OSTIS-2015
(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКИЕ АСПЕКТЫ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ПОИСКОВЫХ ЗАПРОСОВ В СИСТЕМЕ КОРПУС-МЕНЕДЖЕР

Невзорова О.А.[*], Мухамедшин Д.Р.[**], Билалов Р.Р.[**]

[*] *Научно-исследовательский институт «Прикладная семиотика» АН Республики Татарстан, Казанский (Приволжский) федеральный университет, г. Казань, Россия*

onevzoro@gmail.com

[**] *Институт вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета, г. Казань, Россия*

damirmuh@gmail.com
akibro@yandex.ru

В статье рассматриваются семантические аспекты представления и обработки поисковых запросов, применяемые в системе управления лингвистическими данными (корпус-менеджер), предназначенной для работы с электронным корпусом текстов татарского языка.
Ключевые слова: корпус-менеджер; поисковые запросы; морфологические формулы.

Введение

Представление и обработка поисковых запросов в системах, поддерживающих лексико-морфологический поиск в электронном корпусе текстов, предполагает анализ формальной и содержательной семантики поисковых запросов. Основными этапами обработки данных, содержащих семантические аспекты, являются разбор лексической и морфологической составляющих запроса, выявление синтаксических и логических ошибок в запросе, определение классов, содержащих необходимые данные, преобразование поискового запроса в запрос к хранилищу данных. В статье будет рассмотрено два из указанных выше этапов: выявление ошибок и определение классов.

Многие поисковые системы, работающие с национальными лингвистическими корпусами, используют готовые ядра, такие как «Яндекс.Сервер», используемый в поисковой системе Национального корпуса русского языка[1]. Подобные системы представляют собой комплекс из нескольких подсистем, что позволяет производить быстрый и многофункциональный поиск. Система «Яндекс.Сервер» является проприетарной, а её полная версия распространяется на коммерческой основе.

Корпус-менеджер[2], обсуждаемый в настоящей статье, разработан специально для работы с лингвистическими корпусами. Функционал, предлагаемый системой корпус-менеджер, включает в себя поиск лексических единиц, морфологический поиск, лексико-морфологический поиск, поиск синтаксических единиц, поиск n-грамм с учетом грамматики и др. В основе системы корпус-менеджер заложена семантическая модель представления данных корпуса татарского языка. Поиск производится при помощи общедоступных инструментов с открытым исходным кодом: система управления базой данных MariaDB и хранилище данных Redis. Разработанный корпус-менеджер ориентирован в первую очередь на поддержку электронных корпусов тюркских языков, что является весьма актуальным для активно развивающегося направления тюркской корпусной лингвистики.

1. Представление данных в системе корпус-менеджер

В системе корпус-менеджер данные представляются в виде семантической структуры классов, которая представлена на рисунке 1. Базовым классом является класс Документы, включающим в себя подкласс Контексты, который в свою очередь включает в себя подкласс Разборы.

[1] http://www.ruscorpora.ru/

[2] http://corpus.antat.ru/search/

Последний подкласс содержит в себе около 55,7 млн. элементов, делится на несколько подклассов по морфологическим признакам, а также связан с классами Словоформы, Леммы, Морфологические признаки. Между элементами классов Разборы, Контексты и Документы имеются связи «часть-целое» (на рисунке сплошная линия), между элементами класса Словоформы (и Леммы) и Разборы – связи «часть-целое», между элементами классов Разборы и Морфологические признаки – множественные связи «hasGrammaticalFeature» («имеет морфологический признак», на рисунке квадратные точки), между элементами классов Словоформы и Морфологические признаки – множественные связи «usedWithGrammaticalFeature» («употребляется с морфологическим признаком», на рисунке штрих-пунктир), между элементами класса Морфологические признаки и подклассами класса Разборы – связи «isFoundIn» («встречается в», на рисунке круглые точки).

Такая структура позволяет производить поиск с учётом многих лексических и морфологических параметров, а также находить логические ошибки в пользовательских поисковых запросах. Каждый элемент кроме представленных связей имеет определенные свойства, такие как название, позиция в контексте, позиция в документе и т.п. Эти свойства не используются непосредственно в процессе поиска, но могут быть выведены пользователю наряду с остальными.

2. Представление поисковых запросов

Пользовательский поисковый запрос к системе корпус-менеджер представляет собой упорядоченный кортеж, состоящий из кортежей длины 7 (в общем случае), включающий следующие компоненты:

- Вид поиска;
- Лексическая составляющая запроса;
- Морфологическая составляющая запроса;
- Направление поиска;
- Минимальное расстояние между запросами;
- Максимальное расстояние между запросами;
- Тип поиска.

Например, кортеж Q, указанный в (1), является примером поискового запроса. Здесь Q_1 – первый кортеж поискового запроса, состоящий из следующих компонентов: *lemma* – вид поиска (поиск по лемме), китап – лексическая составляющая запроса (*kитап (Tat)/книга*), $N|V$ – морфологическая составляющая запроса (морфологическая формула, которая означает «имя существительное ИЛИ глагол»), *right* – направление поиска (вправо по отношению к предыдущему кортежу запроса), 1 – минимальное расстояние от предыдущего кортежа с учётом направления, 10 – максимальное расстояние от предыдущего кортежа с учётом направления, *exact* – тип поиска (точный); Q_2 – второй кортеж поискового запроса, где изменены следующие компоненты: *бар (Tat)/ «есть», «имеется»* – лексическая составляющая запроса, $V|ADV$ – морфологическая составляющая запроса (морфологическая формула, которая означает «глагол ИЛИ наречие»), другие компоненты подобны компонентам в кортеже Q_1. Кортеж Q состоит из кортежей Q_1 и Q_2.

$$Q_1 = (lemma, \text{китап}, N|V, right, 1, 10, exact)$$

$$Q_2 = (lemma, \text{бар}, V|ADV, right, 1, 10, exact) \quad (1)$$

$$Q = (Q_1, Q_2)$$

Вид, направление, тип поиска, минимальное и максимальное расстояние между запросами имеют детерминированный характер и не требуют дополнительного разбора, тогда как лексическая и морфологическая составляющие запроса имеют стохастический характер, и, чтобы выявить

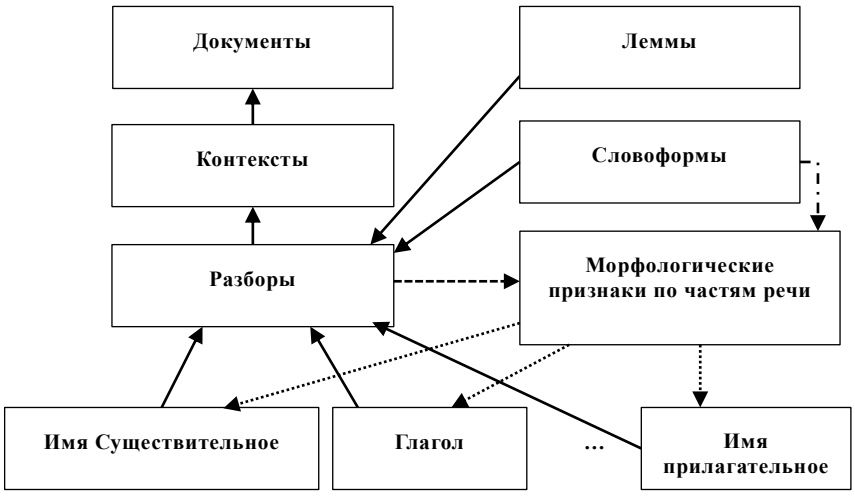

Рисунок 1 – Связи между элементами в системе

семантику этих компонент, необходимо произвести их разбор и обработку.

2.1. Лексическая составляющая запроса

В системе корпус-менеджер лексическая составляющая запроса может представлять собой не только отдельную словоформу или лемму (например, *«китап» (Tat)/книга*), но и их части (например, «кита*») или правила с исключениями (например, «китап -“китаплар”» при поиске по леммам, что означает «все словоформы, имеющие лемму “китап”, за исключением словоформы “китаплар”», *«китаплар» (Tat)/книги*). Такой расширенный синтаксис лексических составляющих запроса требует со стороны системы правильной интерпретации семантики запроса, заложенной пользователем.

При обработке запроса сначала производится поиск лексической составляющей в классе Словоформы или Леммы в зависимости от вида поиска. Элементы класса Словоформы представляют собой двуместный кортеж вида (2), элементы класса Леммы – двуместный кортеж вида (3).

(словоформа, идентификатор) (2)

(лемма, идентификатор) (3)

Поиск производится для всех частей лексической составляющей, разделенных пробельными символами, по первому элементу кортежа. В результате поиска формируются правила для поиска результатов среди элементов класса Разборы. Например, для запроса «кита*» при поиске по словоформам будут найдены все словоформы, начинающиеся с «кита» («китап», «китаплар», «китабы» и т.д.), которые представляют собой подмножество искомых словоформ W. Для этого примера правилом для поиска станет (4), где w – компонент «словоформа» в кортеже элемента класса Разборы.

$$w \in W \quad (4)$$

Если значимая (не исключающая) часть лексической составляющей запроса не существует в соответствующем классе, система сообщает пользователю о синтаксической ошибке.

2.2. Морфологическая составляющая запроса

Определение семантики морфологической составляющей поискового запроса является хоть и тривиальной, но весьма сложной задачей. В системе возможно использовать морфологические формулы, составленные из морфологических обозначений признаков и операций конъюнкции, дизъюнкции и отрицания. Формула может быть вида конъюнкции переменных (5), дизъюнкции переменных (6), отрицания дизъюнкции переменных (7) и произвольной (8).

N,DIR,SG (5)

N|DIR|SG (6)

!(N|DIR|SG) (7)

!N|DIR,SG (8)

В приведенных примерах морфологические формулы имеют следующую семантику:

- (5) – «имя существительное И направительный падеж И единственное число», то есть по этой формуле должны быть найдены все словоформы, которые имеют одновременно морфологические признаки «имя существительное», «направительный падеж» и «единственное число»;
- (6) – «имя существительное ИЛИ направительный падеж ИЛИ единственное число», то есть по этой формуле должны быть найдены все словоформы, которые имеют хотя бы один из морфологических признаков «имя существительное», «направительный падеж», «единственное число»;
- (7) – «НЕ имя существительное ИЛИ направительный падеж ИЛИ единственное число» (может быть преобразовано в «НЕ имя существительное И НЕ направительный падеж И НЕ единственное число»), то есть по этой формуле должны быть найдены все словоформы, которые не имеют ни одного из морфологических признаков «имя существительное», «направительный падеж», «единственное число»;
- (8) – «НЕ имя существительное ИЛИ направительный падеж И единственное число», то есть по этой формуле должны быть найдены все словоформы, которые не имеют морфологического признака «имя существительное», либо имеют одновременно морфологические признаки «направительный падеж» и «единственное число».

В зависимости от типа формулы, обработка поискового запроса производится по различным алгоритмам.

3. Обработка поисковых запросов

Для формул вида (5), (6), (7) задача определения семантики является тривиальной. Для этих типов формулы общий алгоритм обработки отображен на рисунке 2.

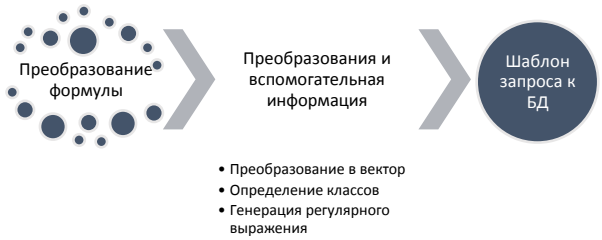

Рисунок 2 - Общий алгоритм обработки морфологических формул

Шаги алгоритма следующие:

1. Преобразование формулы в кортеж морфологических обозначений (разбиение по знаку операции «,» или «|», здесь же удаляются дополнительные символы: скобки, знаки «!» и «~»).
2. Преобразования и вспомогательная информация.
    a. Морфологические обозначения преобразуются в битовый вектор, пригодный для использования в запросе к базе данных.
    b. На основе морфологических обозначений определяются классы, где возможно нахождение искомых данных.
    c. Генерируется регулярное выражение для отметки подходящих сущностей в контекстах.
3. Формирование запроса к базе данных.

3.1. Обработка произвольных формул

Задача правильного определения семантики произвольной формулы требует наиболее общего решения, покрывающего любые сочетания операций в формуле. Таким решением стали поочередные замены частей формулы на простые составляющие вида (5), (6) или (7) в порядке приоритета операций. Замены производятся пока не будет получена простая формула вида (5), (6) или (7), в которых выполнена замена морфологических обозначений признаков на специальные переменные.

Каждая итерация начинается с процесса преобразований. Это попытка выявить соответствие переменной существующему типу простых формул (с возможностью упрощений, где это может оптимизировать скорость запроса к базе данных, например, !(!N,!V) = N|V) и заменить эту переменную сразу. Попытка преобразования предпринимается, если в значении переменной не используются другие переменные. Если попытка преобразования завершилась успешно, текущая итерация прерывается.

Если в формуле есть переменные или формула оказалась произвольного типа, части формулы типа (5), (6) или (7) заменяются на переменные, каждая из которых представляется в формуле в виде «REPLACED_S_N», где N – номер замены. Очередность замен соответствует приоритету логических операций.

Затем в формуле из переменных производится поиск частей типа (5). Такие части заменяются на новые переменные, которые содержат в себе все свойства части (пересечение классов, условия, регулярные выражения).

Последним этапом итерации является замена частей типа (6) (или отдельных переменных). На этом этапе все переменные в формуле заменяются на условия запроса к базе данных, объединяются классы и формируются регулярные выражения.

Итерации выполняются для всех первичных замен со скобками. Результаты последней итерации являются окончательными для сложной формулы и содержат в себе список классов, где возможно нахождение искомых данных, условие для запроса к базе данных, а также регулярное выражения для выделения сущностей в контексте.

Конечный результат работы алгоритмов для обработки всех типов формул представляет собой кортеж, состоящий из преобразованной формулы для работы с базой данных и упорядоченного кортежа, который содержит классы, в которых возможно нахождение искомых результатов.

3.2. Определение классов для лексической составляющей запроса

Чтобы определить множество классов, в которых возможно нахождение данных по искомой лексической составляющей запроса, необходимо воспользоваться связями «usedWithGrammaticalFeature» между элементами классов Словоформы и Морфологические признаки и связями «isFoundIn» между элементами класса Морфологические признаки и подклассами класса разборы. Операция поиска производится для всех частей лексической составляющей запроса и записывается в S_i, где S – множество частей лексической составляющей запроса, i – номер части. В общем случае, получив классы для всех частей лексической составляющей запроса, их необходимо объединить, как показано в (8). Здесь n – это количество частей лексической составляющей запроса, SL – множество классов, в которых возможно нахождение искомых данных.

$$SL = \bigcup_{i=0}^{n-1} s_i \qquad (8)$$

Заметим, что исключающие части лексической составляющей в формировании множества классов участия не принимают, так как из их исключения из запроса не следует исключение классов, в которых могут находиться искомые данные.

3.3. Определение классов для морфологической составляющей запроса

Определение секций для морфологической составляющей запроса уже было показано выше для произвольных формул. Для формул типов (5), (6), (7) определение классов проводится по следующему алгоритму:

1. Множество SM представляет собой множество всех возможных классов, если формула имеет тип (4), иначе задается как пустое множество.
2. Если в формуле существует i-е морфологическое обозначение, определяется номер (далее N) для i-го морфологического обозначения, иначе алгоритм завершает работу.
3. Если морфологическое обозначение определяет морфологический класс, то множество s_i заполняется такими натуральными числами x, что $N * 10 \leq x < (N + 1) * 10$, переход к шагу 5, иначе шаг 4.

4. Поиск множества классов по связи «isFoundIn» между элементом класса Морфологические признаки, соответствующем морфологическому обозначению, и подклассами класса Разборы. Найденное множество записывается как s_i.

5. Если формула имеет тип (6), множество SM объединяется с s_i: $SM = SM \cup s_i$, переход к следующей итерации (шаг 2), иначе шаг 6.

6. Если формула имеет тип (5), множество SM пересекается с s_i: $SM = SM \cap s_i$, переход к следующей итерации (шаг 2), иначе шаг 7.

7. Множество SM задается как дополнение к множеству s_i (по отношению к множеству всех возможных классов): $SM = \overline{s_i}$, переход к следующей итерации (шаг 2).

Общее множество классов S определяется как пересечение множества классов для лексической составляющей запроса и множества классов для морфологической составляющей запроса, как показано в (9).

$$S = SM \cap SL \quad (9)$$

3.4. Выявление логических ошибок

Если множество S пустое, это означает, что пользователь допустил логическую ошибку в запросе, и по этому запросу не могут быть найдены никакие данные, так как не существует ни одного класса, в котором возможно нахождение искомых данных. Логическая ошибка может быть выявлена на любом этапе работы системы, если любое из множеств SL и SM является пустым множеством.

В ходе экспериментов в рамках нагрузочного тестирования и тестирования правильности, 6,88% всех запросов содержало логические ошибки, но они не были выявлены системой. По результатам экспериментов была изменена схема представления данных в системе, в которой на момент проведения экспериментов не было связей между элементами класса Морфологические признаки и подклассами класса Разборы. Внесённые изменения позволили существенно улучшить эффективность алгоритма выявления логических ошибок и сократить количество запросов с логическими ошибками, которые не были выявлены системой.

Примером запроса с логической ошибкой может служить запрос, в котором присутствует морфологическая составляющая (10).

!(N|V),INF_1 (10)

Из морфологической формулы (10) видно, что её первая часть «!(N|V)» имеет вид (7) и означает «НЕ имя существительное или глагол» или «НЕ имя существительное И НЕ глагол». Для этой части результаты могут быть найдены в любом классе, кроме «имя существительное» и «глагол». Вторая часть запроса «INF_1» означает «инфинитив, оканчивающийся на аффикс -ырга», для этой части классом, в котором могут быть найдены результаты, является только класс «глагол». Между частями формулы стоит операция конъюнкции. Это означает, что множество классов, содержащих результаты для формулы, является пересечением множеств классов, содержащих результаты для каждой части формулы. Такое множество является пустым множеством, а значит формула содержит логическую ошибку.

Заключение

Предложенные в настоящей статье методы представления и обработки поисковых запросов реализованы в поисковом модуле системы управления лингвистическими данными, работающей с электронным корпусом татарского языка.

Чтобы проверить пригодность, правильность, согласованность, характер изменения во времени, было проведено комплексное тестирование системы. Тестирование пригодности показало, что предложенные методы полностью решают поставленные задачи. Тестирование правильности, основанное на сравнении с другим эталонным методом представления и обработки запросов, показало, что предложенные методы работают правильно. Тестирование согласованности и характера изменения во времени показало, что предлагаемый синтаксис лексической и морфологической составляющей запроса верно интерпретируется системой, а время, необходимое для обработки и выполнения поискового запроса системой, не превышает 0,05 сек. в 98,71% случаев для лексического поиска, в 77,71% случаев для морфологического поиска и в 98,08% случаев для лексико-морфологического поиска. Во многом, таких результатов удалось добиться благодаря представлению данных в виде семантической сети и предложенным методам представления и обработки поисковых запросов.

Использование семантических сетей для представления данных лингвистических корпусов позволяет покрыть широкий спектр задач. В статье описаны лишь некоторые из них. В перспективе предполагается использование этой же схемы для реализации морфологического анализатора и решения задачи снятия омонимии.

Общий подход к решению задач в поисковой системе для лингвистического корпуса позволяет использовать разработанную систему не только для работы с электронным корпусом текстов на татарском языке, но и с корпусами других языков, без существенных изменений в системе.

Библиографический список

[Suleymanov et al., 2013] Dzhavdet Suleymanov, Olga Nevzorova, Ayrat Gatiatullin, Rinat Gilmullin, Bulat Khakimov National corpus of the Tatar language “Tugan Tel”: Grammatical Annotation and Implementation // Procedia - Social and Behavioral Sciences (2013), vol. 95, pp. 68-74.

[Nevzorova, Salimov, 2012] Nevzorova O., Salimov F. Model of Lexicographical Database: Structure, Basic Functionality,

Implementation // International Journal "Information Models and Analyses". Vol.1. Number 1, 2012. - P.21-27.

[Cormen et al., 2009] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2009). Introduction to Algorithms (3rd ed.). Massachusetts Institute of Technology. Pp. 253–280.

[Davies 2009] Mark Davies The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights // International Journal of Corpus Linguistics. Volume 14. Number 2, 2009, pp. 159-190(32)

[Han 2011] Jing Han Survey on NoSQL database // Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, 26-28 Oct. 2011, pp. 363-366.

[Oracle Corporation 2015] MySQL 5.6 Reference Manual: 19 Partitioning, Oracle Corporation. Web: http://dev.mysql.com/doc/refman/5.6/en/partitioning.html

[Аброскин 2009] Аброскин А. А. Поиск по корпусу: проблемы и методы их решения // Национальный корпус русского языка: 2006-2008. Новые результаты и перспективы. СПб.: Нестор-История, 2009

[Захаров 2002] Захаров В.П. Корпусная лингвистика. Учебно-метод. пособие. – СПб., 2005. – 48 с.

SEMANTIC ASPECTS OF SEARCH REQUEST REPRESENTATION AND PROCESSING IN CORPUS-MANAGER SYSTEM

Nevzorova O.A.*, Mukhamedshin D.R.**, Bilalov R.R.**

** Research Institute of Applied Semiotics of the Academy of Sciences of Tatarstan Republic, Kazan (Volga Region) Federal University, Kazan, Russia*

onevzoro@gmail.com

*** Institute of Computer Mathematics and Information Technologies, Kazan (Volga Region) Federal University, Kazan, Russia*

damirmuh@gmail.com

akibro@yandex.ru

The article provides semantic aspects of search request representation and processing applied in management system of linguistic data (corpus manager), designed to work with corpus of texts of the Tatar language.

Introduction

Corpus manager discussed in this article designed specifically for working with linguistic corpora. Functionality available within the corpus manager system includes a search of lexical units, morphological search, lexical and morphological search, search of syntactic units, search n-grams with the grammar etc. The corpus manager system based on semantic model data representation of corpus of the Tatar language. Designed corpus manager focuses primarily to support electronic corpuses of Turkic languages, which is highly relevant to the rapidly developing areas of Turkic corpus linguistics.

Main part

Data in the corpus manager system presented in the form of the semantic structure of classes. Such structure allows searching considering many lexical and morphological parameters, as well as finding logical errors in user search queries.

User search query to the corpus manager system represented an ordered tuple consisting of tuples of length 7 (generally), which includes the following components: type of search, lexical component of the request, morphological component of the request, search direction, minimum distance between requests, maximum distance between requests, search type.

When processing a request, first searched for lexical component in the Word forms class or Lemmas class depending on the type of search. As a result of the search, system generates rules for the search for results among the elements of the Parsing results class.

Depending on the type of formula, the search query processing performed by various algorithms. For simple formulas, steps of algorithm are as follows: conversion formula to the tuple of morphological signs, conversions and obtaining supporting information for the query to the database, forming query to the database. The problem of the correct defining the semantics of a complex formula requires the most common solutions covering any combinations of operations on a formula. This solution became alternately replacement parts of the formula into simple components in order of operations priority.

To define a set of classes in which is possible to find data on the required lexical component of the request, you must use the relations "usedWithGrammaticalFeature" between elements of the Word forms class and the Grammatical features class and relations "isFoundIn" between elements of the Grammatical features class and subclasses of the Parsing results class. The total set of classes *S* defined as the intersection of a set of classes for the lexical component of the request and a set of classes for the morphological component of the request.

If the set S is empty, it means that user has made a logical error in the request and this request can not be found any data, since there is no class where possible to find the required data.

Conclusion

The use of semantic networks for data representation in linguistic corpora allows covering a wide spectrum of problems. This article describes a few of them. In the feature, it is planned to use the same scheme for the implementation of the morphological analyzer and solving the problem of disambiguation.

A general approach to solving problems in the search engine on linguistic corpora may allow the use of this system not only with electronic corpus of texts of the Tatar language, but also with corpuses of other languages without significant changes in the system.