УДК004.912+004.6+004.4'236

СИСТЕМА ДЛЯ ОБРАБОТКИ КОРПУСОВ ТЕКСТОВ

Бармина Елена Ивановна, Бушуев Роман Николаевич, Котельникова Надежда Владимировна, Ланин Вячеслав Владимирович, Плотникова Оксана Арсеновна

Национальный исследовательский университет «Высшая школа экономики», 614070, Россия, г. Пермь, ул. Студенческая, 38, elenabarmina@ya.ru, bushuev.roman@gmail.com, nadya.kott@gmail.com, lanin@perm.ru, simple_oks@rambler.ru

Статья посвящена описанию разработки системы для обработки корпусов текстов, которая ориентирована на профессионалов в области корпусной лингвистики. В данной работе представлено описание разработки хранилища, компонентов прикладного уровня, веб-редактора для визуального языка лексико-семантических шаблонов и веб-компонента для визуализации результатов. Объектом исследования является текстовые корпуса, предметом – системы для работы с текстовыми корпусами. Целью исследования является разработка новой системы для работы с текстовыми корпусами. В данной статье также приведен анализ существующих решений, более подробно рассмотрены отрицательные и положительные их характеристики. В заключении статьи будут сформулированы выводы о проделанной работе и планы на будущее.

Ключевые слова: корпусная лингвистика, визуализация данных, DSL, облачные хранилища.

Введение

Ежегодно с появление современных технологий объем информации увеличивается. Данное явление также отразилось и на текстовых документах. Ли Джайлсем было проведено исследование, которое показало, что в интернете минимум 114 млн. научных документов на английском языке. Примерно 27 млн. из них хранятся в открытом доступе. Такое огромное количество данных содержит в себе новую полезную информацию, которою необходимо исследовать.

Корпусная лингвистика занимается тем, что создает текстовые корпуса, которые состоят из электронных документов определенной тематики или имеют общие свойства. На их базе лингвисты исследуют новые тексты, с целью выявления в них определенных паттернов и закономерностей.

Большинство известных систем (www.laurenceanthony.net/, wordsmith.org/) для работы с текстовыми корпусами предоставляют возможность хранения и аннотирования текстов, сохранения результатов в файл; поиск слов, словосочетаний, предложений по заданному шаблону. Однако данные системы имеют ряд существенных недостатков. Во-первых, от пользователя требуется владение специальнымязыком, на котором задаются правила

получения результата. Во-вторых, результаты обработки файла в конечном итоге выводятся в текстовом виде, который не является репрезентативным для конечного пользователя.

В-третьих, загружаемый в систему файл должен включать в себя следующие обязательные характеристики: текст без форматирования, который необходимо обработать, аннотация, правила, по которым производилась аннотация текста.

На отечественном сегменте также присутствуют национальные системы (http://opencorpora.org/, http://www.ruscorpora.ru/), которые предназначены только для хранения и поиска необходимой информации в корпусе. Кроме того, они не позволяют пользователям загружать собственные корпусы. Главной задачей данных систем является создание национального корпуса русского языка, поэтому они не в полной мере удовлетворяют потребностям специалистов, у которых есть собственные корпусы для анализа и обработки.

Таким образом, согласно проведенному исследованию было принято решение создать новую систему, в которой будут устранены существующие проблемы. Данная система является достаточно сложной и для ее создания потребуется разработать:

1. Хранилище для сбора всей необходимой информации.
2. API для работы с этим хранилищем.
3. Компоненты прикладного уровня взаимодействия с существующими программными комплексами для аннотирования текста.
4. Веб-редактор для создания лексико-семантических шаблонов.
5. Веб-компонент для визуализации статических данных корпусной лингвистики.

Разработка хранилища

Одним из основных компонентов таких систем является хранилище текстов. Хранилища существующих систем отвечают требованиям масштабируемости, доступности и имеют пользовательский доступ, т.е. пользователь может добавлять тексты и текстовые корпусы, удалять текстовые корпусы и тексты и т.д. Однако эти хранилища не отвечают текущим требованиям.

Во-первых, хранилище должно быть адаптивным, т.е. должно уметь сохранять структуры, которые изначально не были предусмотрены. Во-вторых, хранилище должно иметь программный доступ, т.е. предоставлять возможность другим программистам обращаться к хранилищу через интерфейсы прикладного уровня.

Разработкой текстовых корпусов занимались множество исследователей Т. Маккенри, Б. Даилле, А. Хардие. Они разработали стандартные схемы для хранения текстовых корпусов

[1] [2]. Несмотря на то, что такие схемы являются устаревшими, их можно модернизировать, чтобы удовлетворить современным требованиям.

Разработкой структур для хранения динамических данных занимались исследователи С. Амер-Яхиа, М. Фернандез Л. Ресенде и Р. Фенг. В данных работахописываются структуры для хранения динамических данные в xml формате и ориентированном графе [3] [4][5].

В ходе исследования были выявлены требования к тому, что необходимо хранить в хранилище. Данная схема представлена в упрощенной форме (рис. 1).

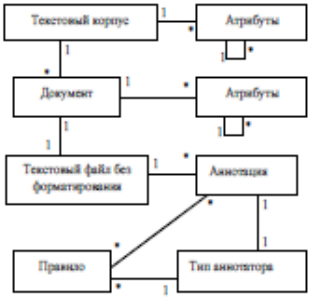

Рис. 8. Схема хранилища

Текстовый корпус должен хранить атрибуты, которые будут добавляться автоматически. Каждый из этих атрибутов может иметь свои собственные атрибуты. Каждый текстовый корпус имеет множество документов, которые в свою очередь характеризуются своими динамическими атрибутами. Каждый из этих атрибутов может иметь свои собственные атрибуты, например, атрибут автор может иметь дополнительные атрибуты, такие как дата рождения, смерти, пол и т.п.

В свою очередь документ имеет свой текстовый файл без форматирования, который будет использован для создания аннотации. Аннотацию можно создавать через разные инструменты (GATE, Clark, JTokeniser), каждый из этих инструментов имеет свои правила, поэтому для каждого из них одно и то же правило задается по-разному.

Для того, чтобы обеспечить доступ к хранилищу с удаленных устройств необходимо развернуть на облачной платформе. В качестве такой платформы для нашей системы было принято решение выбрать “Microsoft Azure”, так как он обладает гибким и удобным интерфейсом необходимый для разработки облачного хранилища.

Разработка компонентов прикладного уровня

Разработка прикладного уровня заключается в том, чтобы связать хранилище и различные аннотирующие инструменты вместе. Инструменты аннотации необходимы для

того, чтобы создавать разметку в тексте. По этой разметке можно будет идентифицировать тексты по различным параметрам. При разметке текстов, текстам приписывается дополнительная информация.

На сегодняшний день самым популярным инструментом для аннотаций текстов – это GATE. Он является системой для обработки естественного языка с открытым исходным кодом, использующая наборы компонентов на языке Java. Система решает такие задачи, как извлечение информации, ручная и автоматическая семантическая аннотация, работа с онтологиями, машинное обучение, анализ потока сообщений в блогах [6].

Существуют различные аннотирующие инструменты (GATE, Clark, JTokenize). Каждый из этих инструментов предоставляет определенный набор методов, который можно будет использовать при аннотировании текстов. Все эти инструменты можно связать одним интерфейсом "IAnnotator" (рис. 2).

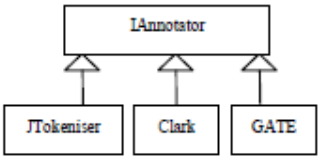

Рис. 9. Компоненты прикладного уровня

Разработка веб-редактор для создания лексико-семантических шаблонов

Для того чтобы начать анализировать текст, необходимо задать правила (шаблонные конструкции), по которым он будет аннотирован.С помощью Java Annotation Patterns Engine (JAPE)обработка текста происходит на основе регулярных выражений и позволяет разрабатывать лексико-семантические шаблоны, по которым ведется поиск.

При создании нового веб-редактораважно учитывать, что большая часть пользователей – это лингвисты, и специалисты, не связанные с компьютерными науками, поэтому разработка подобных шаблонов вызывает у них затруднения из-за сложности языка.

Данный веб-редактор необходим для создания лексико-семантических шаблонов в понятно для пользователя виде. Например, пользователь хочет найти все словосочетания, которые отвечают схеме: «Подлежащее + сказуемое» (рис. 3), система должна выделить все словосочетания, которые отвечают заданной схеме.

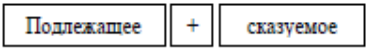

Рис. 10. Словосочетание

Разработка веб-компонента для визуализации статических данных

В большинстве случаев лингвист получает выходные результаты в текстовом документе, и ему необходимо её визуализировать для большей наглядности и для выявления интересных трендов и закономерностей, так как визуальная модель обладает большей когнитивной силой.

Веб-приложения (www.sketchengine.co.uk, cqpweb.lancs.ac.uk), которые существуют сегодня для работы с текстовыми корпусами, обладают сильным лингвистическим аппаратом, но представить в наглядном виде полученную информацию в ходе обработки текста не могут. Функционал большинства из них сложен для понимания. Поэтому лингвисту, который первый раз столкнулся с данным приложением, сложно понять итоговую картину. Статистические данные, полученные в ходе разбора текста сложны для интерпретации, что замедляет процесс анализа документа.

Поставленная задача решается с использованием методов математической статистики, комбинаторики, теории графов и множеств, а также технологии веб-программирования: высокоуровневый язык программирования – JavaScript, язык разметки – HTML5 и языка стилей – CSS3 [7].

Например, пользователь хочет найти соотношение односоставных предложений в тексте и для визуального представления полученных данных он использует диаграмму (рис.4)

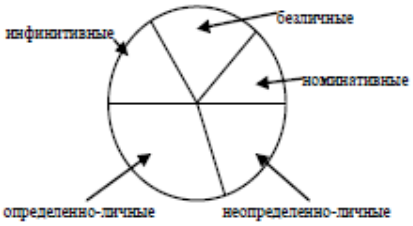

Рис. 11. Тип односоставных предложений

Заключение

В заключении данной статьи можно сказать, что описана последовательность шагов для разработки системы обработки корпусов. В данной системе большинство компонентов являются взаимозаменяемыми. Это делает ее более гибкой, тем самым позволяя модифицировать части системы независимо друг от друга.На данном этапе работы были спроектированы части системы, а также реализована работа хранилища. В дальнейшем планируется продолжить разработку системы и разработать оставшиеся части системы.

Библиографический список

1. *McEnery T. & Daille B.*,Database Design For Corpus Storage: The ET10-63 Data Model,UCREL Technical Papers, Lancaster, 1993.
2. *Stuart D., Aitken B. & Abbott D.*,Content Models for Enhancement and Sustainability: Creating a Generic Framework for Digital Resources in the Arts and Humanities in Metadata and Semantic Research, German: Springer Berlin Heidelberg, 2011.
3. *Amer-Yahia S., Fernandez M., Greer R., and Srivastava D.*, "Logical and Physical Support for Heterogeneous Data," in Eleventh Int. ACM Conference on Information and Knowledge Management, McLean, VA: ACM Press Nov. 2002
4. *Resende L. and Feng R.*, Handling heterogeneous data sources in a SOA environment with service data objects, in Proceedings of the 2007 ACM SIGMOD international conference on Management of data, New York, NY: ACM Press, 2007.
5. *Cunningham H., Maynard D., Bontcheva K.*, et al. Developing Language Processing Components with GATE Version 7. The University of Sheffield.
6. *Amelie Kutter* and *Cathleen Kantner* 2011. How to get rid of the Noise in the Corpus: Cleaning Large Samples of Digital Newspaper Texts. University of Stuttgart, Dept of International Relations & European Integration International Relations Online Working Paper Series.
7. *Паклин Н. Б., Орешков В. И.* Визуализация данных. Отданныхкзнаниям. – 2-еизд. – Спб.: Питер, 2013.

SYSTEM FOR MANIPULATING TEXT CORPORA

Barmina E.I., Bushuev R.N., Kotel'nikova N.V., Lanin V.V., Plotnikova O.A.

National Research University Higher School of Economics,st. Studencheskaya, 38, Perm, Russia, 614070, elenabarmina@ya.ru, bushuev.roman@gmail.com, nadya.kott@gmail.com, lanin@perm.ru, simple_oks@rambler.ru

The topic of research is system for manipulating text corpora. In this article it will observed existing solutions and briefly described advantages and disadvantages of them. We will take a closer look at the aim and objectives. The aim of research is developing system for manipulating text corpora. This research revealed four parts of the work process such as creating data storage, experimental-level component, web editor and visual web application. In conclusion it will be included the outputs of our research and further plans.

Key words: corpus linguistics, data visualization, DSL, cloud storages