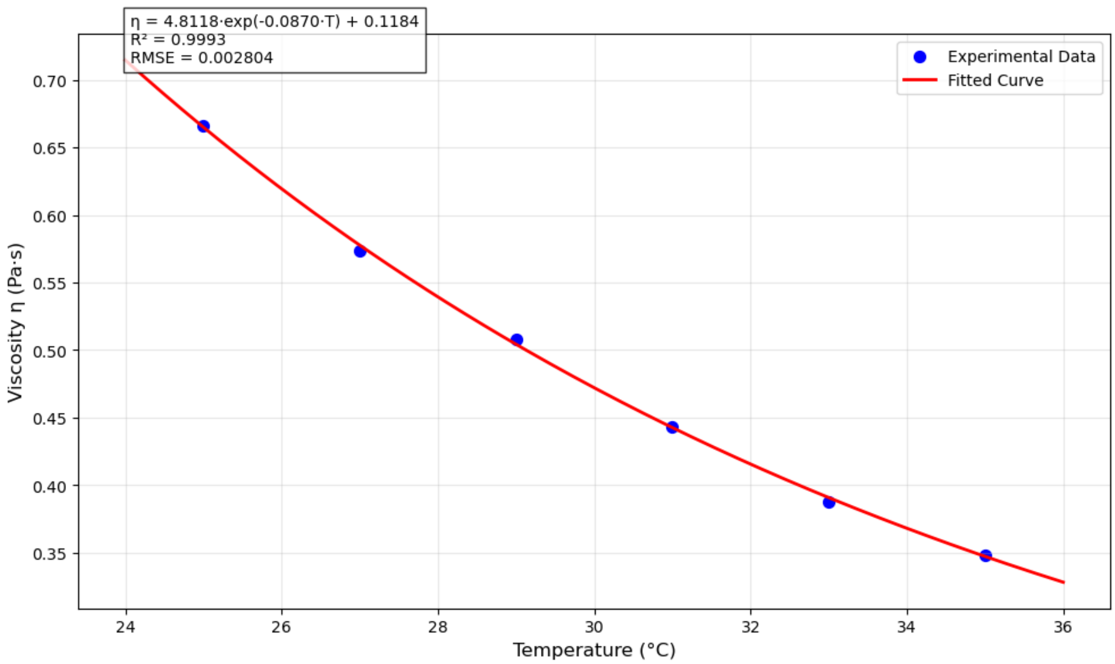
eta_t

December 16, 2024

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

def exp_func(x, a, b, c):
    return a * np.exp(-b * x) + c

temp = np.array([25, 27, 29, 31, 33, 35])
eta1 = np.array([0.6661057331996978, 0.5734885930386372, 0.5083247869099908,
                 0.4434899016082358, 0.3877024202929399, 0.3485997586979679 4])

p0 = [2.0, 0.05, 0.3]
bounds = ([0, 0, 0], [np.inf, np.inf, np.inf])

popt, pcov = curve_fit(exp_func, temp, eta1, p0=p0, bounds=bounds)
a, b, c = popt

temp_fit = np.linspace(min(temp)-1, max(temp)+1, 200)
eta_fit = exp_func(temp_fit, a, b, c)

residuals = eta1 - exp_func(temp, *popt)
ss_res = np.sum(residuals**2)
ss_tot = np.sum((eta1 - np.mean(eta1))**2)
r_squared = 1 - (ss_res / ss_tot)
rmse = np.sqrt(np.mean(residuals**2))

plt.figure(figsize=(10, 6))
plt.scatter(temp, eta1, color='blue', label='Experimental Data', s=50)
plt.plot(temp_fit, eta_fit, 'r-', label='Fitted Curve', linewidth=2)
plt.xlabel('Temperature (°C)', fontsize=12)
plt.ylabel('Viscosity   (Pa·s)', fontsize=12)
plt.grid(True, alpha=0.3)
plt.legend(fontsize=10)

equation = f'  = {a:.4f}·exp(-{b:.4f}·T) + {c:.4f}'
stats = f'R² = {r_squared:.4f}\nRMSE = {rmse:.6f}'
plt.text(0.05, 0.95, equation + '\n' + stats,
```

```
         transform=plt.gca().transAxes,
         bbox=dict(facecolor='white', alpha=0.8))

plt.tight_layout()
plt.show()

print(f"Fit Parameters:")
print(f"a = {a:.6f}")
print(f"b = {b:.6f}")
print(f"c = {c:.6f}")
print(f"R-squared = {r_squared:.6f}")
print(f"RMSE = {rmse:.6f}")
```




```
Fit Parameters:
a = 4.811766
b = 0.087013
c = 0.118437
R-squared = 0.999335
RMSE = 0.002804
```