ПРИМЕНЕНИЕ ЦЕПЕЙ МАРКОВА ДЛЯ РЕШЕНИЯ ЗАДАЧИ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ЭКВИВАЛЕНТНОГО ОБРАЗЦУ ТЕКСТА

Е. Д. Гуменников

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель И. А. Мурашко

Автоматическая генерация эквивалентного текста по предоставленному оригиналу подразумевает автоматическое создание текста, наделенного схожим с оригиналом семантическим наполнением. Решение данной задачи может найти применение во многих отраслях, таких как разработка обучающих лингвистических программ, чатботов, систем автоматической коррекции текста. Исходными данными для этой программы является текст-оригинал, выходными – текст, эквивалентный тому, что был подан в качестве входных данных.

Существует множество подходов для решения задач NLP – от разнообразных алгоритмических способов до применения всевозможных нейронных сетей. В этой работе будет рассмотрено применение цепей Маркова для решения поставленной задачи.

В общем под Марковской цепью понимают специфичную комбинацию событий, где вероятность возникновения последующего события зависит от того, в каком состоянии процесс находится в настоящее время или находился незадолго до рассматриваемого момента. При этом зачастую ранние состояния этого процесса не оказывают влияния на будущий элемент генерируемой последовательности. Цепи Маркова как генератор текста, как правило, реализуются следующим образом: в начале программа подготавливает обучающий текст, причем, чем больше будет этот текст, тем более приемлемым получится результат генерации. Затем из исходного текста алгоритм составляет выборку типа «слово – последующее слово», или «словосочетание – последующие слово», рассчитывается вероятность возникновения слова после той либо иной конструкции и все это записывается как база знаний правил генерации, затем в выборку добавляется коллекция слов, с которых исходный текст может начаться. После из этой коллекции выбирается одно слово и помещается в результирующий текст, таким образом появляется первое слово исходного текста. После этого добавляется новое слово в соответствии с базой знаний генерации, затем алгоритм потеряется от последнего элемента цепочки. Максимальная длина цепочки элементов ограничивается, соответственно, количеством элементов в исходном тексте. В результате этого получается текст, внешне похожий на естественно-языковой. Однако стоит понимать, что смысл текста, сгенерированного с помощью цепей Маркова, будет, к сожалению, отсутствовать, несмотря на то, что в целом слова и даже предложения могут быть взаимосвязаны друг с другом. Однако, если база знаний алгоритма будет основана на текстах с идентичным семантическим значением, высока вероятность того, что сгенерированный текст будет семантически идентичен тексту, на основании которого была собрана база знаний.

Применить цепи Маркова для решения задачи генерации эквивалентного текста по оригинальному тексту можно, реализовав алгоритм, основанный на трех шагах:

1) определить семантическое значение оригинала;

2) выделить базу знаний в соответствии с семантическим значением оригинала;

3) сгенерировать по выбранной базе знаний текст.

Такой подход даст действительно хорошие результаты, однако его реализация станет достаточно сложной.

Первой проблемой, стоящей на пути к реализации описанного алгоритма, будет использование метода определения семантического значения текста оригинала. Автоматическое определение семантического значения – это серьезная задача, не менее сложная чем генерация, хоть и более разработанная.

Вторая проблема – это сбор достаточного объема баз знаний для всевозможных тематик исходного текста. Эта задача воистину непосильна, если, конечно, не ограничиться жестким набором тем, над которыми генератор сможет работать.

Однако есть и другой способ применения цепей Маркова для решения поставленной задачи. Следует подготовить большую базу знаний – статистическую модель языка, на котором будут подаваться оригиналы текстов, эквиваленты которых нужно сгенерировать. Также, чтобы улучшить результаты генерации, не лишним будет воспользоваться базой знаний, содержащей словари синонимов и аналогичных словосочетаний. Далее необходимо выделить ключевые слова и короткие комбинации слов из исходного текста, подобрать им синонимы и эквивалентные сочетания, после чего, исходя из собранной коллекции ключевых слов, отредактировать статистическую модель целевого языка, увеличив вероятность возникновения ключевых слов оригинала и их синонимов в тексте в случае, если исходная вероятность их появления не равна нулю. Затем выполнить стандартный алгоритм генерации текста с помощью цепи Маркова. Схема приведенного метода изображена на рис. 1.

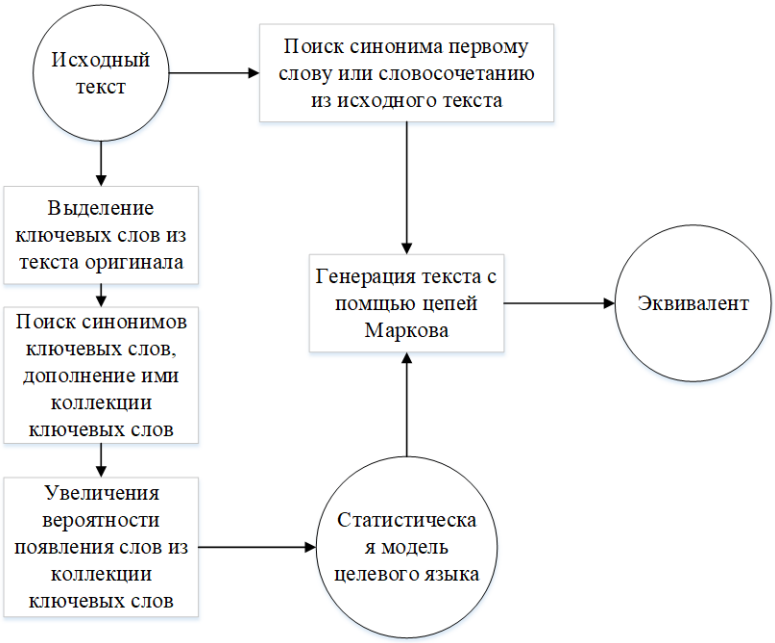

Рис. 1. Схема метода применения цепей Маркова для решения генерации эквивалентных текстов

Применение этого алгоритма требует решения другой задачи NLP – выделения ключевых слов. Наиболее простым методом решения для данной задачи является статистический метод определения ключевых слов. Выборка ключевых слов формируется через расчет частоты появления всех слов в тексте, затем вбираются самые часто встречающиеся слова. Данный метод широко применяется благодаря своей простоте, так как он не требует специфических баз знаний и шаблонов. Однако стоит сказать, что частота употребления слова в тексте – не идеальный параметр для такой

оценки, так как этот признак не является актуальным для некоторого набора слов, например, местоимений, междометий и ряда других слов. Но это достаточно просто преодолеть, составив базу слов, которые необходимо заранее исключить из списка ключевых. Такую базу легко собрать, проанализировав большое количество разношерстных текстов, выделив *N*-е количество наиболее употребляемых слов. Чистый статистический метод может быть очень эффективен применительно к языкам со скудной морфологией, где каждое слово, вероятнее всего, не имеет огромного набора форм, например, в английском языке.

Литература

1. Мурашко, И. А. Оптимизация проектных решений : курс лекций для студентов специальностей 1-40 01 02 / И. А. Мурашко, Д. Е. Храбров. – Гомель : ГГТУ им. П. О. Сухого, 2014. – 94 с.
2. Розанов, А. К. Быстрый алгоритм анализа словоформ естественного языка с трехуровневой моделью словаря начальных форм / А. К. Розанов // Cloud of science. – 2016. – № 1. – С. 115–124.
3. Осминин, П. Г. Построение модели реферирования и аннотирования научно-технических текстов, ориентированной на автоматический перевод : автореф. дис. … канд. филол. наук: 10.02.21 / П. Г. Осминин. – Челябинск : ФГБОУ ВПО ЮУрГУ, 2016. – 108 с.

ПРИМЕНЕНИЕ МЕТОДОВ BIG DATA В ПРОГРАММНОМ КОМПЛЕКСЕ КОНСОЛИДАЦИИ ИНФОРМАЦИИ ОБ АВИАРЕЙСАХ

Ю. Ю. Пашковская

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Т. А. Трохова

В настоящее время одной из актуальных задач в области получения информации об авиасообщениях является задача автоматизации процесса консолидации больших объемов данных, включающая возможность получения выборок различного типа по запросу пользователя. Тема доклада посвящена решению этой актуальной проблемы. Разработанная система предназначена для формирования хранилища больших объемов данных с применением методологий и технологий Big Data. Применение этой системы позволит пользователям получать достоверную информацию об авиаперелетах по разным категориям запросов в короткое время.

Программный комплекс разработан на платформе Spring boot. Одним из отличительных моментов платформы Spring boot является применение паттерна MVC.

Концепция паттерна MVC предполагает разделение приложения на три компонента:

– модель (model);
– представление (view);
– контроллер (controller).

В качестве модели выбраны java классы, описывающие все поля, которые будут располагаться в базе данных. Так называемые pojo объекты для описания методов GET и SET; будет использоваться библиотека генерации кода Lombok. Она позволяет существенно уменьшить объем кода.

Представлением являются страницы html, которые содержат код пользовательского интерфейса в основном на языке html с thymeleaf.

В контроллере предполагается обработка запросов пользователя. Для работы с базой данных реализован репозиторий. Он необходим для того, чтобы работа с ней