

Рубайло Андрей Валерьевич, Косенко Максим Юрьевич

ПРОГРАММНЫЕ СРЕДСТВА ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

В статье рассмотрены существующие инструменты, предназначенные для извлечения именованных сущностей из текстов на естественном языке. Проведено сравнение рассматриваемых инструментов с целью выявления наиболее подходящего из них для решения задачи извлечения именованных сущностей из неразмеченных русскоязычных текстов. Обоснована практическая эффективность Томита-парсера для решения задач по извлечению именованных сущностей из неразмеченных русскоязычных текстов.

Адрес статьи: www.gramota.net/materials/1/2016/12/23.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2016. № 12 (114). С. 87-92. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2016/12/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

IMPROVEMENT OF STATE SUBSIDIES OF DAIRY CATTLE BREEDING

Petrov Aleksei Anatol'evich

*Ulyanovsk State Agricultural Academy named after P. A. Stolypin
paa_78@mail.ru*

The paper analyzes the current state of state subsidies of production and sale of milk. The article shows drawbacks of the existing system of subsidies associated with their conditions. It is proposed to expand availability of subsidies, which will facilitate their obtaining by cost-beneficial producers, increase stability and profitability of milk production. The author analyzes and proves effectiveness of implementation of the proposed measures and conditions.

Key words and phrases: state-run program; state support; subsidies; development of agriculture; dairy cattle breeding.

УДК 004.912

Технические науки

В статье рассмотрены существующие инструменты, предназначенные для извлечения именованных сущностей из текстов на естественном языке. Проведено сравнение рассматриваемых инструментов с целью выявления наиболее подходящего из них для решения задачи извлечения именованных сущностей из неразмеченных русскоязычных текстов. Обоснована практическая эффективность Томита-парсера для решения задач по извлечению именованных сущностей из неразмеченных русскоязычных текстов.

Ключевые слова и фразы: извлечение именованных сущностей; обработка текста; обработка информации; автоматизация; Томита-парсер; Named Entity Recognition; GATE; PullEnti SDK; Eureka Engine.

Рубайло Андрей Валерьевич

Косенко Максим Юрьевич

*Челябинский государственный университет
daiwin@mail.ru; kosenko@csu.ru*

ПРОГРАММНЫЕ СРЕДСТВА ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

1. Введение

За последние 10 лет благодаря активному развитию сети Интернет количество информации значительно выросло. Об этом можно судить по публикациям исследований за 2011 и 2015 года [5; 8]. В 2011 году было опубликовано исследование, согласно которому, в 2007 году человечество хранило на всех своих цифровых устройствах и носителях примерно $2,4 \times 10^{21}$ бит информации, то есть 300 экзабайт [5]. Объем данных, переданных по сети Интернет за 2015 год, находится на отметке 1000 экзабайт и достигнет 2300 экзабайт к 2020 году. Такие данные приводит компания CISCO [8].

Столь большие объёмы информации и стремительные темпы прироста увеличивают потребность в автоматическом извлечении структурированных данных из машиночитаемых документов на естественном языке, так как, чтобы человек мог воспринимать такие объёмы информации, ее необходимо обрабатывать, структурировать, тем самым повышая ее информативность.

Извлечение именованных сущностей (ИС) (Named Entity Recognition) – одна из фундаментальных задач обработки информации, извлечения структурированных данных из неструктурированных документов. Извлечение более сложных структур, таких как отношения или события, нуждается в извлечении ИС как в предшествующей операции обработки текста. Под термином «именованная сущность» (Named Entity) понимается слово или n -грамма определенного типа, имеющая референт, то есть принадлежность кому-то или чему-то.

В данной работе представлен краткий обзор и сравнительный анализ существующих инструментальных средств, используемых для извлечения именованных сущностей. В обзор включены инструментальные системы, которые вплоть до настоящего времени поддерживаются и применяются для обработки текстов.

2. Характеристика существующих решений

2.1. Томита-парсер

Томита-парсер [1] является инструментом извлечения структурированных данных из текста на естественном языке. Система разрабатывается компанией Яндекс и используется в собственных сервисах, таких как Яндекс.Почта, Яндекс.Новости, Яндекс.Авто и Яндекс.Работа.

Томита носит название в честь японского ученого Масару Томита, разработавшего алгоритм GLR (Generalized Left-to-right) [7], который используется в программе для разбора входных данных.

Очередность этапов обработки данных фиксирована: токенизация, морфологический анализ, лексический анализ (распознавание ключевых слов и словосочетаний по внутренним словарям Томиты), синтаксический разбор по заданной грамматике, интерпретация (вывод результатов). Для распознавания необходимых конструкций из текстов используются правила, написанные на языке расширенных контекстно-свободных грамматик.

Для запуска парсера необходимо на вход подать сам анализируемый текст, а также подключить словарь и как минимум одну грамматику. Объем словаря и сложность грамматики зависят от целей анализа. Файл грамматики состоит из шаблонов, написанных на внутреннем языке *Томита-парсера*.

Правила (грамматики) в общем случае имеют вид:

$$S \rightarrow S_1 \dots S_n.$$

В левой части (до \rightarrow) указывается нетерминал S , который строится из терминалов (S_1, S_n) и должен хотя бы один раз встретиться в левой части правила. Если нетерминал встречается только в левой части и никогда – в правой, значит, это вершина грамматики.

Терминалы « S_n » в свою очередь имеют следующую конструкцию:

$$X < P_1, \dots, P_n > \text{interp}(K_1, \dots, K_n),$$

где « X » – это лемма или одно из зарезервированных имен терминалов (*AnyWord, Noun, Verb* и т.д.).

« P_n » – это помета-ограничение, к которым относятся тип ключевого слова или их множество, грамматические характеристики или их объединения, согласование в грамматиках, регулярные выражения и отдельная помета «no_hom», которая требует, чтобы нетерминал состоял из омонимов с одной частью речи.

« K_n » – это имена атрибутов фактов, в которые при интерпретации предаются терминалы « S_n » из выявленного нетерминала « S ».

Эти шаблоны описывают в обобщенном виде цепочки слов, которые могут встретиться в тексте. Кроме того, грамматики определяют, как именно нужно представлять извлеченные факты в итоговом выводе.

Приведем пример грамматики, используемой *Томита-парсером* для выделения из текста конструкций вида «ОАО “Организация”».

orgDiscs -> *Word*<kwtype="org_дискрипторы">;

orgName -> *Word*<h-reg1, quoted>;

ORG -> *orgDiscs interp (+Org.Name) orgName interp (+Org.Name::not_norm)*;

Нетерминал «*orgDiscs*» получает ссылку на пользовательский словарь «org_дискрипторы», который объявлен в корневом словаре.

В словарях содержатся ключевые слова, которые используются в процессе анализа грамматиками. Каждая статья этого словаря задает множество слов и словосочетаний, объединенных общим свойством.

Нетерминал «*OrgName*» ищет все слова, попадающие под терминал *Word*, то есть любые слова, состоящие из букв русского или латинского алфавита, в том числе написанные через дефис, и под две пометы-ограничения:

- 1) «h-reg1» говорит о том, что первый символ слова должен быть в верхнем регистре;
- 2) «quoted» сообщает о том, что слово должно быть заключено в кавычки.

Нетерминал «*ORG*» является вершиной грамматики, так как никогда не встречается в левой части правила. По факту он собирает из двух нетерминалов итоговое правило.

«*interp (+Org.Name)*» обозначает, что данный терминал, идущий перед, при интерпретации результатов будет записан в факт «*Org*» в атрибут «*Name*», а знак «+» – то, что он будет конкатенирован в этом же атрибуте с другим терминалом этого же нетерминала (*ORG*).

Томита-парсер является удобным инструментом для извлечения ИС с помощью правил и словарей. Правила удобно разрабатывать при помощи отладочного вывода деревьев синтаксического разбора и списка срабатываний правил. В *Томите* есть готовые пометы-ограничения, граммы и зарезервированные терминалы, которые экономят много времени при написании правил. Недостатком является то, что набор помет-ограничений нельзя расширять, лемматизация не всегда корректно работает, в особенности с именами собственными, структура выходных файлов не поддается настройке, только выходной формат (*.xml, *.txt, *.proto), встраивание в другие системы и расширение собственными модулями также не предусмотрены.

2.2. GATE

General Architecture for Text Engineering (GATE) [4] – это система обработки естественного языка с открытым исходным кодом, использующая наборы компонентов на языке *Java*. Система изначально была разработана в Университете Шеффилда и развивается с 1995 года.

Система применяется для извлечения информации, ручной и автоматической семантической аннотации, анализа кореферентности, работы с онтологиями (например, *WordNet*), машинного обучения (*Weka, RASP, MAXENT, SVM Light*), анализа потока сообщений в блогах (например, *Twitter*).

К системе *GATE* могут подключаться модули трех типов – языковые ресурсы (это документы и корпуса документов), обрабатывающие ресурсы (*processing resources* – модули, обеспечивающие логику выполнения) и визуальные ресурсы (*visual resources* – отвечают за отображение результатов и взаимодействие с пользователем). В результате работы текст разделяется на отрезки, которым приписываются аннотации с различными характеристиками.

Сама система *GATE*, как и программные модули для нее, написаны на языке *Java*. После обработки есть возможность просмотра результатов и изменения их вручную. Кроме того, можно выбрать аннотации определенного типа и просматривать только их.

Есть возможность обрабатывать тексты в различных форматах: простом текстовом, *HTML, DOC* и др. При этом разметки исходного текста (абзацы, шрифты и др.) сохраняются и могут использоваться при анализе.

Для описания правил в системе *GATE* применяется язык *Jape* [2].

Как и в *Томита-парсере*, правила на языке *Jape* состоят из левой и правой частей.

```

Rule: University1
(
  {Token.string == "University"}
  {Token.string == "of"}
  {Lookup.minorType == city}
):orgName
-->
:orgName.Organisation =
{kind = "university", rule = "University1"}

```

В левой части описывается шаблон, по которому будет проводиться поиск сущности:

```

{Token.string == "University"}
{Token.string == "of"}, описывают конкретные словоформы, которые указывают на сущность определённого типа, так называемые дескрипшены.

```

```

{Lookup.minorType == city}, описывает название города из словаря городов (аннотация Lookup служит для пометки слов, найденных в словарях).

```

Правая часть создает новую аннотацию типа *Organisation* с атрибутом *kind* и значением *rule*, полученным правилом *University1*. Фактически в *rule* уйдет «*University of Toronto*», если «*Toronto*» есть в словаре.

Достоинством языка правил *Jape* является то, что его можно расширять, добавляя новые типы и атрибуты аннотаций, так как в нём отсутствуют встроенные типы и атрибуты, включая лингвистические. Предполагается, что для каждого проекта будут разработаны свои типы и атрибуты, но эта гибкость – и плюс, и минус.

При архитектуре, которая используется в *GATE*, предполагается возможность обработки любых естественных языков, но на деле она не подходит гибким языкам наподобие русского, латинского, немецкого, в которых слова образуются и согласуются путем прибавления приставок, суффиксов, окончаний, так как морфология не обрабатывается. Также недостаток информации о системе требует много сил и времени для разработки своих модулей к программе.

К плюсам системы можно отнести наличие пользовательского интерфейса, свободу, которую предлагает язык правил *Jape*, работу с огромным количеством входных форматов, даже таких «проблемных» для парсинга как *pdf* и *docx*, кроссплатформенность за счет *java*-платформы, на которой система была разработана.

2.3. PullEnti

PullEnti SDK [6] – это комплект средств разработки от компании ООО «Семантик», предназначенный для решения задачи анализа текста и выделения именованных сущностей из неструктурированных русскоязычных текстов в рамках информационных систем, разрабатываемых на *.NET Framework 2.0* и выше. В соревновании *FactRuEval* на конференции «Диалог-2016» данный продукт занял первое место на дорожках T1, T2, T2-m и второе место – на T1-1 [Ibidem]. На практике используется в ряде крупных проектов, таких как «Доктор Ватсон», «Правовая экспертиза», АИС Госдумы РФ, а также сервис сравнения диссертаций в рамках проекта «Диссернет».

SDK состоит из общей и специализированной частей. Общая часть содержит реализацию общих алгоритмов морфологического и синтаксического анализа, а также поддержку модели данных. Специализированная часть состоит из отдельных сборок (анализаторов), реализующих выделение сущностей определённых типов (персоны, организации и др.). Такие сборки подключаются динамически как плагины, благодаря чему система может расширять свой функционал без перекомпилирования. Например, семантический анализ реализован в виде такой отдельной сборки.

При анализе исходный текст разбивается на токены в виде двунаправленного списка. В дальнейшем токены объединяются в «метатокены», представляющие более крупные конструкции, в частности, сущности.

Выделение сущностей основано на правилах. Однако для некоторых типов сущностей можно подгружать внешние словари (онтологии), содержащие описания существующих сущностей, и тогда система при выделении будет пытаться привязываться к внешним сущностям.

Вся морфология реализуется в одной сборке *EP.Morphology.dll*, которая может использоваться абсолютно независимо от *SDK Pullenti*, но не наоборот. Здесь морфология производится для каждого токена отдельно, независимо от окружающих его токенов. То есть это *POS-Tagger*, выдающий для каждого токена словоформы всевозможные морфологические варианты. Вопрос интерпретации и уменьшения множества вариантов решается вне *POS-Tagger*. Например, ядро *SDK Pullenti* при анализе именных групп оставляет только непротиворечивые корневому слову варианты и т.п.

Можно отметить следующие особенности:

- поддерживаются русский, украинский и английский языки;
- для неизвестных слов выдаются варианты;
- производится лексикографическая корректировка. Например, если в слове на кириллице одна буква заменена на латинскую, аналогичную по внешнему виду, то производится соответствующая корректировка. Также корректируются буквы с ударениями, замена апострофа на «ъ», «ё» на «е» и другие случаи;

- для токена предлагается вариант леммы;
- есть функция получения для леммы всех вариантов словоформ;
- словарь для каждого языка занимает в памяти от 100 до 150 Мб, словари можно динамически загружать и выгружать.

Для задач, где требуется обрабатывать множество текстов и хранить получаемые сущности, *SDK* предлагает специальный базовый класс *RepositoryBase*, облегчающий реализацию хранилища сущностей. Данный класс берёт на себя функции отождествления новых данных со старыми данными и поддержки непротиворечивости семантической сети – всё множество выделяемых сущностей представляет собой ориентированный граф, так как значениями атрибутов могут выступать другие сущности.

К плюсам можно отнести то, что система показывает очень хорошие результаты, о чем можно судить по *FactRuEval* и демо-странице проекта.

К минусам можно отнести отсутствие хорошей документации для разработчиков, а та, которая есть, описывает лишь подключение к проекту и архитектуру библиотеки, для *SDK* такой документации недостаточно, требуется как минимум описание методов, аргументов, хотя бы форума с активными участниками. Высокий порог вхождения и минимум документации делают систему недоступной большинству потенциальных пользователей.

2.4. Eureka Engine

Eureka Engine – это коммерческая модульная система лингвистического анализа текстов, предназначенная для извлечения структурированной информации из неструктурированных данных значительных объемов. Используется в решениях для управления корпоративной информацией *HP Autonomy* и системе мониторинга СМИ и социальных медиа *Brand Analytics*.

Eureka Engine включает в себя следующие модули: определение языка сообщения (24 языка), автоматическое определение тональности документа, определение тематики, выделение именованных сущностей и имен собственных (NER), нормализация слов, разметка частей речи.

Доступ к системе платный по подписке и возможен двумя способами:

1. Как внешний сервис с доступом через стандарт *REST API*.
2. Самостоятельная система, разворачиваемая внутри закрытой сети компании.

Использование *REST API* позволяет очень быстро интегрироваться в логику существующих в компании процессов, арендуя обработку данных, с возможностью увеличения квоты обрабатываемых данных переходом на другой тарифный план.

Если у компании высокие требования к защите доступа к данным или ежемесячные потоки информации составляют сотни тысяч и миллионы документов, то разумнее использовать вариант развертывания самостоятельной системы внутри корпоративной сети, что снизит нагрузку на канал связи, ускорит обработку данных, так как задержка при передаче данных на обработку снизится и позволит обезопасить данные внутренними политиками безопасности компании.

На данный момент система включает в себя следующие шесть модулей обработки информации, использующих в работе статистические алгоритмы:

- модуль автоматического определения языка (*LanguageDetector*);
- модуль автоматического определения тональности (*SentiFinder*);
- модуль автоматической классификации текстов (*TextClassifier*);
- модуль нормализации слов русского языка;
- модуль морфологического анализа русского языка;
- модуль автоматического определения именованных сущностей (NER).

Модуль автоматического определения ИС реализован в виде отдельного сервиса, который классифицирует именованные сущности на пять классов и поддерживает два языка: русский и английский:

- физические лица;
- юридические лица;
- географические объекты;
- названия продуктов и брендов;
- именованные события.

Авторы системы гарантируют высокие показатели точности, полноты и *f*-меры для модуля ИС и приводят следующие числовые результаты оценки работы: точность – 86,18%; полнота – 85,78%; *f*-мера – 85,97% [3]. Также говорится о том, что этот модуль не использует словари и тезаурусы, что наводит на мысли об использовании методов машинного обучения, а не *rule-based, knowledge-based* подходов.

Чтобы извлечь именованные сущности с помощью *API*, необходимо отправить *json*-файл по протоколу *REST* на сервера *Eureka Engine* в следующем виде:

```
{ "text": "Генеральная ассамблея ООН приняла 27 марта резолюцию о территориальной целостности Украины. Об этом сообщает Agence France-Presse." }
```

Ответ будет предоставлен также в виде *json*-файла со следующей структурой:

```
[{ "i": 22, "l": 3, "ner": "ORG", "v": "ООН" },  
{ "i": 83, "l": 7, "ner": "GEO", "v": "Украина" },  
{ "i": 109, "l": 20, "ner": "ORG", "v": "Agence France-Presse" }]
```

К плюсам системы можно отнести её простоту для разработчиков при внедрении в свои системы, для малых объемов текстов достаточно приобрести подписку и обрабатывать данные через *Eureka Engine API*.

К минусам можно отнести то, что клиент не имеет доступа к параметрам и внутренним механизмам системы для более точной подстройки, высокую стоимость: самый дешевый тариф – 25 000 рублей за 50 000 документов в месяц [Ibidem].

3. Сравнение обозреваемых решений

Проводимое исследование, в рамках которого написана статья, связано с извлечением ИС из русскоязычных текстов, поэтому поддержка русского языка в продуктах является важным критерием сравнения. Русский язык относится к классу флективных языков, то есть к языкам, у которых словоизменение происходит с помощью флексии (окончания), поэтому автоматическое определение морфологии слов – непростая задача, для решения которой требуется наличие в составе продукта морфологического анализатора. Для того чтобы ИС, состоящие из нескольких слов, корректно выделялись и нормализовались, необходимо учитывать грамматическое согласование между ними.

Продукты, взятые для обзора, хоть и отличаются по типу использования (*SDK*, *SaaS* и т.д.), но так или иначе они предназначены для решения одних задач, поэтому их сравнение является обоснованным и корректным. Результаты проведенного сравнения представлены в Таблице 1.

Таблица 1.

Сравнение обозреваемых решений

Критерий оценки	<i>Томита-парсер</i>	<i>GATE</i>	<i>PullEnti</i>	<i>Eurika Engine</i>
Простота использования (1-5)	4	2	1	5
Поддержка русского языка	+	-	+	+
Метод классификации ИС	Словари и КС-грамматики	Словари и КС-грамматики	Словари и КС-грамматики	?
Морфология	+	-	+	+
Введение новых характеристик	-	+	-	-
Грамматическое согласование	+	-	+	?
Визуальный интерфейс пользователя/разработчика	-	+	-	-

Проведенное сравнение позволило определить, что *Томита-парсер* превосходит представленные в обзоре аналогичные инструменты для решения задачи извлечения именованных сущностей. Работая с данным продуктом, можно за короткие сроки построить систему правил и базу словарей для русскоязычных текстов, сосредоточившись именно на обработке текста, а не на переписывании компонентов программы под себя или диалогах со службой поддержки о некорректности данных из «черного ящика».

Для того чтобы подтвердить доводы о преимуществе *Томиты*, был проведен опыт, в ходе которого была вручную составлена тестовая выборка для 25-ти документов, основанных на публикациях русскоязычных новостных сайтов. Для извлекаемых ИС были определены пять типов: «Дата», «Адрес», «Персона», «Географический объект» и «Организация».

Оценка результатов работы *Томиты-парсера* на тестовых данных проводилась стандартными для оценки классификатора мерами: точностью (*precision*), полнотой (*recall*) и *f*-мерой (Таблица 2).

Таблица 2.

Результаты опыта

Мера оценки	Результат, %
Точность	77,7
Полнота	78,5
<i>f</i> -мера	78,13

Полученные в ходе опыта результаты полноты, точности и *f*-меры считаются высокими для задач классификации, что ещё раз подтверждает эффективность *Томиты* как инструмента для извлечения именованных сущностей.

Заключение

В работе были рассмотрены и сравнены четыре инструмента для автоматического извлечения именованных сущностей из текстов на естественных языках или построения систем с аналогичной целью на их основе.

В результате сравнения существующих продуктов *Томита-парсер* оказался наиболее подходящим, а последующий опыт доказал его практическую эффективность для решения задач по извлечению именованных сущностей из русскоязычных текстов.

Список литературы

1. **Томита-парсер. Руководство разработчика** [Электронный ресурс]. URL: <https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage/> (дата обращения: 01.12.2016).
2. **Cunningham H., Maynard D., Tablan V.** JAPE: a Java Annotation Patterns Engine. Second edition. Sheffield, 2000. 30 p.
3. **Eureka Engine** [Электронный ресурс]. URL: <http://eurekaengine.ru> (дата обращения: 01.12.2016).
4. **General Architecture for Text Engineering** [Электронный ресурс]. URL: <http://www.gate.ac.uk/> (дата обращения: 05.12.2016).

5. **Hilbert M.** The World's Technological Capacity to Store, Communicate, and Compute Information // Science. 2011. Vol. 332. Iss. 6025. P. 60-65.
6. **PullEnti** [Электронный ресурс]. URL: www.pullenti.ru (дата обращения: 04.12.2016).
7. **Tomita M.** LR Parsers for Natural Languages // COLING: 10th International Conference on Computational Linguistics: Proceedings of COLING 84. California, 1984. P. 354-357.
8. **White paper: Cisco VNI Forecast and Methodology, 2015-2020** [Электронный ресурс]. URL: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html> (дата обращения: 30.11.2016).

SOFTWARE TOOLS FOR INFORMATION EXTRACTION FROM NATURAL-LANGUAGE TEXTS

Rubailo Andrei Valer'evich
Kosenko Maksim Yur'evich
Chelyabinsk State University
daiwin@mail.ru; kosenko@csu.ru

The article describes the existing tools for extracting named entities from natural-language texts. A comparison of the considered tools to identify the most suitable of them to solve the task of extracting named entities from non-formatted Russian-language texts is carried out. The authors substantiate practical efficiency of *Tomita-parser* to solve tasks of extracting named entities from non-formatted Russian-language texts.

Key words and phrases: extraction of named entities; word processing; data processing; automation; Tomita-parser; Named Entity Recognition; GATE; PullEnti SDK; Eureka Engine.

УДК 17.018.22

Культурология

В статье анализируются традиционные ценностно-нормативные культурные модели, утрачивание которых сегодня становится главной причиной кризиса духовных оснований современной отечественной культуры. На социальном уровне этот процесс – причина культурной аномии общества. На уровне личностей это – кризис культурной идентичности. Статья направлена на выявление педагогических возможностей традиционной народной татарской культуры и перспектив ее использования в современных поликультурных условиях России на примере опыта Республики Татарстан.

Ключевые слова и фразы: национальная культура; педагогические возможности; поликультурное пространство; татары; Татарстан.

Сайфутдинова Гузель Борисовна, к.и.н.

Казанский государственный энергетический университет
guzelsai@mail.ru

ПЕДАГОГИЧЕСКИЕ ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ ТРАДИЦИОННОЙ НАРОДНОЙ ТАТАРСКОЙ КУЛЬТУРЫ В СОВРЕМЕННОМ ПОЛИКУЛЬТУРНОМ ПРОСТРАНСТВЕ РОССИИ

Педагогический потенциал традиционной народной культуры обуславливается олицетворенными в ней незаменимыми ценностями, актуальными и в наши дни, например, любовь к Родине, природе, семье, родительству, труду, толерантность к наследию своего и других народов. Ценности традиционной культуры разных народов своеобразно проявляются в самобытном наследии этносов: памятники зодчества, народная музыка, танцы, фольклор и др. В Указе Президента РФ «О Стратегии государственной национальной политики Российской Федерации на период до 2025 года» [1] сформулированы важнейшие задачи деятельности государственных структур по укреплению государственного единства и целостности России, сохранению этнокультурной самобытности ее народов. Стратегия направлена на активизацию всестороннего сотрудничества народов Российской Федерации, развития их национальных языков и культур как основы патриотического и духовно-нравственного воспитания. Для решения данного вопроса созданы органы государственного управления культурой и образованием, дома народного творчества, культурно-досуговые центры и другие социально-культурные учреждения. Неизменными стали конкурсы и фестивали, посвященные традиционной культуре, как на российском, так и на региональном уровне. Потребности современной социальной практики обусловили повышение внимания к формированию этнокультурной компетентности кадров, что нашло отражение в новых федеральных государственных образовательных стандартах по направлениям высшего профессионального образования, как социогуманитарным, так и техническим.

Выбор татарской культуры для изучения опыта сохранения, развития и реализации педагогического потенциала традиционной народной культуры был обусловлен тем, что татары – второй по численности этнос России и Татарстана. Давно и успешно осуществляется организуемая и поддерживаемая государством