

Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

Лабораторна робота № 5

Дисципліна: Хмарні технології

Тема: Створення застосунку зв'язування сутностей із застосуванням Azure AI Language

Виконав: Студент групи МІТ-31

Пугач Назар

Мета: Створити застосунку зв'язування сутностей із застосуванням Azure AI Language

Хід роботи

Завдання 5.1: Створити Azure AI Language сервіс.

Ми створювали його в попередній лабораторній роботі.

Завдання 5.2: Створити консольний застосунок.

```
using Azure;
using System;
using System.Globalization;
using Azure.AI.TextAnalytics;

namespace EntityLinkingExample
{
    class Program
    {
        // This example requires environment variables named "LANGUAGE_KEY" and
        "LANGUAGE_ENDPOINT"
        static string languageKey =
        Environment.GetEnvironmentVariable("LANGUAGE_KEY");
        static string languageEndpoint =
        Environment.GetEnvironmentVariable("LANGUAGE_ENDPOINT");

        private static readonly AzureKeyCredential credentials = new
        AzureKeyCredential(languageKey);
        private static readonly Uri endpoint = new Uri(languageEndpoint);

        // Example method for recognizing entities and providing a link to an
        online data source.
        static void EntityLinkingExample(TextAnalyticsClient client)
        {
            var response = client.RecognizeLinkedEntities(
                "Microsoft was founded by Bill Gates and Paul Allen on April 4,
1975, " +
                "to develop and sell BASIC interpreters for the Altair 8800. " +
                "During his career at Microsoft, Gates held the positions of
chairman, " +
                "chief executive officer, president and chief software
architect, " +
                "while also being the largest individual shareholder until May
2014.");
            Console.WriteLine("Linked Entities:");
            foreach (var entity in response.Value)
            {
                Console.WriteLine($"\\tName: {entity.Name},\\tID:
{entity.DataSourceEntityId},\\tURL: {entity.Url}\\tData Source:
{entity.DataSource}");
                Console.WriteLine("\\tMatches:");
                foreach (var match in entity.Matches)
                {
                    Console.WriteLine($"\\t\\tText: {match.Text}");
                    Console.WriteLine($"\\t\\tScore:
{match.ConfidenceScore:F2}\\n");
                }
            }
        }

        static void Main(string[] args)
        {
            var client = new TextAnalyticsClient(endpoint, credentials);
            EntityLinkingExample(client);

            Console.WriteLine("Press any key to exit.");
            Console.ReadKey();
        }
    }
}
```

```
}
```

Рисунок 5.1 – Код застосунку.

Завдання 5.3: Створити Web-застосунок за шаблоном ASP.NET MVC, що пропонує користувачу ввести текст та розпізнати сутності з нього й пов'язати їх із відповідними посиланнями на базу знань. Результати вивести у формі таблиці.

Спочатку створимо модель.

```
using System.Collections.Generic;

namespace lab5.Models
{
    public class TextInputModel
    {
        public string Text { get; set; }
        public List<EntityModel> Entities { get; set; } = new();
    }

    public class EntityModel
    {
        public string Name { get; set; }
        public string Url { get; set; }
        public string DataSource { get; set; }
        public List<EntityMatch> Matches { get; set; } = new();
    }

    public class EntityMatch
    {
        public string Text { get; set; }
        public double ConfidenceScore { get; set; }
    }
}
```

Рисунок 5.2 – Модель.

Змінимо оригінальний контролер під наші потреби.

```
using Azure.AI.TextAnalytics;
using lab5.Models;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;

namespace lab5.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;
        private readonly TextAnalyticsClient _textAnalyticsClient;

        public HomeController(ILogger<HomeController> logger,
            TextAnalyticsClient textAnalyticsClient)
        {
            _logger = logger;
            _textAnalyticsClient = textAnalyticsClient;
        }

        public IActionResult Index()
        {
            return View(new TextInputModel());
        }

        [HttpPost]
        public IActionResult Analyze(TextInputModel model)
```

```

    {
        if (string.IsNullOrEmpty(model.Text))
        {
            ModelState.AddModelError("", "Enter text.");
            return View("Index", model);
        }

        var response =
            _textAnalyticsClient.RecognizeLinkedEntities(model.Text);
        model.Entities = response.Value.Select(entity => new EntityModel
        {
            Name = entity.Name,
            Url = entity.Url?.ToString(),
            DataSource = entity.DataSource,
            Matches = entity.Matches.Select(m => new EntityMatch
            {
                Text = m.Text,
                ConfidenceScore = m.ConfidenceScore
            }).ToList()
        }).ToList();

        return View("Index", model);
    }

    public IActionResult Privacy()
    {
        return View();
    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None,
NoStore = true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
    }
}

```

Рисунок 5.3 – Контролер.

Змінимо відображення.

```

@model lab5.Models.TextInputModel

<h2>Text Analysis and Entity Recognition</h2>

<form asp-action="Analyze" method="post">
    <div class="form-group">
        <label for="Text">Enter text:</label>
        <textarea class="form-control" id="Text" name="Text"
rows="4">@Model?.Text</textarea>
    </div>
    <button type="submit" class="btn btn-primary mt-2">Analyze</button>
</form>

@if (Model?.Entities?.Count > 0)
{
    <h3 class="mt-4">Recognized Entities:</h3>
    <table class="table table-bordered mt-2">
        <thead class="table-dark">
            <tr>
                <th>Name</th>
                <th>Source</th>
                <th>URL</th>
            </tr>
        </thead>
        <tbody>

```

```

@foreach (var entity in Model.Entities)
{
    <tr>
        <td>@entity.Name</td>
        <td>@entity.DataSource</td>
        <td><a href="@entity.Url" target="_blank">Link</a></td>
    </tr>
    <tr>
        <td colspan="4">
            <strong>Matches:</strong>
            <ul>
                @foreach (var match in entity.Matches)
                {
                    <li>@match.Text (Confidence:
@match.ConfidenceScore.ToString("F2"))</li>
                }
            </ul>
        </td>
    </tr>
}
</tbody>
</table>
}

```

Рисунок 5.4 – Відобарження.

Додамо ключі в appsettings.json.

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AzureAI": {
    "LanguageKey": "lanKey",
    "LanguageEndpoint": "lanEndpoint"
  },
  "AllowedHosts": "*"
}

```

Рисунок 5.5 – Ключі.

Завдання 5.4: Перевірка працезданості.

Text Analysis and Entity Recognition

Enter text:

Microsoft was founded by Bill Gates and Paul Allen on April 4, 1975

Analyze

Recognized Entities:

Name	Source	URL
Microsoft	Wikipedia	Link
Matches: <ul style="list-style-type: none">Microsoft (Confidence: 0,48)		
Bill Gates	Wikipedia	Link
Matches: <ul style="list-style-type: none">Bill Gates (Confidence: 0,52)		
Paul Allen	Wikipedia	Link
Matches: <ul style="list-style-type: none">Paul Allen (Confidence: 0,54)		
April 4	Wikipedia	Link

Рисунок 5.6 – Результат.

Висновок: Під час виконання лабораторної роботи я створив застосунок зв'язування сутностей із застосуванням Azure AI Language