УДК 004.62 **DOI: 10.18413/2518-1092-2021-6-1-0-5**

Наумов Р.К.
Железков Н.Э.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФОРМАТОВ ХРАНЕНИЯ ТЕКСТОВЫХ ДАННЫХ ДЛЯ ДАЛЬНЕЙШЕЙ ОБРАБОТКИ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики», Кронверкский пр., д. 49, г. Санкт-Петербург, 197101, Россия

e-mail: ruslan.naumow.dake@gmail.com, nikita.e.zhelezkov@gmail.com

Аннотация
На сегодняшний день одним из перспективных направлений в области информационных технологий является машинное обучение. Оно используется во многих сферах деятельности, в том числе и в анализе текстовых данных. Между этапами сбора данных и их анализом располагается этап хранения данных. Одним из вопросов, требующих тщательного рассмотрения, является выбор формата хранения этих данных. Настоящая статья представляет собой обзор наиболее популярных форматов хранения текстовых данных, используемых в машинном обучении. Определены критерии, по которым произведено сравнение. Результатом работы является сравнительная таблица анализируемых форматов. Исходя из результатов, сделан вывод о наиболее эффективном способе хранения текстовых данных.
Ключевые слова: машинное обучение, текстовые данные, текстовые форматы, сериализация данных.
Для цитирования: Наумов Р.К., Железков Н.Э. Сравнительный анализ форматов хранения текстовых данных для дальнейшей обработки методами машинного обучения // Научный результат. Информационные технологии. – Т.6, №1, 2021. – С. 40-47. DOI: 10.18413/2518-1092-2021-6-1-0-5

Naumov R.K.
Zhelezkov N.E.

COMPARATIVE ANALYSIS OF TEXT DATA STORAGE FORMATS FOR FURTHER PROCESSING BY METHODS OF MACHINE LEARNING

Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 49 Kronverkskiy prospekt, St. Petersburg, 197101, Russia

e-mail: ruslan.naumow.dake@gmail.com, nikita.e.zhelezkov@gmail.com

Abstract
Today, one of the most promising areas in the field of information technology is machine learning. It is used in many areas of activity, including text data analysis. Between the data collection and analysis stages, there is a data storage stage. One of the issues that requires careful consideration is the choice of storage format for this data. This article provides an overview of the most popular text data storage formats used in machine learning. The criteria for the comparison are determined. The result of the work is a comparative table of the analyzed formats. Based on the results, a conclusion is made about the most efficient way to store text data.
Keywords: machine learning, text data, text formats, data serialization.
For citation: Naumov R.K., Zhelezkov N.E. Comparative analysis of text data storage formats for further processing by methods of machine learning // Research result. Information technologies. – T.6, №1, 2021. – P. 40-47. DOI: 10.18413/2518-1092-2021-6-1-0-5

ВВЕДЕНИЕ

Текстовый канал коммуникаций – это способ обеспечения коммуникации в цифровой среде. К нему относят социальные сети, мессенджеры, чаты, электронные письма. Текстовый

канал является основным способом сбора текстовых данных [19], которые в зависимости от цели их получения в дальнейшем либо хранятся и обрабатываются для извлечения необходимой информации [1]. Стоит отметить, что цель получения данных также играет роль в способе их хранения. [12, 13, 14]

Помимо выбора системы хранения данных (СХД) [18] важным фактором в хранении, обработке и передачи данных является формат хранения этих данных. Именно от формата зависит скорость работы с данными, объем хранимых файлов, их взаимодействие с различными системами. В отличие от привычных нам структурированных данных, методы извлечения информации из которых хорошо известны [2], текстовые данные в большинстве случаев являются неструктурированными или слабоструктурированными, то есть не имеют заранее определенной структуры и не организованы в установленном порядке. Несмотря на то, что некоторые авторы описывают подходы к обработке таких данных [3, 4], данный факт приводит к трудностям анализа, особенно в случае использования традиционных программ, предназначенных для работы со структурированными данными. Именно поэтому для машинного обучения [15, 16] текстовые данные переводятся в специализированные форматы.

На сегодняшний день в области машинного обучения наиболее часто встречаются следующие форматы хранения текстовых данных:

- CSV – текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми [5].
- JSON – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается человеком [6].
- XML – простой, очень гибкий текстовый формат, являющийся подмножеством SGML (ISO 8879) [8], который позволяет определять собственные теги и атрибуты [9].

Целью статьи является исследование наиболее популярных форматов хранения данных для их последующего анализа методами машинного обучения. Под текстовыми данными в работе рассматривается выгруженные чаты видеоконференций на платформе Zoom.

ОСНОВНАЯ ЧАСТЬ

Для успешного анализа приведенных форматов необходимо определить ряд критериев, которым формат должен соответствовать в той или иной степени. Авторы статьи [9] для сравнительного анализа форматов обмена данными используют следующие критерии:

1. Размер файла. Файл должен быть небольшим для быстрой загрузки файла в базу данных, а также быстрой передачи по сети;
2. Скорость парсинга (сериализации/десериализации). Формат должен минимизировать расходы на парсинг текста, тем самым увеличивать скорость обработки;
3. Поддержка библиотеками Python. Превосходная гибкость и простота использования Python делают его одним из самых популярных языков программирования. [20] Именно поэтому обучать системы принято на данном языке, а значит данный критерий является неотъемлемым;
4. Удобство к чтению и редактированию. Текст, хранящийся в формате, должен быть прост к восприятию и внесению изменений, то есть формат должен быть «человекочитаемым».

Определив основные критерии, перейдем к подробному рассмотрению каждого из представленных форматов.

Данные в CSV. В этом формате каждая строка файла — это строка таблицы. Несмотря на название формата (Comma-Separated Values – значения, разделённые запятыми), разделителем может быть не только запятая. И хотя у форматов с другим разделителем может быть и собственное название, например, TSV (tab separated values), тем не менее, под форматом CSV понимают, как правило, любые разделители.

Автор [7] в своей книге приводит следующий пример файла в формате CSV:

```
hostname,vendor,model,location
sw1,Cisco,3750,London
sw2,Cisco,3850,Liverpool
```

Рис. 1. Пример файла CSV
Fig. 1. CSV file example

Как видно из рисунка 1 данный формат прост к восприятию. Понятно, что первая строка – это названия столбцов, а последующие – строки. Файл содержит несколько строк, а его объем составит 81 байт.

В стандартной библиотеке Python есть модуль csv, который позволяет работать с файлами в CSV формате. Также для работы с данным форматом доступен класс DictReader.

Данные в JSON. Формат JSON часто используется в API. Кроме того, этот формат позволит сохранить такие структуры данных как словари или списки в структурированном формате и затем прочитать их из файла в формате JSON и получить те же структуры данных в Python [7].

После преобразования файла из рисунка 1 в формат JSON получится следующий файл:

```
{
"hostname": "sw1",
"vendor": "Cisco",
"model": 3750,
"location": "London"
},
{
"hostname": "sw2",
"vendor": "Cisco",
"model": 3850,
"location": "Liverpool"
}
```

Рис. 2. Пример файла JSON
Fig. 2. JSON file example

Из рисунка 2 видно, что те же самые текстовые данные, рассмотренные ранее принимают более громоздкий формат. Строк и специальных обозначений становится больше, следовательно растет объем хранимого файла. Теперь объем текстовых данных равен 203 байта. Однако, структурированность данных устраняет сложности в обработке информации.

Как и в случае с CSV, в Python есть модуль, который позволяет легко записывать и читать данные в формате JSON.

Данные в XML. XML – это самодокументируемый формат, который описывает структуру и имена полей так же, как и значения полей.

Первоначальные текстовые данные в формате XML выглядит следующим образом:

```
<Records>
  <Record>
    <Row
      A="hostname"
      B="vendor"
      C="model"
      D="location"
    />
  </Record>
  <Record>
    <Row
      A="sw1"
      B="Cisco"
      C="3750"
      D="London"
    />
  </Record>
  <Record>
    <Row
      A="sw2"
      B="Cisco"
      C="3850"
      D="Liverpool"
    />
  </Record>
```

Рис. 3. Пример файла XML
Fig. 3. XML file example

Из рисунка 3 можно сделать вывод, что файл XML среди трех рассматриваемых форматов является самым «тяжелым». Он содержит больше всего строк и специальных символов, а его вес равен 455 байт. Понимание текстовых данных, хранимых в XML файле, по сравнению с предыдущими форматами становится более затрудненным.

В Python документы XML могут быть обработаны несколькими способами. Язык имеет традиционные парсеры DOM и SAX, но автор работы [10] сфокусировался на другой библиотеке под названием ElementTree. Модуль ElementTree входит в стандартную библиотеку Python и находится в xml.etree.ElementTree.

Авторы статьи [5] подробно провели анализ эффективности скорости сериализации-десериализации файла в форматах XML и JSON. Парсером JSON был выбран FastJSON, его преимущества подробно рассмотрел другой автор [17]. Парсером XML авторы статьи выбрали стандартный механизм от Майкрософт – Microsoft XML Parser. Для определения скорости они использовали файл конфигурации базы данных PDM системы в формате XML. Результаты приведены в таблице:

Таблица 1

Сравнение форматов сериализации-десериализации

Table 1

Comparison of serialization-deserialization formats

Метод и средства сериализации-десериализации	Время операции сериализации, миллисекунды	Время операции десериализации, миллисекунды
JSON (FastJSON)	279	644
XML (Microsoft XML Parser)	773	461

Разница времен десериализации обусловлена эффективностью алгоритма сжатия данных. При обработке небольших объемов данных JSON будет более компактным, однако при работе с более сложными структурами эффективность сжатия XML будет выше, чем у JSON. [5]

Авторы статьи [11] провели исследование, в котором определяли, эффективный способ хранения данных. За исходные данные в работе выбраны текстовые файлы, документы, не прошедшие обработку для последующего анализа. Авторы отмечают, что данные могут содержать пробелы или дубликаты, ошибочные значения, противоречия. Такие данные требуют преобразования к виду, удобному для применения аналитических алгоритмов, то есть применения процедур и методов, позволяющих извлечь данные из разнообразных источников, преобразовать их в единый формат, нормировать данные, избавиться от пробелов, дубликатов, ошибок данных. В результате исследования они получили следующее:

Таблица 2

Анализ эффективности CSV формата

Table 2

CSV format performance analysis

Способ хранения	Время добавления 10 000 записей, сек	Время получения 100 000 записей, сек	Время обновления 10 000 записей, сек
CSV файл	4,6	1,576	9

Исследователи выяснили, что самым эффективным способов хранения таких данных является CSV файл. Исходя из таблицы 2, можно отметить очень высокую скорость работы CSV формата с текстовыми данными.

РЕЗУЛЬТАТ ИССЛЕДОВАНИЯ

По результатам анализа трех форматов хранения текстовых данных построена сравнительная таблица (Таблица 3), включающая критерии, описанные ранее.

Таблица 3

Сравнительная таблица форматов

Table 3

Format Comparison Table

Критерий	CSV	JSON	XML
Размер файла, байт	81	203	455
Среднее время операции над данными, мс	45	461,5	617
Поддержка библиотеками Python	Да	Да	Да
Человекочитаемость	5	4	3

По результатам проведенного исследования, отраженным в таблице 3, видно, что формат CSV для одинаковых данных по сравнению с форматами JSON и XML занимает меньший объем память компьютера. Авторы статей [9, 11] объясняют это тем, что форматы JSON и XML поддерживает иерархичность структур, то есть упрощает хранение связанных данных. В свою очередь CSV-файлы изначально не могут представлять иерархические или реляционные данные. Связи между данными обычно обрабатываются с использованием нескольких CSV-файлов. Однако, сообщения, получаемые из текстового канала связи можно организовать в простую плоскую таблицу.

Файлы в формате JSON чаще используются в решениях обмена горячими данными. Документы JSON часто отправляются с веб- и мобильных устройств, выполняющих онлайн-транзакции, устройств Интернета вещей. [21]

Одной из основных проблем XML-файлов является крайняя детализация, тем самым данные из таких файлов обрабатываются дольше. Необходимость такой детализации обусловлена расширяемостью формата. Таким образом, разработчики используют XML для хранения динамичных данных.

На примере представленных работ можно заметить зависимость эффективности форматов – чем больше формат требует объема, тем медленнее происходит работа с данными. Отметим, что рассматриваемые форматы достаточно популярны, так как каждый имеет модуль в последних версиях ЯП Python.

Из результатов, полученных в исследовании, можно сделать вывод, что для хранения текстовых данных, полученных из чата видеоконференции Zoom, целесообразно использование формата CSV. Получаемые данные не имеют сложной структуры, поэтому хранение их в CSV-файлах приведет к меньшим потерям в объемах занимаемой памяти и более эффективной обработке информации.

ЗАКЛЮЧЕНИЕ

В данной статье рассматривались три формата хранения текстовых данных: CSV, JSON и XML. Были заданы критерии для определения наиболее подходящего формата для дальнейшей обработки методами машинного обучения. Исследование проводилось на основе работ разных авторов. Оценка объема файла и «человекочитаемости» в конкретном формате на основе работы одного из авторов [7] показала преимущества формата CSV. Авторами других статей [5, 11] оценен критерий скорости работы с данными, в котором наилучшим образом показали себя CSV и JSON форматы. Также исследование показало, что рассматриваемые форматы довольно популярны среди разработчиков Python. Сделан вывод по зависимости эффективности форматов от требуемого объема для одинакового набора текстовых данных.

В настоящей работе, собрана лишь начальная информация для комплексного и доказательного рассмотрения вопроса эффективности различных форматов хранения текстовых данных. Отметим, что работа может быть дополнена более подробным рассмотрением критериев форматов, а также внедрением новых критериев оценки эффективности. Результаты данного исследования могут быть использованы в дальнейшем определении структуры и системы хранения данных, сериализации и передачи данных.

Список литературы

1. Извлечение информации из разноструктурированных данных и её приведение к целевой схеме / Брюхов Д.О., Ступников С.А., Калиниченко Л.А., Вовченко А.Е. // Selected Papers of the XVII International Conference on Data Analytics and Management in Data Intensive Domains. 2015. С. 81–90.
2. Бедарев Н.В., Войнов А.А. Тексты на естественном языке и методы извлечения структурированных данных // Международная научно-технологическая конференция студентов и молодых ученых «Молодёжь. Наука. Технологии». 2018. №2. С. 37-42.
3. Борисов А.В. Современные решения и подходы к обработке массивов неструктурированной текстовой информации в области больших данных // Проблемы современной науки и образования. 2017. С. 49-52.
4. Петрова И.Ю., Горянин С.В. Информационно-аналитическая система EcoHealth для хранения и анализа структурированных и неструктурированных больших данных // Инженерно-строительный вестник Прикаспия: научно-технический журнал 2017. № 3 (21). С. 66–71.
5. Погодин Г.В., Фиго Д.М., Васильев Э. Н. Сериализации структур данных для хранения и передачи в информационных системах. Методы и средства // «Молодежь в науке». Сборник докладов 16-й научно-технической конференции. 2017. №2. С. 231-236.
6. CSV URL: https://ru.wikipedia.org/wiki/CSV (дата обращения: 11.12.2020).
7. Самойленко Н. Python для сетевых инженеров Выпуск 3.0. URL: https://pyneng.readthedocs.io/_/downloads/ru/latest/pdf/ (дата обращения: 12.12.2020)
8. Extensible Markup Language (XML) URL: https://www.w3.org/XML/ (дата обращения: 12.12.2020).
9. Канаев К.А., Фалеева Е.В., Пономарчук Ю.В. Сравнительный анализ форматов обмена данными, используемых в приложениях с клиент-серверной архитектурой // Фундаментальные исследования. – 2015. – № 2-25. – С. 5569-5572.
10. Пилигрим М. Погружение в Python 3. 2010.
11. Сучкова Е.А., Николаева Ю.В. Разработка оптимальной структуры хранения данных для систем поддержки принятия решений // Кибернетика и программирование. – 2016. № 4. С. 58-64.

12. Романов А.С. Модель базы данных для хранения текстов и их характеристик // Доклады Томского государственного университета систем управления и радиоэлектроники. 2008. №1. С. 70-73.

13. Шевелев О.Г. Представление набора текстов в реляционной базе данных для целей лингвистического анализа. 2004.

14. Довбенко А.В. Хранение данных в NoSQL системах на примере MongoDB. 2015.

15. Коротеев М.В., Коротеев К.М. Обзор некоторых современных тенденций в технологии машинного обучения // E-Management. 2018. С. 26-35.

16. Руйчева А.П. Развитие машинного обучения // Современные технологии в образовании: материалы международной научно-практической конференции. 2017. Ч. 1. С. 232-237.

17. Mison: A Fast JSON Parser for Data Analytics / Li, Yinan, Katsipoulakis N., Chandramouli, B., Goldstein J., Kossman D. 2017.

18. Савин И.В. Анализ систем хранения данных // Известия Тульского государственного университета Технические науки. 2019. С. 193-196.

19. Басов О.О., Саитов И.А. Основные каналы межличностной коммуникации и их проекция на инфокоммуникационные системы // Труды СПИИРАН. (7), С. 122–140.

20. Растет популярность Python открытые системы. М.: Открытые системы, 2019. С. 5-11.

21. Langdale G., Lemire D. Parsing gigabytes of JSON per second // The VLDB Journal: The International Journal on Very Large Data Bases. 2019. 28(6). pp. 941.

Reference

1. Information Extraction from Multistructured Data and its Transformation into a Target Schema / Briukhov D.O., Stupnikov S.A., Kalinichenko L.A., Vovchenko A.E. // Selected Papers of the XVII International Conference on Data Analytics and Management in Data Intensive Domains. 2015. pp. 81–90.

2. Bedarev N.V., Voinov A.A. Natural Language Texts and Methods of Structured Data Extraction // International scientific and technological conference of students and young scientists “Youth. The science. Technologies". 2018. No. 2. pp. 37-42.

3. Borisov A.V. Modern solutions and approaches to processing arrays of unstructured textual information in the field of big data // Problems of modern science and education. 2017.pp. 49-52.

4. Petrova I. Y., Goryanin S.V. Information and analytical system EcoHealth for storage and analysis of structured and unstructured big data // Engineering and construction bulletin of the Caspian Sea region: scientific and technical journal 2017. No. 3 (21). pp. 66–71.

5. Pogodin G.V., Figo D.M., Vasiliev E.N. Serialization of data structures for storage and transmission in information systems. Methods and means // "Youth in Science". Collection of reports of the 16th scientific and technical conference. 2017. No. 2. P. 231-236.

6. CSV URL: https://ru.wikipedia.org/wiki/CSV (date of the request: 11.12.2020).

7. Samoylenko N. Python for Network Engineers Release 3.0 URL: https://pyneng.readthedocs.io/_/downloads/ru/latest/pdf/ (date of the request: 12.12.2020).

8. Extensible Markup Language (XML) URL: https://www.w3.org/XML/ (date of the request: 12.12.2020).

9. Kanaev K.A., Faleeva E.V., Ponomarchuk Y.V. Comparative Analysis of Data Exchange Formats for Applications with Client-Server Architecture // Basic research. – 2015. – No. 2-25. – p. 5569-5572.

10. Pilgrim M. Dive into Python 3. 2010

11. Suchkova E.A., Nikolaeva Yu.V. Development of an optimal data storage structure for decision support systems // Cybernetics and programming. – 2016. No. 4. pp. 58-64.

12. Romanov A.C. Database model for storing texts and their characteristics // Reports of the Tomsk State University of Control Systems and Radioelectronics. 2008. No. 1. pp. 70-73.

13. Shevelev O.G. Representation of a set of texts in a relational database for the purpose of linguistic analysis. 2004.

14. Dovbenko A.V. Data storage in NoSQL systems on the example of MongoDB. 2015.

15. Koroteev M.V., Koroteev K. Review of Some Contemporary Trends in Machine Learning Technology // E-Management. 2018. pp. 26-35.

16. Ruycheva, A.P. Development of machine learning // Modern technologies in education: materials of an international scientific and practical conference. 2017. Part 1.pp. 232-237.

17. Mison: A Fast JSON Parser for Data Analytics / Li, Yinan, Katsipoulakis N., Chandramouli, B., Goldstein J., Kossman D. 2017.

18. Savin I.V. Analysis of data storage systems // Bulletin of the Tula State University Technical Sciences. 2019. pp. 193-196.

19. Basov O.O., Saitov I.A. Main channels of interpersonal communication and their projection onto infocommunication systems // Proceedings of SPIIRAS. (7), pp. 122–140.

20. The popularity of Python open-source systems is growing. Moscow: Open Systems, 2019.S. 5-11.

21. Langdale G., Lemire D. Parsing gigabytes of JSON per second // The VLDB Journal: The International Journal on Very Large Data Bases. 2019. 28(6). pp. 941.

Наумов Руслан Кириллович, инженер, студент, мегафакультет трансляционных информационных технологий
Железков Никита Эдуардович, инженер, студент, мегафакультет трансляционных информационных технологий

Naumov Ruslan Kirillovich, engineer, student, Faculty of Translational Information Technologies
Zhelezkov Nikita Eduardovich, engineer, student, Faculty of Translational Information Technologies