www.volsu.ru

МОДЕЛИРОВАНИЕ,
ИНФОРМАТИКА И УПРАВЛЕНИЕ

DOI: https://doi.org/10.15688/mpcm.jvolsu.2019.4.4

УДК 004.91, 81'33, 004.42
ББК 32.973, 81.1

Дата поступления статьи: 02.07.2019
Дата принятия статьи: 25.10.2019

АВТОМАТИЗАЦИЯ МОРФОЛОГИЧЕСКОЙ РАЗМЕТКИ АРХИВНЫХ ДОКУМЕНТОВ [1]

Анатолий Сергеевич Комендантов

Студент института математики и информационных технологий,
Волгоградский государственный университет
anatoly.komendantov@gmail.com, matf@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Александр Георгиевич Матвеев

Студент института математики и информационных технологий,
Волгоградский государственный университет
sna4es@gmail.com, matf@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Андрей Владимирович Светлов

Кандидат физико-математических наук, доцент кафедры математического анализа и теории функций,
Волгоградский государственный университет
a.v.svetlov@gmail.com, andrew.svetlov@volsu.ru, matf@volsu.ru
https://orcid.org/0000-0002-8764-6132
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Аннотация. Работа посвящена описанию созданной авторами статьи надстройки над утилитой для стемминга MyStem И. Сегаловича. Приложение добавляет к возможностям утилиты удобный графический интерфейс, простой для освоения и интуитивно понятный пользователям, не специализирующимся в информационных технологиях. Оно перехватывает вывод утилиты MyStem, специальным образом переформатирует и анализирует его. Кроме того, приложение имеет функционал для снятия омонии вручную, если автоматическими средствами морфологические характеристики слова определены

неверно. Основное назначение данного приложения — подготовка морфологической разметки документов архивного фонда «Михайловский станичный атаман» для создания лингвистического корпуса. В ходе работы над приложением была решена задача корректной обработки текстов, содержащих устаревшие кириллические символы.

Ключевые слова: автоматизация лингвистического анализа, автоматизация морфологического анализа, утилита MyStem, графический интерфейс, программная оболочка, корпусная лингвистика.

Введение

В наши дни до сих пор существует большое количество исторически ценных документов, которые не были включены в научный оборот. Такие документы, содержащие ценные для истории русской культуры и русского языка сведения, в основном хранятся в архивах и не используются широкой научной общественностью из-за их малой доступности и отсутствия качественных инструментов для их автоматизированного анализа. Таким примером являются документы фонда «Михайловский станичный атаман» (1734–1836 гг.), хранящиеся в Государственном архиве Волгоградской области.

Данные документы были написаны с использованием скорописи первой половины XVIII в., для которой не существует компьютерных средств для распознавания. Коллектив ученых ВолГУ под руководством профессора О.А. Горбань провел транслитерацию текста, при этом была сохранена орфография оригинала: соблюдены выносные буквы строка в строку (буквы, которые пишутся над строкой, хотя и составляют часть слова, остающегося в строке), сохранен титл (надстрочный знак над сокращенно написанным словом или над буквой, обозначающей цифру), все надстрочные и подстрочные символы, все пометки на полях были перенесены в сноски (см.: [1; 2; 4; 5]). Следующим этапом на пути к автоматизированному анализу в аспекте корпусной лингвистики стала адаптация текста: выносные буквы даны в строку, титла раскрыты, диграфы устранены, добавлены пробелы после предлогов. В нынешнем виде эти документы стали пригодны для компьютерной обработки без необходимости создания специализированных средств, корректно работающих, например, с выносными буквами и титлами — можно использовать существующие инструменты для обработки текстов, нужно только убедиться, что они адаптированы для устаревшей лексики и графики. Это существенно более простая задача, описанию решения которой и посвящена данная статья.

1. Постановка задачи

Как уже было замечено выше, основной целью данного проекта является создание корпуса документов фонда «Михайловский станичный атаман». Методы корпусной лингвистики представляются наиболее оптимальными в данном случае, так как предполагают обработку большого количества текстов с целью решения самых разнообразных лингвистических задач. Наша группа присоединилась к коллективу филологов для обеспечения технической и программной части проекта. Главной задачей для нас является создание (или адаптация существующего) «движка» корпуса документов, то есть программного обеспечения, решающего задачи хранения базы данных размеченных текстов, выполнения запросов к этой базе, а также предоставления пользователям удобного

интерфейса для работы, не требующего специальной квалификации в области информационных технологий. Однако вне зависимости от выбранного или созданного «движка» все тексты перед их включением в корпус должны пройти специальную подготовку, а точнее — разметку. От «движка» будет зависеть лишь формат представления этой разметки, а следовательно, решать задачи разметки и разработки программного обеспечения корпуса можно параллельно. Настоящая статья посвящена вопросу автоматизации разметки архивных документов.

Существуют следующие типы разметки текстов:

– Морфологическая разметка: указание такой информации, как часть речи, род, число, падеж, наклонение и прочее.

– Синтаксическая разметка: разбор того, как связаны между собой слова в предложении и какую синтаксическую роль они имеют.

– Семантическая разметка: анализ текста и слов в нем на основе контекста.

– Анафорическая разметка: в какой-то степени смысловой анализ текста. Состоит в поиске повторений звуков, слов и их частей.

– Просодическая разметка: анализ ударений и интонаций в тексте, зачастую сопровождается дискурсной разметкой, показывающей паузы, оговорки, повторы и т. д.

По мере движения от начала к концу данного списка увеличивается участие человека в подготовке разметки рассматриваемого типа. Далее речь пойдет только о морфологической разметке, которую современными программными средствами можно осуществлять практически полностью автоматически — основная функция исследователя здесь будет заключаться в верификации результата, так как в случае наличия омонимии приложение может ошибиться в выборе наиболее вероятного варианта. После анализа необходимых функций разрабатываемой утилиты получился следующий список требований к ней:

- простой, удобный и понятный интерфейс;
- корректная работа с текстами на русском языке, в том числе с устаревшей лексикой и графикой;
- возможность сохранить результаты анализа в удобном формате;
- возможность выгрузки обработанных текстов в автоматизированную систему хранения данных.

Существует множество различных программ для автоматизации морфологического анализа. Однако лишь некоторые из них корректно работают с текстами на русском языке, единицы имеют графический интерфейс, и ни одна не смогла корректно обработать текст войсковой грамоты из упомянутого архива. Подробнее о функциональных возможностях и сравнительном анализе стеммеров см. в [3].

Таким образом, при всем современном многообразии программного обеспечения не существует таких систем для разметки текстов, которые имели бы графический интерфейс и позволяли производить автоматизированный морфологический анализ текстов на русском языке. Также существующие продукты не поддерживают устаревшие кириллические символы. Например, морфологический анализатор MyStem игнорирует буквы зело, омега, и краткое и другие и принимает их за пробелы. Из-за этого неверно определяет начальную форму слова и грамматическую информацию. При этом оказалось, что

MyStem корректно обрабатывает устаревшую лексику, если она передается в программу с использованием современной кириллицы.

Программа MyStem была разработана Ильей Сегаловичем и впервые представлена в 1998 году. В основе программы используется метод поиска основы слова. На первом шаге происходит «отсечение» окончаний и проверка в дереве основ и словаре. Если слово не словарное, происходит переход к следующему шагу и повторная проверка. Кроме того, на каждом шаге строится гипотеза о возможном изменении слов, и если она была построена ранее, ее вес увеличивается. Когда в результате «отсечения» в основе остается всего 2 символа, гипотезы ранжируются в порядке убывания веса, а если вес гипотезы в 5 или более раз меньше самого большого веса, она отсеивается. Так же последние версии MyStem могут выводить всю грамматическую информацию, и, если такая информация имеется, частоту использования слова в IPM (instances per million) [6]. При этом программа не имеет визуального интерфейса, поэтому не слишком удобна для использования исследователями, не имеющими навыков работы с командной строкой.

Таким образом, было принято решение разработать собственное приложение как надстройку над MyStem. Помимо удобного интерфейса наша программа имеет опцию для работы с устаревшей кириллицей, при включении которой буквы зело, омега, и краткое и другие заменяются на «кс», «о» и «и» соответственно, и только после этого текст передается для анализа в MyStem, а затем в обработанном документе производится обратная замена символов. Также поддерживается экспорт размеченного текста во внешнюю систему хранения и обработки данных. Для реализации графического интерфейса была использована платформа JavaFX (OpenJFX) на основе Java для создания приложений, содержащая достаточно инструментов для разработки красивого и при этом функционального и удобного интерфейса. Во время работы непосредственно с самой платформой используется декларативный язык программирования JavaFX Script.

2. Описание принципов работы приложения и алгоритмов

Интерфейс разработанного приложения стандартный для всех Windows-приложений, и потому интуитивно понятен. Открытие файла производится привычным образом, после чего в главное окно выводится уже размеченный текст. При работе с документами, содержащими устаревшие кириллические символы, следует выбрать соответствующую опцию в меню Правка → Настройки. Однако, чтобы увидеть морфологическую информацию в размеченном тексте, нужно либо переключиться на вкладку табличного представления документа, либо поставить галочку «Морфологич. информация» в нижней части окна. Кроме того, дополнительное меню отображается при нажатии на правую кнопку мыши над выбранным словом — так можно увидеть список альтернативных вариантов морфологического разбора и в случае ошибки выбрать в качестве основного другой вариант или задать собственный. Таким образом решается одна из главных проблем при автоматизированной подготовке текстов для корпуса — снятие омонимии. К сожалению, во многих случаях в русском языке полная автоматизация данного процесса невозможна, и необходимо вмешательство пользователя. Описываемое приложение, таким образом, предлагает удобный интерфейс устранения омонимии вручную (см. рис. 1–3).

Если говорить непосредственно о внутренней логике программы, то при включении опции преобразования устаревшей кириллицы после выбора файла происходит сохранение исходного текста в два ArrayList<String> по слову в элемент. Затем в одном из них

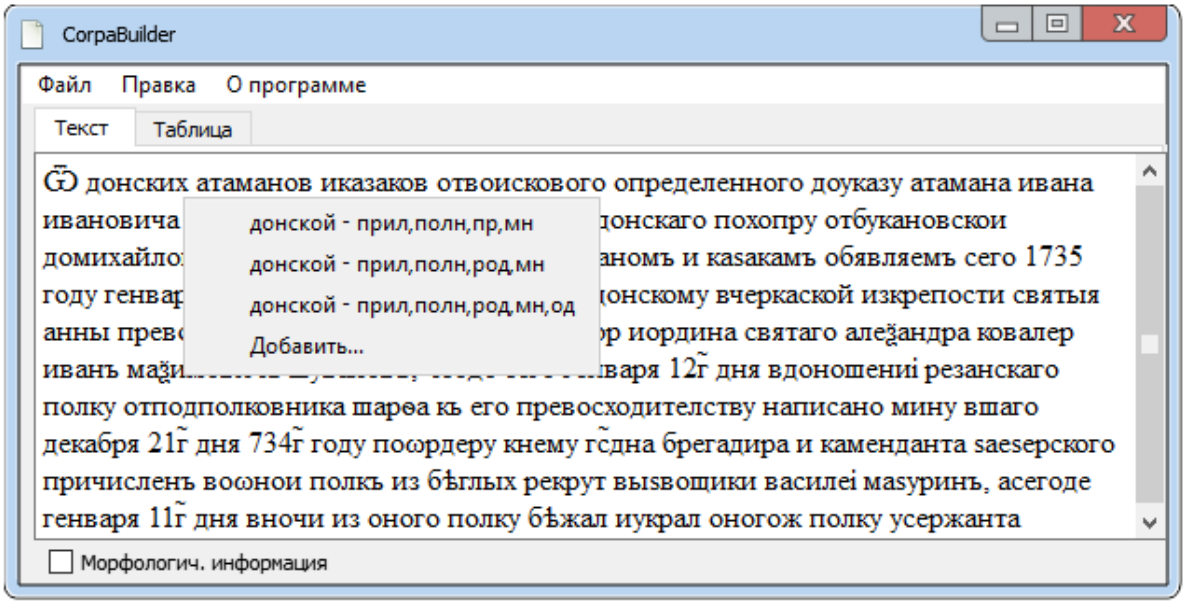

Рис. 1. Интерфейс снятия омонимии вручную

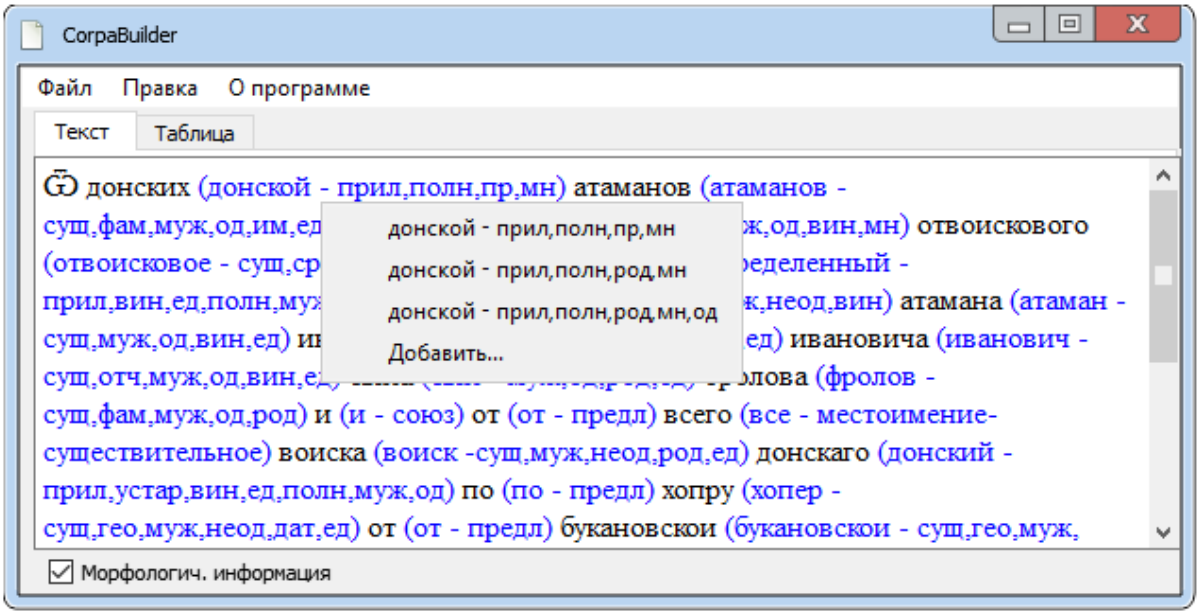

Рис. 2. Интерфейс приложения с опцией отображения морфологической информации

происходит замена всех устаревших букв на современные аналоги. Содержимое массива с замененными символами сохраняется во временный файл, который помещается в директорию временных файлов ОС. Этот файл обрабатывается MyStem, после чего наше приложение анализирует его выходной поток и выводит его на экран. Последний этап (анализ и вывод) описан ниже, а на рисунках 4 и 5 представлены блок-схема и диаграмма последовательностей, более детально описывающие алгоритм работы приложения.

Вызов утилиты MyStem для полученного файла производится с опциями

```
-cisd  --eng-gr --generate-all --format json.
```

Ключ «c» означает копирование всего ввода на вывод, включая слова и межсловные

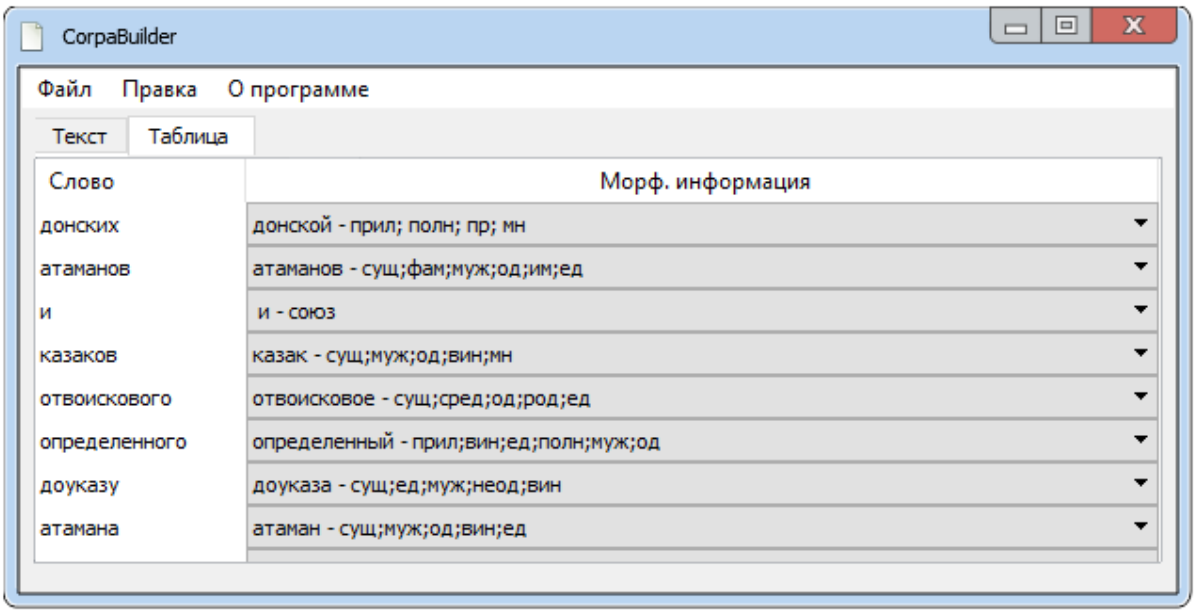

Рис. 3. Табличный интерфейс приложения

промежутки. Это требуется для возврата к полному представлению текста. Опция «`i`» используется для печати грамматической информации, «`s`» — для печати маркера конца приложения, «`d`» — включает контекстное снятие омонимии, «`--eng-gr`» — переключает вывод грамматической информации на английский язык, «`--generate-all`» — используется для генерации всех возможных гипотез для незнакомых слов, «`--format json`» — для вывода результата в JSON. После окончания работы MyStem получается результат в виде строки JSON. Затем происходит ее маппинг с помощью JSON-процессора Jackson в объект Word, который имеет те же поля, что и JSON, выводимый MyStem. После маппинга на выходе получаем массив объектов Word.

Данные о грамматической информации слова выводятся в формате, который не очень удобен для дальнейшей работы с ними. Например, разбор слова «человек» выглядит так:

```
{"analysis":[{"lex":"человек","gr":"S,m,anim=acc,pl"},
{"lex":"человек", "gr":"S,m,anim=gen,pl"},
{"lex":"человек","gr":"S,m,anim=nom,sg"}], "text":"человек"}.
```

Как видно, в структуре возможных разборов слова отсутствует явное разделение на падеж, род, число и прочие признаки. Поэтому было принято решение выполнять дополнительную обработку значения «`gr`», представляя каждое свойство слова, перечисленное в нем, в виде соответствующего enum.

Затем, если включена опция замены старославянских символов, в каждый объект Word, согласно индексу, добавляется оригинальное слово.

Полученный массив экземпляров класса Word помещается в объект класса ObservableList с помощью метода observableArrayList класса FXCollections из библиотеки JavaFX для последующего вывода на экранную форму. В зависимости от вкладки вывод происходит в виде текста или таблицы.

Таблица представляет собой объект TableView из JavaFX, колонки таблицы — TableColumn. Для первой колонки применим метод word.setCellValueFactory(new

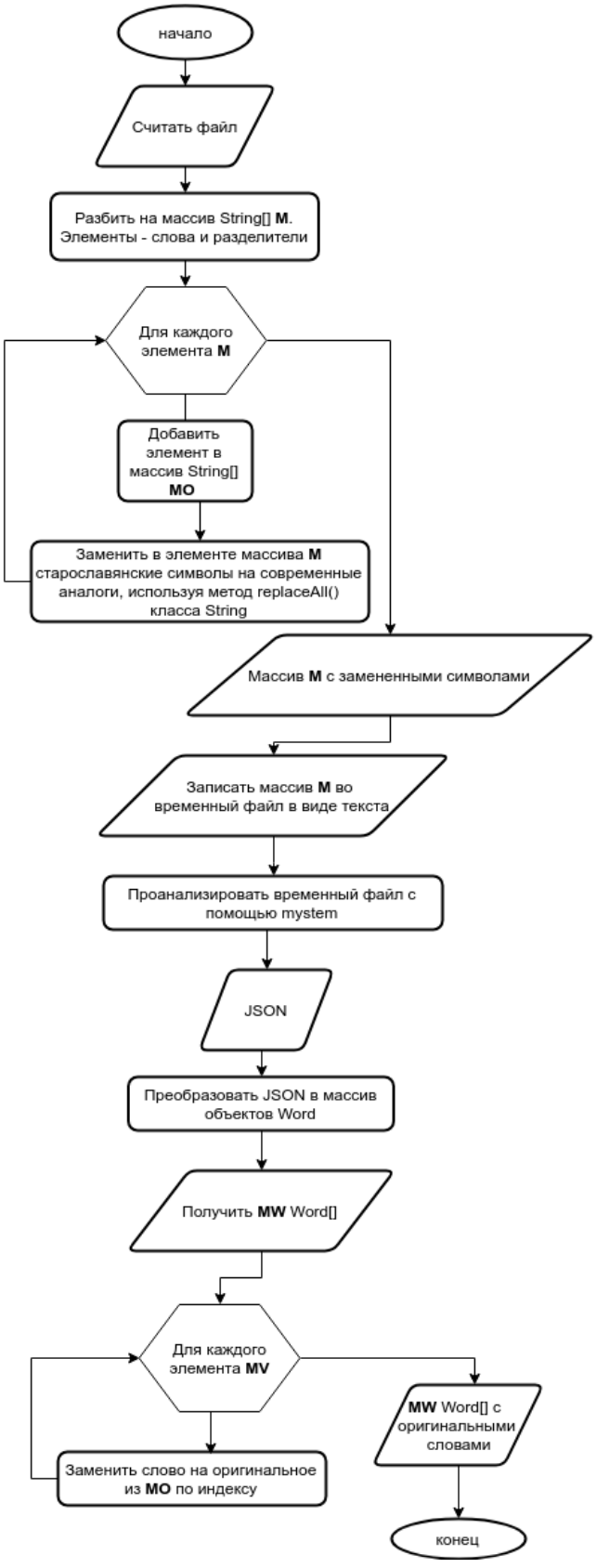

Рис. 4. Блок-схема алгоритма

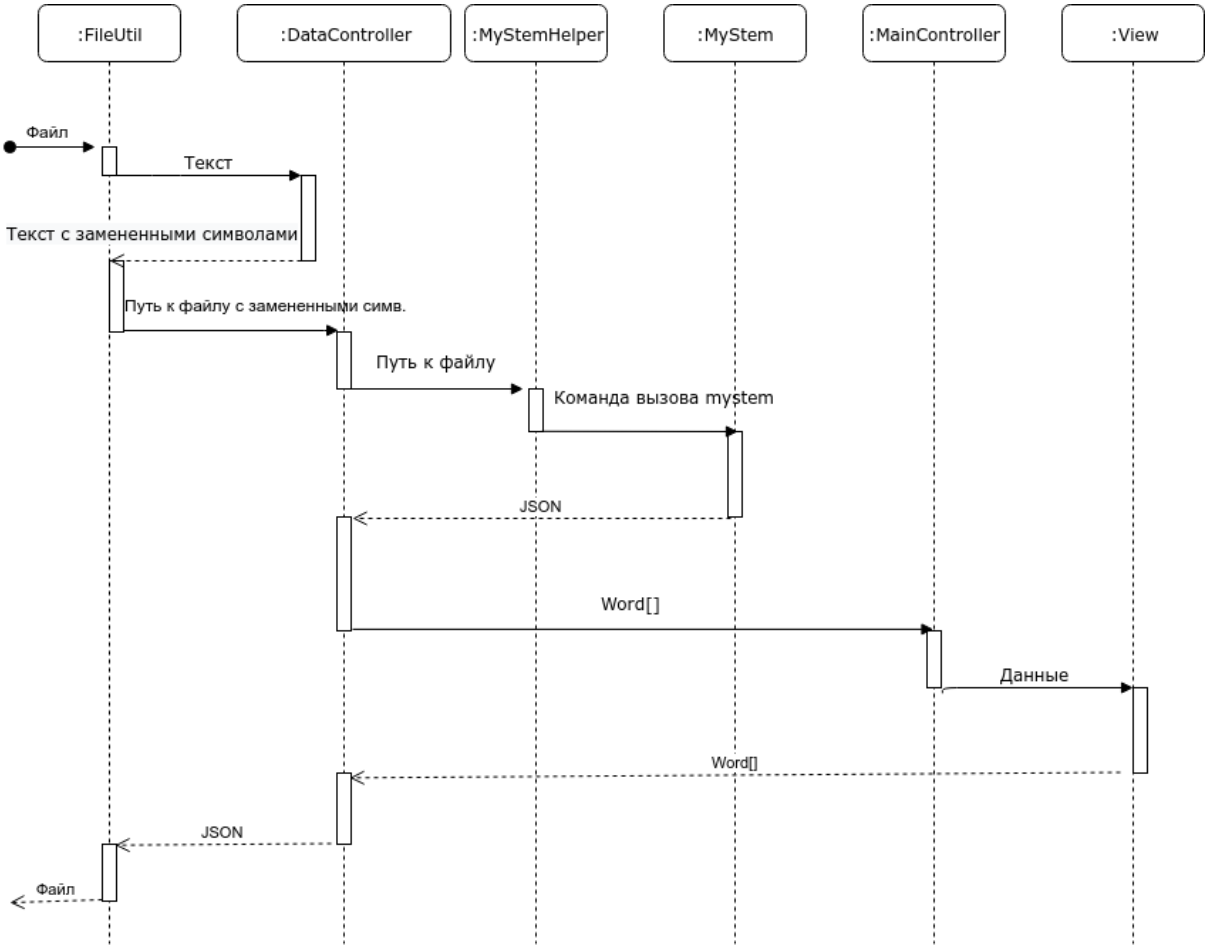

Рис. 5. Диаграмма последовательностей приложения

PropertyValueFactory<>("originalText")). Значением столбца в таком случае будет значение поля originalText объекта Word.

Далее, для поддержки редактирования поля выполним следующие действия:

```
word.setOnEditCommit((TableColumn.CellEditEvent<Word, String>
event) -> {
TablePosition<Word, String> pos = event.getTablePosition();
String newText = event.getNewValue();
Word editableWord =
event.getTableView().getItems().get(pos.getRow());
editableWord.setText(newText);
});
```

Таким образом, при редактировании слова новое значение будет сохраняться в объект класса Word.

Подобным образом осуществляется редактирование и остальных колонок. Один из разборов можно выбрать в ComboBox колонки «Морф. информация». Так устроено ручное снятие омонимии. Последний разбор (с индексом, равным длине массива) имеет наибольшую вероятность. По умолчанию поле selectedAnalysis объекта Word имеет значение, равное длине массива. При выборе одного из пунктов ComboBox это поле приобретает значение индекса выбранного пункта. В случае если ни один из разборов не

является верным, пользователь может воспользоваться пунктом «Добавить. . . ». В таком случае в поле analysis объекта класса Word, которое представляет собой массив объектов Analysis, добавляется новый объект с указанной в окне информацией.

Подобным образом устроена работа вкладки «текст». Для реализации нестандартного поведения (при щелчке правой кнопкой мыши на слово открывается меню с вариантами разбора и пункт для добавления своего разбора) был создан нестандартный (пользовательский) элемент интерфейса, в котором каждое слово представлено отдельным объектом и у каждого такого объекта существует свое меню, которое открывается при нажатии на него правой кнопкой мыши. Интерфейс приложения и описанные возможности были представлены выше на рисунках 1–3.

Заключение

Таким образом, после завершения работы с программой мы имеем исходный файл, дополненный морфологической разметкой. В настоящее время разработка программного обеспечения корпуса еще не завершена, поэтому требуемый формат представления морфологической информации пока не определен. По этой причине сохранение документа производится в формате JSON. Точнее, сохраняется массив JSON-объектов Word с содержащимися в них JSON-объектами analysis, с индексом, равным значению поля selectedAnalysis в объекте Word. Например, слово «человек» будет представлено так:

```
[
{
"text":"человек",
"inital":"человек",
"partOfSpeech":"S",
"gender":"m",
"anim":"anim",
"case":"nom",
"tense":"sg"
}
]
```

Данный формат прост и лаконичен, поэтому без труда может быть трансформирован в любой другой вид, если при разработке «движка» корпуса будет отдано предпочтение какому-то другому формату.

ПРИМЕЧАНИЕ

[1] Работа выполнена при финансовой поддержке гранта РФФИ № 19-012-00246.

СПИСОК ЛИТЕРАТУРЫ

1. Балясова, Е. С. Войсковые грамоты XVIII в.: лингвистический корпус / Е. С. Балясова. // Теоретические и прикладные аспекты корпусных исследований : тез. науч. конф. — Электрон. текстовые дан. — Режим доступа: https://volsu.ru/upload/medialibrary/904/2016-konferentia-tezises-corpus.pdf. — Загл. с экрана.

2. Балясова, Е. С. Региональные архивные документы XVIII века в аспекте корпусной лингвистики / Е. С. Балясова, Е. М. Шептухина // Коммуникативные аспекты современной лингвистики и лингводидактики : материалы Междунар. науч. конф. — Волгоград : Изд-во ВолГУ, 2017. — С. 31–37.

3. Светлов, А. В. Автоматизация процесса получения лингвистической информации: современные возможности / А. В. Светлов, А. С. Комендантов // Вестник Волгоградского государственного университета. Серия 2, Языкознание. — 2017. — Т. 16, № 2. — С. 39–46. — DOI: https://doi.org/10.15688/jvolsu2.2017.2.4.

4. Шептухина, Е. М. Войсковые грамоты середины XVIII века в аспекте категории модальности / Е. М. Шептухина, О. А. Горбань // Вестник Волгоградского государственного университета. Серия 2, Языкознание. — 2015. — № 5 (29). — С. 7–18. — DOI: http://dx.doi.org/10.15688/jvolsu2.2015.5.1.

5. Шептухина, Е. М. Этапы создания лингвистического корпуса войсковых грамот XVIII—XIX вв. архивного фонда «Михайловский станичный атаман» ГАВО / Е. М. Шептухина, О. А. Горбань // Гуманитарное образование и наука в техническом вузе : сб. докл. Всерос. науч.-практ. конф. с междунар. участием. — Ижевск : Изд-во Ижев. гос. техн. ун-та им. М.Т. Калашникова, 2017. — С. 428–431.

6. Segalovich, I. A fast morphological algorithm with unknown word guessing inducedby a dictionary for a web search engine. / I. Segalovich // Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications. — Las Vegas : CSREA Press, 2003. — P. 273–280.

REFERENCES

1. Balyasova E.S. *Voyskovye gramoty XVIII v.: lingvisticheskiy korpus* [Don Cossack Army Charters of the 18th Century: Linguistic Corpus]. *Teoreticheskie i prikladnye aspekty korpusnykh issledovaniy: tez. nauch. konf.* URL: https://volsu.ru/upload/medialibrary/904/2016-konferentia-tezises-corpus.pdf.

2. Balyasova E.S., Sheptukhina E.M. Regionalnye arkhivnye dokumenty XVIII veka v aspekte korpusnoy lingvistiki [18th Century Regional Archival Documents in the Aspect of Corpus Linguistics]. *Kommunikativnye aspekty sovremennoy lingvistiki i lingvodidaktiki: materialy Mezhdunar. nauch. konf.* Volgograd, Izd-vo VolGU, 2017, pp. 31-37.

3. Svetlov A.V., Komendantov A.S. Avtomatizatsiya protsessa polucheniya lingvisticheskoy informatsii: sovremennye vozmozhnosti [Automation of the Process for Obtaining Linguistic Information: State-Of-The-Art Capabilities]. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 2, Yazykoznanie* [Science Journal of Volgograd State University. Linguistics], 2017, vol. 16, no. 2, pp. 39–46. DOI: https://doi.org/10.15688/jvolsu2.2017.2.4.

4. Sheptukhina E.M., Gorban O.A. Voyskovye gramoty serediny XVIII veka v aspekte kategorii modalnosti [Don Cossack Army Charters of the Mid 18th Century Via the Category of Modality]. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 2, Yazykoznanie* [Science Journal of Volgograd State University. Linguistics], 2015, no. 5 (29), pp. 7-18. DOI: http://dx.doi.org/10.15688/jvolsu2.2015.5.1.

5. Sheptukhina E.M., Gorban O.A. Etapy sozdaniya lingvisticheskogo korpusa voyskovykh gramot XVIII—XIX vv. arkhivnogo fonda «Mikhaylovskiy stanichnyy ataman» GAVO [Stages of Creating the Linguistic Corpus of Don Cossack Army Charters of the 18th—19th Centuries Archive Fund «Mikhailovsky Stanichny Ataman»]. *Gumanitarnoe obrazovanie i nauka v tekhnicheskom vuze: cb. dokl. Vseros. nauch.-prakt. konf. s mezhdunar. uchastiem.* Izhevsk, Izd-vo Izhev. gos. tekhn. un-ta im. M.T. Kalashnikov, 2017, pp. 428-431.

6. Segalovich I. A fast morphological algorithm with unknown word guessing inducedby a dictionary for a web search engine. *Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications*. Las Vegas, CSREA Press, 2003, pp. 273-280.

AUTOMATION OF MORPHOLOGICAL TAGGING OF ARCHIVAL DOCUMENTS

Anatoly S. Komendantov

Student, Institute of Mathematics and IT,
Volgograd State University
anatoly.komendantov@gmail.com, matf@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Alexander G. Matveev

Student, Institute of Mathematics and IT,
Volgograd State University
sna4es@gmail.com, matf@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Andrey V. Svetlov

Candidate of Physical and Mathematical Sciences, Associate Professor,
Department of Mathematical Analysis and Function Theory,
Volgograd State University
a.v.svetlov@gmail.com, andrew.svetlov@volsu.ru, matf@volsu.ru
https://orcid.org/0000-0002-8764-6132
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Abstract. The paper provides the description of the add-on to MyStem stemming tool by I. Segalovich. We designe the application to add to MyStem a convenient graphical interface that is easy to learn and intuitive for users who do not specialize in information technology. It turned out that MyStem correctly processes outdated vocabulary if it is passed into the program using modern Cyrillic. In addition to the convenient interface, our program has the option to work with the outdated Cyrillic alphabet, when for instance, the letters zelo and omega are replaced by “ks” and “o” respectively, and only then the text is transferred for analysis to MyStem, and then the characters are replaced back in the processed document. So our add-on intercepts the output of MyStem tool, reformats and analyzes it in a special way. In addition, the application has functionality for removing homonyms manually if the program was not correct with automatic tagging of morphological characteristics of a word. The main purpose of this application is to prepare morphological tagging of documents of the archival fund “Mikhailovsky Stanichny Ataman” to create a linguistic corpus. During the work on the application, we solved the problem with correct processing of texts containing outdated Cyrillic characters. To implement a functional and user-friendly graphical interface, we use JavaFX platform (OpenJFX).

Key words: automation of linguistic analysis, automation of morphological analysis, MyStem tool, graphical interface, software shell, corpus-based linguistics.