

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Кафедра системного программирования

Боровиков Никита Александрович

СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР
БИБЛИОГРАФИЧЕСКОЙ ИНФОРМАЦИИ ДЛЯ
JABREF

Бакалаврская работа

Научный руководитель:
к.ф.-м.н., ст. преп. Луцив Д.В.

Рецензент:
Инженер программист, ООО «Ф-Лайн Софтвер»
Павлов С. С.

Санкт-Петербург

2019

SAINT PETERSBURG STATE UNIVERSITY

Mathematics & Mechanics Faculty

Software Engineering Chair

Borovikov Nikita

JABREF BIBLIOGRAPHIC INFORMATION PARSER

Graduation Thesis

Scientific supervisor:

Ph.D., s.l Luciv D.

Reviewer:

Software Engineer «First Line Software»

Pavlov S.

Saint Petersburg

2019

Оглавление

Оглавление	3
Введение	4
Постановка задачи	6
Литературный обзор	7
Менеджеры библиографической информации	7
Синтаксические анализаторы библиографической информации	9
JabRef	12
BibTeX	12
Архитектура проекта	15
Реализация синтаксического анализатора	17
Особенности реализации	17
Интеграция в систему	20
Интерфейс	22
Апробация	26
Заключение	30
Список литературы	31

Введение

Компьютер является неотъемлемой частью жизни современного человека. В любой сфере деятельности, будь то досуг или работа, мы стараемся максимизировать объем информации представленной в электронном формате. Не важно, идет речь о прогнозе погоды, составлении отчета или составлении меню ресторана — хранение информации в электронном виде значительно упрощает ее хранение, поиск и распространение.

Вышеописанные обстоятельства естественным образом приводят к ежегодному возрастанию числа электронных статей. Благодаря таким сервисам как Google Scholar и Российский Индекс Научного Цитирования, поиск нужных статей стал достаточно тривиальной задачей. Однако в случаях, когда человек занимается написанием научной работы, перед ним, кроме прочего, встает задача изучения значительного числа работ других авторов. Для упрощения этой задачи и частичной автоматизации управления множеством библиографических ссылок, обычно используются менеджеры библиографии.

Менеджер библиографии — это система, позволяющая пользователям создавать и переиспользовать библиографические ссылки. После создания ссылки ее можно использовать при формировании списков литературы. Кроме этого, система предоставляет удобный доступ к сохраненным статьям, что значительно упрощает работу с ними. Наиболее популярными вариантами являются Zotero, Mendeley, Citavi и JabRef. У каждого есть как преимущества так и недостатки.

Имея возможность удобного хранения библиографической информации, хочется также не тратить время на создание библиографической ссылки путем ручного заполнения полей. Этого можно избежать, если ресурс, содержащий работу, на которую человек пытается создать ссылку, предоставляет возможность извлечь метаданные, однако в случае, когда ссылка представлена простым текстом, этот вариант неприменим.

Отсюда родилась идея внедрения в менеджер библиографической информации синтаксического анализатора для автоматизации процесса преобразования произвольной библиографической ссылки из простого текста в формат, поддерживаемый менеджером.

Таким образом, цель данной дипломной работы состоит в интеграции синтаксического анализатора библиографической информации в менеджер библиографии JabRef.

Постановка задачи

Целью данной работы является внедрение синтаксического анализатора в систему управления библиографической информацией JabRef, для приведения библиографических ссылок в произвольном формате к формату BibTeX.

Для достижения цели были выделены следующие основные задачи:

1. Ознакомиться со структурой JabRef и изучить методы интеграции расширений
2. Реализовать синтаксический анализатор в JabRef
3. Провести апробацию проделанной работы

Литературный обзор

Менеджеры библиографической информации

Существует более десяти сервисов, предназначенных для управления библиографией, однако лишь несколько из них подходят для выполнения поставленных задач в силу отсутствия в открытом доступе исходного кода. В данной главе будут разобраны наиболее популярные библиографические менеджеры согласно системе Google Trends.

Zotero

Один из наиболее популярных менеджеров и один из немногих, удовлетворяющих условиям популярности и открытого исходного кода. Написан на JavaScript. Обладает широким спектром возможностей[2]:

- Хранение данных в облаке. 300 МБ для бесплатной версии и неограниченное в платной
- Русскоязычный интерфейс
- Поддержка расширения для браузера
- Доступны библиографические стили AAA, APA, Chicago, IEEE, Harvard, MLA, Vancouver и другие
- Работа без подключения к интернету.

Mendeley

Еще один популярный менеджер библиографии, однако без открытого исходного кода. Обладает следующими характеристиками[3]:

- Возможность загружать статьи непосредственно с таких платформ, как Science Direct
- Отслеживание новых публикаций выбранных авторов
- Доступны библиографические стили AAA, APA, Chicago, IEEE, Harvard, MLA, Vancouver и другие
- Мобильный клиент
- Внутренняя база данных проекта
- Нет возможности работать без подключения к сети.

Citavi

Как и Mendeley — является проприетарной программой и находится в списке только для более полного освещения возможностей подобных программ. Широко распространен в Германии, где используется по университетским лицензиям в большей части университетов. Важными аспектами являются[4]:

- Расширения для Mozilla Firefox и Google Chrome, а также Adobe Reader и Adobe Acrobat
- Ограничение бесплатной версии в 100 библиографических записей на базу.
- Поддержка только платформы Windows
- Возможность работать с собственными заметками

JabRef

Второй из рассматриваемых менеджеров с открытым исходным кодом. Написан на Java. По спектру возможностей схож с Zotero, однако обладает рядом отличий[5]:

- Использует BibTeX в качестве нативного формата, однако поддерживает импорт и экспорт во многие другие форматы
- Поддерживает расширение для браузера
- Не имеет облачного хранилища
- Не имеет ограничений по количеству создаваемых библиографических записей

Сравнение

В силу того, что поставленные ограничения свели выбор менеджера библиографии к двум относительно схожим вариантам, выбор был сделан исходя из языка программирования. Так было принято решение внедрить синтаксический анализатор в JabRef.

Синтаксические анализаторы библиографической информации

Есть три основных подхода к решению задачи синтаксического анализа библиографической ссылки. Регулярные выражения, правила и обучаемые алгоритмы.

Регулярные выражения — формальный язык поиска и редактирования подстрок внутри исходной строки. В его основе лежит использование метасимволов(символов, соответствующих другим символам, множествам символов или символьным последовательностям). Регулярное выражение

строится из шаблона, включающего в себя терминалы — обычные символы — и нетерминалы — метасимволы.

Решение заключается в поиске во входной строке подстрок соответствующих заранее заданным шаблонам. Найденные соответствия в случае корректного сопоставления и будут являться искомыми сущностями.

Алгоритмы основанные на правилах не имеют принципиальных структурных отличий относительно алгоритмов на регулярных выражениях и по сути своей являются просто более удобочитаемыми версиями первых.

Одним из рассмотренных решений был Citation-Parser[6]. Эта основанная на правилах программа на языке Python предназначена для определения следующих сущностей: author, title, journal, pages, publisher, year.

Данное решение является весьма простым и преследует цель определения нескольких строго заданных форматов. Например, в библиографии должен быть строгий порядок сущностей. Автор не может идти после года, а год после заголовка. Все решение основывается на предположении, что порядок заполнения является правильным, и сводится к проверке каждого слова в строке на соответствие некоторым признакам, свойственным той или иной сущности.

Эндрю Маккалума (Andrew McCallum) в своей статье Information Extraction[7], для выполнения задачи по парсингу библиографии используется метод основанный на скрытых марковских моделях.

Скрытая марковская модель — это модель, похожая по своему принципу работы, на марковский процесс с неизвестными параметрами, и цель которой — предсказывание этих неизвестных параметров.

В рамках данной работы было принято решение использовать парсер на основе регулярных выражений и правил. Благодаря этому можно будет получить относительно неплохо работающий алгоритм, сосредоточившись на решении задачи интеграции парсера в проект.

JabRef

При работе над проектом с открытым исходным кодом крайне важно досконально изучить его особенности в силу того, что над проектом ведет работу множество не связанных друг с другом людей. Для того, чтобы внесенные изменения не нарушали общую работу системы, был проведен глубокий анализ проекта. В этой главе будут описаны основные особенности, так или иначе повлиявшие на реализацию.

BibTeX

Основным форматом библиографических ссылок, используемым в JabRef является формат BibTeX. В разрезе данной работы нас интересуют библиографические базы, а именно — используемый формат записей. В JabRef реализованы все стандартные типы записей. В таблице 1 представлен список всех записей используемых в JabRef с указанием обязательных и дополнительных полей, распознаваемых стандартными программами-обработчиками.

Тип записи	Описание	Обязательные поля	Дополнительные поля
article	Статья из журнала	author, title, journal, year	volume, number, pages, month, note, key
book	Определенное издание книги	author/editor, title, publisher, year	volume, series, address, edition, month, note, key, pages
booklet	Печатная работа, без издателя или организатора	title	author, howpublished, address, key, month, year, note
conference	Синоним inproceedings, оставлено для совместимости с Scribe	author, title, booktitle, year	editor, pages, organization, publisher, address, month, note, key
inbook	Часть книги, возможно без названия. Например, глава, диапазон страниц	author/editor, title, chapter/pages, publisher, year	volume, series, address, edition, month, note, key
incollection	Часть книги, имеющая собственное название	author, title, booktitle, year	editor, pages, organization, publisher, address, month, note, key
inproceedings	Тезис (труд) конференции.	author, title, booktitle, year	editor, series, pages, organization, publisher, address, month, note, key
manual	Техническая документация.	title	author, organization, address, edition, month, year, note, key
mastersthesis	Магистерская диссертация.	author, title, school, year	address, month, note, key
misc	Использовать, если другие типы не подходят.	-	author, title, howpublished, month, year, note, key

Тип записи	Описание	Обязательные поля	Дополнительные поля
phdthesis	Кандидатская диссертация.	author, title, school, year	address, month, note, key
proceedings	Сборник трудов (тезисов) конференции.	title, year	editor, publisher, organization, address, month, note, key
techreport	Отчёт, опубликованный организацией	author, title, institution, year	type, number, address, month, note, key
unpublished	Документ, имеющий автора и название, но формально не опубликованный	author, title, note	month, year, key

Таблица 1: Формат записей

Каждая запись содержит набор полей, далее приведен список стандартных полей, однако использовать можно произвольный набор символов, который будет просто игнорироваться стандартными программами.

- address: Адрес издателя (обычно просто город, но может быть полным адресом для малоизвестных издателей)
- annotate (в JabRef — abstract): Аннотация для библиографической записи.
- author: Имена авторов (если больше одного, то разделяются and)
- booktitle: Наименование книги, содержащей данную работу.
- chapter: Номер главы
- crossref: Ключ кросс-ссылки (позволяет использовать другую библио-запись в качестве названия, например, сборника трудов)
- edition: Издание
- editor: Имена редакторов (оформление аналогично авторам)

- eprint: Спецификация электронной публикации, обычно используется в технических отчетах
- howpublished: Способ публикации, если нестандартный
- institution: Институт, вовлеченный в публикацию
- journal: Название журнала, содержащего статью
- key: Скрытое ключевое поле, задающее порядок сортировки (если «author» и «editor» не заданы).
- month: Месяц публикации (может содержать дату). Если не опубликовано — создания.
- note: Любые заметки
- number: Номер журнала
- organization: Организатор конференции
- pages: Номера страниц, разделённые запятыми или двойным дефисом. Для книги — общее количество страниц.
- publisher: Издатель
- school: Институт, в котором защищалась диссертация.
- series: Серия, в которой вышла книга.
- title: Название работы
- type: Тип отчёта, например «Заметки исследователя»
- url: WWW-адрес
- volume: Том журнала или книги.
- year: Год публикации (если не опубликовано — создания)

Архитектура проекта

Проекты с открытым исходным кодом имеют некоторые особенности. В силу того, что любой желающий может вносить изменения, могут возникнуть сложности с качеством кода и проработанностью архитектуры. Для развивающихся проектов без строгого контроля это может вылиться в отсутствие четкой структуры, наличие устаревшего кода и различия в стилях написания кода.

Все эти проблемы накапливаясь приводят к значительному усложнению понимания кода, особенно для новых участников. В данный момент в JabRef происходит реструктуризация архитектуры и приведение ее к принципам MVC[8].

Кроме основных модулей, отвечающих за модель, логику и графический интерфейс, в проекте также выделяются модули для сохранения конфигурационных параметров и консольный интерфейс[9]. Все классы, которые еще не приведены к этому формату по умолчанию относятся к модулю графического интерфейса. На рис. 1 показана схема взаимодействия модулей.

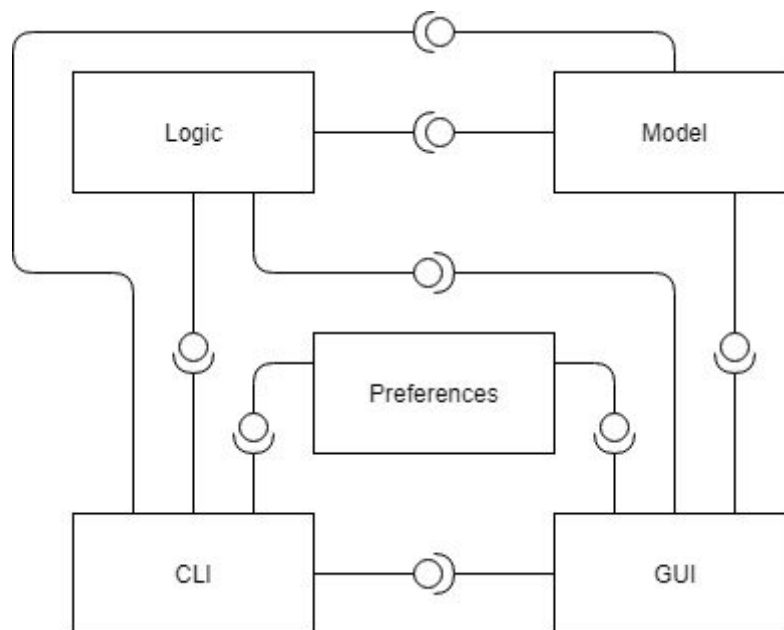


Рис. 1: Взаимодействие модулей

Реализация синтаксического анализатора

Особенности реализации

В качестве синтаксического анализатора было принято решение использовать модернизированную версию из курсовой работы. Подход состоит из нескольких шагов.

В первую очередь производится предобработка строки. Удаляются переносы, лишние пробелы и производятся другие незначительные изменения для минимизирования зашумленности текста.

Вторым шагом идет последовательный поиск и замена однозначно определяемых частей библиографической ссылки — таких как автора, год, страницы и прочее — на теги. Вся найденная информация вместе с этим приводится к единому формату и заносится в соответствующие поля формируемой библиографической записи.

Последним шагом оставшаяся информация дробится на части, каждая из которых в исходном тексте была разделена знаком препинания. Исходя из их порядка, длины и состава и положения относительно тегов, делается предположение, о том, чем является эта часть.

Далее приведено описание реализации поиска некоторых полей.

Авторы

Поиск каждого автора производится отдельно, с условием, что между авторами должен быть символ разделитель. Это может быть запятая или союз. Формат записи, в котором участвует фамилия и инициалы достаточно просто распознать, но в случае использования полной формы имени приходится делать допущение, что имя автора указано с большой буквы. В противном случае, найденное сочетание слов может относиться уже к заголовку работы. Каждый автор заменяется тегом, после чего на третьем шаге все цепочки найденные между тегами авторов, также классифицируются как авторы, если удовлетворяют минимальным ограничениям по количеству слов и формату записи.

Дата

Поиск даты также является достаточно тривиальной задачей. Год будем рассматривать, как четырехзначное число, больше 1800 и не позже текущего года +1, для того чтобы не пропустить предстоящие публикации. Месяц и день либо указаны рядом с годом либо не указаны. Достаточно проверить на наличие рядом с годом двух чисел в диапазоне 1-31 и 1-12 соответственно с типичными для дат разделителями. Кроме того возможен вариант разделения пробелом, но тогда месяц должен быть написан словами, что не является проблемой в силу того, что список возможных месяцев также заранее известен. Помимо прочего учтена возможность отсутствия дня в случае если месяц указан словом.

Страницы

Из оставшихся элементов, многосимвольными числами с достаточно высокой вероятностью являются страницы или номера выпусков. При этом страницы практически всегда указаны последними из этих двух. Предусмотрена возможность записи как одного числа так и диапазона.

Ссылка

Отдельно стоит упомянуть возможность наличия url'a в библиографической ссылке. Определить его наличие достаточно просто можно проверкой на наличие в строке протокола и прямых слэшей. Однако правильно распознать всю ссылку от начала до конца — куда менее тривиальная задача.

Остальные поля нельзя однозначно определить посредством регулярных выражений, но можно сделать вывод исходя из уже полученных данных.

Так заголовок скорее всего находится сразу после авторов и заканчивается первой запятой или точкой. В некоторых форматах в качестве разделителя заголовка и остальной части ссылки используется двойной прямой слэш, что также помогает идентифицировать это поле.

Следующим элементом с высокой вероятностью является издатель или журнал. Отличить друг от друга их можно по двум косвенным признакам: наличию(отсутствию) слова журнал и наличию(отсутствию) еще одного элемента библиографии. Если такой есть, то это почти наверняка город, а значит предыдущий пункт скорее является издателем и наоборот.

От выбора между журналом и издателем зависит также тип создаваемой BibTeX сущности — article и book соответственно.

Интеграция в систему

Реализация синтаксического анализатора сама по себе не является необходимым результатом. Основной задачей является внедрение полученного решения в систему.

В процессе изучения JabRef было выяснено, что внешние плагины больше не поддерживаются. Начиная с версии 3.0 команда разработчиков JabRef решила отказаться от поддержки плагинов в связи с тем, что для их реализации использовался Java Plug-in Framework, на данный момент являющийся устаревшим. В качестве решения было предложено два варианта. Первый — использовать альтернативное программное обеспечение Jigsaw. Однако, для стабильной работы Jigsaw необходимо использование Java 9, что не является возможным в текущей реализации. Вторым предложенным вариантом была интеграция подключаемых модулей внутрь основной системы. В связи с этим архитектурным решением, всем желающим расширить функциональность следует вносить изменения непосредственно в сам проект.

Изменения на уровне модели отслеживаются на остальных уровнях благодаря шаблону проектирования EventBus, который в данном случае реализуется через использование класса TelemetryClient, относящегося к службе Azure Application Insights.

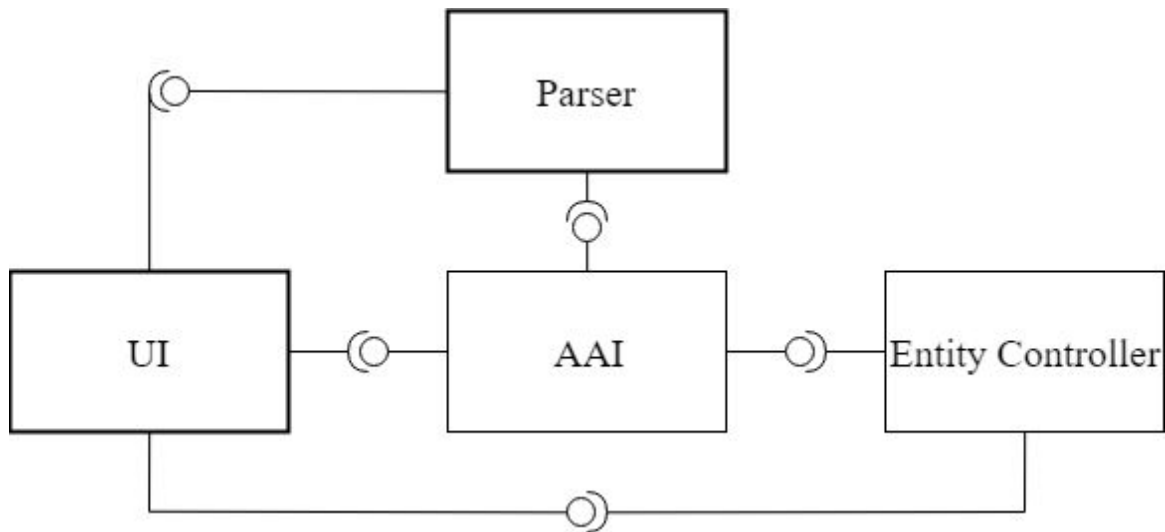


Рис. 2: Настройка связи модулей через TelemetryClient

Application Insights — это расширяемая служба управления производительностью приложений. В данном случае используется для отслеживания событий на уровне модели и отправляющая их на другие модули, что позволяет уменьшить зависимость уровней друг от друга.

После интеграции синтаксического анализатора на уровне логики и настройки связи новой сущности с остальными модулями через TelemetryClient, появилась необходимость имитировать последствия нажатия пользователем кнопки New Article, нажатие которой вызывает логику создания новой сущности типа Article. Однако в нашем случае вместо новой сущности необходимо было подставлять сущность уже созданную нами — правильного типа и с заполненными полями после работы синтаксического анализатора.

В целях избежать дублирования кода, было принято решение изменить логику создания новой сущности, вынеся некоторые методы на более абстрактный уровень. В силу неполной структурированности кода это потребовало внесения изменений в еще несколько не связанных с задачей классов для сохранения стабильности работы.

Последней вставшей задачей было реализация графического интерфейса. Для удобства пользователя и сохранения интуитивной понятности интерфейса было принято решение использовать простое окно для ввода текста с условием, что для перехода к следующему шагу, поле должно быть не пустым.

Еще одним шагом, призванным повысить удобство использования было внесение исходной строки введенной пользователем и строки прошедшей синтаксический анализатор в поле комментария к библиографической сущности. Это позволит быстро проставить поля, которые парсер не смог распознать и так же быстро поправить ошибки, если распознавание прошло не верно.

Интерфейс

Далее приведено пошаговое представление графического интерфейса с описанием выполняемых действий.

На Рис. 3 мы видим стандартную форму для заполнения полей новой сущности в ручном режиме.

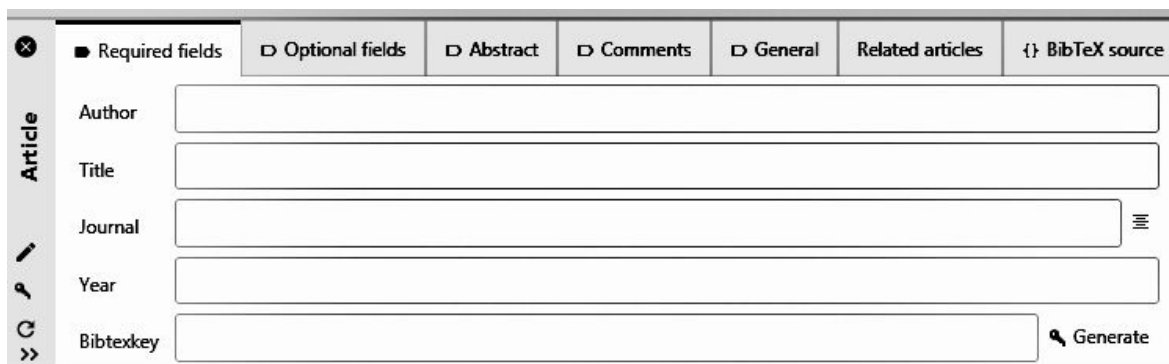


Рис. 3: Стандартное окно новой сущности типа Article

Для доступа к инструменту автоматического извлечения полей необходимо в разделе Tools панели инструментов выбрать пункт Extract BibTeX from plain text, как показано на Рис. 4.

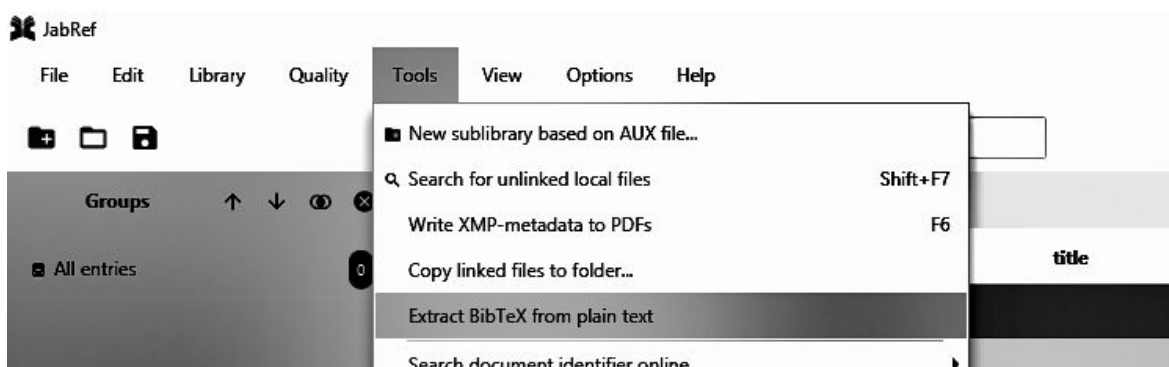


Рис. 4: Новый пункт в инструментах

После этого откроется окно с полем для ввода текста, неактивной кнопкой Extract и кнопкой Cancel, которое можно увидеть на Рис. 5.1.

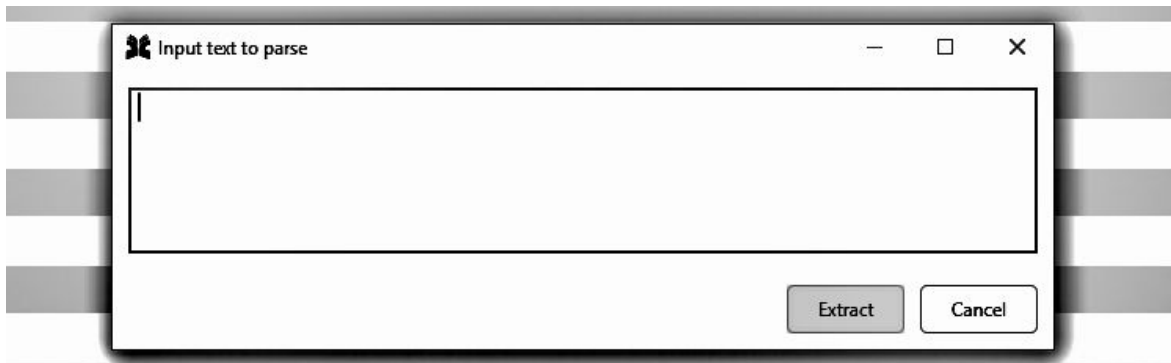


Рис. 5.1: Окно для ввода текста библиографической ссылки

На Рис. 5.2 показано, как после ввода текста, кнопка Extract становится активной. Ее нажатие приведет к вызову логики синтаксического анализа и извлечения библиографических полей из введенного текста.

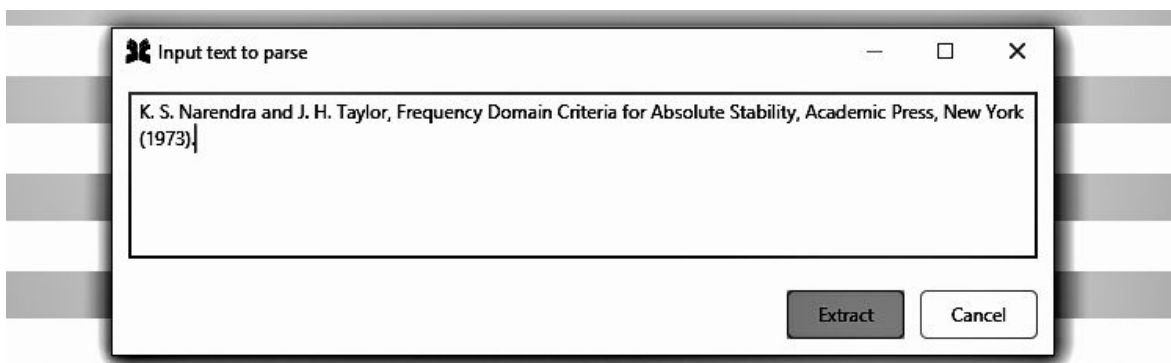


Рис. 5.2: Заполненное окно для ввода

В приведенном примере парсер обнаружил 2 текстовые подстроки помимо заголовка. Что, по вышеописанной логике преобразует сущность из Article в Book, как показано на Рис. 6.

The screenshot shows a web interface for entering book information. At the top, there are tabs for 'Required fields', 'Optional fields', 'Abstract', 'Comments', 'General', 'Related articles', and 'BibTeX source'. The 'Required fields' tab is active. On the left, a vertical sidebar contains the word 'Book' and several icons. The main form area contains the following fields:

- Title: Frequency Domain Criteria for Absolute Stability
- Publisher: Academic Press
- Year: 1973
- Author: Narendra, K. S. and Taylor, J. H.
- Editor: (empty)
- Bibtexkey: (empty)

A 'Generate' button is located at the bottom right of the form.

Рис. 6: Заполненные поля сущности Book

Для удобства исправления ошибок, допущенных синтаксическим анализатором, исходный текст вместе с версией с тегами приводится в поле Comment. Это позволяет просто перераспределять вставленные значения полей. Рис. 7.

This screenshot shows the 'Comment' field of the book entry form. The field contains the following text:

```
[author_tag] [author_tag] [title_tag] [publisher_tag] [address_tag] [year_tag]
K. S. Narendra and J. H. Taylor, Frequency Domain Criteria for Absolute Stability, Academic Press, New York (1973).
```

The 'Comments' tab is selected at the top of the interface.

Рис. 7: Заполненное поле Comment

Апробация

Данная система была апробирована с использованием библиографических ссылок различных форматов с искусственным внедрением зашумления.

Для тестов на вход программе подаются правильно оформленные библиографические ссылки. После этого происходит от одного до четырех случайных изменений из списка:

- К произвольному символу справа добавляется перенос строки,
- У произвольного символа меняется регистр,
- Случайный символ удаляется,
- Случайный знак препинания из множества (.,|/) заменяется на другой из того же множества.

Описание примеров:

- Поле отмечено зеленым, если программа распознала его верно.
- Поле отмечено желтым, если программа распознала его не полностью или эта часть осталась нераспознанной.
- Поле отмечено красным, если программа распознала его неверно.

Далее приведен краткий список с примерами, подаваемых алгоритму библиографических ссылок и полученный результат.

Input:

K. S. Narendra and J. H. Taylor, Frequency Domain Criteria for Absolute Stability, Academic Press, New York (1973).

Output:

```
@Book{  
  title   = {Frequency Domain Criteria for Absolute Stability},  
  publisher = {Academic Press},  
  year    = {1973},  
  author  = {Narendra, K. S. and Taylor, J. H.},  
  address = {New York},  
}
```

В первом примере все поля распознаны верно.

Input:

Lambert Diane, Liu Chuanhai. Adaptive Thresholds: Monitoring Streams of Network Counts Online // Journal of the American Statistical Association. — 2006. — no. 101. — P. 78–88.

```
@Article{  
  author = {Lambert, Diane and Liu, Chuanhai},  
  title  = {Adaptive Thresholds: Monitoring Streams of Network Counts Online},  
  journal = {Journal of the American Statistical Association},  
  year   = {2006},  
  pages  = {78–88},  
  comment = {— no. 101. — P.},  
}
```

Во втором примере, все основные поля распознаны верное, однако осталась нераспознанная информация.

Input:

G. Leonov and N. Kuznetsov, Nonlinear Mathematical Models of Phase-Locked Loops. Stability and Oscillations. Cambridge Scientific Publisher, 2014.

Output:

```
@Book{  
  title = {Nonlinear Mathematical Models of Phase-Locked Loops},  
  publisher = {Stability and Oscillations},  
  year = {2014},  
  author = {Leonov, G. and Kuznetsov, N.},  
  address = {Cambridge Scientific Publisher},  
}
```

В третьем примере из-за наличия точки в заголовке, часть его была принята за отдельно поле. Из-з этого смещения, часть заголовка была распознана как publisher и часть как adress, что привело к значительному искажению.

Апробация проводилась на 100 корректных библиографических ссылках с искусственным зашумлением. Для каждой ссылки изначально были размечены присутствующие в ней поля. Далее в качестве меры используется расстояние Левенштейна:

- Каждое найденное поле считается верно положительным результатом, если соответствует истине не менее, чем на 90%.
- Результат считается ложно положительным, если соответствует истине менее чем на 90% или содержит данные из других полей.
- Верно отрицательным считается полностью пустое поле, в случае его отсутствия в исходном тексте.

Метрики результатов приведены на Рис. 8.

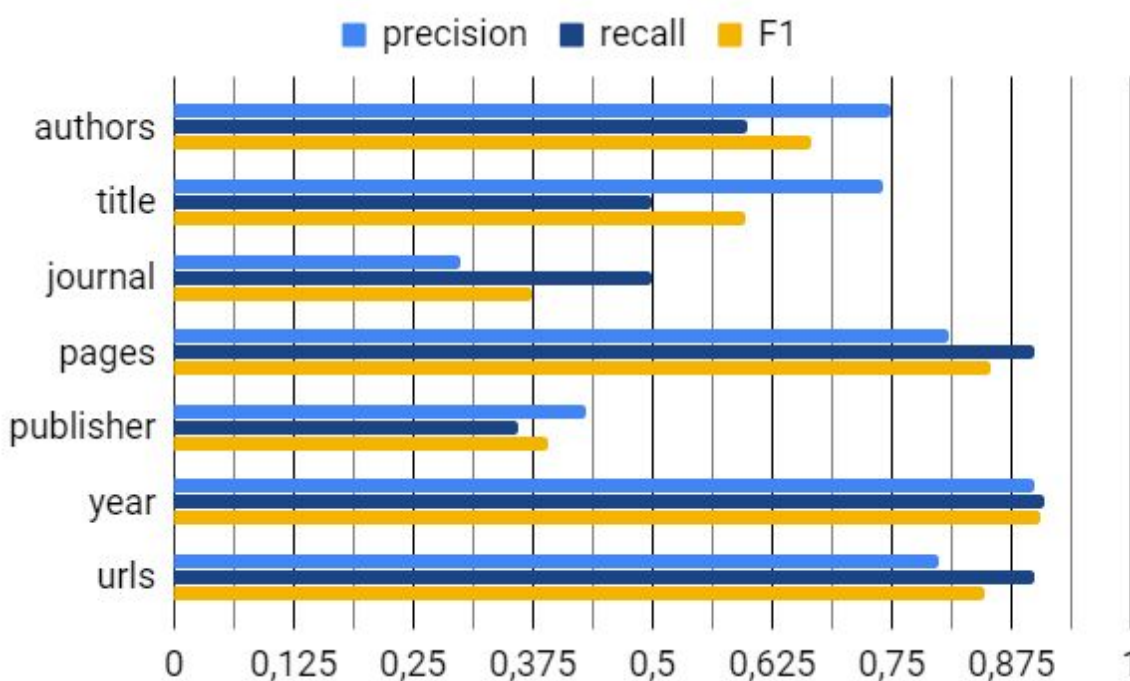


Рис. 8: Результаты апробации

Заключение

В рамках данной дипломной были получены следующие результаты:

1. Изучена система JabRef. Разобраны принципы работы с библиографической информацией, используемые форматы данных. Найдены способы добавления расширений в систему.
2. На основе текущей реализации произведена интеграция синтаксического анализатора в JabRef. Добавлена возможность автоматического создания библиографической ссылки в формате BibTeX с извлечением данных из простого текста. Создан Pull Request в репозиторий проекта.
3. Произведена апробация проделанной работы на библиографических ссылках различного формата с добавлением и без добавления искусственного зашумления.

Список литературы

- [1] Коршунов О. П. Библиографическая информация как научное понятие Советская библиография. 1985. С. 31-42.
- [2] Zotero: <https://zotero.org> (online; accessed: 01.02.2019)
- [3] Henning, V., Reichelt, J. (2008). "Mendeley - A Last.fm For Research?". IEEE Fourth International Conference on eScience. 7 (12): 327–328
- [4] Citavi: <https://www.citavi.com/en/products> (online; accessed: 01.02.2019)
- [5] Feyer S., Siebert S., Gipp B., Aizawa A., Beel J. (2017) Integration of the Scientific Recommender System Mr. DLib into the Reference Manager JabRef. In: Jose J. et al. (eds) Advances in Information Retrieval. ECIR 2017. Lecture Notes in Computer Science, vol 10193. Springer, Cham
- [6] Citation-Parser: <https://github.com/manishbisht/Citation-Parser> (online; accessed: 01.02.2019)
- [7] Andrew McCallum Information Extraction: Distilling Structured Data from Unstructured Text, Queue - Social Computing, Volume 3 Issue 9, November 2005, Pages 4
- [8] Сергей Рогачев, Обобщенный Model-View-Controller, rsdn.org, 2007
- [9] JabRef Documentation:
<https://github.com/JabRef/jabref/wiki/High-Level-Documentation> (online; accessed: 01.02.2019)
- [10] D. Tkaczyk, A. Collins, P. Sheridan, J. Beel. Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers. In Proceedings of ACM JCDL, 2018, USA.