Аметист' – Ковров И.П." selected.
- Кому (To):** A dropdown menu with "Арканьянский И.В." and "Симонова С.А." selected.
- Временный № (Temporary number):** A text field containing "вр-871 от 16.06.2016".
- Автор (Author):** A text field containing "Аватор Филиппенко И. Н.".
- Подразделение регистрации (Registration department):** A dropdown menu with "Направление СДУ" selected.
- Категории (Categories):** A dropdown menu with "Кредитование" selected.
- Добавить дело (Add case):** A button.

В полученных JSON-файлах атрибуты документа как правило представляют собой категориальные переменные или текст. Категориальные переменные задаются в виде *UUID* [9], который является уникальным значением вида xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxxxx, где M и N задают версию идентификатора, а x является произвольной латинской буквой или цифрой. Каждому документу соответствует *резолуция* — файл, хранящий в себе последовательность заданий в порядке их создания. Каждое задание характеризуется *атрибутами* и *исполнителем*. Исполнители бывают двух видов: *Appointed* (тот, кто был назначен на задание при его создании) и *Executes* (тот, кто непосредственно выполнял задание). Специфика данных такова, что *Appointed* и *Executes* обычно совпадают. Поэтому в представленной реализации анализируются только те, кто был назначен на задание. Однако в общем случае данный подход можно расширить.

4. Реализация

Для осуществления поставленной задачи была произведена лемматизация слов, из которых состоит описание документа. Для реализации этого использовались библиотеки обработки естественного языка Natural Language Toolkit [7] и PyMyStem3 [10]. Далее для каждого исполнителя с $1..k$ уровня исполнения любого документа в словарь заносится набор слов из описания документов как показано в листинге 1.

После того, как известен набор слов, появлявшихся в описаниях документов исполнителей, можно применить поиск ассоциативных правил алгоритмом Apriori, представленный в листинге 2.

Listing 1: Псевдокод создания словаря исполнителей

```
people = {}
for doc in dataBase:
    assignees = doc.resolution['Executes'] # получение цепочки исполнителей документа
    for assignee in assignees:
        people[assignee] += doc.description
```

Listing 2: Псевдокод алгоритма Apriori

```
 $L_1 = \{large\ 1 - itemsets\}$ 
 $k = 2$ 
while  $L_{k-1} \neq \{\}$ :
     $C_k = \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k - 1\} \not\subseteq L_{k-1}\}$ 
    for transactions  $t \in T$ :
         $C_t = \{c \mid c \in C_k \wedge c \subseteq t\}$ 
        for candidates  $c \in C_t$ :
             $count[c] = count[c] + 1$ 
     $L_k = \{c \mid c \in C_k \wedge count[c] \geq \epsilon\}$ 
     $k = k + 1$ 
return  $\bigcup_k L_k$ 
```


5. Эксперименты

Полученная реализация была протестирована на данных от компании Digital Design. Данные представляли собой 131215 документов Правительства Мурманской области. Ниже в таблице представлены ассоциативные правила для нескольких исполнителей с наибольшим числом заданий.

Support — выраженное в процентах отношение числа документов, в которых встретились указанные термы и данный исполнитель, к общему числу рассматриваемых документов (в моем случае выборка из трех исполнителей с наибольшим числом заданий).

Confidence — выраженное в процентах отношение числа документов, на которых был назначен указанный исполнитель, к числу документов, в которых встретились указанные термы.

Таблица 1: Ассоциативные правила для Дмитриенко

| # | [terms] → Дмитриенко | Confidence | Support |
|---|---|------------|---------|
| 1 | ['губернатор', 'данные', 'оперативный', 'перечень', 'поручение'] | 1.9269 | 71.2215 |
| 2 | ['перечень', 'поручение', 'совещание'] | 2.0614 | 70.229 |
| 3 | ['данные', 'оперативный', 'перечень', 'совещание'] | 1.9606 | 70.994 |
| 4 | ['контроль', 'поручение', 'снятие'] | 1.2043 | 65.7492 |
| 5 | ['2011', 'губернатор', 'данные', 'оперативный', 'перечень', 'поручение', 'совещание'] | 1.3836 | 65.1715 |
| 6 | ['контроль', 'перечень', 'поручение', 'снятие'] | 1.1091 | 68.9895 |

Таблица 2: Ассоциативные правила для Портная

| # | [terms] → Портная | Confidence | Support |
|---|---|------------|---------|
| 1 | ['2011', 'перечень', 'поручение'] | 1.0979 | 37.3333 |
| 2 | ['исполнение', 'продление', 'срок'] | 1.1987 | 69.2556 |
| 3 | ['направлять', 'рф'] | 1.0123 | 99.13 |
| 4 | ['данные', 'оперативный', 'перечень', 'поручение', 'совещание'] | 1.2697 | 98.12 |

Заключение

В ходе выполнения работы были получены следующие результаты:

- Исследованы данные, предоставленные компанией Digital Design;
- Проведен обзор предметной области и изучены существующие подходы к решению задачи;
- Реализован алгоритм Apriori;
- Полученная реализация протестирована на предоставленных данных.

Список литературы

- [1] Agrawal Rakesh, Imieliński Tomasz, Swami Arun. Mining Association Rules Between Sets of Items in Large Databases // SIGMOD Rec. — 1993. — . — Vol. 22, no. 2. — P. 207–216. — URL: <http://doi.acm.org/10.1145/170036.170072>.
- [2] Agrawal Rakesh, Srikant Ramakrishnan. Fast Algorithms for Mining Association Rules in Large Databases // Proceedings of the 20th International Conference on Very Large Data Bases. — VLDB '94. — San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1994. — P. 487–499. — URL: <http://dl.acm.org/citation.cfm?id=645920.672836>.
- [3] Digital Design. — <https://digdes.ru/>. — Дата обращения: 07.05.2018.
- [4] DocsVision. — <http://www.docsvision.com>. — Дата обращения: 07.05.2018.
- [5] Manning Christopher D., Raghavan Prabhakar, Schütze Hinrich. Introduction to Information Retrieval. — New York, NY, USA : Cambridge University Press, 2008. — ISBN: 0521865719, 9780521865715.
- [6] McKinney Wes. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. — First edition. — Beijing : O'Reilly, 2013. — ISBN: 9781449319793 1449319793.
- [7] NLTK. — <https://www.nltk.org/>. — Дата обращения: 07.05.2018.
- [8] Sharma Meera, Kumari Madhu, Singh V. B. Bug Assignee Prediction Using Association Rule Mining // Proceedings, Part IV, of the 15th International Conference on Computational Science and Its Applications – ICCSA 2015 - Volume 9158. — New York, NY, USA : Springer-Verlag New York, Inc., 2015. — P. 444–457. — URL: http://dx.doi.org/10.1007/978-3-319-21410-8_35.

- [9] UUID. — <https://ru.wikipedia.org/wiki/UUID>. — Дата обращения: 07.05.2018.
- [10] pymystem3. — <https://github.com/nlpub/pymystem3/>. — Дата обращения: 07.05.2018.
- [11] Чуриков Никита. Предсказание атрибутов документов в системе документооборота. — 2017.