КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

УДК 80/81

ПРОБЛЕМА РАСШИРЕНИЯ ФУНКЦИОНАЛА ИНФОРМАЦИОННОГО ЛИНГВИСТИЧЕСКОГО РЕСУРСА

О. В. Донина, В. В. Филатов

Воронежский государственный университет

Поступила в редакцию 31.01.2019 г.

Аннотация. В статье предлагается решение задачи расширения функционала информационного лингвистического ресурса (а именно: определение оптимального расположения новых данных, настройка доступа к БД и обновление системы пополнения базы данных), возникшей в связи с использованием новых источников данных в рамках расширяющейся теории криптоклассного анализа. В результате работы: 1) возможность получения раздельного доступа к источникам информации реализована в виде разделов системы, доступ к каждому из которых осуществляется через главное меню; 2) стала доступна загрузка данных в новые разделы системы; 3) реализована возможность использования .xls и .xlsx в качестве форматов загружаемых файлов и т.д. Указанные изменения позволили адаптировать систему для осуществления работы с данными, полученными из лингвистических корпусов GloWbE, NOW и iWeb, а также сделать ее более функциональной и удобной для использования.

Ключевые слова: информационный лингвистический ресурс, PHP, SQL, JavaScript, язык разметки HTML, язык формирования внешнего вида документа CSS, технология AJAX, корпусные исследования, криптоклассный анализ.

Annotation. The paper proposes a solution to the problem of expanding the functionality of the linguistic resource (namely, determining the optimal location of new data, setting up access to the database and updating the database replenishment system). Some implemented changes are the following: 1) the possibility of obtaining separate access to information sources is implemented in the form of system sections, 2) data loading into new system sections has become available; 3) the possibility to use .xls and .xlsx as uploadable file formats has been implemented, etc. These changes made the database more adaptable to work with data obtained from the linguistic corpora such as GloWbE, NOW and iWeb, as well as to make it more functional and convenient for users.

Keywords: linguistic resource, PHP, SQL, JavaScript, HTML, CSS, AJAX, corpora studies, cryptotype.

ВВЕДЕНИЕ

В процессе проведения исследований в области языковой категоризации, предполагающих обработку больших массивов лингвистических данных, возникла необходимость систематизирования полученных результатов. Для выполнения данной задачи был создан лингвистический информационный ресурс «COEL» (Cryptotypes of the English Language) «Криптоклассы английского языка».

В результате изучения литературы по вопросам разработки лингвистических информационных ресурсов [1, 2, 3], мы пришли к выводу о том, что современные системы имеют следующую архитектуру:

1. База данных;
2. Веб-приложение;
3. Пользовательский интерфейс.

Использование данной структуры является наиболее логичным подходом к созданию лингвистических систем, так как работа конечного продукта не зависит от конфигурации пользовательской машины. Более того,

при изучении строения системы «COEL» было выявлено, что данный ресурс обладает аналогичной общей архитектурой.

Указанная информационная системы была разработана в качестве инструмента для исследовательской работы в рамках теории криптоклассного анализа на материале данных сочетаемости абстрактных имен английского языка. Возможности «COEL» позволяют сравнивать количественные характеристики полученных в ходе исследования данных и проводить статистический анализ.

Структура рассматриваемой системы подходит для работы с информацией, полученной из корпуса COCA (Corpus of Contemporary American English). Однако в связи с появлением новых корпусов (т.е. лингвистических баз данных) на площадке «corpus.byu.edu» [https://corpus.byu.edu/corpora.asp] появилась потребность расширения функционала информационной системы «COEL» для внедрения и разграничения данных, полученных из ресурсов «GloWbE» (The corpus of Global Web-based English), «NOW» (News On the Web) и «iWeb».

В данной работе будет описан порядок действий, связанных с подготовкой системы «COEL» к работе с новыми данными.

МАТЕРИАЛЫ И МЕТОДЫ

Стоит описать характер данных, содержащих в ИС «COEL», которая была специально создана для работы с большим количеством отобранного лингвистического материала и его анализа в рамках криптоклассной теории. Под криптоклассом понимается единица описания языковой категоризации, а точнее «скрытая языковая категория, характеризующаяся тождеством семантического признака составляющих ее слов и проявляющаяся в особенностях лексико-синтаксической сочетаемости» [4, с. 14–15]. Значения имен, составляющих криптокласс, различаются, но их дискурсивное поведение является однотипным, что обусловлено проявлением категориального признака. Одни имена – эталоны, для которых признак данного класса не является скрытым. Например, слово needle (иголка) является эталоном криптокласса «Res Acutae» («острое»). Другие имена – те, которые обозначают не только предметы физического мира, но и непредметные, абстрактные сущности, системообразующий признак которых скрыт. Такие имена характеризуются по аналогии с эталонами. Так, употребление имени pain в таких словосочетаниях, как «an acute pain» и т.д., указывает на способность боли (pain) (в системе такие имена обозначаются *nouns*) быть острой (acute) (в системе такие слова называются *classifiers*), что говорит о вхождении данного имени в криптокласс «Res Acutae» («острое»).

Функциями «COEL» является хранение, обработка, редактирование, расчет и отображение данных о распределении имен существительных по криптоклассам английского языка. Данные, собранные в «COEL», могут быть использованы для сравнения количественных характеристик имен, что позволяет искать объяснение причины распределения существительных по различным криптоклассам английского языка. Более того, информация, извлеченная из системы, используется для определения меры межъязыковой эквивалентности [5], последующей визуализации результатов [6], в том числе методами Data Mining [7], а также может способствовать определению вероятности появления в дискурсе того или иного криптокласса или новых метафорических высказываний.

Структура рассматриваемой системы подходит для работы с информацией, полученной из корпусов, содержащих примеры употребления одного варианта английского (таких, как COCA [8]). Однако в связи с увеличением числа исследуемых ресурсов появилась необходимость адаптировать «COEL» для размещения и работы с данными, извлеченными из корпусов GloWbE [9], NOW [10] и iWeb [11], которые содержат примеры словоупотреблений различных вариантов английского языка. В рамках статьи будет рассмотрено, как нами были решены 2 задачи расширения функционала системы, связанного с появлением новых данных: 1) определение оптимального расположения новых данных и

настройка доступа к БД и 2) обновление системы пополнения базы данных.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Одной из основных структурных составляющих информационной системы «COEL» является база данных, состоящая из следующих таблиц: 1. compatibility; 2. contexts; 3. classifiers; 4. cryptoclasses; 5. cryptoclasses_ classifiers; 6. nouns; 7. stats; 8. tmp.

Каждая из этих таблиц выполняет свою функцию. Так, таблицы *nouns* и *classifiers* содержат информацию о существительных и классификаторах соответственно. *Compatibility* хранит данные о том, как сочетаются указанные выше группы существительных и классификаторов, а *contexts* содержит список примеров данной сочетаемости.

Работает это следующим образом. Таблицы базы данных системы «COEL» содержат по несколько полей. Так, в таблице *compatibility* данные размещены в виде трех столбцов: noun_id (идентификатор существительного), classifier_id (идентификатор классификатора) и id (идентификатор пары). Условие существования идентификатора пары (id) a и b – наличие в базе данных контекста, в котором классифицируемым существительным является a, а классификатором – b. Получая запрос, в котором идентификаторы существительного и классификатора, например, равны 533 (Pain) и 351 (Acute) соответственно, система определяет идентификатор данной пары (id). Таблица contexts содержит поле pair_id, информация в котором соответствует id из compatibility. Поле context таблицы contexts содержит контексты, каждый из которых соответствует своему идентификатору пары (pair_id).

При расширении функционала существующие таблицы были переименованы с использованием префикса, который указывает на источник данных (т. е. корпус), а таблицы, предназначенные для хранения новых данных, были созданы в той же БД с использованием префиксов по аналогичному принципу. По своей структуре новые таблицы идентичны предназначенным для хранения данных, полученных из COCA.

Таким образом, список таблиц обновился и выглядит следующим образом (табл.1).

Для реализации возможности перехода по разделам системы «COEL», меню, всегда доступное пользователю, было дополнено выпадающим списком (рис. 1), используя который, пользователь может в любой момент выбрать нужный источник данных.

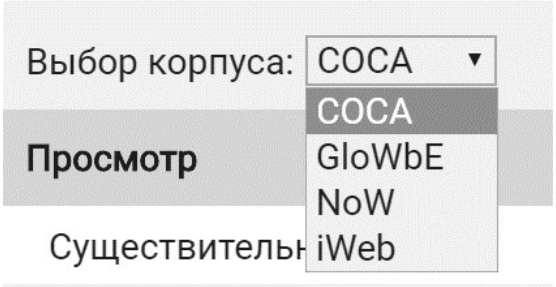

Рис. 1. Выпадающее меню

Таблица 1

Обновленный список таблиц базы данных

Данные из COCA	Данные из GLOWBE	Данные из NOW	Данные из iWeb
coca_compatibility	glowbe_compatibility	now_compatibility	nwbc_compatibility
coca_contexts	glowbe_contexts	now_contexts	nwbc_contexts
coca_classifiers	glowbe_classifiers	now_classifiers	nwbc_classifiers
coca_cryptoclasses	glowbe_cryptoclasses	now_cryptoclasses	nwbc_cryptoclasses
coca_cryptoclasses_ classifiers	glowbe_cryptoclasses_ classifiers	now_cryptoclasses_ classifiers	nwbc_cryptoclasses_ classifiers
coca_nouns	glowbe_nouns	now_nouns	nwbc_nouns
coca_stats	glowbe_stats	now_stats	nwbc_stats
coca_tmp	glowbe_tmp	now_tmp	nwbc_tmp

```
<select class="change-db-selector" name="database" onChange="document.forms.change_db.submit()">
    <option value="1"<?php if (!$_SESSION['database'] || $_SESSION['database'] == 1) echo ' selected'?>>COCA </option>
    <option value="2"<?php if ($_SESSION['database'] == 2) echo ' selected' ?>>GloWbE </option>
    <option value="3"<?php if ($_SESSION['database'] == 3) echo ' selected' ?>>NoW </option>
    <option value="4"<?php if ($_SESSION['database'] == 4) echo ' selected' ?>>iWeb </option>
</select>
```

Рис. 2. Программный код раскрывающегося списка

На рисунке (рис. 2) показано, что выпадающее меню (контейнер select) содержит четыре тега option, каждый из которых является пунктом данного меню. При использовании данного выпадающего списка значение переменной сессии ($_SESSION) меняется на число, соответствующее номеру выбранного раздела системы.

За подключение к БД и выполнение SQL-запросов отвечает программный код, содержащийся в файле class_db.php. Функции, расположенные в данном файле, были дополнены конструкциями «if», инструкции которых выполняются при изменении значения переменной сессии. Результатом выполнения инструкции любой из представленных на рисунке ниже конструкций «if» (рис. 3) является присвоение переменной prefix значения, равного префиксу в названии таблиц базы данных с символом подчеркивания.

```
if ($_SESSION['database'] == 1) {
    $prefix = "coca_";
}
if ($_SESSION['database'] == 2) {
    $prefix = "glowbe_";
}
if ($_SESSION['database'] == 3) {
    $prefix = "now_";
}
if ($_SESSION['database'] == 4) {
    $prefix = "iweb_";
}
```

Рис. 3. Значения переменной prefix

Названия таблиц в запросах дополняются префиксом [pr]. Например, в запросе к временной таблице, доступной на странице «Загрузить из файла», используется название [pr]tmp. При переходе между разделами системы происходит автоматическое изменение значения переменной prefix на соответствующее выбранному разделу. Далее функция «str_replace» находит все «[pr]» в запросах и заменяет их на значение $prefix.

Выбор такого подхода к реализации системы переключения между разделами системы «COEL» обусловлен, в первую очередь, отсутствием необходимости повторного подключения к базе данных при смене раздела системы. К тому же, данный способ отличается простотой реализации: таблицы не содержат внешних ключей, связи между ними определены в программном коде и обновляются при выполнении той или иной функции, а использование префиксов позволяет продолжить работать с одним web-приложением.

Загрузка больших объемов данных в «COEL» доступна в разделе «Загрузить из файла». Требуемый формат файла с данными – .csv. Использование файлов данного формата является стандартным методом пополнения содержимого баз данных. Это прежде всего связано с широкой поддержкой .csv большинством существующих программных систем, а также меньшим размером по сравнению с .xls, что позволяет достаточно быстро проводить импортирование данных.

Так как при проведении криптоклассного анализа лингвистами все результаты сохраняются в форматах .xls и .xlsx, было принято решение реализовать возможность загрузки файлов таких форматов в системе «COEL». При этом форматом файла, используемого для загрузки информации в базу данных, должен оставаться .csv. Это значит, что файл, созданный пользователем, должен быть автоматически конвертирован.

Процесс загрузки контекстов в базу данных проходит в несколько этапов, а именно:

1. загрузка на сервер;
2. обработка и формирование временной таблицы;
3. проверка данных временной таблицы на предмет наличия в базе данных;
4. обновление содержимого базы данных.

Файл load_db.php (страница «Загрузить из файла») содержит форму (рис. 4).

```
<form method="POST" action="convert.php" enctype="multipart/form-data">
      <input class="select-file" name="up" size="18" type="file" value=""/>
      <input type="hidden" value="convert" name="page" />
      <div>Введите имя криптокласса: </div>
      <input class="cryptoclass-name-input" type="text" name ="name"/>
      <input type="submit" value="Загрузить" />
</form>
```

Рис. 4. Форма загрузки файла на сервер

```
$lines = file ( $uploadfile );
foreach ( $lines as $line ) {
        $words = explode('п', $line );
        if ( $words[0] != '') $noun = $words[0];
           $noun = ucfirst(strtolower(trim($noun)));
           if ( $words[1] != '') $classifier = $words[1];
              $classifier = ucfirst(strtolower(trim($classifier)));
              $context = $words[2];
              $db -> insert('[pr]tmp', 'noun, classifier, context, cryptoclass_name', ...);
}
```

Рис. 5. Код, осуществляющий анализ csv-файла и составление таблицы

Форма – это элемент кода, предназначенный для обмена информацией между пользователем и сервером. В качестве одного из атрибутов тега form выступает action. Данный атрибут является ссылкой на файл convert.php, к которому происходит обращение при отправке данных формы на сервер. В convert.php содержится программный код, отвечающий за считывание файла .csv и формирование временной таблицы в базе данных (рис. 5).

Переменной words присваивается разрыв строки при помощи разделителя «п». Учитывая, что .csv отображает табличные данные в текстовом виде, назовем данные, ограниченные символом «п», ячейками. Таким образом, переменная words – это каждая ячейка в строке. Ячейкам 0, 1 и 2 присваивается переменная, индекс которой соответствует значению столбца (например, $words[1]). Так, переменной words с индексом [0] является ячейка «имя», а переменной words[1] – классификатор криптокласса. Первый символ в данных ячейках преобразуется в верхний регистр (функция «ucfirst»), остальные данные – в нижний (функция «strtolower»), а пробелы удаляются (функция «trim»). Далее производится формирование временной таблицы «[pr]tmp». Префикс, подставляемый программой, зависит от того, какой раздел выбран в меню. Таблица раздела «СОСА» состоит из следующих видимых пользователю столбцов: Имя, Классификатор, Контекст, Криптокласс. Ячейки первых трех столбцов заполняются данными, полученными из файла .csv. Данные в последнем столбце – название криптокласса, введенное вручную.

Для построения системы конвертирования было принято решение использовать файл convert.php. Это позволило реализовать конвертер, с которым не нужно работать отдельно. Пользователь может решить сам, какой файл загрузить (.csv, .xls или .xlsx), система должна автоматически определить его формат и решить, что делать дальше.

Для разработки конвертера была использована библиотека PHPExcel [Balliauw]. С ее помощью можно считывать и записывать такие форматы, как .xls, .xlsx, .csv, .xml и др.

Программный код, выполняющий обработку загружаемых файлов .xls и .xlsx, изображен на рисунке (рис. 6).

Если форматом загружаемого файла является .xls или .xlsx, выражение include включает и выполняет файл PHPExcel.php. Далее в зависимости от расширения файла выбирается «считыватель» (Reader): «Excel5» или «Excel2007». Задача «считывателя» – анализ и извлечение данных. Для того, чтобы «считыватель» анализировал только текстовые дан-

```
include('PHPExcel/Classes/PHPExcel.php');
$fileType = PHPExcel_IOFactory::identify(PATH . $uploadfile);
$objReader = PHPExcel_IOFactory::createReader('Excel5'); // 'Excel2007' для .xlsx
$objReader->setReadDataOnly(true);
$objPHPExcel = $objReader->load(PATH . $uploadfile);

$objWriter = PHPExcel_IOFactory::createWriter($objPHPExcel, 'CSV');
$objWriter->setDelimiter('n');
$objWriter->setSheetIndex(0);
$objWriter->save($uploadfile . '.csv');
$uploadfile = $uploadfile . '.csv';
```

Рис. 6. Код, осуществляющий конвертирование файлов .xls(x)

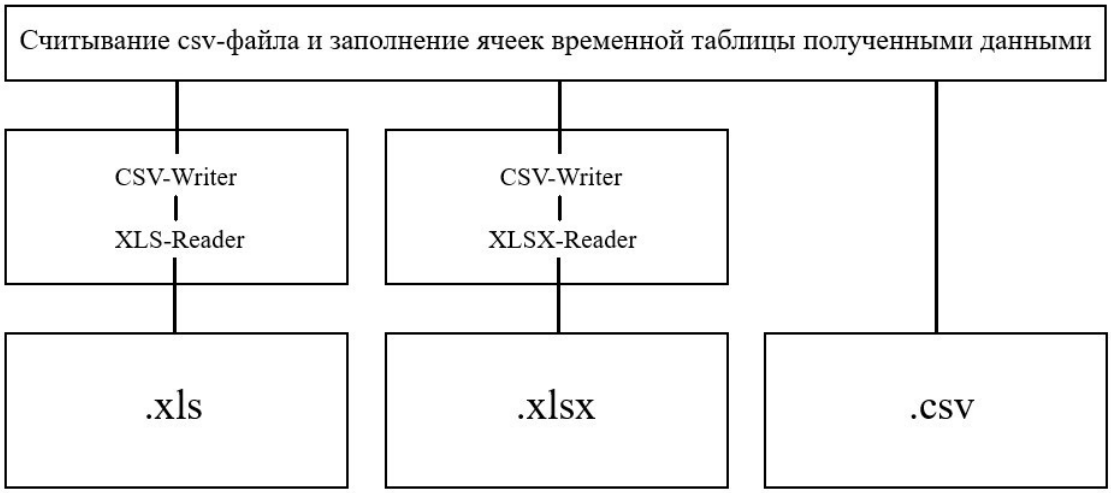

Рис. 7. Процесс загрузки данных во временную таблицу

ные, игнорируя такие параметры, как размер шрифта или цвет заливки, ему присваивается параметр «setReadDataOnly(true)». После этого происходит считывание загружаемого файла.

Далее выбирается модуль записи (Writer). В нашем случае на выходе необходимо получать файл формата .csv для последующего его разбиения на ячейки и загрузки в базу данных, поэтому был выбран модуль «CSV». В качестве разделителя устанавливается символ «п», а в качестве первого столбца устанавливается нулевой. Затем новый вариант загружаемого файла сохраняется в формате .csv и получает имя, соответствующее имени старого файла, но дополняется новым расширением. Например, файл Example.xlsx получит название Example.xlsx.csv. Значение переменной uploadfile, то есть имя загружаемого файла, также обновляется. Теперь, когда новый файл создан, он будет обработан по тому же принципу, что и все файлы .csv.

Таким образом, для реализации загрузки файлов .xls или .xlsx в процесс обработки данных потребовалось добавить один этап (рис. 7).

Теперь в первую очередь происходит определение формата загружаемых файлов, считывание текстовой информации и запись в новый csv-файл, а затем разбиение и размещение в таблице базы данных. При этом принцип обработки .csv не изменился.

К загрузке данных в таблицы разделов «GloWbE», «NOW» и «iWeb» требуется другой подход, в связи с тем, что в источниках данных, используемых в исследовании, содержится несколько вариантов английского языка. Таблицы «glowbe_tmp», «now_tmp» и «iweb_tmp» были дополнены столбцом «language» (во временной таблице в веб-интерфейсе он называется «Язык»).

При условии, что переменная сессии не равна единице (это говорит о том, что выбран раздел базы данных, отличающийся от

```
$lines = file($uploadfile);
foreach ($lines as $line) {
        $words = explode('п', $line);
        if ($words[0] != '') $noun = $words[0];
           $noun = ucfirst(strtolower(trim($noun)));
           if ($words[1] != '') $classifier = $words[1];
              $classifier = ucfirst(strtolower(trim($classifier)));
              if ($words[2] != '') $language = $words[2];
                 $language = trim($language);
                 $context = $words[3];
                 $db->insert('[pr]tmp', 'noun, classifier, language, context, cryptoclass_name', ...);
}
```

Рис. 8. Считывание .csv и составление временной таблицы в новых разделах системы

Временная таблица				
Существительное	Классификатор	Язык	Контекст	Криптокласс

Рис. 9. Поля временной таблицы в разделе «Загрузить из файла»

«COCA»), считыватель csv-файлов получает дополнительный индекс ячейки. На рисунке (рис. 8) показано, что контекст в данных таблицах расположен на последней позиции в строке.

Третьей ячейкой является «language». В данном случае функции «ucfirst» и «strtolower» не используются, так как название варианта языка представлено в виде аббревиатуры. Если данные в загружаемом файле получены из «GloWbe», «NOW» или «iWeb», то поля временной таблицы на странице «Загрузить из файла» выглядят следующим образом (рис. 9), а требования к их последовательности в загружаемом файле обновляются.

Для реализации возможности загружать и хранить данные, полученные из новых ресурсов, требуется внести изменения в структуру базы данных. Для размещения в системе «COEL» информации о вариантах английского языка в базе данных были созданы таблицы «glowbe_languages», «now_languages» и «iweb_languages» с полями «id», «name» и

```
CREATE TABLE glowbe_languages (
  id INT AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  short_name VARCHAR(3) NOT NULL,
  PRIMARY KEY (id)
  )
```

Рис. 10. Создание таблицы в базе данных на языке SQL

«short_name». Создание таблицы выполняется на языке SQL следующим образом (рис. 10).

На рисунке (рис. 11) изображена часть таблицы «glowbe_languages», зарегистрированная в результате выполнения указанного выше запроса и ручного заполнения данных.

В строках первого поля, которое заполняется автоматически, содержатся идентификаторы языка. В записях второго столбца размещены данные о полном названии варианта английского языка. Информация, содержащаяся в строках третьего поля необходима для загрузки контекстов в базу данных, т.к. в данные о вариантах английского языка в системе «COEL» было решено представлять в

id	name	short_name
1	British English	GB
2	American English	US
3	Irish English	IE
4	Canadian English	CA
5	Australian English	AU
6	New Zealand English	NZ
7	Jamaican English	JM

Рис. 11. Таблица «glowbe_languages»

```
$cnt1 = $db->count('[pr]nouns', "name = '$noun'");
if (!$cnt1) {
    $db->insert('[pr]nouns', 'name', "$noun");
    $noun_id = mysqli_insert_id($db->link);
}
else {
    echo "noun '$noun' exists...<br/>";
    $noun_id = $db->select_assoc('id', '[pr]nouns', "name = '$noun'", '', '1');
    $noun_id = $noun_id['id'];
}
```

Рис. 12. Проверка существительных на предмет существования в базе данных

```
$cnt4 = $db->count('[pr]compatibility', "noun_id =$noun_id AND classifier_id=$classifier_id");
if (!$cnt4) {
    $db->insert('[pr]compatibility', 'noun_id, classifier_id', "$noun_id|><|$classifier_id");
    $pair_id = mysqli_insert_id($db->link);
}
else {
    echo "pair '$noun - $classifier' exists...<br/>";
    $pair_id = $db->select_assoc('id', '[pr]compatibility', "noun_id = $noun_id AND classifier_id=$classifier_id", '', '1');
    $pair_id = $pair_id['id'];
}
```

Рис. 13. Код, выполняющий проверку наличия в базе данных комбинации

виде аббревиатур, как это сделано в корпусах М. Дэвиса, выступающих в качестве источника материала.

В таблицы «glowbe_compatibility», «now_compatibility» и «iweb_compatibility», содержащие поля «id», «noun_id» и «classifier_id» было добавлено поле «language_id», содержимым которого являются числовые данные, а точнее идентификатор варианта языка, полученный из соответствующих таблиц [pr] languages. Таблица [pr]compatibility обновляется при загрузке в базу данных новой информации с помощью раздела «Загрузить из файла». За распределение данных временной таблицы ([pr]temp) по постоянным таблицам отвечает программный код, содержащийся в файле commit_temp.php. Его задачей является проверка загружаемых данных на наличие в базе и, в случае их отсутствия, размещение в таблицах.

На рисунке (рис. 12) изображен участок кода, выполняющий проверку существительных в загружаемых таблицах. Если существительное не найдено, оно регистрируется в таблице «[pr]nouns». Если оно уже существует в базе, то пользователь получит краткое сообщение, говорящее об этом, и загрузка продолжится. Такую же проверку проходят и классификаторы. Далее комбинации из существительных и классификаторов, расположенных в каждой строке загружаемого файла, например, acute (острый) и pain (боль), проходят проверку на наличие в базе данных. В случае отсутствия пары, данные о ней в виде идентификаторов имени и классификатора, а также идентификатора самой комбинации появляются в таблице «[pr]compatibility».

Для загрузки данных из новых корпусов необходимо реализовать проверку информации о варианте языка. Создание комбинации из имени, классификатора и варианта английского (рис. 13) проводится по тому же принципу, что и комбинация имя-классификатор. Если переменной сессии является не единица, а также комбинация (тройка) не зарегистрирована, информация о ней появляется в таблице «[pr]compatibility».

Однако для осуществления проверки наличия комбинации, как и в случаях с классификаторами и существительными, необходимо получить информацию о возможном существовании загружаемых данных в системе. При разработке системы проверки информации, загружаемой в новые разделы «COEL», необходимо учесть, что корпуса, которые выступают в качестве источников информации, содержат фиксированное количество вариантов английского языка. Простая проверка на предмет наличия соответствий в БД с возможностью регистрации новых данных

```
if ($_SESSION['database'] != 1) {
   $cnt6 = $db->count('[pr]languages', "short_name = '$language'");
   if (!$cnt6) {
      echo "<script type='text/javascript'>alert('Один (или более) язык не зарегистрирован в базе данных!');</script>";
      break;
    }
    else {
      echo "language '$language' exists...<br/>";
      $language_id = $db->select_assoc('id', '[pr]languages', "short_name = '$language'", '', '1');
      $language_id = $language_id['id'];
    }
}
```

Рис. 14. Код, выполняющий проверку данных о языке на предмет отсутствия ошибки

```
$context = mysqli_real_escape_string($db->link, $context);
$cnt5 = $db->count('[pr]contexts', "combination_id =$combination_id AND context='$context'");
if (!$cnt5) {
    $db->insert('[pr]contexts', 'combination_id, context', "$combination_id|><|$context");
    $context_id = mysqli_insert_id($db->link);
}
else {
    echo "context '" . mb_substr($context, 0, 30, "utf-8") . "'... exists...<br/>";
    $context_id = $db->select_assoc('id', '[pr]contexts', "combination_id = $combination_id AND context='$context'", '', '1');
                  $context_id = $context_id['id'];
}
```

Рис. 15. Программный код, выполняющий размещение данных в [pr]contexts

проигнорирует опечатку в аббревиатуре, что вызовет создание новой тройки и неправильное размещение одного (или нескольких) контекстов.

Во избежание ошибок в работе системы, в программный код проверки (рис. 14) было решено включить участок разметки, содержащий javascript-код, который вызывает предупреждение о том, что как минимум один из вариантов языка в загружаемой таблице отсутствует в базе данных. Пользователь увидит данное предупреждение и в разделе «СОСА», который не содержит информацию о различных вариантах английского. Это сделано для того, чтобы избежать неправильной загрузки данных. После вызова предупреждения оператор break досрочно завершает цикл проверки и прерывает операцию загрузки контекстов, не внеся изменений.

Так как комбинация из существительного, классификатора и языка не является парой, поле pair_id во всех таблицах [pr]contexts было изменено на combination_id. Названия переменных (рис. 15) и параметры SQL-запросов были обновлены в соответствии с данным изменением. Если в процессе выполнения проверок ошибки выявлены не были, объявляются переменные, используемые для загрузки информации в таблицу [pr]contexts. Далее происходит пополнение таблицы контекстов новыми данными. Программный код, выполняющий размещение информации в [pr]contexts, изображен на рисунке ниже (рис. 15).

Таким образом, проверка языка позволяет защитить от некоторых возможных случаев загрузки неправильных данных, а построение комбинаций на базе трех параметров дает возможность сделать поиск по контекстам более функциональным.

ЗАКЛЮЧЕНИЕ

В процессе выполнения действий по расширению функционала лингвистического ресурса «COEL», описанных в статье, были внесены следующие основные изменения:

1. Для осуществления возможности хранения материала, полученного из корпусов GloWbE, NOW и iWeb (которые содержат примеры словоупотреблений в различных вариантах английского языка), было произведено расширение базы данных системы «COEL». Возможность получения раздельного доступа к источникам информации реализована в виде разделов системы, доступ к каждому из которых осуществляется через главное меню.

2. В систему загрузки контекстов словоупотреблений (страница «Загрузить из фай-

ла») были внесены изменения, в результате которых стала доступна загрузка данных в новые разделы системы. Также была реализована возможность использования .xls и .xlsx в качестве форматов загружаемых файлов.

3. Страница «Контексты» новых разделов была дополнена выпадающим списком «Язык», используя который пользователь может фильтровать получаемую информацию по варианту английского языка. Функции отображения содержимого данного меню, а также меню «Существительное», были обновлены, в результате чего пункты указанных списков теперь изменяются в зависимости от примененных параметров. Для хранения информации, полученной на странице «Контексты», и работы с ней в офлайн-режиме была реализована возможность выгрузки на локальное хранилище.

Указанные изменения позволили адаптировать систему для осуществления работы с данными, полученными из GloWbE, NOW и iWeb, а также сделать ее более функциональной и удобной для использования.

СПИСОК ЛИТЕРАТУРЫ

1. *Саженин, И. И.* К вопросу о построении базы данных прагматически маркированной лексики / И. И. Саженин // Вестник Новосибирского государственного педагогического университета. – 2015. – № 5. – С. 142–155.

2. *Баранов, В. А.* Полное собрание сочинений М. В. Ломоносова в Интернете: подготовка электронной коллекции и функциональные возможности модулей корпуса / В. А. Баранов // Уч. зап. Казанского ун-та. Серия: Гуманитарные науки. – 2010. – Вып. 6. – С. 223–234.

3. *Gatiatullin, A.* Multilingual Database of Turkic Color Names: Structure and Design / A. Gatiatullin, M. Kurmanbakiev, B. Khakimov // Proceedings of the International Conference Turkic Languages Processing: TurkLang. – 2015. – P. 224–232.

4. *Борискина, О. О.* Теория языковой категоризации: национальное языкознание сквозь призму криптокласса / О. О. Борискина, А. А. Кретов. – Воронеж : Воронежский государственный университет, 2003. – 211 с.

5. *Донина, О. В.* Криптоклассные данные для определения меры межъязыковой эквивалентности / О. В. Донина // Вестник Воронеж. гос. ун-та. Сер. Лингвистика и межкультурная коммуникация. – 2015. – № 1. – С. 108-110.

6. *Донина, О. О.* Способы визуализации результатов криптоклассного исследования / О. О. Донина // Вестник Воронеж. гос. ун-та. Сер. Лингвистика и межкультурная коммуникация. – 2015. – № 3. – С. 105–112.

7. *Донина, О. В.* Применение методов data mining для решения лингвистических задач / О. В. Донина // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2017. – № 1. – С. 154–160.

8. *Davies, M.* Corpus of Contemporary American English (COCA) [Электронный ресурс] / M. Davies. – 2008. – Режим доступа: https://corpus.byu.edu/COCA/ (дата обращения: 12.12.2018).

9. *Davies, M.* The corpus of Global Web-based English (GloWbE) [Электронный ресурс] / M. Davires. – 2013. – Режим доступа: https://corpus.byu.edu/glowbe/ (дата обращения: 17.12.2018).

10. *Davies, M.* News On the Web (NOW) [Электронный ресурс]. – 2016. – Режим доступа: https://corpus.byu.edu/now/ (дата обращения: 14.12.2018).

11. *Davies, M.* The iWeb corpus [Электронный ресурс]. – 2018. – Режим доступа: https://corpus.byu.edu/iweb/ (дата обращения: 29.12.2018).

Донина О. В. – канд. филол. наук, преподаватель кафедры теоретической и прикладной лингвистики, факультет романо-германской филологии, Воронежский государственный университет.
E-mail: olga-donina@mail.ru

Филатов В. В. – студент 1 курса магистратуры факультета компьютерных наук, Воронежский государственный университет.
E-mail: vladislavfilatov159@gmail.com

Donina O. V. – Candidate of Philology, Lecturer, Department of Theoretical and Applied Linguistics, Romance and Germanic Philology Faculty, Voronezh State University.
E-mail: olga-donina@mail.ru

Filatov V. V. – 1st year master's student, Computer Science Faculty, Voronezh State University.
E-mail: vladislavfilatov159@gmail.com