```
from unittest.mock import Mock, patch

from transformers import AutoModelForCausalLM, AutoTokenizer

from swarms import ToolAgent



def test_tool_agent_init():
    model = Mock(spec=AutoModelForCausalLM)
    tokenizer = Mock(spec=AutoTokenizer)
    json_schema = {
        "type": "object",
        "properties": {
            "name": {"type": "string"},
            "age": {"type": "number"},
            "is_student": {"type": "boolean"},
            "courses": {"type": "array", "items": {"type": "string"}},
        },
    }
    name = "Test Agent"
    description = "This is a test agent"

    agent = ToolAgent(
        name, description, model, tokenizer, json_schema
    )
```

```
    assert agent.name == name
    assert agent.description == description
    assert agent.model == model
    assert agent.tokenizer == tokenizer
    assert agent.json_schema == json_schema


@patch.object(ToolAgent, "run")
def test_tool_agent_run(mock_run):
    model = Mock(spec=AutoModelForCausalLM)
    tokenizer = Mock(spec=AutoTokenizer)
    json_schema = {
        "type": "object",
        "properties": {
            "name": {"type": "string"},
            "age": {"type": "number"},
            "is_student": {"type": "boolean"},
            "courses": {"type": "array", "items": {"type": "string"}},
        },
    }
    name = "Test Agent"
    description = "This is a test agent"
    task = (
        "Generate a person's information based on the following"
        " schema:"
    )
```

```
    agent = ToolAgent(
        name, description, model, tokenizer, json_schema
    )
    agent.run(task)

    mock_run.assert_called_once_with(task)



def test_tool_agent_init_with_kwargs():
    model = Mock(spec=AutoModelForCausalLM)
    tokenizer = Mock(spec=AutoTokenizer)
    json_schema = {
        "type": "object",
        "properties": {
            "name": {"type": "string"},
            "age": {"type": "number"},
            "is_student": {"type": "boolean"},
            "courses": {"type": "array", "items": {"type": "string"}},
        },
    }
    name = "Test Agent"
    description = "This is a test agent"

    kwargs = {
        "debug": True,
```

```
    "max_array_length": 20,
    "max_number_tokens": 12,
    "temperature": 0.5,
    "max_string_token_length": 20,
}

agent = ToolAgent(
    name, description, model, tokenizer, json_schema, **kwargs
)

assert agent.name == name
assert agent.description == description
assert agent.model == model
assert agent.tokenizer == tokenizer
assert agent.json_schema == json_schema
assert agent.debug == kwargs["debug"]
assert agent.max_array_length == kwargs["max_array_length"]
assert agent.max_number_tokens == kwargs["max_number_tokens"]
assert agent.temperature == kwargs["temperature"]
assert (
    agent.max_string_token_length
    == kwargs["max_string_token_length"]
)
```