Прототип системы автоматического определения несоответствий в текстовых данных в реальном времени

И. В. Никифоров[1], Н. В. Воинов[2], П. Д. Дробинцев[3]
Санкт-Петербургский политехнический университет Петра Великого
[1]igor.nikiforovv@gmail.com, [2]voinov@ics2.ecd.spbstu.ru, [3]drob@ics2.ecd.spbstu.ru

***Аннотация*. Новые современные технологии обработки больших данных сделали возможным внедрение таких аспектов информационной безопасности, как прогнозирование и выявление подозрительных неавторизованных действий еще до того, как они фактически произойдут. Например, определение нестандартных банковских транзакций, проверка соответствия выписанного рецепта и курса лечения диагнозу, соответствие темы работы ее содержимому и т.д. Для реализации описанной функциональности необходима система для автоматического определения несоответствий на основе проверки корректности данных. А поскольку объемы данных в современном мире уже исчисляются петабайтами и время на их обработку – очень важный критерий, подобная система должна функционировать в режиме реального времени. В работе рассмотрены метод и реализующий его алгоритм в рамках прототипа системы автоматического определения несоответствий в неформализованном тексте на естественном языке в режиме реального времени.**

Ключевые слова: определение несоответствий в текстовых данных; семантическое сходство; обработка данных в реальном времени; Word2Vec; Apache Storm

I. Существующие методы и программные средства обнаружения несоответствий

Для решения задачи автоматического определения несоответствий можно выделить три основных метода: методы определения выбросов, методы машинного обучения с учителем и методы машинного обучения без учителя. Рассмотрим каждую из данных категорий.

Простейшие методы определения выбросов работают для одномерных численных выборок. Самым тривиальным способом определения выбросов является использование 3σ интервала, то есть вычисление для исходного набора верных данных математического ожидания m и среднеквадратического отклонения, после чего считать выбросами те приходящие данные, которые не попадают в интервал [m–3σ; m+3σ]. Данный метод не обладает высокой степенью точности, так как математическое ожидание не является робастной [1] характеристикой. Соответственно, в реальных системах данный метод практически не используется.

В отличие от математического ожидания, медиана и квантили являются робастными характеристиками выборки. Весьма распространен метод [2], предложенный Джоном Тьюки и основанный на межквартильном расстоянии.

Для многомерных численных выборок данных возможно использовать методы машинного обучения без учителя. Одним из самых распространённых и надёжных методов кластеризации без учителя является k-means – метод k-средних. Главная идея этого метода заключается в минимизации суммы квадратичных отклонений расстояний векторов от центров кластеров [3].

Одним из методов машинного обучения с учителем является логистическая регрессия – статистическая модель, позволяющая предсказать вероятность наступления некоторого события.

Среди программных систем, реализующих представленные выше методы, можно выделить следующие категории:

- системы, построенные на базе специализированных платформ (например, фреймворк Neo4j [4]);
- системы, направленные на использование в определенных сферах деятельности (например, CPA detective [5] в маркетинговой отрасли или Software AG Banking: Fraud detection [6] в финансовых компаниях);
- специализированные системы, реализующие методы поиска определенных артефактов в текстовых данных [7, 8].

Вышеперечисленные системы являются системами реального времени и используют методы интеллектуального анализа данных для мониторинга информации. Однако ни одна из систем не работает с неформализованным текстом, а если и работает, то принятие решения о корректности данных требует значительных временных вычислений.

II. Концепция системы автоматического определения несоответствий в текстовых данных

Принцип работы системы заключается в предварительном анализе входной информации перед записью непосредственно в целевое хранилище, например, базу данных. О найденных несоответствиях выдается предупреждение пользователю. Если несоответствий не обнаружено, то данные напрямую записываются в хранилище.

Новый пакет данных поступает на вход модуля приемки. На основе конфигурации системы происходит разбиение входных данных на структуры и блоки. Каждый блок – это атомарно анализируемая порция, которая в общем случае представлена одномерным массивом разнотипных значений. В то же время на основе конфигурации система запрашивает нужные для сравнения данные из целевого хранилища. В отличие от входных данных, из хранилища возвращается эталонный набор блоков. После чего задача определения выбросов сводится к сравнению входного блока с массивом эталонных блоков из хранилища. Результатом работы системы является вердикт для пользователя.

Какие именно поля данных должны проходить проверку на выбросы, определяется пользователем при создании конфигурационного файла. На основе него происходит разбор входных данных и выбор соответствующих методов для анализа отдельных полей входных данных.

A. Метод определения семантической близости текстов

Модулем системы, представляющим особый интерес, является модуль анализа и сравнения данных.

Для текстовой неформализованной информации под несоответствием будем понимать семантическое несоответствие входного текста эталонной тематике, заданной другим текстом или предложением. Например, несовпадение смысла заголовка содержимому письма, несоответствие текста дипломной работы или статьи ее заголовку, несоответствие назначенного лекарства или диагноза анамнезу, спам в электронной почте и т.д.

Для определения подобного несоответствия предлагается метод, основанный на вычислении косинусной близости между входным анализируемым предложением и остальными предложениями в тексте. Для реализации подобного метода необходимо преобразовывать предложения естественного языка в численные значения для дальнейшей обработки математическими методами. В данной работе используется алгоритм Word2Vec [9], который эффективно преобразует представления слов в векторное пространство.

Word2Vec – это набор алгоритмов обработки естественного языка, использующий двухслойные нейронные сети для векторного представления слов. Отличительной особенностью Word2Vec является наличие готовой языковой модели, которая содержит информацию о том, какие слова есть в языке и как они соотносятся друг с другом.

B. Алгоритм поиска несоответствий

Вначале входное предложение и входной текст преобразуются в нормальную форму (например, «собаки» – в «собака», «сделал» – в «сделать»). Для этого используется сторонний пакет jmorphy2, позволяющий приводить слова к нормальной форме, ставить слово в нужную форму (менять падеж, число), возвращать грамматическую информацию о слове.

Затем слова из нормализованного текста, которых нет в языковой модели, и предложения, все слова в которых отсутствуют в языковой модели (например, предложения, написанные на другом языке, предложения, состоящие из малоизвестных слов), удаляются. Далее каждое из слов приводится к векторной форме с помощью алгоритма Word2Vec. Алгоритм обучается так, что слова близкие по контексту имеют высокую косинусную близость. Сумма векторов слов даёт вектор слова, несущего «общий смысл». Например «король» + «женщина» + «ребёнок» = «принцесса». Это свойство используется для решения поставленной задачи.

Для каждого предложения полученные вектора суммируются, что даёт общий вектор предложения, показывающий итоговый смысл. Далее составляются две выборки, первая выборка содержит косинусные расстояния между каждыми двумя предложениями текста, а вторая выборка - косинусные расстояния между входным предложением и каждым предложением входного текста. Затем к полученным выборкам применяется критерий Манна-Уитни [10]. Критерий используется для оценки различий между двумя независимыми выборками по уровню какого-либо признака, измеренного количественно, а также позволяет выявлять различия в значении параметра между малыми выборками. В данном случае между значениями смыслового отклонения предложений текста друг от друга и значениями отклонения входного предложения от предложений текста. В случае, если критерий возвращает значение, меньшее заданного заранее пользователем значения α, алгоритм отвергает гипотезу о равенстве значений, тем самым утверждая, что входной текст и входное предложение по смыслу отличаются.

Общая схема алгоритма представлена на рис. 1.

Данный алгоритм был реализован в программной библиотеке, позволяющей определять, соответствует ли по смыслу входное предложение входному тексту. Программным интерфейсом библиотеки является класс JSentenceDetection с единственным публичным методом detect. На вход метод принимает предложение и текст и возвращает логическое значение, показывающее, соответствует ли по смыслу предложение тексту.

III. Реализация прототипа системы

Для эффективной работы система автоматического определения несоответствий поддерживает обработку данных в реальном времени. Для реализации такой обработки используется современный фреймворк обработки данных Apache Storm [11].

Рис. 1. Схема алгоритма определения несоответствий

Сокращение времени обработки достигается за счет распараллеливания независимых друг от друга модулей. Система стремится к эффективному расслоению на подзадачи, которые могут исполняться параллельно без задержки общего потока данных. Фреймворк, на основе которого осуществляется построение топологии исполнения, осуществляет распараллеливание автоматически на заранее сконфигурированное количество блоков.

В прототипе системы модули получения входных данных, извлечения данных по конфигурации из целевого хранилища, анализа, записи в хранилище после анализа, а также взаимодействия с пользователем не имеют общей логики, поэтому эти части разделяются на отдельные элементы в общей топологии системы. Наиболее существенный недостаток топологии – большая загруженность узла анализа данных по сравнению с остальными модулями. Такая проблема возникает из-за того, что процесс анализа данных по сравнению с остальными процессами наиболее долгий. Пока появляются новые данные, поток до узла анализа данных загружен, а после данного узла простаивает. Решением проблемы является разбиение узла на более мелкие подзадачи, т.к. методы проверки данных работают независимо друг от друга, а их результаты суммируются лишь в конце всего потока для вынесения общего вердикта.

IV. Результаты апробации

Рассмотрим несколько примеров применения системы. Используем в качестве входного текста статью о влиянии кофе на артериальное давление, а в качестве входного предложения: «У людей, страдающих гипертонией, давление может резко и сильно вырасти до критических значений, угрожающих здоровью».

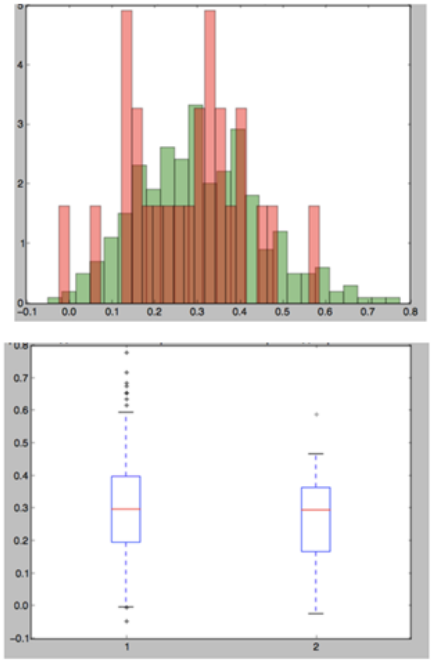

Рис. 2. Результаты для входного предложения с похожим смыслом

После применения разработанной системы получены результаты, представленные на рис. 2. На верхней части рисунка красная гистограмма относится к распределению значений соответствия входного предложения предложениям из текста, а зеленая – к распределению значений соответствия предложений текста друг другу. На нижней части рисунка первая диаграмма размаха показывает распределение значений соответствия предложений текста друг другу, а вторая – распределение значений соответствия входного предложения предложениям из текста. Из гистограмм и диаграмм размаха видно, что предложение соответствует тексту по смыслу. Что подтверждается результатом работы теста со значением равным 0.454577.

Теперь рассмотрим входное предложение «Кредитный рейтинг России в 2016 году достигнет рекордных величин», оставив входной текст неизменным.

Из графиков, представленных на рис. 3, видно смысловое несоответствие предложения и текста, что подтверждается результатом работы теста со значением равным 0.

V. Заключение

В работе был предложен метод автоматического определения несоответствий в текстовых данных, реализованный в прототипе программной системы. Метод основан на алгоритме Word2Vec для представления слов в векторном пространстве. Основной технологией, реализующей поддержку обработки в режиме реального времени, является фреймворк Apache Storm.

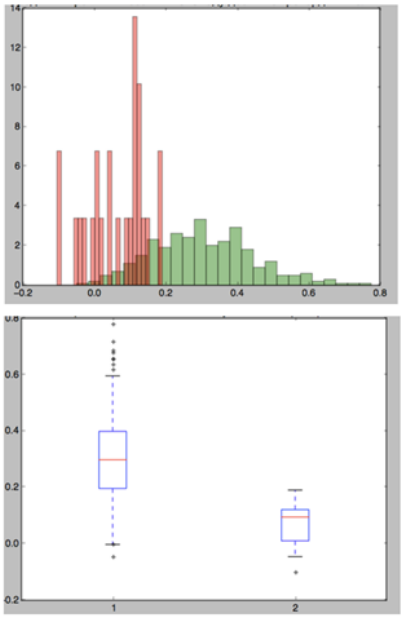

Рис. 3. Результаты для входного предложения с непохожим смыслом

В рамках работы была создана библиотека определения корреляции между предложениями неформализованного текста. Её работоспособность проверена на множественных наборах данных. Библиотеку можно использовать для различных систем определения несоответствия в текстовых данных.

Список литературы

[1] Hampel F.R., Ronchetti E.M., Rousseeuw P.J., Stahel, W.A. Robust Statistics. The Approach Based on Influence Functions. New York: John Wiley and Sons, 1986.

[2] Tukey J.W. Exploratory data analysis. Reading, PA: Addison-Wesley, 1977.

[3] MacKay D. Information Theory, Inference and Learning Algorithms. Cambridge: Cambridge University Press, 2003, pp. 284–292.

[4] Neo4j Official website. Available at https://neo4j.com/

[5] CPA detective Official website. Режим доступа: http://cpadetective.com/technology.html.

[6] Software AG Banking fraud detection. Режим доступа: http://softwareag.com/us/solutions/banking/fraud_detection/overview/default.asp

[7] Koznov D.V., Romanovsky K.Y. DocLine: A method for software product lines documentation development. Programming and Computer Software. 2008, vol. 34, i. 4, pp. 216-224. DOI: https://doi.org/10.1134/S0361768808040051

[8] Koznov D.V., Luciv D.V., Basit H.A., Lieh O.E., Smirnov M.N. Clone detection in Reuse of software documentation. Lecture Notes in Computer Science. 2015, vol. 9609, pp. 170-185. DOI: https://doi.org/10.1007/978-3-319-41579-6_14

[9] Официальный сайт Word2Vec. Режим доступа: https://code.google.com/archive/p/word2vec/

[10] Mann H.B., Whitney D.R. On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistic. 1947, vol. 18, i. 1, pp. 50-60. DOI: 10.1214/aoms/1177730491

[11] Официальный сайт Apache Storm. Режим доступа: http://storm.apache.org