Веб-платформа для фольклорных исследований на основе онтологии предметных областей

В. А. Лисин¹, Е. А. Сидорова²

¹ Новосибирский государственный университет
Новосибирск, Россия

² Институт систем информатики им. А. П. Ершова СО РАН
Новосибирск, Россия

Аннотация

Рассматривается веб-платформа, обеспечивающая размещение фольклорных материалов и проведение научных исследований. Фольклорные исследования связаны с изучением аудио- и видеоматериалов, фиксирующих воспроизведение элементов народного творчества на национальных языках, создание текстовых записей с переводами и комментариями на языке общего пользования (в данной работе переводы представлены на русском языке), построение картины мира на основе источников. Для структурирования и представления контента используется подход на основе онтологий, который позволяет описывать не только ресурсы, но и предметные знания в стиле Semantic Web, т. е. с помощью иерархий классов, объектов и связей между ними. Основной особенностью фольклорных исследований является необходимость синхронизации переводов (создание параллельных корпусов текстов) и разметки текстов сущностями предметной области (семантическая разметка). При этом каждый корпус сопоставляется определенной народности и имеет как свой национальный язык, так и свою уникальную систему понятий об окружающем мире. Такое представление предъявляет множество нестандартных требований к платформе, таких как работа с произвольными языками, поддержка множеств онтологий, обеспечение создания и редактирования национальных предметных онтологий, семантическая разметка текстов, представление, навигация и поиск по разнородным ресурсам. Разработанная платформа предоставляет все необходимые инструменты для исследований, включая инструменты для разработки онтологий национальных предметных областей и ручного аннотирования текстов в режиме реального времени несколькими специалистами. Размещение ресурсов на платформе осуществляется на основе онтологии ресурсов, включающей такие понятия как корпус, видео- и аудиоресурсы, графическое изображение, персона, географическое место, жанр текста и т. п. Онтологии предметных областей представлены в виде иерархии, где на верхнем уровне размещается онтология универсалий, общая для всех фольклорных исследований, а наследуемые онтологии специализируются для каждого представленного национального корпуса. Веб-приложение построено на основе фреймворка Python Django и библиотеки TypeScript React, хранение данных реализовано с помощью базы данных Postgres.

Ключевые слова

онтология фольклора, редактор онтологии, инструмент разметки корпуса

Благодарности

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 20-412-540001

Для цитирования

Лисин В. А., Сидорова Е. А. Веб-платформа для фольклорных исследований на основе онтологии предметных областей // Вестник НГУ. Серия: Информационные технологии. 2021. Т. 19, № 2. С. 53–64. DOI 10.25205/1818-7900-2021-19-2-53-64

Web Platform for Folklore Research Based on Domain Ontology

V. A. Lisin¹, E. A. Sidorova²

¹ Novosibirsk State University
Novosibirsk, Russian Federation

² A. P. Ershov Institute of Informatics Systems SB RAS
Novosibirsk, Russian Federation

Abstract

In this paper, we take a close look at a web platform that provides the tools necessary for working with folklore materials and conducting scientific research based on them. Folklore studies consist of working with audio and video materials, which contain the reproduction of elements of folk art in national languages, creating specific text recordings with translation and comments, written in a public language, and building a picture of the worlds based on available resources. To structure and present this content, we use an ontology-based approach, which allows linguists to describe not only the resources, but also subject knowledge in the Semantic Web style, i.e. using hierarchies of classes, objects and relationships between them. The main feature of folklore research is the need for synchronization of translations, which is achieved by creating a parallel corpora of texts, and the ability to label texts with entities of the subject area, which is called semantic markup. Moreover, each corpus is connected with a certain nationality and has both its own national language and unique system of concepts of the world around it. Such representation imposes many non-standard requirements for the platform, such as working with arbitrary languages, supporting many ontologies, ensuring the creation and editing of national subject ontologies, semantic text markup, presentation, navigation, and search across heterogeneous resources. The developed platform provides all the necessary tools for research, including tools for the development of ontologies in specific national subject areas and manual annotation of texts in real time by several specialists. Resources of the web-platform are located in the resource ontology, which includes such concepts as corpus, video resource, audio resource, graphic image, person, geographical location, genre of text, etc. Ontologies of subject areas are presented in the form of a hierarchy, where the ontology of universals, common to all folklore studies, is located at the top level. At the same time, inherited ontologies are specialized for each represented national corpus. The web application is built with Python Django framework and the TypeScript React library. Data storage is implemented using the Postgres database.

Keywords

folklor ontology, ontology editor, semantic markup tool

Acknowledgements

This work was carried out with the financial support of the Russian Foundation for Basic Research within in the framework of the scientific project no. 20-412-540001

For citation

Lisin V. A., Sidorova E. A. Web Platform for Folklore Research Based on Domain Ontology. *Vestnik NSU. Series: Information Technologies*, 2021, vol. 19, no. 2, p. 53–64. (in Russ.) DOI 10.25205/1818-7900-2021-19-2-53-64

Введение

Фольклорные исследования связаны с изучением аудио- и видеоматериалов, фиксирующих воспроизведение элементов народного творчества на национальных языках, создание текстовых записей с переводами и комментариями на языке общего пользования (в данной работе переводы представлены на русском языке), построение картины мира на основе источников. Для структурирования и представления контента в современных научных исследованиях все чаще используется подход на основе онтологий, который позволяет описывать не только ресурсы, но и предметные знания в стиле Semantic Web, т. е. с помощью иерархий классов, объектов и связей между ними. Структурированный (размеченный) контент в дальнейшем может быть использован в качестве базы для исследования того или иного аспекта языка, статистического анализа, проверки гипотез.

Для проведения подготовки и работы над материалом существует специальное программное обеспечение для создания корпусов. Функционал данных продуктов варьируется в зависимости от цели исследования и типа материала, начиная от создания коллекций коротких текстовых фрагментов и до формирования хорошо аннотированных корпусов текстов, в которых разметка включает типизацию текстовых фрагментов и выстраивание отношений ме-

жду ними на основе модели пользователя. Сама разметка может включать или не включать в себя метаданные – информацию о самом источнике (автор, дата, жанр и т. п.)

Так, система Brat [1] представляет собой веб-платформу для аннотирования текста, в частности добавления тегов к уже существующим текстовым документам. Простая типизация и классификация объектов в тексте позволяет создавать необходимые данные для распознавания именованных сущностей, бинарных отношений и простых информационных запросов. Данный сервис также поддерживает аннотации типа *n*-арных ассоциаций, которые используются для создания связей между любым числом фрагментов, участвующих в определенных отношениях. Детальное описание типов и свойств аннотаций может быть расширено использованием атрибутов, которые применимы к любому объекту корпуса. В свободном доступе также присутствуют такие приложения, как LightTag [2], TagTog [3] и Ova [4].

Несмотря на то что множество из приведенных систем поддерживает привязку фрагмента текста к нескольким меткам или возможность построения связей между метками, большинство из них не обеспечивают построение связей между фрагментами текста, поддержку нескольких разметок, комментирование или разнообразие уровней доступа к системе.

Данные веб-сервисы имеют аналогичный набор функций и один общий недостаток – их инструментария недостаточно для проведения полноценной исследовательской работы.

В данной работе рассматривается веб-платформа, обеспечивающая размещение фольклорных материалов и проведение научных исследований.

Особенности исследования фольклора

Процесс разметки, в общем случае, заключается в выделении и приписывании фрагментам текста определенных тегов. Тип разметки зависит от решаемой задачи, она может быть морфологической, синтаксической, анафорической, семантической, дискурсной и т. п. При фольклорных исследованиях помимо классических лингвистических задач, связанных с анализом языка народности, на котором представлен текст (изначально аудиозапись или видеозапись) фольклора, возникает задача выявления картины мира, основных понятий и сущностей и разметка на их основе фольклорного текста.

Одной из отличительных особенностей фольклорного ресурса является то, что источник, как правило, представлен на национальном языке. Это определяет специфику хранения и представления пользователю материалов. Каждый текстовый ресурс состоит из двух частей: текст на оригинальном языке и его перевод на язык общего пользования. При разметке или просмотре текста пользователь должен иметь возможность сразу видеть перевод для каждого значимого фрагмента, а при добавлении такого ресурса – осуществлять параллельную разметку.

Разрабатываемая система направлена на предметно-ориентированную семантическую разметку. Процесс разметки происходит на основе иерархии онтологий предметных областей. На основе размеченных данных пользователь имеет возможность проведения онтологического поиска и подтверждения результатов данного поиска фрагментами текста.

Аннотация текста включает множество «вхождений» элементов онтологии в текст или множество фрагментов размеченных элементами онтологии – экземплярами классов и отношений. Разметке подвергается только оригинал текста, переводная часть синхронизируется с оригиналом (в нашем подходе – построчно), и при необходимости ее разметка с некоторым приближением может быть получена за счет сопоставления размеченных фрагментов. Разметка отношений опирается на связующие слова и осуществляется между уже размеченными сущностями онтологии.

На рис. 1 приведен пример разметки фрагмента текста из произведения А. С. Пушкина «Евгений Онегин». В примере размечены два экземпляра сущностей «Персона» и «Река». Экземпляр «Онегин» был привязан к фрагменту текста на позиции X(1), экземпляр «Нева» – к позиции X(8). Было построено отношение между соответствующими фрагментами текста

на основе связующего слова на позиции 5. На основе этого отношения была создана связь между двумя экземплярами онтологии под названием «Родился в».

Рис. 1. Пример размеченного фрагмента текста
Fig. 1. Markup example on a text fragment

Параллельно разметке текста пользователь имеет возможность пополнить текущую онтологию ПрО сущностями, встречающимися в рассматриваемой предметной области. В роли таких сущностей могут выступать действующие лица, предметы быта, места, абстрактные понятия и т. п. При пополнении онтологии необходимо соблюдение строгого формализма, который в то же время обеспечивает гибкие средства описания предметной области, а также установления между ними семантических связей. Понятия предметной области выстраиваются в иерархию «общее – частное» с наследованием свойств по этой иерархии [5; 6]. В дальнейшем, достаточно указать требуемую единицу онтологии для ее привязки к отдельному фрагменту текста. Построение связей между сущностями ПрО подразумевает ведение списка возможных типов связей между ее объектами и классами. Такой подход позволяет специалисту формализовать множество возможных отношений, устраняя риск построения связей с похожими именами, что в дальнейшем может негативно повлиять на процедуру поиска.

Использование инструмента, основанного на web-технологиях, предоставляет специалистам возможность работы над текстом в параллельном режиме. В этом случае каждый участник процесса может создать свою версию разметки параллельного текста, не прерывая процесс работы другого участника. В ходе работы доступен просмотр прогресса сотрудника с дальнейшей возможностью объединить персональные разметки.

При разметке текста инструмент выделяет фрагменты, с которыми работал пользователь, наглядно представляя информацию об объектах и классах в тексте, связях между ними и прикрепленных к тексту ресурсах. Если разметка текста осуществляется на основе онтологии ПрО, то «внешняя» информация о ресурсе формируется на основе онтологии ресурсов. Таким образом, экземплярами онтологии ресурсов, прикрепленными к тексту, могут быть лица, проводившие предварительную обработку отображаемого материала, места записи текста, а также информация, касающаяся первоисточника материала. Онтология ресурсов была построена в соответствии со стандартом CIDOC CRM [7]. Основная роль CIDOC CRM

заключается в обеспечении обмена информацией и интеграции между разнородными источниками информации о культурном наследии. Он направлен на предоставление семантических определений и пояснений, необходимых для преобразования разрозненных или локализованных источников информации в согласованный глобальный ресурс.

Архитектура системы разметки

Для анализа возможных вариантов реализации системы были рассмотрены следующие программные средства:

Laravel представляет собой бесплатное программное средство с открытым исходным кодом для разработки веб-приложений модели MVC (табл. 1).

Характеристика Laravel

Таблица 1

Table 1

Laravel properties

Преимущества	Недостатки
Наличие встроенного сборщика скриптов и scss	Функционал фреймворка работает через фасады, что приводит к ошибкам IDE-систем при нахождении методов и свойств классов
Встроенный шаблонизатор	Отсутствие встроенных генераторов интерфейсов
Гибкое формирование route	Низкая производительность

Выбор Laravel обосновывается в ситуациях, когда разработчиком предъявляются особые требования к «frontend», что свидетельствует о большем вкладе средств и времени на разработку интерфейсов приложения, а также полного разделения «frontend» от «backend».

Yii является высокоэффективным компонентно-структурным PHP-фреймворком для разработки крупных приложений (табл. 2).

Характеристика Yii

Таблица 2

Table 2

Yii properties

Преимущества	Недостатки
Низкий старт разработки	Строгое форматирование route
Встроенные решения для интерфейсов	Плохое развитие
Наличие генератора моделей и контроллеров	Неразрывность «backend» и «frontend»

Выбор Yii можно осуществить в ситуациях, когда нет требований к «frontend» части и дальнейшему развитию системы и проект нужно выполнить в сжатые сроки.

Универсальный фреймворк с открытым исходным кодом Spring является лидером рейтинга популярности среди средств разработки на языке JAVA. Данное средство разработки стало широко распространенным благодаря выступлению в качестве альтернативы и замены модели Enterprise JavaBeans, предоставляя большую свободу создания проектов (табл. 3).

Таблица 3

Характеристика Spring

Table 3

Spring properties

Преимущества	Недостатки
Слабо связанная система	Конфигурации в xml
Принцип инверсии управления	Сложность реализации
	Низкая производительность

Данный фреймворк специализирован на создании гибких систем, так как слабая связь его компонентов позволяет легко изменить реализацию, а инверсия управления дает возможность использования Spring в любом приложении, работающем с «spring-core». Из-за сложности проектов и конфигурации с помощью XML страдает скорость разработки и появляется необходимость использования фреймворка Spring Boot, работающего под основным фреймворком Spring.

Django появился в 2005 г. и постепенно стал одним из лучших фреймворков, который позволяет множеству разработчиков выполнять ту или иную работу в течение нескольких минут (табл. 4).

Таблица 4

Характеристика Django

Table 4

Django properties

Преимущества	Недостатки
Легкая масштабируемость	Необходимость знания всей системы для разработки
Защита от ошибок безопасности	Монолитность
Быстрота развертки проекта	Низкая производительность

Django является подходящим решением для быстрого и качественного создания проектов, но его монолитность способна ограничить рост проекта в случаях, когда инструментов недостаточно для реализации необходимого функционала.

В качестве программных средств реализации был выбран фреймворк Django.

Данное ПО находится в свободном доступе и заметно ускоряет процесс проектирования и разработки благодаря десятку дополнительных функций. В их число входят пакеты аутентификации пользователей, карты сайта, администрирования содержимого и работы с каналами. Данные модули позволяют при необходимости легко масштабировать проект.

Также, используя Django, разработчик гарантирует защиту от ошибок безопасности, ставящих под угрозу веб-приложение: кросс-сайт подлоги, инъекции, кросс-сайтовый скриптинг и т. д. Ключом данных гарантий являются отсутствие чистого SQL-кода в приложении и система пользовательской аутентификации, предустановленная в проекте.

Данный фреймворк наилучшим образом подходит для работы со средними и высокими трафиками, такими как трафики сетей средних и крупных организаций.

Набор функций, предоставляемый данным ПО, удовлетворяет требованиям, представленным в исходных данных работы, что нивелирует недостаток монолитности.

Важно отметить, что данная платформа была разработана для решения определенного набора задач в сложной и многоуровневой новостной организации. В центре набора задач стояли три важные основы:

- 1) предоставление возможности использования простого интерфейса для работы с базами данных, текстовым форматированием и информационными ресурсами;
- 2) управление UI с помощью инструментов языка разметки HTML, CSS, а также языков программирования JavaScript и его вариаций;
- 3) быстрая и надежная система обновления и устранения проблем работы системы в короткие сроки.

Ключом предоставления данных возможностей является разработка компонентов данных, дизайна и бизнес-логики с использованием метода «loose coupling». Данная технология обеспечивает управления блоками веб-приложения независимо друг от друга (рис. 2).

Рис. 2. Архитектура разрабатываемого приложения

Fig. 2. Architecture of the web application

Было произведено разделение логики разрабатываемого приложения на два ключевых элемента: интерфейс приложения на основе React и логику приложения на основе Django REST. Данное разделение осуществлено на основе парадигмы SoC – тип архитектуры, который позволяет разработчику проводить изменения системы в ее изолированных участках, уменьшая время простоя основного функционала сервиса.

Особенности реализации веб-платформы

Для разработки системы были установлены следующие функциональные требования:

- поддержка хранения параллельных корпусов фольклорных текстов, включающих оригинальные и переведенные тексты, а также комментарии;
- поддержка комментирования специалистами фрагментов текста в процессе его разметки;
- поддержка редактора онтологий, который позволяет вводить новые классы, их атрибуты и связи;

- поддержка хранения изображений, аудио- и видеоресурсов и их связей с разметкой текстов;
- поддержка иерархии онтологий;
- поддержка разметки корпуса текстов на основе онтологии.

Для хранения разнородной информации в базе данных Postgres был выбран пакет «db_file_storage», поддерживающий файлы, такие как текст, изображения, аудиозаписи и видеозаписи. DBFS представляет собой стороннюю библиотеку, позволяющую сохранять файлы непосредственно в базу данных.

Для просмотра текстов и отображения метаданных корпуса разработанная панель навигации обеспечивает возможность наглядного отображения дерева корпусов с их содержимым. Менеджер окон предоставляет формы для редактирования сущностей в реальном времени. На рис. 3 представлено окно с информацией об экземпляре класса «Богатырские кони». Левая часть окна содержит в себе список атрибутов данного экземпляра, наследуемый из его класса. Правая часть отображает значения данных атрибутов.

Атрибут	Значение
Имя	Конь Тохонойн Ганса Тоолэна
Класс	Богатырские кони
Характеристика	Положительный
Значение	Главный
Обладает умением	
Масть	изредка-буланый
Особенность	как облако подражает

Строки:

Для связи строки и объекта перетаскивайте номер строки в данное поле

Строка #2 Чер пүдгерге,

Отношения:

Рис. 3. Пример окна объекта онтологии предметной области
Fig. 3. Window interface example of an object in a domain ontology

Реализованная система аутентификации построена на принципе использования токенов авторизации. При регистрации нового пользователя в системе его аккаунт подтверждается администратором ресурса, так как без одобрения (активации) пользователю не будет выдан токен – зашифрованная строка, содержащая в себе имя пользователя и его пароль. Токены позволяют добавить один уровень косвенности для аутентификации. Вместо того чтобы аутентифицироваться с именем пользователя и паролем для каждого защищенного ресурса, пользователь аутентифицирует этот путь один раз (в течение ограниченного сеанса), получает ограниченный по времени токен и использует его для дальнейшей аутентификации во время сеанса. Преимущества данного метода заключаются в том, что пользователь может передать токен, как только он его получит, в другую автоматизированную систему, которой

он готов доверять в течение ограниченного времени, но не передавать данные об имени пользователя и пароль.

Функция привязки отрезков текста к объектам и классам корпуса основана на использовании инструмента «drag» и созданного менеджера окон. Инструменты по выделению созданных связей, находящихся в тексте, и отображению объектов и классов, привязанных к тексту, наглядно демонстрируют пользователю прикрепленные экземпляры и классы к текстовому ресурсу. На рис. 4 представлен фрагмент параллельного текста с выделенными в нем экземплярами классов. Зеленый маркер у номера строки сигнализирует о наличии связи данной строки с элементом онтологии. При наведении курсора мыши на данный индикатор появляется информационное окно, содержащее данные о классе и экземпляре. Также при наведении происходит выделение всех связей в тексте в соответствующих фрагментах.

Страница:	1	Число строк на странице:	50	Всего строк:	1590
0	Амдыгы төлдүң* алында полча,	Прежде нынешнего поколения было,			
1	Объект: Конь Тохонайн	Позже давнего поколения было.			
2	Ганса Тоолэна	В то время, когда земля сотворялась,			
3	Класс: Богатырские арга полчаттыр.	Когда земля-вода схватывались.			
4	кони Мешалкой облужуп,	Мешалкой земля делилась.			
5	Камыспа суғ пөлүшчиган тем полтур.	Ковшом вода отделялась.			
6	Көгериш-кел, көк өлең өсчаттыр.	Зеленя, молодая трава вырастала.			
7	Ағаш пашторыңга тамчылаш-келип, чарылыш-кел,	В то время, когда на деревьях, набухая-проклёвываясь,			
8	ездиги на исчитқан тем полтур.	Зеленые листья выросли.			

Рис. 4. Пример разметки и визуализации связи в параллельном тексте
Fig. 4. Markup and visualization example of a relation in a parallel text

В соответствии с требованиями была реализована система классов и объектов. Иерархическое дерево классов и объектов привязано к каждой корневой ветке дерева корпусов. Данный тип реализации позволяет использовать индивидуальные онтологии для каждого типа корпусов.

Для реализации поддержки связей объектов базы данных с любыми объектами была реализована архитектура на основе tag Django ORM. Данный подход заключается в использовании таблицы посредника, предоставляемой ядром фреймворка Django. Этот посредник привязан ко всем объектам в базе данных и хранит информацию об индивидуальном ключе объекта и индивидуальном ключе таблицы, к которой он привязан.

Для удобства пользователя реализована возможность добавления произвольного числа типов связей. Типы связей, типы ресурсов, авторы и места не имеют конкретной привязки к определенному корпусу для их использования во всей платформе.

Окно иерархии классов и объектов, расположенное на экране просмотра и разметки параллельного текста определенного корпуса текстов, обеспечивает быструю навигацию и выбор требуемых сущностей (классов) и экземпляров.

Инфраструктура как услуга

Учитывая малое число ожидаемых одновременных подключений, был рассмотрен вариант развертывания проекта на облачном сервисе. Окружение развертывания представляет собой среду, в которой размещается веб-приложение и обеспечивается его запуск и доступ к функционалу. Данная среда состоит из:

- аппаратных решений;
- операционной системы;

- хранилища данных;
- сервера приложений.

Несмотря на то что реализация систем возможна на собственных аппаратных решениях, общим подходом к развертыванию является применение типа удаленного доступа к вычислительным средствам под названием «Инфраструктура как Услуга» (IaaS). Большое число IaaS поставщиков позволяют выбрать предустановленные полноценные рабочие окружения, такие как Django, поддерживая его как часть своего пакета «Платформа как Услуга» (PaaS).

Были рассмотрены основные хостинг-провайдеры услуг запуска Django приложений, представленные в табл. 5. При анализе учитывались основные факторы, влияющие на выбор хостинга:

- горизонтальное и вертикальное масштабирование;
- наличие модулей анализа работы;
- географическое местоположение серверов;
- цена за соответствующий тарифный план.

Таблица 5

Хостинг-провайдеры

Table 5

Hosting providers

Название	Масштабирование	Модули	Местоположение	Цена / мес., руб.
Liquid Web	горизонтальное и вертикальное	нет	США, Россия	1300
o2switch	горизонтальное и вертикальное	да	Франция	370
MilesWeb	горизонтальное и вертикальное	нет	США, Англия	570
Heroku	горизонтальное и вертикальное	да	США, Германия, Россия	1200

В результате анализа средств для развертывания системы был выбран сервис Heroku. Функционал платформы позволяет разработчику проводить горизонтальное и вертикальное масштабирование проекта, арендуя новые виртуальные контейнеры и подключая дополнительные модули к системе, а богатая библиотека подключаемых средств наблюдения и анализа работы веб-приложений предоставляет инструменты для детального мониторинга активности. Heroku является популярным облачным инструментом «Платформа как Сервис». Благодаря развитой веб-инфраструктуре платформа устраняет необходимость в балансировки нагрузки и обслуживании серверов.

Принцип обработки Django веб-приложений заключается в работе нескольких изолированных Unix-контейнеров, организующих окружение для системы. Данные контейнеров имеют эфемерную файловую систему, проводящую самообновление и самоочистку при каждом перезапуске. Heroku, используя встроенный балансировщик нагрузок, равномерно распределяет поступающий трафик среди всех виртуальных контейнеров.

Взаимодействие с хостингом Heroku было выполнено при помощи терминала, предоставляемого платформой. Данный инструмент позволяет контролировать версии приложения, загружать и выгружать файлы конфигураций и вести историю внесенных изменений.

Для разработки приложения была использована основанная на файлах база данных SQLite. Несмотря на то что ее применение невозможно в опубликованном проекте, при развертывании приложения база данных была автоматически мигрирована в Postgres посред-

вом инструментов, предоставляемых хостингом. В результате сервер веб-платформы по созданию и редактированию семантически размеченных корпусов был развернут на облачном сервисе с режимом доступа <https://neofront.herokuapp.com/>.

Заключение

В работе представлены результаты исследований, связанных с обеспечением семантической разметки фольклорных текстов на основе онтологии. Разработано веб-приложение по разметке параллельных корпусов на основе онтологий ресурсов и предметных областей. Онтологии предметных областей представлены в виде иерархии, где на верхнем уровне размещается онтология универсалий, общая для всех фольклорных исследований, а наследуемые онтологии специализируются для каждого представленного национального корпуса.

Множество нестандартных требований к платформе, такие как работа с произвольными языками, поддержка множества онтологий, обеспечение создания и редактирования национальных предметных онтологий, семантическая разметка текстов, представление, навигация и поиск по разнородным ресурсам, были обеспечены.

Разработанная платформа предоставляет все необходимые инструменты для исследований, включая инструменты для разработки онтологий национальных предметных областей и ручного аннотирования текстов в режиме реального времени несколькими специалистами. Размещение ресурсов на платформе осуществляется на основе онтологии ресурсов, включающей такие понятия, как корпус, видео- и аудиоресурсы, графическое изображение, персона, географическое место, жанр текста и т. п.

Дальнейшее развитие платформы предполагает использование графовых баз данных для размещения онтологий предметных областей и онтологии ресурсов.

Список литературы

1. **Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, Jun'ichi Tsujii.** Brat: a Web-based Tool for NLP-Assisted Text Annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, 2012, p. 102–107.
2. **Tobias Daudert.** A Web-based Collaborative Annotation and Consolidation Tool. In: Proceedings of the 12th Conference on Language Resources and Evaluation, 2020, p. 7053–7059.
3. **Juan Miguel Cejuela, Peter McQuilton, Laura Ponting, Steven J Marygold, Raymund Stefancsik, Gillian H Millburn.** Tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database the Journal of Biological Databases and Curation*, 2014, no. 2014, p. bau033.
4. **Mathilde Janier, John Lawrence, Chris Reed.** OVA+: An Argument Analysis Interface. *Frontiers in Artificial Intelligence and Applications*, 2014, no. 266, p. 463–464.
5. **Кустова Г. И., Ляшевская О. Н., Падучева Е. В., Рахилина Е. В.** Семантическая разметка лексики в Национальном корпусе русского языка: принципы, проблемы, перспективы // Национальный корпус русского языка: 2003–2005. Результаты и перспективы. М., 2005. С. 155–174.
6. **Гриневиц А. А.** Онтология «Образы медвежьих песен хантов» для портала «Фольклор народов Сибири» // Вестник музыкальной науки. 2016. № 3 (13). С. 95–99.
7. **Meghini C., Doerr M.** A first-order logic expression of the CIDOC Conceptual Reference Model. *International Journal of Metadata, Semantics and Ontologies*, 2018, no. 13 (2), p. 131–149.

References

1. **Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, Jun'ichi Tsujii.** Brat: a Web-based Tool for NLP-Assisted Text Annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, 2012, p. 102–107.
2. **Tobias Daudert.** A Web-based Collaborative Annotation and Consolidation Tool. In: Proceedings of the 12th Conference on Language Resources and Evaluation, 2020, p. 7053–7059.
3. **Juan Miguel Cejuela, Peter McQuilton, Laura Ponting, Steven J Marygold, Raymund Stefancsik, Gillian H Millburn.** Tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database the Journal of Biological Databases and Curation*, 2014, no. 2014, p. bau033.
4. **Mathilde Janier, John Lawrence, Chris Reed.** OVA+: An Argument Analysis Interface. *Frontiers in Artificial Intelligence and Applications*, 2014, no. 266, p. 463–464.
5. **Kustova G. I., Lyashevskaya O. N., Paducheva E. V., Rakhilin E. V.** Semantic markup of vocabulary in the National Corpus of the Russian language: principles, problems, prospects. In: National corpus of the Russian language: 2003–2005. Results and prospects. Moscow, 2005, p. 155–174. (in Russ.)
6. **Grinevich A. A.** Ontology “Images of Khanty bear songs” for the portal “Folklore of the peoples of Siberia”. *Bulletin of Musical Science*, 2016, no. 3 (13), p. 95–99.
7. **Meghini C., Doerr M.** A first-order logic expression of the CIDOC Conceptual Reference Model. *International Journal of Metadata, Semantics and Ontologies*, 2018, no. 13 (2), p. 131–149.

Материал поступил в редколлегию

Received

11.03.2021

Сведения об авторах

Лисин Владислав Александрович, студент магистратуры, Новосибирский государственный университет (Новосибирск, Россия)
vladlisin2@gmail.com

Сидорова Елена Анатольевна, кандидат физико-математических наук, старший научный сотрудник, Институт систем информатики им. А. П. Ершова Сибирского отделения Российской академии наук, Лаборатория ИИ (Новосибирск, Россия)
Isidorova@iis.nsk.su

Information about the Authors

Vladislav A. Lisin, Master's Student, Novosibirsk State University (Novosibirsk, Russian Federation)
vladlisin2@gmail.com

Elena A. Sidorova, PhD, Senior Researcher, A. P. Ershov Institute of Systems informatics of the Siberian Branch of the Russian Academy of Sciences, AI laboratory (Novosibirsk, Russian Federation)
Isidorova@iis.nsk.su