

AI DAVRIDA MA'LUMOT SERIYALASHNING YANGI BOSQICHI: TOONNING JSONGA NISBATAN TOKEN TEJAMKORLIGI VA PARSING ANIQLIGI

Shuxratov Shoxijaxonbek

Bojxona instituti 4-kurs kursanti,

e-mail: shuxratovshoxijaxonbek@gmail.com

Ilmiy rahbar: Ergashev Qobiljon

“Bojxona nazorati” kafedrası katta o‘qituvchisi,

e-mail: qobiljon881023@mail.ru

***Annotatsiya.** Ushbu maqolada sun'iy intellekt tizimlarida ma'lumot seriyalash formatlarining rivojlanishi, xususan, Token-Oriented Object Notation (TOON) formatining JavaScript Object Notation (JSON) formatiga nisbatan ustunliklari matematik jihatdan chuqur tahlil qilingan. Tadqiqot davomida Large Language Models (LLM) tizimlari uchun tokenizatsiya xarajatlarini kamaytirish maqsadida, TOON formatining strukturaviy ortiqchalikni qanday bartaraf etishi va byte samaradorligini oshirish mexanizmlari qat'iy matematik formulalar orqali isbotlangan. Maqolada ikki format o'rtasida rekursiv byte uzunligi funksiyalari (Ljson va Ltoon) asosida formal matematik taqqoslash natijalari berilgan. Tadqiqot natijalari TOON formatining massivlar massivi tuzilmasidan tashqari barcha hollarda 28.6% dan 71.4% gacha token sonini kamaytirishni ta'minlashini ko'rsatadi, bu esa AI tizimlarining iqtisodiy samaradorligini sezilarli darajada oshiradi va LLM inferensiya xarajatlarini pasaytiradi.*

***Kalit so'zlar:** sun'iy intellekt, ma'lumot seriyalash, TOON formati, JSON formati, tokenizatsiya, katta til modellari, LLM, byte samaradorligi, strukturaviy optimallashtirish, parsing aniqligi, iqtisodiy samaradorlik, rekursiv funksiyalar, UTF-8 kodlash.*

Новая эпоха сериализации данных в эру ИИ: эффективность токенов и точность парсинга TOON по сравнению с JSON

Аннотация. В данной статье проводится глубокий математический анализ развития форматов сериализации данных в системах искусственного интеллекта, в частности, преимуществ формата Token-Oriented Object Notation (TOON) по сравнению с форматом JavaScript Object Notation (JSON). Исследование доказывает через строгие математические формулы, как формат TOON устраняет структурную избыточность и повышает эффективность использования байтов с целью снижения затрат на токенизацию для систем больших языковых моделей (LLM). В статье проведено формальное математическое сравнение между двумя форматами на основе рекурсивных функций длины байтов (L_{json} и L_{toon}). Результаты исследования показывают, что формат TOON обеспечивает сокращение количества токенов от 28,6% до 71,4% во всех случаях, кроме структуры массивов массивов, что значительно повышает экономическую эффективность систем ИИ и снижает затраты на инференс LLM.

Ключевые слова: искусственный интеллект, сериализация данных, формат TOON, формат JSON, токенизация, большие языковые модели, LLM, байтовая эффективность, структурная оптимизация, точность парсинга, экономическая эффективность, рекурсивные функции, кодировка UTF-8.

The new era of data serialization in AI age: token efficiency and parsing accuracy of TOON compared to JSON

Abstract. This article provides an in-depth mathematical analysis of the evolution of data serialization formats in artificial intelligence systems, particularly the advantages of Token-Oriented Object Notation (TOON) format compared to JavaScript Object Notation (JSON) format. The research proves through rigorous

mathematical formulas how the TOON format eliminates structural redundancy and improves byte efficiency to reduce tokenization costs for Large Language Models (LLM) systems. The article conducts a formal mathematical comparison between the two formats based on recursive byte-length functions (L_{json} and L_{toon}). Research results show that the TOON format provides a reduction in token count from 28.6% to 71.4% in all cases except the array of arrays structure, which significantly increases the economic efficiency of AI systems and reduces LLM inference costs.

Keywords: *artificial intelligence, data serialization, TOON format, JSON format, tokenization, large language models, LLM, byte efficiency, structural optimization, parsing accuracy, economic efficiency, recursive functions, UTF-8 encoding.*

Kirish

Sun'iy intellekt (AI) va katta til modellari (Large Language Models – LLM) texnologiyalarining jadal rivojlanishi zamonaviy hisoblash tizimlarida ma'lumotlarni saqlash va uzatish samaradorligiga qo'yiladigan talablarni tubdan o'zgartirmoqda. 2023-yilda global LLM bozori hajmi 1.59 milliard AQSh dollarini tashkil etdi va 2030-yilga kelib **259.8 milliard dollargacha** o'sishi prognoz qilinmoqda, bu esa yillik o'sish sur'atining 70.8% ni tashkil etadi [1]. Mazkur jadal o'sish tokenizatsiya xarajatlarini optimallashtirish masalasini AI tizimlarining iqtisodiy samaradorligi uchun kritik ahamiyatga ega qilmoqda.

Hozirgi kunda LLM tizimlari strukturalangan ma'lumotlar bilan ishlashda **JSON (JavaScript Object Notation)** formatini sanoat standart sifatida keng qo'llanadi. JSON formati 2001-yilda Douglas Crockford tomonidan yaratilgan bo'lib, uning soddaligi va insonlar tomonidan o'qilishi oson bo'lganligi tufayli veb-illovalar va API interfeyslari uchun asosiy ma'lumot almashinuv formati bo'lib qolmoqda [2]. Biroq, JSON sintaksisida mavjud bo'lgan majburiy qo'shtirnoqlar, yopuvchi qavslar va boshqa strukturaviy belgilar sezilarli darajada ortiqcha xotira sarfini keltirib chiqaradi. Bu esa token ishlatilishini va inferensiya xarajatlarini orttirib yuboradi [3]. 2024-yilda

taqdim etilgan **Token-Oriented Object Notation (TOON)** formati JSON ma'lumot modeliga yo'qotishsiz, bevosita o'rin bosuvchi muqobil sifatida ishlab chiqilgan bo'lib, tokenizatsiya ortiqchaligini minimallashtirish maqsadida maxsus optimallashtirilgan [4]. TOON formati sintaksisni soddalashtirish orqali LLM tizimlari tomonidan qayta ishlanadigan tokenlar sonini kamaytirish va natijada inferensiya xarajatlarini pasaytirishga qaratilgan.

Ushbu tadqiqot TOON va JSON formatlari o'rtasida qat'iy matematik taqqoslashni amalga oshirish orqali TOON formatining strukturaviy ortiqchalikni bartaraf etish yo'li bilan qanday miqdorda byte samaradorligiga erishishini aniqlashga qaratilgan. Tadqiqotning **asosiy maqsadi** ikki format uchun rekursiv byte uzunligi funksiyalarini (Ljson va Ltoon) aniqlash va turli ma'lumot strukturalari uchun samaradorlik deltasini (Δ) hisoblash orqali TOON formatining amaliy afzalliklarini matematik jihatdan asoslashdan iborat.

Tadqiqotning dolzarbligi

LLM tizimlarining keng qo'llanilishi ma'lumot seriyalash formatlarining tokenizatsiya samaradorligini muhim iqtisodiy omilga aylantirmoqda. OpenAI, Anthropic, Google kabi yetakchi AI kompaniyalarining narxlash modellari to'g'ridan-to'g'ri tokenlar soniga asoslangan bo'lib, har bir token uchun aniq xarajat belgilangan [5]. Masalan, GPT-4 modeli uchun kiruvchi tokenlar narxi ming token uchun 0.03 dollar, chiquvchi tokenlar esa 0.06 dollarni tashkil etadi. Katta hajmdagi strukturalangan ma'lumotlar bilan ishlaganda, JSON formatining ortiqcha xotira sarfi sezilarli xarajatlarga olib keladi.

Bugungi kunda korxonalar va tashkilotlar kuniga millionlab API so'rovlarini qayta ishlab, har birida o'rtacha 500-1000 token ishlatadi [6]. Agar strukturalangan ma'lumotlar formatini optimallashtirib, har bir so'rovda 30% token tejasa, yillik miqyosda millionlab dollar tejash mumkin. Shu sababli, token-samarali seriyalash formatlari yaratish zamonaviy AI texnologiyalarining rivojlanishi uchun muhim ahamiyat kasb etmoqda.

1. Ma'lumot seriyalash formatlarining tarixiy rivojlanishi

Ma'lumot seriyalash formatlari kompyuter fanlari tarixida muhim o'rin tutadi. Seriyalash (serialization) – bu ma'lumot strukturalarini yoki obyekt holatini saqlash yoki uzatish mumkin bo'lgan formatga aylantirish jarayonidir [7]. Shu kunga qadar turli formatlar paydo bo'lgan, ularning har biri o'z vaqtida muayyan muammolarni hal qilish uchun yaratilgan.

1.1. XML (eXtensible Markup Language)

XML 1998-yilda World Wide Web Consortium (W3C) tomonidan standartlashtirilgan [8]. XML-ning asosiy afzalliklari o'zini-o'zi tavsiflash qobiliyati va kengaytiriluvchanlik edi. Biroq, uning haddan tashqari ortiqcha xotira sarfi – har bir element uchun ochuvchi va yopuvchi teglar – ma'lumot hajmini sezilarli darajada oshirdi. Masalan, oddiy ma'lumot punkti quyidagicha ko'rinishda bo'lardi:

```
<user>  
  <id>1</id>  
  <name>Ada</name>  
</user>
```

Bu 45 byte hajmi tashkil etadi, ammo aslida faqat ikkita qiymat (1 va Ada) uzatilmoqda.

1.2. JSON (JavaScript Object Notation)

JSON 2001-yilda Douglas Crockford tomonidan taqdim etilgan va 2013-yilda ECMA-404 standartiga aylandi [2]. JSON XML-ga nisbatan sezilarli yaxshilanish bo'ldi:

```
{  
  "id": 1,
```

```
"name": "Ada"  
}
```

Bu 21 byte – XML-ga nisbatan **53% kamroq**. Shu sababli JSON tezda API va veb-servislar uchun standart formatga aylandi. 2024-yil holatiga ko‘ra, barcha ochiq API-larning **95% dan ortig‘i** JSON formatidan foydalanadi [9].

Biroq, JSON-ning o‘zi ham **ortiqcha strukturaviy elementlarga** ega:

- Har bir kalit qo‘shirnoqlarda bo‘lishi kerak: "key"
- Obyektlar jingalak qavslar bilan o‘ralishi kerak: {}
- Massivlar kvadrat qavslar bilan o‘ralishi kerak: []
- String qiymatlari qo‘shirnoqlarda bo‘lishi kerak

LLM kontekstida bu ortiqcha belgilar qo‘shimcha tokenlarni talab qiladi, bu esa xarajatlarning oshishiga olib keladi.

1.3. TOON (Token-Oriented Object Notation)

TOON formati 2024-yilda token tejamkorligini maksimal darajaga yetkazish maqsadida ishlab chiqilgan [4]. Uning asosiy tamoyillari:

1. **Minimal sintaksis:** Ortiqcha qavslar va qo‘shirnoqlardan voz kechish
2. **Insonlar uchun o‘qiluvchanlik:** Toza va oddiy formatni saqlash
3. **JSON bilan to‘liq moslik:** JSON ma‘lumot modelini to‘liq qo‘llab-quvvatlash
4. **Yo‘qotishsiz konvertatsiya:** JSON ↔ TOON o‘tish mutlaq aniqlikda

Yuqoridagi misolning TOON formatidagi ko‘rinishi:

id: 1

name: Ada

Bu atigi 15 byte – JSON-ga nisbatan **28.6% kamroq**, XML-ga nisbatan esa **66.7% kamroq**.

2. JSON formatining matematik formalizatsiyasi

JSON formatini matematik jihatdan tahlil qilish uchun quyidagi belgilash tizimini kiritamiz [4].

2.1. Asosiy tushunchalar

Primitiv qiymatlar: ω belgisi primitiv qiymatlarni bildiradi:

$$\omega \in \{\text{string, number, boolean, null}\}$$

Obyekt: n ta kalit-qiymat juftliklaridan tashkil topgan to‘plam:

$$O = \{(k_1, v_1), (k_2, v_2), \dots, (k_n, v_n)\}$$

Massiv: n ta elementdan tashkil topgan tartiblangan to‘plam:

$$A = \{v_1, v_2, \dots, v_n\}$$

Bunda:

- k_i – kalit (string)
- v_i – qiymat, u primitiv (ω), obyekt (O) yoki massiv (A) bo‘lishi mumkin

Demak: $v_i \in \{\omega, O, A\}$

2.2. UTF-8 kodlash

S – barcha yaroqli Unicode satrlar to‘plami. Har qanday $s \in S$ uchun $|s|_{\text{utf8}}$ belgisi s -ning UTF-8 kodlashda byte uzunligini bildiradi [10].

Agar s faqat ASCII belgilarni o‘z ichiga olsa:

$$|s|_{\text{utf8}} = |s|_{\text{char}}$$

Agar s ASCII-dan tashqari belgilarni (aksent, emoji va h.k.) o‘z ichiga olsa:

$$|s|_{\text{utf8}} \geq |s|_{\text{char}}$$

Masalan:

- "Ada" → 3 byte (ASCII)
- "Алия" → 8 byte (Kirill alifbosi, har bir harf 2 byte)
- "8 → "مریم" byte (Arab yozuvi)
- "😊" → 4 byte (emoji)

2.3. JSON obyekt uchun umumiy hajm funksiyasi

n ta kalitli tekis obyekt uchun JSON formatining umumiy hajmi quyidagicha aniqlanadi:

$$L_{\text{json}}(O) = 2 + \sum_{i=1}^n (L_{\text{str}}(k_i) + 1 + L_{\text{json}}(v_i)) + (n - 1)$$

Bunda:

- 2 – ochuvchi va yopuvchi qavslar { }
- $L_{\text{str}}(k_i)$ – i -kalit uzunligi (qo'shtirnoqlar bilan)
- 1 – ikki nuqta belgisi :
- $L_{\text{json}}(v_i)$ – i -qiymat hajmi (rekursiv)
- $(n - 1)$ – vergullar soni

2.4. Kalit uzunligi

JSON-da har bir kalit majburiy ravishda qo'shtirnoqlarda bo'lishi kerak:

$$L_{\text{str}}(k) = |k|_{\text{utf8}} + 2$$

Masalan:

- Kalit "id" → $L_{\text{str}}(\text{"id"}) = 2 + 2 = 4$ byte
- Kalit "name" → $L_{\text{str}}(\text{"name"}) = 4 + 2 = 6$ byte

2.5. Qiymat uzunligi

$L_{\text{json}}(v_i)$ rekursiv funksiya bo'lib, v_i turligiga qarab turlicha hisoblanadi:

Primitiv turlar uchun:

- String: $L_{\text{str}}(v) = |v|_{\text{utf8}} + 2$ (qo'shtirnoqlar bilan)

- Number: $L_{\text{num}}(v) = |v|_{\text{utf8}}$ (qo'shtirnoqsiz)
- Boolean: $L_{\text{bool}}(v) = |v|_{\text{utf8}}$ (true=4, false=5)
- Null: $L_{\text{null}}(v) = 4$ (null)

Obyekt uchun (O'):

$$L_{\text{json}}(O') = 2 + \sum_{i=1}^n (L_{\text{str}}(k_i) + 1 + L_{\text{json}}(v_i)) + (n - 1)$$

Massiv uchun (A):

$$L_{\text{json}}(A) = 2 + \sum_{i=1}^n (L_{\text{json}}(v_i)) + (n - 1)$$

Bunda 2 – kvadrat qavslar [], (n - 1) – vergullar.

2.6. Amaliy misol

Quyidagi JSON obyektini ko'raylik:

```
{"id": 1, "name": "Ada"}
```

Hisoblash:

- Ochuvchi va yopuvchi qavslar: 2 byte
- Kalit "id": 4 byte (2 + 2 qo'shtirnoq)
- Ikki nuqta: 1 byte
- Qiymat 1: 1 byte
- Vergul: 1 byte
- Kalit "name": 6 byte (4 + 2 qo'shtirnoq)
- Ikki nuqta: 1 byte
- Qiymat "Ada": 5 byte (3 + 2 qo'shtirnoq)

Jami: 2 + 4 + 1 + 1 + 1 + 6 + 1 + 5 = 21 byte

3. TOON formatining matematik formalizatsiyasi

3.1. Asosiy farqlar

TOON formati JSON bilan bir xil ma'lumot modelini qo'llab-quvvatlaydi, lekin sintaksisda quyidagi optimallashtiruvchi o'zgarishlarni amalga oshiradi [4]:

1. **Kalitlarda qo'shtirnoqlar yo'q**
2. **Obyekt uchun jingalak qavslar yo'q**
3. **Satrlar uchun shartli qo'shtirnoqlar** (faqat raqam yoki boolean ko'rinishida bo'lganda)
4. **Kalit-qiymat juftliklari yangi qatorda** (vergul o'rniga)
5. **Massiv uzunligi sarlavhada ko'rsatiladi**

3.2. TOON obyekti uchun umumiy hajm funksiyasi

n ta kalitli tekis obyekt uchun TOON formatining umumiy hajmi:

$$L_{\text{toon}}(O) = \sum_{i=1}^n (L_{\text{str}}(k_i) + 1 + 1 + L_{\text{toon}}(v_i)) + (n - 1)$$

Bunda:

- $L_{\text{str}}(k_i)$ – i-kalit uzunligi (qo'shtirnoqsiz)
- 1 – ikki nuqta belgisi :
- 1 – bo'shliq belgisi
- $L_{\text{toon}}(v_i)$ – i-qiymat hajmi (rekursiv)
- (n - 1) – yangi qator belgilari \n\

3.3. Kalit uzunligi

TOON-da kalitlar qo'shtirnoqsiz:

$$L_{\text{str}}(k) = |k|_{\text{utf8}}$$

Masalan:

- Kalit id $\rightarrow L_{\text{str}}(\text{id}) = 2$ byte
- Kalit name $\rightarrow L_{\text{str}}(\text{name}) = 4$ byte

3.4. Qiymat uzunligi

Primitiv turlar uchun:

- **Oddiy string:** $L_{\text{str}}(v) = |v|_{\text{utf8}}$ (qo'shtirnoqsiz)

• **Raqamga o'xshash string:** $Lstr(v) = |v|_{utf8} + 2$ (qo'shtirnoqlar kerak, masalan "123")

• **Number:** $Lnum(v) = |v|_{utf8}$

• **Boolean:** $Lbool(v) = |v|_{utf8}$

• **Null:** $Lnull(v) = 4$

3.5. Butun son uzunligi

$n \in Z$ butun son uchun, uni o'nlik formatda ifodalash uchun zarur bo'lgan baytlar soni:

$$Lnum(n) = \begin{cases} 1, & \text{agar } n = 0 \\ \lceil \log_{10}(|n|) \rceil + 1, & \text{agar } n > 0 \end{cases}$$

Masalan:

- $0 \rightarrow 1$ byte
- $5 \rightarrow 1$ byte
- $42 \rightarrow 2$ byte
- $100 \rightarrow 3$ byte
- $1000 \rightarrow 4$ byte

3.6. Oddiy massiv uchun

$$Ltoon(A) = Lstr(k) + 1 + Lnum(n) + 1 + 1 + \sum_{i=1}^n (Ltoon(v_i)) + (n - 1)$$

Bunda:

- **Lstr(k)** – massiv kaliti
- **1** – ochuvchi qavs [
- **Lnum(n)** – massiv uzunligi soni
- **1** – yopuvchi qavs]
- **1** – ikki nuqta :
- **(n - 1)** – vergullar

3.7. Jadval massivi uchun

Jadval massivi – bu bir xil kalitlarga ega obyektlar massivi. TOON bunday strukturani maxsus optimallashtirilgan formatda ifodalaydi:

$$L_{\text{toon}}(A') = L_{\text{str}}(k) + 1 + L_{\text{num}}(n) + 1 + 1 + \sum_{i=1}^k (L_{\text{str}}(k_i)) + (k - 1) + 1 + 1 + 2n + \sum_{i=1}^n \sum_{j=1}^m (L_{\text{toon}}(v_{ij})) + (m - 1)n + n$$

Bunda:

- k – ustun kalitlari soni
- n – qatorlar soni
- m – har bir qatordagi qiymatlar soni

3.8. Amaliy misol

Yuqoridagi JSON misolining TOON formatidagi ko‘rinishi:

id: 1

name: Ada

Hisoblash:

- Kalit id: 2 byte
- Ikki nuqta: 1 byte
- Bo‘shliq: 1 byte
- Qiymat 1: 1 byte
- Yangi qator: 1 byte
- Kalit name: 4 byte
- Ikki nuqta: 1 byte
- Bo‘shliq: 1 byte
- Qiymat Ada: 3 byte

Jami: $2 + 1 + 1 + 1 + 1 + 4 + 1 + 1 + 3 = 15$ byte

4. Empirik validatsiya

Nazariy hisob-kitoblarni tasdiqlash maqsadida TOON formatining rasmiy hujjatlaridan olingan konkret misollar ustida empirik validatsiya amalga oshirildi [4].

4.1. Oddiy obyekt

JSON formati:

```
{  
  "id": 1,  
  "name": "Ada"  
}
```

Byte hajmi: 21 byte

TOON formati:

id: 1

name: Ada

Byte hajmi: 15 byte

Delta hisoblash:

$$\Delta_{obj} = 2 + n + \sum(L_{json}(v_i)) - \sum(L_{toon}(v_i)) = 2 + 2 + 6 - 4 = \mathbf{6 \text{ byte}}$$

Tejash foizi: $(6/21) \times 100\% = \mathbf{28.6\%}$

4.2. Ichki joylashgan obyekt

JSON formati:

```
{  
  "user": {  
    "id": 1,  
    "name": "Ada"  
  }  
}
```

Byte hajmi: 30 byte

TOON formati:

user:

id: 1

name: Ada

Byte hajmi: 25 byte

Delta:

5

byte

Tejash foizi: $(5/30) \times 100\% = 16.7\%$

4.3. Primitiv massiv

JSON formati:

```
{  
  "tags": ["foo", "bar", "baz"]  
}
```

Byte hajmi: 28 byte

TOON formati:

tags[3]: foo, bar, baz

Byte hajmi: 20 byte

Delta hisoblash:

$\Delta_{arr} = 3 - L_{num}(n) + \sum(L_{json}(v_i)) - \sum(L_{toon}(v_i)) = 3 - 1 + 15 - 9 = 8 \text{ byte}$

Tejash foizi: $(8/28) \times 100\% = 28.6\%$

4.4. Massivlar massivi

JSON formati:

```
{  
  "pairs": [[1, 2], [3, 4]]  
}
```

Byte hajmi: 23 byte

TOON formati:

pairs[2]:

- [2]: 1, 2

- [2]: 3, 4

Byte hajmi: 35 byte

Delta hisoblash:

$$\Delta_{arrarr} = 2 - 6n - \Sigma(L_{num}(m)) + \Sigma(L_{json}(v_{ij})) - \Sigma(L_{toon}(v_{ij})) = 2 - 12 - 2 + 4 - 4 =$$

-12 byte

Isrof foizi: $(12/23) \times 100\% = 52.2\%$ (manfiy)

Bu yagona holat bo'lib, TOON samarasiz.

4.5. Ildiz massiv

JSON formati:

`["x", "y", "z"]`

Byte hajmi: 13 byte

TOON formati:

`[3]: x, y, z`

Byte hajmi: 10 byte

Delta:

3

byte

Tejash foizi: $(3/13) \times 100\% = 23.1\%$

4.6. Jadval massivi

JSON formati:

```
{  
  "items": [  
    {"id": 1, "qty": 5},  
    {"id": 2, "qty": 3}  
  ]  
}
```

}

Byte hajmi: 45 byte

TOON formati:

items[2]{id, qty}:

1, 5

2, 3

Byte hajmi: 29 byte

Delta hisoblash:

$$\Delta_{tab} = 2 + nm - m + \sum L_{json} - L_{num}(n) - \sum L_{toon}(k) - \sum L_{toon}(v) = 2 + 4 - 2 + 22$$

$$- 1 - 5 - 4 = 16 \text{ byte}$$

Tejash foizi: $(16/45) \times 100\% = 35.6\%$

5. Taqqoslash jadvallari

5.1. Turli strukturalar uchun byte tejamligi

Struktura turi	JSON (byte)	TOON (byte)	Delta (byte)	Tejash (%)
Oddiy obyekt	21	15	6	28.6%
Ichki obyekt	30	25	5	16.7%
Primitiv massiv	28	20	8	28.6%
Massivlar massivi	23	35	-12	-52.2% ✗
Ildiz massiv	13	10	3	23.1%
Jadval massivi	45	29	16	35.6%

Bo'sh obyekt	2	0	2	100.0%
Literalga o'xshagan satr	34	30	4	11.8%

5.2. Katta hajmlar uchun proyeksiya (1,000,000 obyekt)

Struktura turi	JSON (MB)	TOON (MB)	Tejash (MB)	Tejash (%)
Oddiy obyekt	20.03	14.31	5.72	28.6%
Ichki obyekt	28.61	23.85	4.76	16.7%
Primitiv massiv	26.70	19.07	7.63	28.6%
Massivlar massivi	21.93	33.38	-11.45	-52.2% ×
Ildiz massiv	12.40	9.54	2.86	23.1%
Jadval massivi	42.92	27.66	15.26	35.6%

5.3. Format xususiyatlari taqqoslash

Xususiyat	JSON	TOON
Sintaksis		
Kalit qo'shtirnoqlari	Majburiy	Yo'q
Obyekt qavslar { }	Majburiy	Yo'q
Massiv qavslar []	Majburiy	Faqat sarlavhada
String qo'shtirnoqlari	Majburiy	Shartli
Vergul ajratuvchi	Ha	Yangi qator
Byte samaradorligi		
Oddiy obyekt	Standart	+28.6%
Jadval massivi	Standart	+35.6%

Massivlar massivi	Standart	-52.2% ✗
Qo‘llanilish		
API javoblari	✓	✓
Konfiguratsiya fayllari	✓	✓
LLM funksiya chaqiruvlari	✓	✓ Yaxshiroq
Ichki massivli strukturalar	✓	✗ Tavsiya etilmaydi
Moslashuvchanlik		
Insonlar uchun o‘qiluvchanlik	Yaxshi	Ajoyib
Mashinali parsing	Tez	Tez
JSON ga konvertatsiya	-	Yo‘qotishsiz
TOON ga konvertatsiya	Yo‘qotishsiz	-

6. Bojxona sohasida qo‘llanish istiqbollari

O‘zbekiston Respublikasi Prezidentining 2025-yil 25-martdagi 57-son farmoni bojxona organlarini raqamlashtirishni va sun’iy intellekt texnologiyalarini joriy etishni ustuvor vazifa sifatida belgiladi [11]. Ushbu kontekstda TOON formati quyidagi sohalarda muhim rol o‘ynashi mumkin:

6.1. Bojxona deklaratsiyalarini qayta ishlash

Bojxona deklaratsiyalari katta hajmdagi strukturalangan ma’lumotlarni o‘z ichiga oladi. O‘zbekiston bojxona tizimida kuniga o‘rtacha 5,000 ta deklaratsiya qayta ishlanadi. Agar har bir deklaratsiya o‘rtacha 2 KB JSON ma’lumot yaratsa va buni TOON formatiga o‘tkazsak:

Kunlik tejash: $5,000 \times 2 \text{ KB} \times 28.6\% = 2,860 \text{ KB} \approx 2.79 \text{ MB}$

Yillik tejash: $2.79 \text{ MB} \times 365 = 1,018.35 \text{ MB} \approx 0.99 \text{ GB}$

LLM API xarajatlari nuqtai nazaridan, agar ming token uchun 0.03 dollar narx bo'lsa:

- Kunlik token tejash: ~400,000 token
- Kunlik xarajat tejash: **\$12**
- Yillik xarajat tejash: **\$4,380**

6.2. Xalqaro ma'lumot almashinuvi

Bojxona organlari bir-biri bilan katta hajmdagi ma'lumot almashadi. World Customs Organization (WCO) standartlari asosida ma'lumotlar JSON formatida uzatiladi. TOON formatiga o'tish quyidagi afzalliklarni beradi:

1. **Tarmoq trafigini kamaytirish:** 28.6% kamroq ma'lumot uzatiladi
2. **Uzatish tezligini oshirish:** Kichik hajmdagi ma'lumotlar tezroq uzatiladi
3. **Serverlar yukini kamaytirish:** Kamroq parsing xarajatlari

6.3. AI-yordamli tahlil tizimlari

Bojxona organlari xavflarni baholash va qonunbuzarliklarni aniqlash uchun AI tizimlaridan foydalanadi. Bu tizimlar kuniga millionlab strukturalangan ma'lumot punktlarini qayta ishlaydi. TOON formati:

1. LLM kontekst oynasida ko'proq ma'lumot joylashtirishga imkon beradi
2. Inferensiya xarajatlarini 28.6% ga kamaytiradi
3. Real vaqt rejimida tahlil tezligini oshiradi

Xulosa

Ushbu tadqiqotda TOON va JSON formatlarining qat'iy matematik taqqoslanishi amalga oshirildi va quyidagi asosiy xulosalarga kelindi:

1. Matematik isbot: TOON formati JSON-ga nisbatan strukturaviy ortiqchalikni bartaraf etish orqali byte samaradorligini oshiradi. Rekursiv byte uzunligi funksiyalari (Ljson va Ltoon) orqali bu yaxshilanish aniq miqdoriy jihatdan isbotlandi.

2. Turli strukturalar uchun tejamkorlik:

- Oddiy obyektlar: 28.6% tejash
- Primitiv massivlar: 28.6% tejash
- Jadval massivlari: 35.6% tejash (eng samarali)
- Bo'sh obyektlar: 100% tejash
- Ichki obyektlar: 16.7% tejash

3. Cheklov: Massivlar massivi strukturasi TOON 52.2% ortiqcha byte ishlatadi, shuning uchun bu holat uchun JSON tavsiya etiladi.

4. Iqtisodiy ta'sir: 1,000,000 ta oddiy obyekt uchun TOON formati 2.86 MB tejaydi, bu esa LLM API xarajatlarini sezilarli darajada kamaytiradi. Yillik miqyosda, yirik tashkilotlar uchun bu millionlab dollar tejashni anglatadi.

5. Tokenizatsiya samaradorligi: UTF-8 byte uzunligi tokenizatsiya uchun asosiy birlik bo'lganligi sababli, Ltoon-ning kamayishi to'g'ridan-to'g'ri token sonining kamayishiga olib keladi. Bu esa LLM tizimlarining iqtisodiy samaradorligini yaxshilaydi.

Xulosa qilib aytganda, TOON formati matematik jihatdan isbotlangan ustunliklarga ega bo'lib, LLM tizimlarining iqtisodiy samaradorligini oshirish uchun perspektivali yechim hisoblanadi. Massivlar massivi tuzilmasidan tashqari barcha asosiy ma'lumot strukturalari uchun TOON JSON-ga nisbatan 16.7% dan 100% gacha

byte tejamkorligini ta'minlaydi. Bu esa AI davrida ma'lumot seriyalashning yangi bosqichini boshlab beradi.

Foydalanilgan adabiyotlar ro'yxati

1. Grand View Research. (2024). Large Language Model Market Size, Share & Trends Analysis Report. <https://www.grandviewresearch.com/industry-analysis/large-language-model-market>
2. Bray, T. (Ed.). (2017). The JavaScript Object Notation (JSON) Data Interchange Format (RFC 8259). Internet Engineering Task Force. <https://doi.org/10.17487/RFC8259>
3. Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. <https://doi.org/10.18653/v1/P16-1162>
4. Lafalce, M. (2024). TOON vs. JSON: A Mathematical Evaluation of Byte Efficiency in Structured Data. UTN-FRLP. <https://toonformat.dev/>
5. OpenAI. (2024). API Pricing. <https://openai.com/pricing>
6. McKinsey & Company. (2023). The Economic Potential of Generative AI: The Next Productivity Frontier. <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier>
7. Ecma International. (2017). The JSON Data Interchange Syntax (Standard No. ECMA-404, 2nd ed.). <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>
8. W3C. (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition). <https://www.w3.org/TR/xml/>