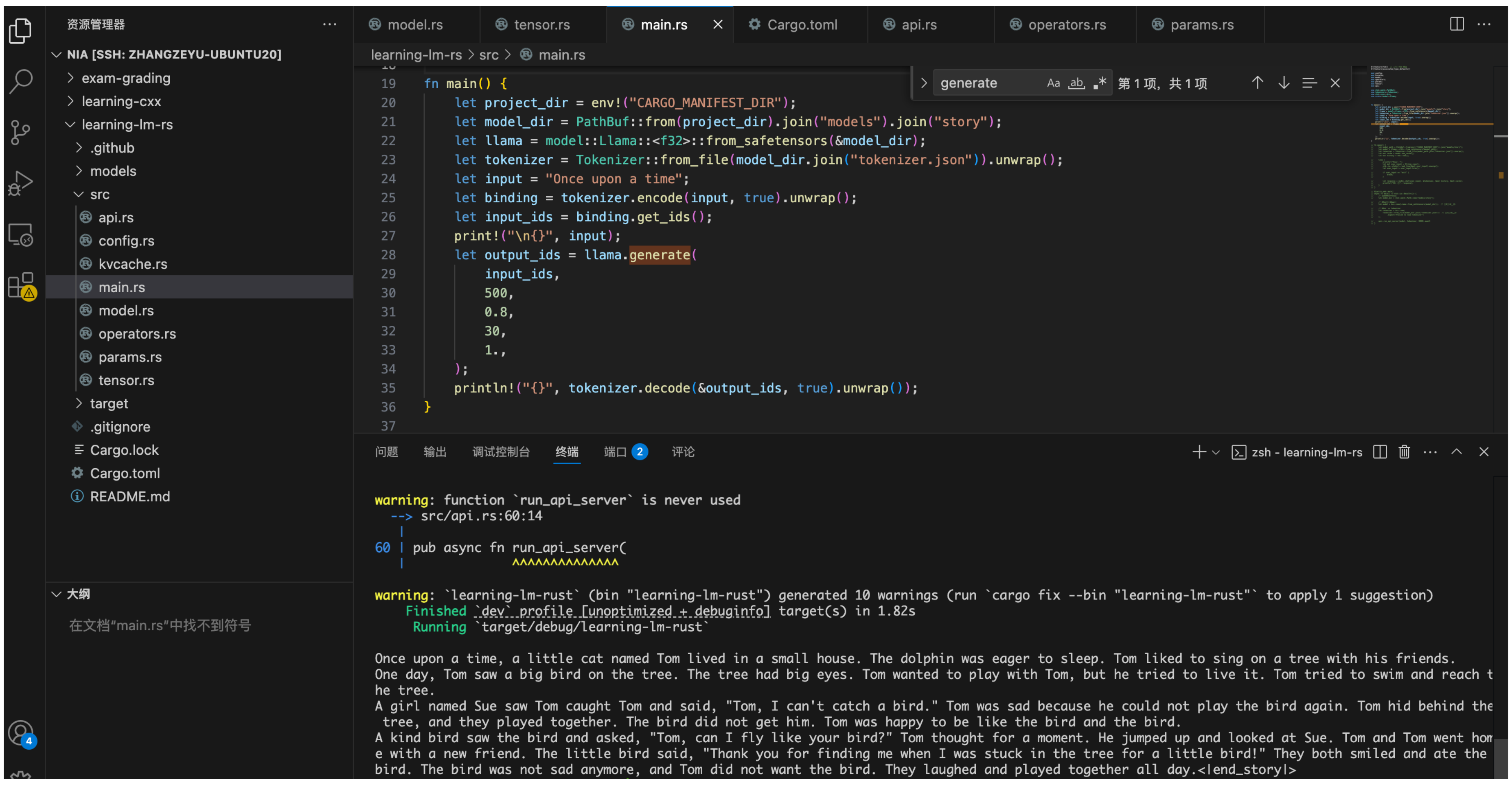
Self-Attention

对于GQA注意力，选择处理方法是将矩阵视为多个向量，按照对应关系手动进行索引和向量乘。将Q视作Q[seq_len][q_head][dim]、KV视作[total_seq_len][k_head][dim]形状进行遍历。

文本生成 & AI对话

生成函数generate首先初始化一个kvcache，并通过循环函数来进行forward。AI对话的实现首先组织Jinja2模板结构，使用tokenizer编码后调用generate完成对话功能。

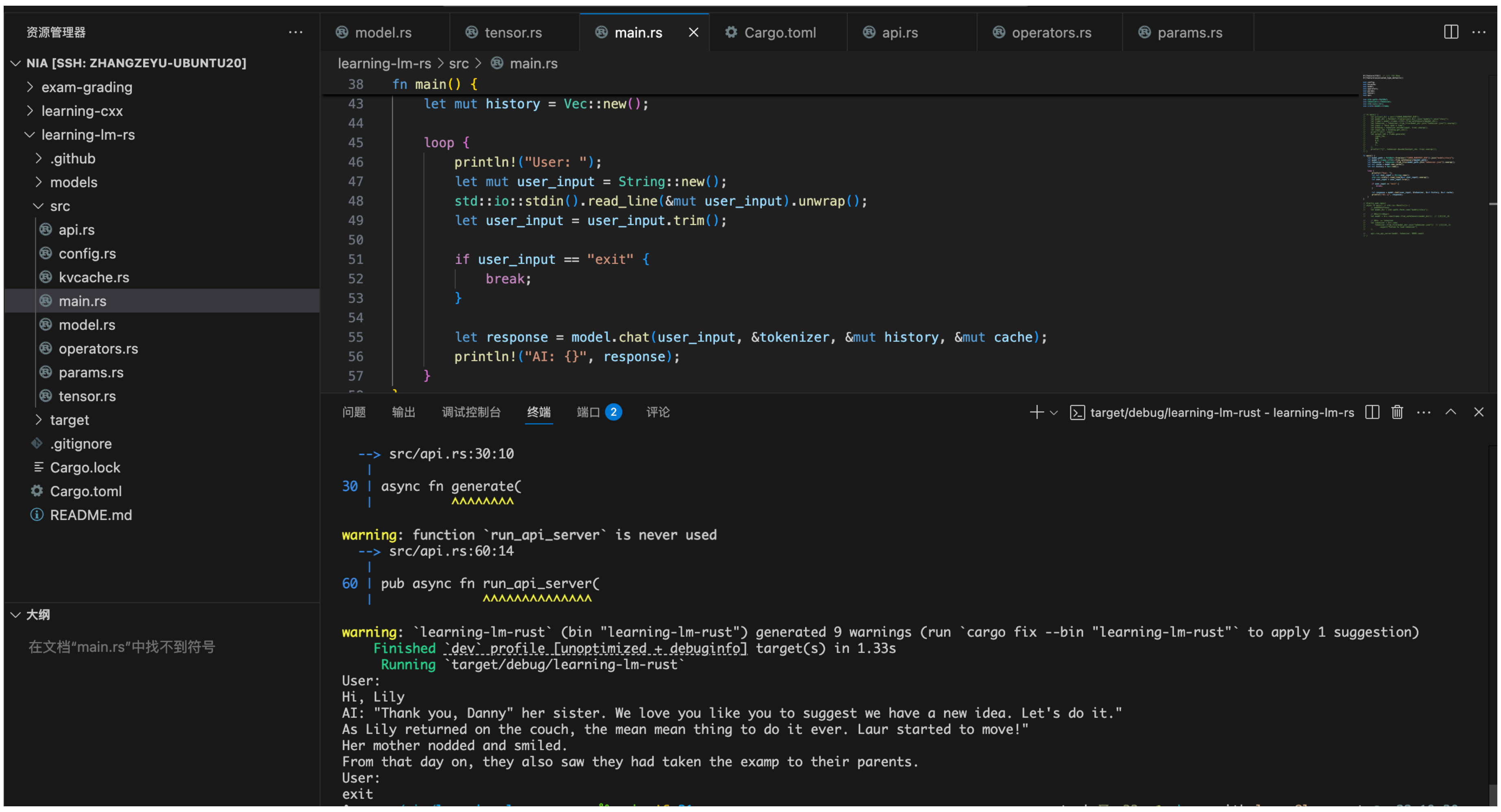
基础文本生成：



```rust
fn main() {
    let project_dir = env!("CARGO_MANIFEST_DIR");
    let model_dir = PathBuf::from(project_dir).join("models").join("story");
    let llama = model::Llama::<f32>::from_safetensors(&model_dir);
    let tokenizer = Tokenizer::from_file(model_dir.join("tokenizer.json")).unwrap();
    let input = "Once upon a time";
    let binding = tokenizer.encode(input, true).unwrap();
    let input_ids = binding.get_ids();
    print!("\n{}", input);
    let output_ids = llama.generate(
        input_ids,
        500,
        0.8,
        30,
        1.,
    );
    println!("{}", tokenizer.decode(&output_ids, true).unwrap());
}
```

```
warning: function `run_api_server` is never used
  --> src/api.rs:60:14
   |
60 | pub async fn run_api_server(
   |              ^^^^^^^^^^^^^^

warning: `learning-lm-rust` (bin "learning-lm-rust") generated 10 warnings (run `cargo fix --bin "learning-lm-rust"` to apply 1 suggestion)
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 1.82s
     Running `target/debug/learning-lm-rust`

Once upon a time, a little cat named Tom lived in a small house. The dolphin was eager to sleep. Tom liked to sing on a tree with his friends.
One day, Tom saw a big bird on the tree. The tree had big eyes. Tom wanted to play with Tom, but he tried to live it. Tom tried to swim and reach t
he tree.
A girl named Sue saw Tom caught Tom and said, "Tom, I can't catch a bird." Tom was sad because he could not play the bird again. Tom hid behind the
 tree, and they played together. The bird did not get him. Tom was happy to be like the bird and the bird.
A kind bird saw the bird and asked, "Tom, can I fly like your bird?" Tom thought for a moment. He jumped up and looked at Sue. Tom and Tom went hom
e with a new friend. The little bird said, "Thank you for finding me when I was stuck in the tree for a little bird!" They both smiled and ate the
bird. The bird was not sad anymore, and Tom did not want the bird. They laughed and played together all day.<|end_story|>
```

AI对话：



```rust
fn main() {
    let mut history = Vec::new();

    loop {
        println!("User: ");
        let mut user_input = String::new();
        std::io::stdin().read_line(&mut user_input).unwrap();
        let user_input = user_input.trim();

        if user_input == "exit" {
            break;
        }

        let response = model.chat(user_input, &tokenizer, &mut history, &mut cache);
        println!("AI: {}", response);
    }
```

```
  --> src/api.rs:30:10
   |
30 | async fn generate(
   |          ^^^^^^^^

warning: function `run_api_server` is never used
  --> src/api.rs:60:14
   |
60 | pub async fn run_api_server(
   |              ^^^^^^^^^^^^^^

warning: `learning-lm-rust` (bin "learning-lm-rust") generated 9 warnings (run `cargo fix --bin "learning-lm-rust"` to apply 1 suggestion)
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 1.33s
     Running `target/debug/learning-lm-rust`
User:
Hi, Lily
AI: "Thank you, Danny" her sister. We love you like you to suggest we have a new idea. Let's do it."
As Lily returned on the couch, the mean mean thing to do it ever. Laur started to move!"
Her mother nodded and smiled.
From that day on, they also saw they had taken the examp to their parents.
User:
exit
```

项目扩展：混合精度推理

在tensor中实现类型转换：

```rust
use half::f16;
#[derive(Debug, Clone, Copy)]
pub enum DType {
    F16,
    F32
}
```

```
impl<T: Copy + Clone + Default + Into<f32>> Tensor<T> {
    pub fn to_dtype(&self, dtype: DType) -> Tensor<f32> {
        match dtype {
            DType::F16 => {
                let converted: Vec<f16> = self.data()
                    .iter()
                    .map(|x| f16::from_f32((*x).into()))
                    .collect();
                let f32_data: Vec<f32> = converted.iter().map(|x| x.to_f32()).collect();
                Tensor::new(f32_data, &self.shape)
            }
            DType::F32 => Tensor::new(
                self.data().iter().map(|x| (*x).into()).collect(),
                &self.shape
            )
        }
    }
}
```

设计思想：在内存敏感操作使用FP16存储，计算敏感操作保持FP32精度。转换embedding查找过程使用FP16：

```
 let table_f16 = self.params.embedding_table.to_dtype(DType::F16);
 OP::gather(&mut residual, input, &table_f16);
```

项目扩展：网络服务 API

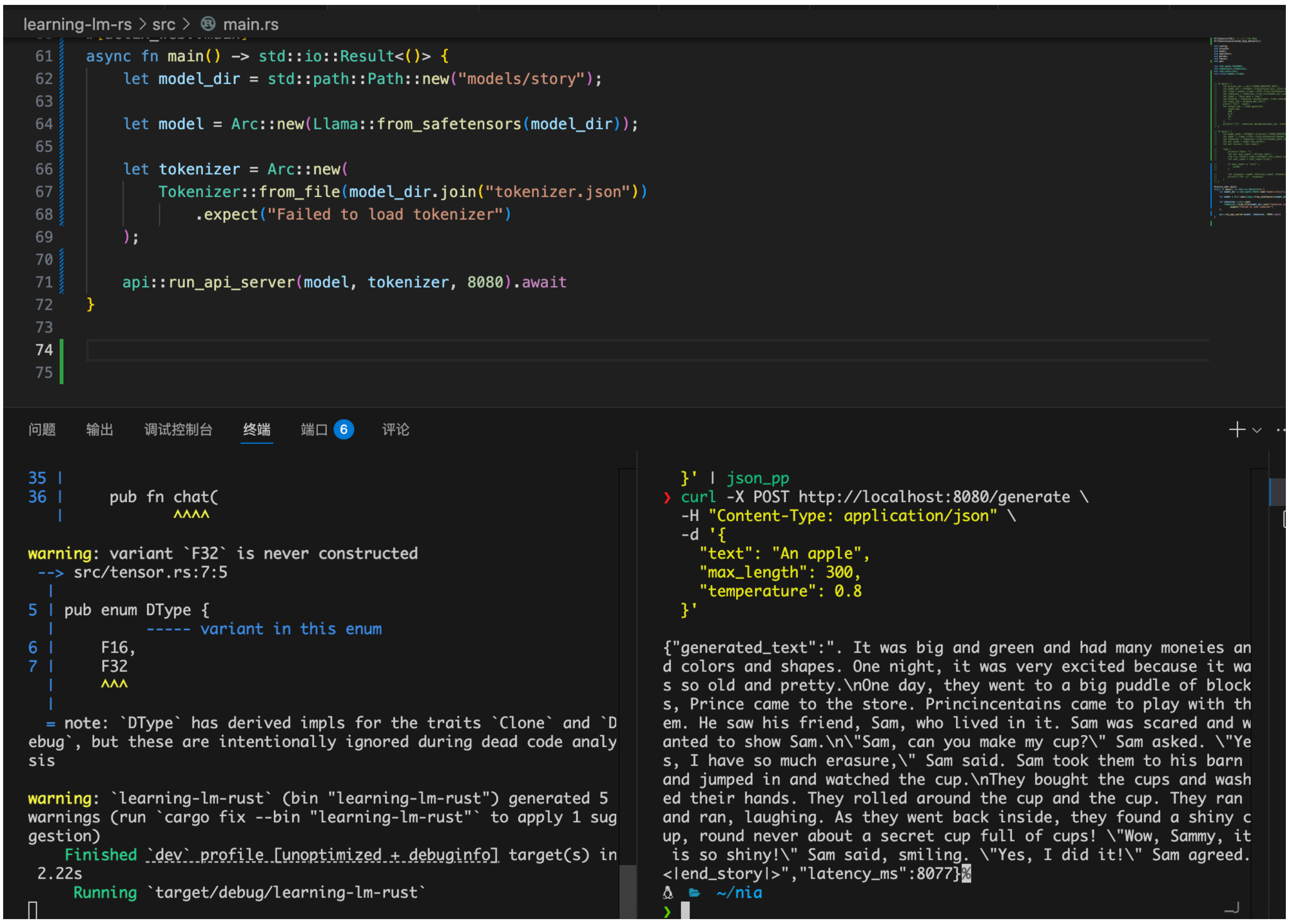
基于Actix-web来实现网络服务API模块，首先设计请求、响应体结构和共享结构来处理JSON数据格式的序列化和反序列化：

```
#[derive(Deserialize)]
struct GenerateRequest {
    text: String,
    max_length: Option<usize>,
    temperature: Option<f32>,
}

#[derive(Serialize)]
struct GenerateResponse {
    generated_text: String,
    latency_ms: u64,
}

struct AppState {
    model: Arc<Llama<f32>>,
    tokenizer: Arc<Tokenizer>,
}
```

主要api函数步骤分为：编码输入、执行推理、解码输出步骤，请求参数为：

```
{
  "text": "必填，输入提示文本",
  "max_length": "可选，默认100",
  "temperature": "可选，默认0.7"
}
```

请求API：



```rust
async fn main() -> std::io::Result<()> {
    let model_dir = std::path::Path::new("models/story");

    let model = Arc::new(Llama::from_safetensors(model_dir));

    let tokenizer = Arc::new(
        Tokenizer::from_file(model_dir.join("tokenizer.json"))
            .expect("Failed to load tokenizer")
    );

    api::run_api_server(model, tokenizer, 8080).await
}


```

```
35 |
36 |     pub fn chat(
   |            ^^^^

warning: variant `F32` is never constructed
 --> src/tensor.rs:7:5
  |
5 | pub enum DType {
  |          ----- variant in this enum
6 |     F16,
7 |     F32
  |     ^^^
  |
  = note: `DType` has derived impls for the traits `Clone` and `D
ebug`, but these are intentionally ignored during dead code analy
sis

warning: `learning-lm-rust` (bin "learning-lm-rust") generated 5
warnings (run `cargo fix --bin "learning-lm-rust"` to apply 1 sug
gestion)
    Finished `dev` profile [unoptimized + debuginfo] target(s) in
 2.22s
     Running `target/debug/learning-lm-rust`
```

```
  }' | json_pp
❯ curl -X POST http://localhost:8080/generate \
  -H "Content-Type: application/json" \
  -d '{
    "text": "An apple",
    "max_length": 300,
    "temperature": 0.8
  }'

{"generated_text":". It was big and green and had many moneies an
d colors and shapes. One night, it was very excited because it wa
s so old and pretty.\nOne day, they went to a big puddle of block
s, Prince came to the store. Princincentains came to play with th
em. He saw his friend, Sam, who lived in it. Sam was scared and w
anted to show Sam.\n\"Sam, can you make my cup?\" Sam asked. \"Ye
s, I have so much erasure,\" Sam said. Sam took them to his barn
and jumped in and watched the cup.\nThey bought the cups and wash
ed their hands. They rolled around the cup and the cup. They ran
and ran, laughing. As they went back inside, they found a shiny c
up, round never about a secret cup full of cups! \"Wow, Sammy, it
 is so shiny!\" Sam said, smiling. \"Yes, I did it!\" Sam agreed.
<|end_story|>","latency_ms":8077}%
  ~/nia
❯
```