УДК 004.82 + 004.62

Медведев Арсений Эдуардович, магистрант.
Белгородский государственный национальный исследовательский университет
Medvedev Arsenii Eduardovich,
Belgorod National Research University

Ядута Анна Зауровна, к.т.н.,
Белгородский государственный национальный исследовательский университет
Yaduta Anna Zaurovna,
Belgorod National Research University

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТОКЕНИЗАЦИИ ДАННЫХ В ФОРМАТАХ TOKEN-ORIENTED OBJECT NOTATION (TOON) И JAVASCRIPT OBJECT NOTATION (JSON)
COMPARATIVE ANALYSIS OF DATA TOKENIZATION IN TOKEN-ORIENTED OBJECT NOTATION (TOON) AND JAVASCRIPT OBJECT NOTATION (JSON) FORMATS

Аннотация. В статье проводится сравнительный анализ токенизации данных в форматах Token-Oriented Object Notation (TOON) и JavaScript Object Notation (JSON) на различных токенизаторах. Сравнение проводится по критериям сохранения семантики и длины закодированной последовательности. Результаты подтверждают потенциал TOON как эффективного формата данных для больших языковых моделей (LLMs).

Abstract. The article presents a comparative analysis of data tokenization for the Token-Oriented Object Notation (TOON) and JavaScript Object Notation (JSON) formats using various tokenizers. The comparison is based on the criteria of semantic preservation and the length of the encoded sequence. The results confirm the potential of TOON as an efficient data format for Large Language Models (LLMs).

Ключевые слова: TOON, JSON, токенизация, LLM.

Keywords: TOON, JSON, tokenization, LLM.

Большие языковые модели (LLMs) всё чаще используются для обработки структурированных данных, однако традиционные форматы, такие, как JavaScript Object Notation (JSON) [1], неэффективны с точки зрения токенизации (процесса преобразования текста в числа для обработки моделью LLM), так как избыточные символы увеличивают длину входных последовательностей, за счёт чего возрастает стоимость вычислений, снижается пропускная способность контекстного окна и повышается вероятность искажения при генерации.

Token-Oriented Object Notation (TOON) – это альтернативный формат представления данных, оптимизированный для эффективной токенизации [2], который сохраняет семантическую совместимость с JSON, но обеспечивает бóльшую компактность за счёт упрощённого синтаксиса. Для оценки практической применимости данного формата мы проводим эксперименты на табличных данных (однородные списки записей с идентичной структурой) и иерархических данных, включающей несколько уровней вложенности и смешанные типы. В проведённом эксперименте мы использовали токенизаторы PreTrainedTokenizerFast, LlamaTokenizerFast и Qwen2TokenizerFast из библиотеки Transformers [3], а также «o200k_base» из библиотеки Tiktoken [4].

Основой для сравнительного анализа TOON и JSON является их семантическая равнозначность в контексте восприятия большими языковыми моделями. Сжатие данных алгоритмом TOON не приводит к потере или искажению информационной целостности.

В формате JSON семантическая связь «ключ — значение» является локальной и явной, так как каждый элемент данных v_i в массиве объектов сопровождается своим ключом k_i,образуя пару (k_i, v_i).Это обеспечивает высокую надежность и читаемость структуры, однако приводит к значительной синтаксической избыточности R_{syn}, что негативно сказывается на токенизации. Поскольку ключи повторяются в каждом из объектов массива, даже при кодировании каждого ключа одним токеном, их суммарный вклад в длину последовательности линейно возрастает. Общая синтаксическая избыточность может быть оценена как:

$$R_{syn} = N * \sum_{j=1}^{m} t(k_j), \quad (1)$$

где m – количество уникальных ключей в схеме объекта, $t(k_j)$ – число токенов, необходимых для кодирования ключа k_j, а N – количество объектов в массиве.

Фигурные скобки, двоеточия и запятые увеличивают информационный шум: например, фрагмент «: » (двоеточие с последующим пробелом) может разбиваться на два токена в зависимости от структуры словаря токенизатора, а числовые и булевы литералы нередко фрагментируются из-за близости к синтаксическим разделителям.

Формат TOON устраняет эти недостатки, реализуя подход разделения схемы и данных (schema-payload separation). Семантические ключи выносятся в глобальный заголовок, а данные преобразуются в упорядоченные кортежи значений, что повышает вероятность их целостной токенизации. Эквивалентность форматов достигается за счет позиционного сопоставления. Для любой записи j и поля i значение $V_{j,i}$ интерпретируется моделью как принадлежащее ключу K_i на основании равенства их индексов относительно разделителей. Формально, семантическое соответствие задаётся как:

$$\forall j \in [1, N], \forall i \in [1, m]: \; S(V_{j,i}) = K_i \leftrightarrow p(V_{j,i}) = p(K_i), \quad (2)$$

где $S(x)$ – схема отображения индекса в ключ, $p(x)$ – позиция элемента x в последовательности, индексированной относительно фиксированного набора разделителей, а m и N – количество полей в схеме и записей соответственно.

На Рисунке 1 показаны данные в форматах JSON и TOON, поочерёдно имеющие табличную, иерархическую и смешанную структуры.

JSON:

```
{
  "users": [
    { "id": 1, "name": "Alice" },
    { "id": 2, "name": "Bob" }
  ]
}
```

TOON:

```
users[2]{id,name}:
  1,Alice
  2,Bob
```

```
{
  "user": {
    "name": "Alice",
    "role": "admin"
  }
}
```

```
user:
  name: Alice
  role: admin
```

```
{
  "total": 3,
  "active": true,
  "items": ["a", "b", "c"]
}
```

```
total: 3
active: true
items[3]: a,b,c
```

Рисунок 1. Семантическая эквивалентность данных в форматах JSON и TOON

С точки зрения архитектуры трансформеров, переход от JSON к TOON изменяет правило работы механизма внимания (self-attention). В случае JSON модель опирается на "ближний контекст" (локальное соседство ключа и значения). В случае TOON модель задействует механизм дальнодействующих зависимостей (long-range dependencies), удерживая во внимании заголовок схемы при генерации или анализе каждого последующего токена данных.

Таким образом, спецификация формата указывает на то, что TOON сохраняет строгую типизацию, используя синтаксические маркеры для примитивов, аналогичные JSON (булевы значения, числа, null). То есть, можно утверждать, что эти форматы являются взаимозаменяемыми для однородных структур данных, что позволяет использовать метрику коэффициента сжатия как объективный показатель эффективности.

Для оценки практической эффективности формата TOON относительно JSON в условиях реального использования был проведён эксперимент с использованием четырех различных токенизаторов:

- PreTrainedTokenizerFast (используется в моделях GigaChat);
- Qwen2TokenizerFast (используется в моделях Qwen);
- LlamaTokenizerFast (используется в моделях YandexGPT);
- o200k_base (используется в моделях ChatGPT).

В качестве тестовой информации использовались два типа синтетических данных, отражающих практические сценарии использования ИИ-систем:

- Табличная структура в виде однородного массива из 5 объектов карточек товара, содержащих идентичный набор полей: id, name, category, price, in_stock (true или false), created_at;
- Иерархическая структура в виде смешанного объекта конфигурации Android-приложения, характеризующегося глубокой вложенностью и неоднородностью полей.

Результаты измерения количества закодированных токенов для форматов JSON и TOON представлены на Рисунке 2.

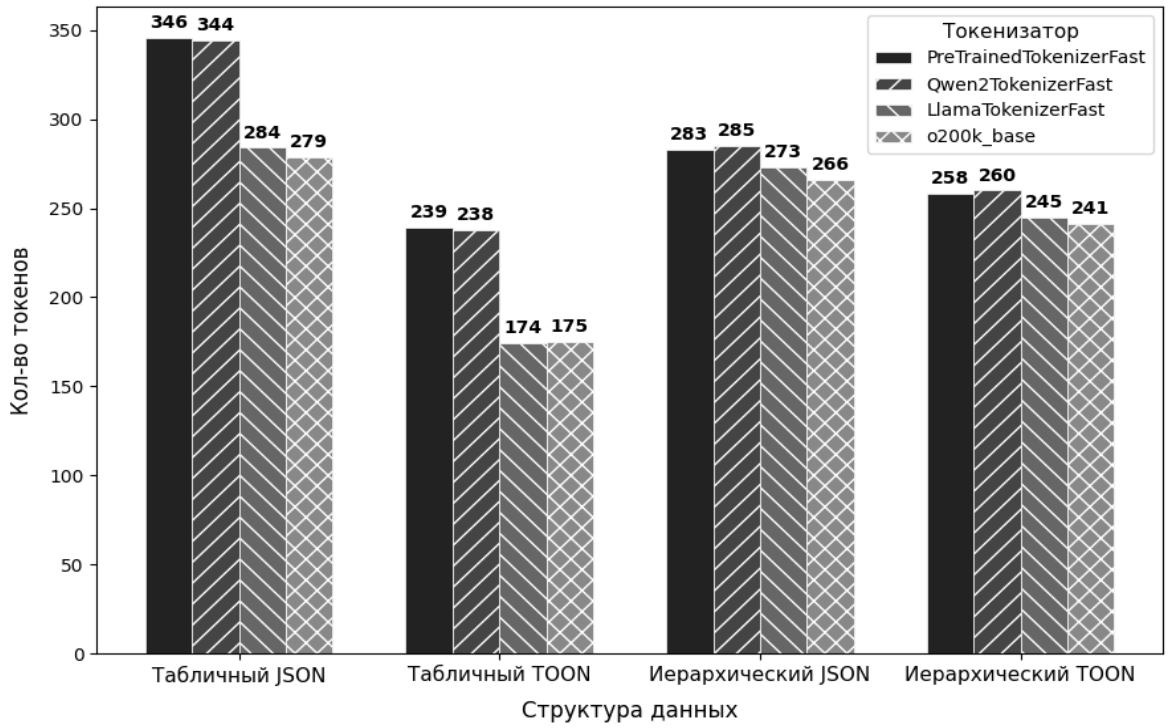

Рисунок 2. Результаты токенизации данных в форматах JSON и TOON

Анализ графика выявляет чёткую зависимость эффективности сжатия от структуры исходных данных. Наибольшее сокращение объёма контекста достигается при обработке табличных массивов, так как в этом случае у формата TOON вынесены семантические ключи в глобальный заголовок, и их повторение в теле данных отсутствует. Для высокоэффективных токенизаторов (Llama, o200k_base) объём токенов снижается с 279-284 до 174-175, а для менее оптимизированных – с 344-346 до 238-239. Таким образом, коэффициент сжатия составляет 30-40%, что подтверждает целесообразность применения TOON в сценариях с однородными данными.

В случае иерархических структур преимущество TOON существенно снижается. Среднее сокращение объёма составляет лишь 9–10 %: например, для Llama – с 273 до 245 токенов, а для Qwen – с 285 до 260. Такая деградация обусловлена необходимостью частого локального переопределения схемы внутри вложенных объектов, что по затратам приближается к явному дублированию ключей. Тем не менее, даже в этом наименее благоприятном сценарии TOON сохраняет небольшое, но устойчивое преимущество в компактности и не приводит к ухудшению показателей по сравнению с JSON.

Сравнительный анализ самих токенизаторов также позволяет сделать вывод об устойчивости TOON. Несмотря на то, что показатели Llama и o200k_base превосходят результаты Qwen и PreTrained по плотности упаковки информации, относительное сжатие TOON по отношению к JSON сохраняется, то есть оно является устойчивой характеристикой, не зависящей от реализации токенизатора или размера словаря конкретной языковой модели.

Сравнительный анализ показал, что Token-Oriented Object Notation обеспечивает устойчивое сокращение объёма токенизации при полном сохранении семантической целостности данных. На табличных структурах экономия достигает в среднем 35%, что делает формат особенно эффективным для однородных коллекций – логов, справочников, результатов запросов и т.д. Результат воспроизводится независимо от токенизатора, подтверждая кросс-архитектурную применимость подхода.

Для глубоко вложенных иерархических структур выигрыш снижается в среднем до 9% из-за накладных расходов на локальное переопределение схем, однако TOON остаётся строго компактнее JSON даже в худших случаях.

Таким образом, TOON представляет собой практически значимое решение для снижения вычислительных и финансовых затрат в LLM-ориентированных системах – в особенности в Retrieval-Augmented Generation (RAG) [5], агентных архитектурах и генерации структурированного вывода. В условиях роста сложности моделей и жёстких ограничений на контекстное окно, проектирование форматов с приоритетом на эффективную токенизацию становится важной задачей в современных системах искусственного интеллекта.

Список литературы:

1. Bray T. The JavaScript Object Notation (JSON) Data Interchange Format: RFC 8259. – Fremont: IETF, 2017. – 18 с. – URL: https://datatracker.ietf.org/doc/html/rfc8259 (дата обращения: 25.11.2025).

2. Token-Oriented Object Notation (TOON) – Compact, human-readable, schema-aware JSON for LLM prompts. – GitHub Repository, 2025. – URL: https://github.com/toon-format/toon (дата обращения: 25.11.2025).

3. Wolf T., Debut L., Sanh V. et al. Transformers: State-of-the-Art Natural Language Processing // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online. Association for Computational Linguistics, 2020. – P. 38–45. – DOI: 10.18653/v1/2020.emnlp-demos.6.

4. OpenAI. Tiktoken is a fast BPE tokeniser for use with OpenAI's models. – GitHub Repository. – 2025. – URL: https://github.com/openai/tiktoken (дата обращения: 25.11.2025).

5. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // arXiv preprint arXiv:2005.11401. – 2020. – 19 p. – URL: https://arxiv.org/abs/2005.11401 (дата обращения: 25.11.2025).