Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный университет имени М. В. Ломоносова»

Филологический факультет

Кафедра теоретической и прикладной лингвистики

Карпенко Кирилл Михайлович

Автоматическая обработка и преобразование библиографических ссылок

Курсовая работа
III курс бакалавриата

Научный руководитель:
доктор филологических наук
Гращенков Павел Валерьевич

Москва
2018

Оглавление

Введение

Научная литература содержит огромные объёмы библиографической информации. Возникающие при обработке этой информации практические задачи внесения информации об изданиях в библиографические базы данных, поиска её с использованием информационно-поисковых систем, преобразования библиографических объектов из одного формата цитирования в другой зачастую подразумевают обработку её вручную (в особенности это справедливо в отношении последней задачи). Процессы решения этих задач являются довольно трудоёмкими и затратными по времени. По этой причине представляется актуальной задача автоматического преобразования библиографических объектов (записей и ссылок) в их структурированные эквиваленты, иными словами, извлечения информации об индивидуальных компонентах каждого объекта (об именах авторов, названии работы, информации об издании и т. д.). Предлагаемая система призвана частично решить данную задачу через автоматизацию извлечения этой информации.

Настоящая система разработана для автоматической сегментации списков библиографических ссылок в соответствии с заданными в пользовательском файле конфигурации правилами. Особенностью этой системы является её независимость от какого-либо конкретного формата цитирования, что обеспечивается именно возможностью настройки сегментации пользовательскими файлами конфигурации.

Задача извлечения компонентов библиографической ссылки относится к области извлечения данных (data extraction). Классическая задача извлечения информации состоит в извлечении структурированных данных из неструктурированной или слабоструктурированной информации; в более практических терминах ее можно определить как нахождение значений заранее определённых полей шаблона [Manning & Schütze 1999]; последнее определение, на наш взгляд, хорошо показывает прямое отношение решаемой в настоящей работе задачи разложения библиографической ссылки на значения её компонентов к области извлечения данных. С другой стороны, хотя библиографические ссылки являются по своей сути – и ввиду своей стандартизированности – структурированными объектами, возможные варианты их структуры являются чрезвычайно многочисленными ввиду большого числа составляющих их компонентов, опциональности большинства из них, слабой

определенности формата некоторых элементов (например, названия цитируемого издания). Кроме того, предписанный стандартами формат далеко не всегда в полной мере соблюдается при составлении ссылки. Все это делает задачу извлечения структурированных данных из библиографических ссылок во многих случаях нетривиальной.

Частной задачей в области извлечения данных, к которой также имеет отношение решаемая задача, является задача распознавания именованных сущностей (named entity recognition). Так, именованными сущностями являются многие элементы библиографической ссылки (наименованиями персон являются имена авторов и редакторов, наименованиями географических локаций – названия мест издания, организаций – издательств и т. д.). Рассматриваемая в настоящей работе задача, однако, в целом является несколько менее сложной, чем задачи, типичные для этой области, поскольку, как отмечалось выше, библиографические ссылки при всем многообразии существующих форматов их оформления являются достаточно стандартизированными объектами, а потому представляется принципиально возможной их обработка с довольно высокой точностью без привлечения более сложных методов, применяющихся в области распознавания именованных сущностей.

Для извлечения данных широко применяется метод сопоставления с образцом (pattern matching); наиболее известным и мощным средством в рамках этого метода являются регулярные выражения [Li *et al.* 2008], использование которых в этой сфере подробно описано в научной и учебной литературе (см., например, [Jurafsky & Martin 2000], [Grishman 2005], [Pratt-Hartmann 2010]). Регулярные выражения лежат в основе работы данной программы и используются в ней для сопоставления строкового представления библиографической ссылки на входе с возможными вариантами представления в ней ее компонентов, генерируемых на основе пользовательского файла конфигурации.

Использование регулярных выражений для работы с библиографическими ссылками было рассмотрено нами в работе [Карпенко 2017]; предложенная в этой работе система преобразовывала ссылки из формата ГОСТ Р 7.0.5—2008 в формат на латинице, используемый в журнале «Вопросы языкознания», и сопоставление с образцом наиболее активно использовалось в модулях сегментации и автозамены этой системы. В определённой мере этот подход продолжен в настоящей системе;

существенным её отличием от предыдущей, впрочем, является попытка реализовать сегментацию по произвольной заданной пользователем структуре ссылки в данном формате цитирования, а не только по структуре, соответствующей формату ГОСТ Р 7.0.5—2008. Кроме того, в недавней работе [Колмогорцев, Сараев 2017] подход, основанный на использовании регулярных выражений, был реализован для решения задачи распознавания библиографических записей в формате ГОСТ 7.1—2003 в русскоязычных текстах.

Устройство системы

Данная система состоит из двух основных модулей: модуля построения списка всех возможных вариантов библиографической ссылки на основании файла конфигурации (класс RegexCollectionBuilder) и модуля сегментации ссылки на основе этого списка (класс Segmentator). Также присутствуют служебные модули (чтения параметров запуска, чтения файла конфигурации и файла со списком ссылок, записи выходного файла).

На вход в программу поступают два текстовых файла:

1. Первый файл является файлом конфигурации и содержит три группы элементов: определения различных узлов выстраиваемой в ходе работы программы вспомогательной грамматики; определения полей, которые может содержать ссылка (FIELDS); определения ссылки полностью (ENTRIES).

2. Второй файл содержит список библиографических ссылок, которые будут подвергнуты сегментации в ходе работы алгоритма.

Программа последовательно считывает определения в каждой из трех вышеописанных групп строк файла конфигурации и строит на их основе вспомогательную грамматику.

Определения в первой группе содержат левую и правую части; в левой части определяется название того или иного узла, в правой части – его составляющие; в роли терминальных составляющих выступают регулярные выражения, в роли нетерминальных – уже определенные ранее узлы. Для каждой нетерминальной составляющей в определении происходит обращение к списку выстроенных этим же алгоритмом на предыдущих шагах регулярных выражений – определений данной нетерминальной составляющей. Для каждого элемента этого списка далее осуществляется ветвление алгоритма; к каждому варианту конкатенацией добавляются тем же способом все варианты следующего в записи нетерминального узла и так далее, пока не будут обработаны все определения составляющих в записи; последние затем конкатенируются, и полученное регулярное выражение заносится вместе с соответствующим ему названием только что определенного узла в список определений (алгоритм, таким образом, является рекурсивным).

Определения во второй группе (FIELDS) содержат имена нетерминальных узлов из списка ранее определённых в первой группе, которые являются полями; таким образом, при сегментации будет осуществляться извлечение значений соответствующих узлов.

Определения в третьей группе (ENTRIES) содержат возможные варианты структуры конечной библиографической ссылки, поступающей на вход; эти варианты определяются через нетерминальные узлы, определенные в первой группе, и через терминальные узлы, представленные регулярными выражениями. Алгоритм работает по принципу, аналогичному описанному выше для определений первой группы в части обработки индивидуальных узлов и рекурсии; на этом шаге, однако, строится список служебных объектов класса RegexInstruction, каждый из которых содержит одно возможное представление определения ссылки в виде регулярного выражения вместе с наименованием типа ссылки, содержащемся в левой части соответствующего определения в файле конфигурации, и с информацией о том, какие поля в этом представлении определены и как их можно извлечь из данного регулярного выражения (для этого при его построении используются захватывающие скобки для создания группы и записывается ее номер в данном регулярном выражении).

Пример синтаксиса, используемого в файле конфигурации, представлен на рис. 1.

По окончании работы модуля класса RegexCollectionBuilder, таким образом, имеется массив объектов класса RegexInstruction, описывающих возможные варианты представления библиографической ссылки в соответствии с описанными в файле конфигурации правилами.

Модуль сегментации получает на вход файл с ссылками и созданный в первом модуле массив объектов класса RegexInstruction. Алгоритм применяет к каждой ссылке все возможные определенные варианты ее представления и при успехе сопоставления ссылки с регулярным выражением создает служебный объект класса Entry, содержащий соответствия между полями, определенные для этого регулярного выражения в соответствующем объекте RegexInstruction, и их значениями, извлеченными из обрабатываемой ссылки. Для каждой ссылки затем определяется объект Entry, содержащий наибольшее количество непустых полей; этот финальный для каждой

ссылки шаг призван выбрать наилучший с точки зрения полноты (recall) вариант сегментации, извлекающий значения наибольшего числа возможных полей. Этот объект добавляется в массив, из которого по окончании работы алгоритма для всех ссылок из входного файла строится выходной файл формата .json, который и является результатом работы программы.

```
last-name = "[А-Я][А-Яа-я\-]+"
first-name = "[А-Я][А-Яа-я\-]+"
initial = "[А-Я]\."
editor-marker = "((ред\.)|(отв\. ред\.))"
and-others = "и др\."
book-title = ".+"
place-pub = "[А-Я][А-Яа-я\- ]+"
place-pub = "((М\.)|(СПб\.)|(Л\.))"
publisher = "[А-Яа-яA-Za-z\-\. ]+"
year-pub = "\d{4}"
edition-edited-text = " \(испр\. и доп\.\)"
edition-edited-text = " \(доп\. и испр\.\)"
edition = edition-num
edition = edition-num edition-edited-text
... ...
FIELDS
author-name
editor-name
book-title
pages-in-book
page-nums
place-pub
publisher
year-pub
edition
... ...
ENTRIES
book = author-name space "(" editor-marker period space ")?" book-title
period space place-pub colon space publisher comma space year-pub period
book = author-name space "(" editor-marker period space ")?" book-title
period space edition period "?" place-pub colon space publisher comma space
year-pub period
... ...
```

Рис. 1. Пример синтаксиса, используемого в пользовательских файлах конфигурации

Пример фрагмента входного файла со ссылками и построенных алгоритмом из этого фрагмента объектов, содержащих значения полей, представлен на рис. 2.

```
Муравьева И. А. Типология инкорпорации. Дисс. ... докт. филол. наук. М.:
РГГУ, 2004.
Срезневский И. И. Материалы для словаря древнерусского языка по
памятникам. В 3-х томах. СПб.: Типография Императорской Академии Наук, 1893.
```


```
{
  "book-title": "Типология инкорпорации. Дисс. ... докт. филол. наук",
  "author-name": "Муравьева И. А.",
  "place-pub": "М.",
  "publisher": "РГГУ",
  "year-pub": "2004"
},
{
  "book-title": "Материалы для словаря древнерусского языка по
  памятникам. В 3-х томах",
  "author-name": "Срезневский И. И.",
  "place-pub": "СПб.",
  "publisher": "Типография Императорской Академии Наук",
  "year-pub": "1893"
},
```


Рис. 2. Фрагменты входного файла с библиографическими ссылками и объектов выходного json-файла с результатами сегментации

Пример окна командной строки, сопровождающего запуск и работу программы, представлен на рис. 3.

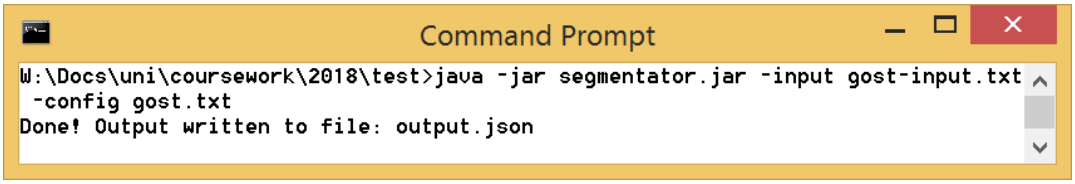

Рис. 3. Пример работы программы в командной строке

Оценка качества. Анализ результатов

Для оценки качества был использован корпус, составленный из 91 затекстовой библиографической ссылки, в которой ручной разметкой было выделено 535 элементов. Для обработки корпуса был использован файл конфигурации для затекстовых библиографических ссылок в формате ГОСТ 7.0.5—2008, правила в котором были составлены с помощью анализа ссылок, сходных по структуре с использованными в тестовом корпусе. Было проведено сравнение элементов ссылок входного файла, выделенных вручную, с элементами, полученными в ходе работы программы выходного json-файла.

Выявлено, что использование программы для сегментации данных ссылок дало результат в 84 распознанных ссылки, содержащих 483 непустых элемента. Анализ распознанных программой ссылок показал, что корректно было распознано 463 элемента, а на уровне полных ссылок – 64 ссылки. При этом корректно распознанными элементами признавались только те, содержание которых полностью совпадало с соответствующими им элементами, размеченными вручную; корректно распознанными ссылками – только те, которые содержат все элементы, присутствующие в ручной разметке соответствующих ссылок, при этом при сегментации не было распознано лишних элементов и все распознанные элементы были распознаны корректно.

Значения метрик полноты (recall), точности (precision) и F-меры представлены в таблице 1.

Таблица 1. Значения метрик для тестового корпуса

	n	p (точность)	r (полнота)	F1-мера
Элементы	535	0,96	0,87	0,91
Полные ссылки	91	0,76	0,70	0,73

Главный результат анализа неверно распознанных элементов и ссылок состоит в том, что большинство причин неверного распознавания могут быть устранены уточнением файлов конфигурации. Так, 7 ссылок, не распознанных алгоритмом вовсе, являются ссылками на электронные ресурсы, доклады на конференциях и не вышедшие

в печать работы; в файле конфигурации не были учтены эти типы ссылок. Из распознанных неверно 17 ссылок 8 (около половины) являются ссылками типа «книга, коллектив редакторов» и содержат элементы «разделитель – одиночная косая черта» и «коллективный редактор»; этот тип также не учтен в файле конфигурации. В других случаях в ссылках содержался URL-адрес (3 ссылки); отсутствовало наименование издательства (2 ссылки); использованы полные личные имена, а не инициалы (*Зализняк Анна А.*); фамилия в имени автора содержала пробелы, а не только дефисы (как в двойных фамилиях) в качестве разделителей (*Ле Гийю де Пенанрос Э.*); содержались не учтённые в файле конфигурации подстроки отдельных полей (*Vol.* в сегменте «том» – помимо *Т.*, *Том*). Все эти ошибки распознавания вызваны дефектами в использованном файле конфигурации и могут быть устранены редактированием существующих и добавлением новых определений.

Привлекает внимание – и представляет больший интерес – немногочисленная группа ошибок, связанных с опечатками в ссылках и их некорректным (не соответствующим стандарту цитирования) форматированием. Предполагается, что файл конфигурации строится для сегментации ссылок, полностью соответствующих нужному формату цитирования; по этой причине, а также поскольку работа алгоритма основана на применении регулярных выражений, ошибки распознавания ссылок, содержащих опечатки, являются закономерными и ожидаемыми.

Заключение

Разработанная система позволяет автоматизировать сегментацию библиографических ссылок в соответствии с заданными пользователем правилами их построения.

Конструкция системы принципиально позволяет осуществлять сегментацию вне зависимости от языка, на котором составлены ссылки, и какого-либо конкретного формата цитирования. В то же время корректность (accuracy) результатов, которая может быть достигнута с использованием этой системы, по большей части напрямую зависит от качества используемых файлов конфигурации.

Следует отметить, что, как выяснилось нами при написании пробных файлов конфигурации, правила достаточно быстро становятся либо многочисленными, либо весьма громоздкими (в зависимости от стиля работы при составлении правил). Как нам представляется, при работе с библиографическими ссылками с использованием регулярных выражений многочисленности и сложности правил невозможно избежать полностью – в частности, из-за их весьма сложной структуры и большого количества опциональных элементов; то же отмечают, например, авторы работы [Колмогорцев, Сараев 2017]. Предлагаемая система, тем не менее, является полезной в рамках данного подхода, поскольку позволяет задавать регулярные выражения преимущественно на уровне терминальных узлов, представляющих элементарные компоненты ссылки, и на их основе строить более сложные компоненты, таким образом, уменьшая затраты труда и времени со стороны пользователя и позволяя ему строить более сложные правила, чем возможно при попытке построения ссылки из регулярных выражений непосредственно, предпринятой нами в [Карпенко 2017] (хотя и для решения задачи, несколько отличной от данной).

Существенным ограничением данной программы является то, что в рамках синтаксиса файлов конфигурации не реализованы в полной мере некоторые характеристики контекстно-независимых грамматик: так, в правой части определений нельзя использовать нетерминалы, содержащиеся в левой части (иными словами, нельзя определять элемент через самого себя). Реализация такой функции позволила бы существенно упростить для пользователя разработку файлов конфигурации и сделать возможным написание намного более сложных правил.

Перспективным для дальнейшей работы в этом направлении представляется совмещение подхода, основанного на обработке ссылок с использованием регулярных выражений, с вероятностными подходами (основанными, например, на скрытых моделях Маркова – HMM). Так, некоторые последовательности символов содержатся в определенных компонентах или на их границах с большей вероятностью, чем в других сегментах ссылки. Машинное обучение модели позволило бы выявить подобные закономерности. Представляется, что совмещение такого подхода с рассмотренным в настоящей работе позволило бы повысить точность определения отдельных компонентов ссылок.

Список литературы

1. ГОСТ Р 7.0.5—2008. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления. – Утв. и введен в действие приказом Ростехрегулирования от 28.04.2008 № 95-ст. М.: Стандартинформ, 2008.
2. Карпенко 2017 – Карпенко К. М. Система автоматической транслитерации библиографических ссылок на русском языке (Курсовая работа). М.: МГУ, 2017.
3. Колмогорцев, Сараев 2017 – Колмогорцев С. В., Сараев П. В. Извлечение библиографии из текстов регулярными выражениями // Новые информационные технологии в автоматизированных системах. 2017. № 20.
4. Jurafsky & Martin 2000 – Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall PTR.
5. Li et al. 2008 – Li, Y., Krishnamurthy, R., Raghavan, S., & Vaithyanathan, S. (2008). Regular Expression Learning for Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, 21—30. Stroudsburg, PA, USA: Association for Computational Linguistics.
6. Manning & Schütze 1999 – Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press.
7. Pratt-Hartmann 2010 – Pratt-Hartmann, I. (2010). Computational Complexity in Natural Language. In A. Clark, C. Fox & S. Lappin (Eds.), *The Handbook of Computational Linguistics and Natural Language Processing* (pp. 429—454). Oxford, UK: Wiley-Blackwell.
8. Grishman 2005 – Grishman, R. (2005). Information Extraction. In R. Mitkov (Ed.), *The Oxford Handbook of Computational Linguistics* (pp. 545—559). Oxford, 2005: Oxford University Press.