УДК 004.415.2
DOI: 10.15827/0236-235X.129.038-046
Дата подачи статьи: 30.10.19
2020. Т. 33. № 1. С. 038–046

Модуль разрешения морфологической неоднозначности: архитуктура и организация базы данных

Д.Р. Мухамедшин [1,2], *аспирант, damirmuh@gmail.com*
Д.Ш. Сулейманов [1,2], *д.т.н., профессор, ipsanrt@mail.ru*

[1] *Институт прикладной семиотики Академии наук Республики Татарстан, г. Казань, 420111, Россия*
[2] *Казанский федеральный университет, г. Казань, 420008, Россия*

Системы управления корпусными данными помогают решать целый ряд актуальных задач, связанных с компьютерной лингвистикой. В частности, это обработка исходных документов, их автоматическая морфологическая разметка, хранение и изменение корпусных данных с морфологической разметкой, выполнение поисковых запросов (выборок данных), интеграция с другими приложениями.

При автоматической морфологической разметке текстов электронного корпуса часто возникает проблема, когда одна и та же словоформа может быть размечена двумя и более наборами морфологических свойств, что называется морфологической неоднозначностью. Авторами был разработан модуль для ручного снятия морфологической неоднозначности, который может помочь в решении проблемы. В данной статье рассмотрена архитектура модуля разрешения морфологической неоднозначности системы управления корпусными данными, разработанной для управления татарским корпусом, и описана организация БД, основанная на транзакционном подходе к хранению данных.

В статье подробно описана архитектура программной части модуля, который является частью системы управления корпусными данными «Туган Тел». Благодаря архитектуре системы управления корпусными данными подобные модули могут быть быстро интегрированы в систему и не влиять на работу других модулей.

Описанная в статье организация БД модуля разрешения морфологической неоднозначности позволяет перейти к решению других задач компьютерной лингвистики, таких как автоматическое разрешение морфологической неоднозначности и улучшение автоматической морфологической разметки текстов электронного корпуса.

Ключевые слова: *морфологическая неоднозначность, корпусные данные, архитектура системы, БД, СУБД.*

Автоматическая морфологическая разметка достаточно больших объемов текстов неизбежно приводит к появлению примеров морфологической неоднозначности [1]. Одна и та же словоформа может быть размечена алгоритмом различными наборами морфологических свойств. Без информации о семантике контекста при автоматической морфологической разметке невозможно принять решение о корректности того или иного набора морфологических свойств в каждом конкретном случае морфологической неоднозначности. Имеется ряд способов, которые могут помочь в снятии проблемы, одним из них является ручное разрешение морфологической неоднозначности экспертом [2, 3].

Разработанный авторами модуль является частью системы управления татарским корпусом (http://tugantel.tatar) [4] и направлен на снятие морфологической неоднозначности в ресурсах татарского национального корпуса [5]. Корректная морфологическая разметка текстов позволяет более качественно выявлять закономерности языка, использовать корпус для различных задач компьютерной лингвистики.

Основной функционал модуля

Процесс разрешения морфологической неоднозначности делится на несколько этапов, в каждом из которых принимают участие пользователи с различными ролями. Модуль ограничивает права доступа для следующих ролей: «Гость», «Аннотатор», «Главный аннотатор», «Администратор» при помощи модели ShieldModel, являющейся частью системы управления корпусом «Туган Тел», и модели AmbiguityUserModel.

Пользователь с правами доступа «Гость» имеет право просматривать страницу авторизации и отправлять данные при помощи формы авторизации, введя в нее свои логин и пароль. При попытке авторизации данные обрабатываются контроллером и моделью авторизации

модуля AmbiguityAuthModel. При попытке воспользоваться другим функционалом модуля система сообщит об ошибке – «Доступ запрещен».

Пользователь с правами доступа «Аннотатор» имеет право просматривать и производить действия с примерами для разрешения морфологической неоднозначности, назначенными для группы пользователей авторизованного пользователя, а также пропущенныеми в процессе разрешения. При просмотре примеров есть возможность выбрать определенный документ (см. http://www.swsys.ru/uploaded/image/2020-1/2020-1-dop/13.jpg), фрейм или просмотреть все примеры.

Список документов или фреймов для разрешения морфологической неоднозначности формируется в модели AmbiguitySamplesModel, соответственно, при помощи методов getDocs и getRules. Список примеров для разрешения морфологической неоднозначности также формируется в модели AmbiguitySamplesModel при помощи метода search, который, в свою очередь, возвращает набор экземпляров класса AmbiguitySampleModel. Далее список выводится в браузере пользователя, как это показано на рисунке (см. http://www.swsys.ru/uploaded/image/2020-1/2020-1-dop/17.jpg). Пока пользователь работает над разрешением морфологической неоднозначности, модуль записывает все его действия при помощи модели AmbiguityTrackingModel.

- Working – общая работа над примерами; используется для анализа полезного времени, потраченного пользователем на разрешение морфологической неоднозначности, при этом данное действие автоматически останавливается, если пользователь отвлекается (переключается в другое окно, не производит никаких действий на странице продолжительное время).
- Choose – выбор корректного, по мнению пользователя, разбора из предложенного списка.
- Miss – пропуск примера; к нему в дальнейшем можно вернуться в списке пропущенных примеров.
- Remove – удаление примера; необходимо, если в исходном тексте имеются артефактные включения и их нужно удалить.
- Correct – корректирование примера, заключающееся в редактировании содержимого словоформы или ручном вводе морфологического разбора;
- Copy, Cut, Paste – действия с буфером обмена (записываются в случае копирования, вырезания или вставки текста на странице);
- Blur, Focus – действия с окном страницы (записываются в случае переключения на другое окно) и возврат в текущее окно браузера.

После завершения работы над контекстом пользователь сохраняет все изменения. Сохраненный контекст исчезает из списка, и, когда в списке остаются пять контекстов, автоматически загружаются следующие контексты. Таким образом, процесс разрешения морфологической неоднозначности реализуется на всем множестве неоднозначных контекстов коллекции текстов. Для удобства модуль также выводит количество оставшихся неразрешенных контекстов.

Пользователь с правами «Главный аннотатор», кроме функционала, доступного пользователям с правами «Аннотатор», может также производить проверку результатов разрешения морфологической неоднозначности, выполненного аннотаторами первого уровня. Также пользователю с правами «Главный аннотатор» доступен функционал добавления и редактирования фреймов (см. http://www.swsys.ru/uploaded/image/2020-1/2020-1-dop/14.jpg), которые содержат наборы правил для подбора примеров морфологической неоднозначности. Функции добавления и редактирования фреймов обрабатываются моделью AmbiguityRuleModel.

В интерфейсе добавления фрейма можно указать необходимые для сохранения и выборки примеров из корпуса перечисленные далее параметры.

- Название фрейма, отражающее содержание примеров или их назначение.
- Тип поиска: по словоформе или по лемме. Также поддерживается поиск по списку словоформ или лемм в соответствии с синтаксисом (словоформы или леммы можно перечислить, разделив их знаком «|»).
- Формула поискового запроса. Указывается в таком же формате, как в URL результатов поиска поискового модуля системы управления корпусными данными «Туган Тел».
- Ограничение – количество примеров, которое необходимо получить в создаваемом фрейме. Может быть любым неотрицательным числом. Ограничение «0» модуль интерпретирует как отсутствие ограничений и добавит максимально возможное количество примеров, соответствующих параметрам, указанным во фрейме.

- Включить словоформы без многозначности. Установка данного флажка добавляет в генерируемые примеры словоформы, в которых нет многозначности, то есть при автоматической морфологической разметке алгоритмом предложен один морфологический разбор.
- Подкорпус. Чтобы подобрать примеры из определенного подкорпуса, необходимо выбрать его из списка. По умолчанию используется подкорпус для разрешения многозначности. В списке также имеется буферный подкорпус, созданный для безопасной работы с копией основного корпуса в модуле ручного снятия многозначности.
- Документ (документы). Чтобы подобрать примеры из определенных документов, необходимо указать их в одном из форматов:

– список идентификаторов документа, по одному в строке (идентификаторы показываются в результатах поиска поискового модуля системы управления корпусными данными);

– список названий файлов документов, по одному в строке (показываются в результатах поиска поискового модуля системы управления корпусными данными);

– начало содержания документа (можно указать первые 20–30 строк из искомого документа, этот способ можно использовать, если идентификатор или название файла документа неизвестны).

- Группа пользователей. Можно указать, для какой группы пользователей будет доступно разрешение примеров создаваемого фрейма.

Пользователю с правами «Главный аннотатор» и «Администратор» также доступен функционал управления пользователями, включающий функции добавления, редактирования пользователей и групп пользователей. Действия, связанные с пользователями и группами пользователей, обрабатываются моделями AmbiguityUserModel и AmbiguityGroupModel соответственно.

Архитектура модуля

Для программной реализации модуля разрешения морфологической неоднозначности, как и для системы управления корпусными данными «Туган Тел», используется концепция MVC (Model-View-Controller) [6], которая была несколько изменена для решения задач системы. Основная схема выполнения любой задачи, поставленной системе, показана на рисунке 1.

Работа системы начинается с запроса (Request), который приходит в основной контроллер (MainControl). Основной контроллер обеспечивает безопасность при помощи компонента ShieldModel, который, в свою очередь, использует объект очереди (Queue). Если ShieldModel считает запрос безопасным, контроль передается специальному контроллеру в зависимости от типа запрашиваемой задачи.

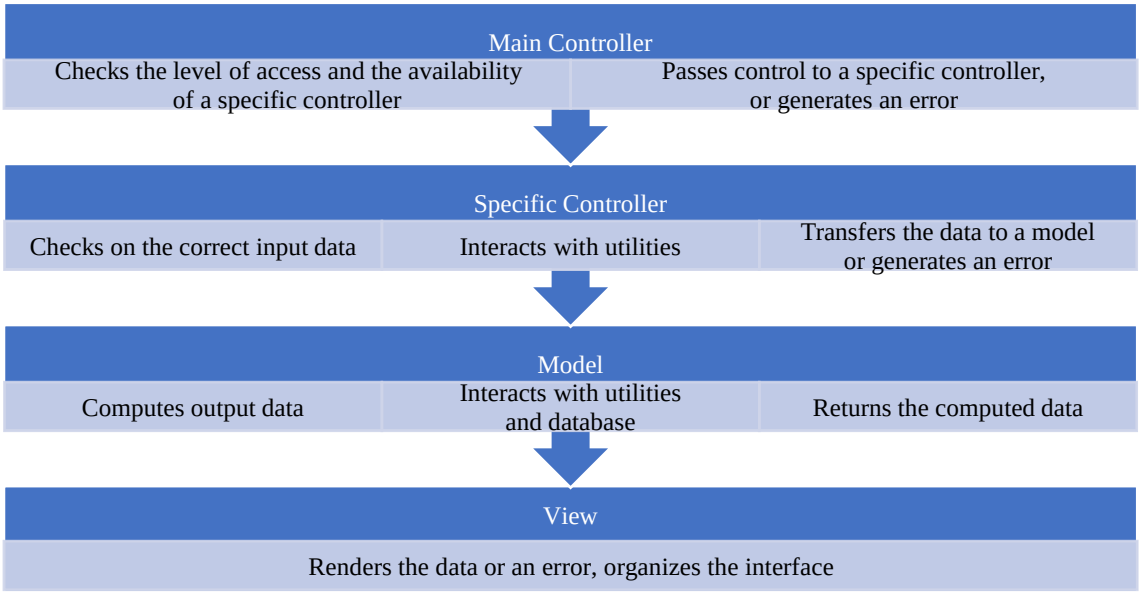

Рис. 1. Абстрактная схема выполнения задач в системе

Fig. 1. The abstract plan of the problem execution in the system

В модуле разрешения морфологической неоднозначности используются 11 специальных контроллеров: AmbiguityDocumentControl – контроллер документов, AmbiguityGroupControl и AmbiguityGroupsControl – контроллеры групп пользователей, AmbiguityHistoryControl – контроллер истории, AmbiguityRuleControl и AmbiguityRulesControl – контроллеры фреймов, AmbiguitySamplesControl – контроллер примеров, AmbiguityStatisticsControl – контроллер статистики, AmbiguityTrackingControl – контроллер отслеживания, AmbiguityUserControl и AmbiguityUsersControl – контроллеры пользователей.

После проверки и фильтрации данных система передает контроль в модель, соответствующую запрошенному действию: AmbiguityDocumentModel, AmbiguityGroupModel, AmbiguityGroupsModel, AmbiguityHistoryModel, AmbiguityRuleModel, AmbiguityRulesModel, AmbiguitySamplesModel, AmbiguityStatisticsModel, AmbiguityTrackingModel, AmbiguityUserModel, AmbiguityUsersModel. Все модели используют модель БД DB, которая, в свою очередь, использует модель Cache для кэширования.

После выполнения действия моделью контроль возвращается к контроллеру, откуда данные передаются в Вид (View). Последний использует данные и соответствующие шаблоны страниц для генерации документа HTML или возвращает данные в формате JSON в зависимости от запрошенного формата выходных данных.

Архитектура БД

Хранение корпусных данных. В системе управления корпусными данными «Туган Тел» основные данные корпусов хранятся в таблицах СУБД MySQL [7]:

– Docs – таблица документов электронного корпуса;

– Sentences – таблица предложений (контекстов), которые являются частями документов электронного корпуса;

– Wordforms – таблица обратного индекса, используемая для быстрого поиска по корпусу, где хранится информация о словоформах и их морфологических разборах.

Таблица документов представляется простой структурой для хранения содержимого и главных метаданных документов электронного корпуса. Структура таблицы документов представлена в таблице 1.

Таблица 1

Структура таблицы документов электронного корпуса

Table 1

The document table structure of the electronic corpus

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор документа
filename	VARCHAR (UNIQUE)	Уникальное исходное имя файла документа
title	VARCHAR	Название документа
author	VARCHAR	Автор документа
text	LONGTEXT	Содержимое документа

Структура таблицы предложений (контекстов) предназначена для хранения содержимого предложений (контекстов), которые являются частями документов электронного корпуса, и представлена в таблице 2.

Таблица 2

Структура таблицы предложений (контекстов)

Table 2

The suggestion table structure (contexts)

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор предложения (контекста)
doc	INT (REFERENCES `docs` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION)	Идентификатор документа электронного корпуса. Столбец связан со столбцом ‘id’ таблицы документов электронного корпуса
pos	INT	Позиция предложения (контекста) в документе
text	LONGTEXT	Содержимое предложения (контекста)

Таблица обратного индекса [8] хранит в себе информацию о всех морфологических разборах каждой словоформы документа электронного корпуса. Структура таблицы обратного индекса показана в таблице 3.

БД модуля разрешения морфологической неоднозначности. Данные модуля разрешения морфологической неоднозначности также хранятся в таблицах СУБД MySQL. Архитектура БД модуля позволяет хранить все транзакции пользователей, а не конечный результат – документы и предложения со снятой морфоло-

Таблица 3

Структура таблицы обратного индекса

Table 3

The reverse index table structure

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор морфологического анализа
word	MEDIUMINT UNSIGNED	Уникальный идентификатор словоформы. Связи между идентификаторами и словоформами хранятся в Redis
lemma	MEDIUMINT UNSIGNED	Уникальный идентификатор леммы. Связи между идентификаторами и леммами хранятся в Redis
doc	MEDIUMINT UNSIGNED	Уникальный идентификатор документа. Документы хранятся в отдельной таблице
pos	MEDIUMINT UNSIGNED	Позиция словоформы в документе
sentence	INT UNSIGNED	Уникальный идентификатор контекста. Контексты хранятся в отдельной таблице
sentence_pos	SMALLINT UNSIGNED	Позиция словоформы в контексте
morph1	BIGINT UNSIGNED	Первые 64 морфологических свойства
morph2	BIGINT UNSIGNED	Последующие 64 (до 64) морфологических свойства

Таблица 4

Структура таблицы пользователей

Table 4

The user table structure

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор пользователя
login	VARCHAR	Логин пользователя, используемый для авторизации
password	VARCHAR	Пароль пользователя, хранящийся в зашифрованном виде и используемый для авторизации
name	VARCHAR	Имя пользователя
email	VARCHAR	E-mail пользователя, используемый для оповещений
permissions	BIGINT	Права пользователя, для каждой роли пользователей используется константа
status	ENUM (‘active’, ‘banned’, ‘trash’)	Статус пользователя: активный, доступ закрыт, удален

гической неоднозначностью. Такой подход позволяет получить результаты на различных этапах работы в различных разрезах, перекладывая формирование конечного результата на отдельный модуль сохранения результатов разрешения морфологической неоднозначности. Кроме того, благодаря такому подходу модуль дает возможность анализировать работу каждого пользователя и оценивать результативность его работы при помощи модели статистики AmbiguityStatsModel.

Структура таблицы пользователей ‘ambiguity_users’ позволяет хранить все данные пользователей модуля. В таблице 4 указаны столбцы таблицы пользователей и их назначение. В частности, эта таблица дает возможность идентифицировать и разграничить права каждого пользователя модуля.

Для реализации функционала групп пользователей используются две таблицы: ‘ambiguity_groups’ для хранения информации о группах пользователей, ‘ambiguity_groups_users’ для хранения связей между пользователями и группами пользователей. Их структуры показаны в таблицах 5 и 6. В таблице связей используется составной индекс типа UNIQUE [9] по обоим столбцам, чтобы исключить возможность дублирования связей.

Таблица 5

Структура таблицы групп пользователей

Table 5

The group user table structure

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор группы пользователей
owner_id	INT	Идентификатор пользователя, являющегося владельцем группы
name	VARCHAR	Название группы пользователей
permissions	BIGINT	Права группы пользователей

Таблица 6

Структура таблицы связей между пользователями и группами пользователей

Table 6

The connection table structure between the users and the group users

Имя столбца	Тип	Описание
user_id	INT	Идентификатор пользователя
group_id	INT	Идентификатор группы пользователей

В таблице фреймов ‘ambiguity_rules’ хранятся правила формирования примеров морфологической неоднозначности. Структура ее и назначение столбцов показаны в таблице 7.

Таблица 7

Структура таблицы фреймов

Table 7

The frame table structure

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор фрейма
user_id	INT	Идентификатор пользователя, создавшего фрейм
name	VARCHAR	Название фрейма
settings	TEXT	Набор правил, хранящийся в формате JSON
status	ENUM (‘processing’, ‘processed’)	Статус фрейма: в процессе формирования примеров, процесс формирования примеров завершен

На основе фреймов модуль формирует примеры для разрешения морфологической неоднозначности, которые записываются в таблицу ‘ambiguity_samples’. Структура ее представлена в таблице 8.

При работе пользователей над примерами модуль записывает каждое действие, произведенное как над примерами морфологической неоднозначности, так и в процессе работы над ними. Такие транзакции хранятся в двух таблицах: в ‘ambiguity_history’ хранятся основные транзакции над примерами, в ‘ambiguity_tracking’ – другие действия пользователей для анализа процесса работы (см. http://www.swsys.ru/uploaded/image/2020-1/2020-1-dop/18.jpg).

Представленные таблицы позволяют полностью реализовать функционал модуля разрешения морфологической неоднозначности. Для обеспечения целостности и согласованности данных в БД также имеются связи [10] между столбцами, которые показаны на рисунке 2.

Заключение

Архитектура модуля и организация БД, описанные в данной статье, используются в модуле разрешения морфологической неоднозначности системы управления корпусными данными, который работает совместно с электронным корпусом татарского языка «Туган Тел».

Представленные методы хранения данных о разрешении примеров морфологической неоднозначности обеспечивают необходимую полноту данных для дальнейшего сохранения результатов в виде документов со снятой морфологической неоднозначностью.

Использование транзакционного подхода к хранению данных позволяет решить целый ряд проблем. В данной статье описаны лишь некоторые из них. В будущем планируется исполь-

Таблица 8

Структура таблицы примеров морфологической неоднозначности

Table 8

The example table structure of the morphological disambiguation

Имя столбца	Тип	Описание
id	INT (AUTO-INCREMENT)	Уникальный идентификатор примера
user_id	INT	Идентификатор пользователя, создавшего пример
group_id	INT	Идентификатор группы пользователей, для которой назначен пример
rule_id	INT	Идентификатор фрейма, на основе которого создан пример
subcorpus	VARCHAR	Префикс подкорпуса, в котором находится пример
doc_id	INT	Идентификатор документа, в котором находится пример
sentence_id	INT	Идентификатор предложения (контекста), в котором находится пример
word_pos	INT	Позиция словоформы примера
is_resolved	TINYINT	Флаг разрешения примера: 1 – пример разрешен, 0 – пример не разрешен

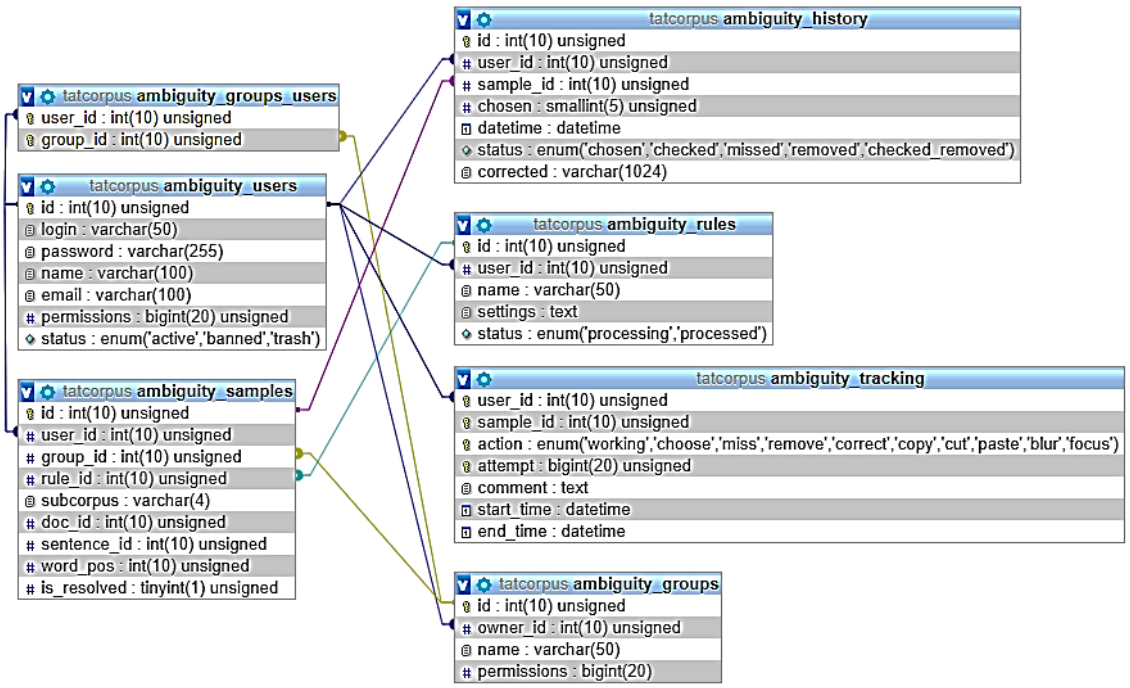

Рис. 2. Связи между столбцами таблиц в БД модуля разрешения морфологической неоднозначности

Fig. 2. The connections between the table columns in the database of the morphological disambiguation module

зовать полученные в процессе работы данные в качестве обучающих для автоматизации разрешения морфологической неоднозначности при помощи методов нейронных сетей [11]. Именно хранение детальных данных о каждом действии пользователя может способствовать достижению высоких результатов при автоматическом разрешении морфологической неоднозначности.

Представленный подход к решению проблем морфологической неоднозначности для лингвистических корпусов дает возможность использовать разработанный модуль и систему управления корпусными данными в целом не только для Электронного корпуса текстов на татарском языке, но и для корпусов других языков без существенных изменений в системе.

Литература

1. Gilmullin R., Gataullin R. Morphological analysis system of the tatar language. Proc. 9th ICCCI, Springer Publ., Nicosia, Cyprus, 2017, pp. 519–528. DOI: 10.1007/978-3-319-67077-5_50.

2. Gataullin R., Suleimanov D., Khakimov B., Gilmullin R. Context-based rules for grammatical disambiguation in the Tatar language. Proc. 9th ICCCI, Springer, Nicosia, Cyprus, 2017, pp. 529–537. DOI: 10.1007/978-3-319-67077-5_51.

3. Хакимов Б.Э., Гильмуллин Р.А., Гатауллин Р.Р. Разрешение грамматической многозначности в корпусе татарского языка // Учен. зап. Казан. ун-та. Сер. Гуманит. науки. 2014. Т. 156. № 5. С. 236–244.

4. «Tugan Tel» Tatar National Corpus Homepage. URL: http://tugantel.tatar/?lang=en (дата обращения: 01.10.2019).

5. Suleimanov D., Gatiatullin A., Khakimov B., Nevzorova O., Gilmullin R. National corpus of the Tatar language “Tugan Tel”: grammatical annotation and implementation. Procedia – Social and Behavioral Sciences, 2013, vol. 95, pp. 68–74. DOI: 10.1016/j.sbspro.2013.10.623.

6. Leff A., Rayfield J.T. Web-application development using the model/view/controller design pattern. Proc. Fifth IEEE Intern. EDOC Conf., 2001, pp. 118–127. DOI: 10.1109/EDOC.2001.950428.

7. DuBois P. MySQL. Pearson Education, 2008, 1224 p.

8. Croft W.B., Metzler D., Strohman T. Search engines: Information retrieval in practice. Addison-Wesley, 2010, vol. 520, pp. 131–141.

9. MySQL 5.6 Reference Manual 19.6.1 Partitioning Keys, Primary Keys, and Unique Keys. URL: https://dev.mysql.com/doc/refman/5.6/en/partitioning-limitations-partitioning-keys-unique-keys.html (дата обращения: 01.10.2019).

10. MySQL 5.6 Reference Manual 13.1.17.6 Using FOREIGN KEY Constraints. URL: https://dev.mysql.com/doc/refman/5.6/en/create-table-foreign-keys.html (дата обращения: 01.10.2019).

11. Gilmullin R., Khakimov B., Gataullin R. A neural network approach to morphological disambiguation based on the lstm architecture in the national corpus of the tatar language. Proc. Intern. Workshop CMLS, 2018, vol. 2303, p. 32. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060582291&partnerID=40&md5=c4269526b6dae5e311f7c40033a904cb (дата обращения: 01.10. 2019).

Software & Systems Received 30.10.19
DOI: 10.15827/0236-235X.129.038-046 2020, vol. 33, no. 1, pp. 038–046

Module for morphological disambiguation: architecture and database organization

D.R. Mukhamedshin [1,2], *Postgraduate Student, damirmuh@gmail.com*
D.Sh. Suleimanov [1,2], *Dr.Sc. (Engineering), Professor, ipsanrt@mail.ru*

[1] *Institute of Applied Semiotics of the Tatarstan Academy of Sciences, Kazan, 420111, Russian Federation*
[2] *Kazan Federal University, Kazan, 420008, Russian Federation*

Abstract. Corpus data management systems today help to solve a number of urgent problems connected with the computer linguistics. Particularly, this is the source documents processing, automatic morphological markup of them, storage and modification of corpus data with morphological markup, the search inquiry execution (selecting data), and integration with other applications.

In the automatic morphological text markup of the electronic corpus, there is a problem when two or more sets of morphological properties marked out the same word form. It is a morphological ambiguity. The authors developed a module for manual resolution of morphological ambiguity, which can help in solving this problem. This article discusses the architecture of the module for resolving the morphological ambiguity of the corpus data management system developed for managing the Tatar corpus, and describes the organization of the database based on the transactional approach to data storage.

The article proposes in detail the architecture of the rupture modulus of the software morphological ambiguousness of corpus data. It is for the tartaric corpus management. In addition, the author described the database organization. It relies on the transactional way to the strategy for the data storage.

The paper in detail describes the architecture of the module software, which is part of the Tugan Tel corpus data management system. Due to the architecture of the corpus data management system, such modules can quickly integrate into the system and have no effect on the other module operation.

The module database organization for the morphological disambiguation described in the paper, allow going over for solving the other computer linguistic problems, such as automatic morphological disambiguation and improving automatic morphological markup of texts in the electronic corpus.

Keywords: morphological ambiguity, corpus data, system architecture, database, DBMS.

References

1. Gilmullin R., Gataullin R. Morphological analysis system of the Tatar language. *Proc. 9th ICCCI*. Springer, 2017, pp. 519–528. DOI: 10.1007/978-3-319-67077-5_50.

2. Gataullin R., Suleimanov D., Khakimov B., Gilmullin R. Context-based rules for grammatical disambiguation in the tatar language. *Proc. 9th ICCCI*. Springer, 2017, pp. 529–537. DOI: 10.1007/978-3-319-67077-5_51.

3. Khakimov B.E., Gilmullin R.A., Gataullin R.R. The elimination of ambiguity in the corpus of the Tatar language. *Uchenye Zapiski Kazanskogo Univ. Gumanitarnye Nauki*. 2014, vol. 156, no. 5, pp. 236–244 (in Russ.).

4. «*Tugan Tel» Tatar National Corpus Homepage*. Available at: http://tugantel.tatar/?lang=en (accessed 01.10.2019).

5. Suleimanov D., Gatiatullin A., Khakimov B., Nevzorova O., Gilmullin R. National corpus of the Tatar language "Tugan Tel": grammatical annotation and implementation. *Procedia-Social and Behavioral Sciences*. 2013, vol. 95, pp. 68–74. DOI: 10.1016/j.sbspro.2013.10.623.

6. Leff A., Rayfield J.T. Web-application development using the model/view/controller design pattern. *Proc. Fifth IEEE Intern. EDOC Conf*. 2001, pp. 118–127. DOI: 10.1109/EDOC.2001.950428.

7. DuBois P. *MySQL*. Pearson Education. 2008, 1224 p.

8. Croft W.B., Metzler D., Strohman T. Search engines: Information retrieval in practice. *Addison-Wesley*. 2010, vol. 520, pp. 131–141.

9. *MySQL 5.6 Reference Manual 19.6.1 Partitioning Keys, Primary Keys, and Unique Keys* Available at: https://dev.mysql.com/doc/refman/5.6/en/partitioning-limitations-partitioning-keys-unique-keys.html (accessed 01.10.2019).

10. *MySQL 5.6 Reference Manual 13.1.17.6 Using FOREIGN KEY Constraints.* Available at: https://dev.mysql.com/doc/refman/5.6/en/create-table-foreign-keys.html (accessed October 01, 2019).

11. Gilmullin R., Khakimov B., Gataullin R. A neural network approach to morphological disambiguation based on the lstm architecture in the national corpus of the tatar language. *Proc. Intern. Workshop CMLS*. 2018, vol. 2303, p. 32. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060582291&partnerID=40&md5=c4269526b6dae5e311f7c40033a904cb. (accessed October 01, 2019).

Для цитирования

Мухамедшин Д.Р., Сулейманов Д.Ш. Модуль разрешения морфологической неоднозначности: архитуктура и организация базы данных // Программные продукты и системы. 2020. Т. 33. № 1. С. 038–046. DOI: 10.15827/0236-235X.129.038-046.

For citation

Mukhamedshin D.R., Suleimanov D.Sh. Module for morphological disambiguation: architecture and database organization. *Software & Systems*. 2020, vol. 33, no. 1, pp. 038–046 (in Russ.). DOI: 10.15827/0236-235X.129.038-046.