Content Recommendation Module

Traction Project Internship

Goals

1. Develop a simple web crawler to crawl a website
2. Parse the website content
3. Process and classify the parsed content
4. Load the data into a graph format
5. Synchronize the graph data with a bot in DialogFlow

Planning

Diagrams; the situation

- **User:**

I am looking for graph implementation on Content Modeling

- **Bot:**

I didn’t understand your question but I found some content that might be helpful:

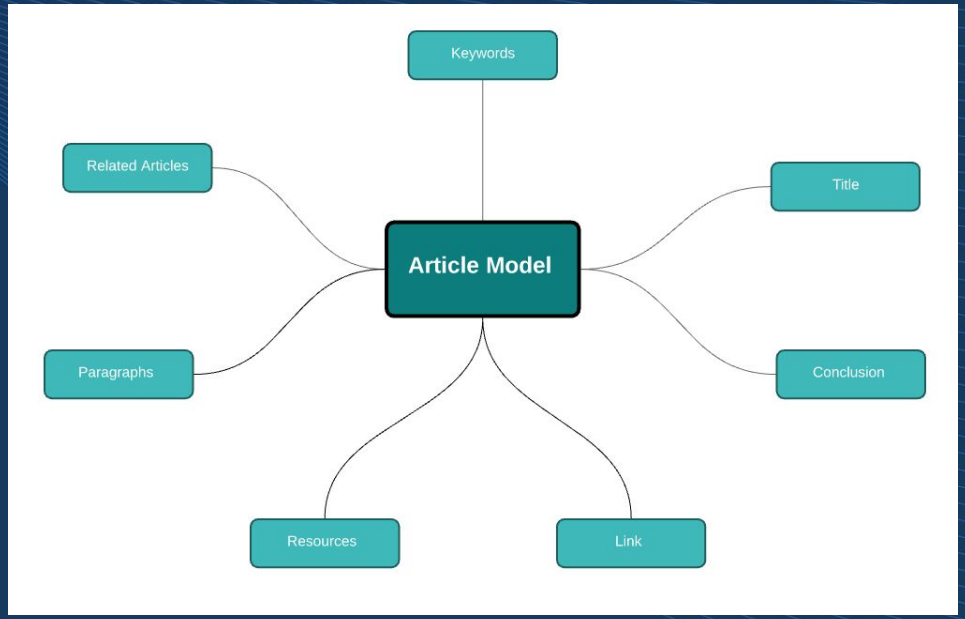
Paragraph
Link to Article 1

Paragraph
Link to Article 2

- **Bot:**

Was that helpful? Yes / No

Diagrams; article node



- **Link**
- **Title**

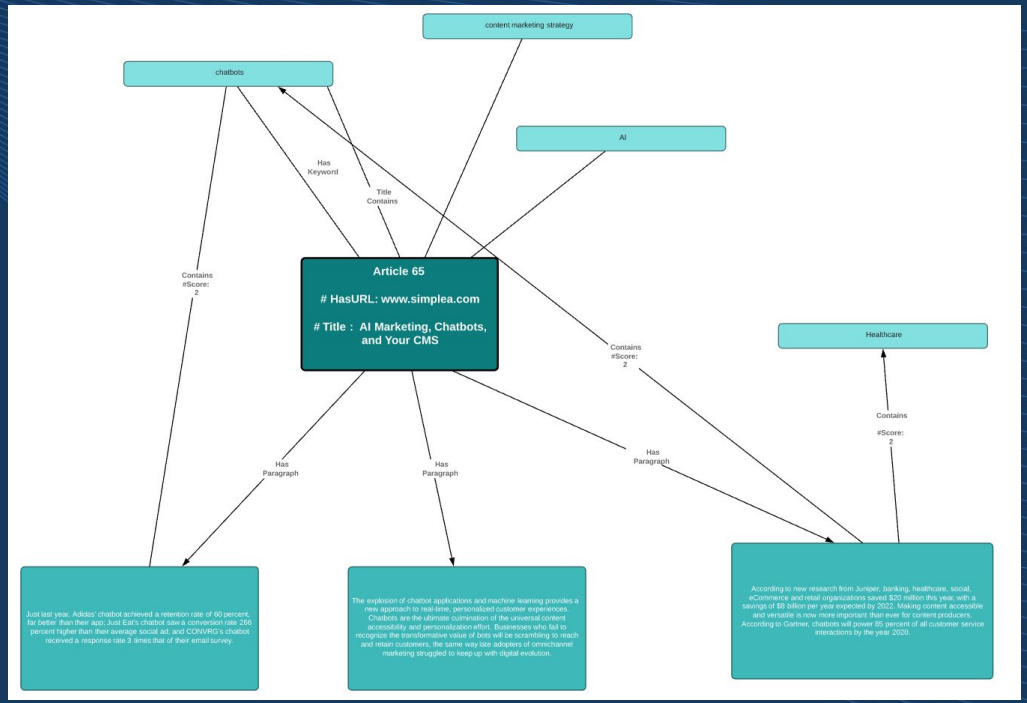
  <title>
- **Keywords**

  <meta name=”keywords”>
- **Paragraphs**

  <p>
- **Conclusion**

  <meta name=”description”>

Diagrams;


content marketing strategy
chatbots
AI
Has Keyword
Title Contains
Article 65
HasURL: www.simplea.com
Title : AI Marketing, Chatbots, and Your CMS
Contains #Score: 2
Contains #Score: 2
Healthcare
Contains #Score: 2
Has Paragraph
Has Paragraph
Has Paragraph
Just last year, Adidas' chatbot achieved a retention rate of 60 percent, far better than their app; Just Eat's chatbot saw a conversion rate 266 percent higher than their average social ad; and CONVRG's chatbot received a response rate 3 times that of their email survey.
The explosion of chatbot applications and machine learning provides a new approach to real-time, personalized customer experiences. Chatbots are the ultimate culmination of the universal content accessibility and personalization effort. Businesses who fail to recognize the transformative value of bots will be scrambling to reach and retain customers, the same way late adopters of omnichannel marketing struggled to keep up with digital evolution.
According to new research from Juniper, banking, healthcare, social, eCommerce and retail organizations saved $20 million this year, with a savings of $8 billion per year expected by 2022. Making content accessible and versatile is now more important than ever for content producers. According to Gartner, chatbots will power 85 percent of all customer service interactions by the year 2020.

Construction

Projects

- **ABot.RecommenderSystem**

ASP.NET Core project that has GraphQL API implemented to query the database.

  - Neo4j graph platform
  - Neo4j .NET driver

- **ABot.Spider**

Console application that works in a linear way to collect the data and save it into the database.

  - DotnetSpider library

Program sequence

```
// 1. Start web crawling and parsing
var spider = DotnetSpider.Spider.Create<SimpleaSpider>();
spider.Run();

// 2. Process the parsed data
FileHandler fh = new FileHandler(spider.Id, executionPath, projectPath);
var file = fh.textFile;
DataProcessor dp = new DataProcessor(executionPath, projectPath);
var articles = dp.GetArticles(file);

// 3. Look for changes, updates, and news from last crawled session
var updates = fh.SameDiff(dp);

// 4. Store data
dp.StoreData(updates);

// 5. Save a backup json file with crawled data
fh.StoreFile();
fh.DeleteFile(spider.Id);
```

- Program.cs

1. Creates and runs a spider
2. The data is stored as JSON
3. The data is processed by a simple form of normalization and word count
4. The data is stored in a graph database.

Program sequence

```
public class SimpleaSpider : DotnetSpider.Spider
{

    protected override void Initialize()
    {
        NewGuidId();
        Scheduler = new QueueDistinctBfsScheduler();
        DownloaderSettings.Type = DownloaderType.HttpClient;
        AddDataFlow(new SimpleaDataParser()).AddDataFlow(new JsonFileStorage());
        AddRequests("https://simplea.com/Articles/facing-content-supply-chain-problems");
    }

    class SimpleaDataParser : DataParser
    {
        public SimpleaDataParser()
        {
            CanParse = DataParserHelper.CanParseByRegex("simplea\\.com/Articles");
            QueryFollowRequests = DataParserHelper.QueryFollowRequestsByXPath(".");
        }

        protected override Task<DataFlowResult> Parse(DataFlowContext context)
        {
            if (context.Response != null)
            {
                context.AddItem("URL", context.Response.Request.Url);
                context.AddItem("Title", context.GetSelectable().XPath("//title").GetValue());
                context.AddItem("Keywords", context.GetSelectable().XPath("//meta[@name='keywords']/@content").GetValue());
                context.AddItem("Summary", context.GetSelectable().XPath("//meta[@name='description']/@content").GetValue());
                var pTags = context.GetSelectable().XPath("//p").Nodes();
                int nr = 1;
                foreach (var p in pTags)
                {
                    context.AddItem("Paragraph " + nr, p.GetValue());
                    nr++;
                }
            }
            return Task.FromResult(DataFlowResult.Success);
        }
    }
```

```
[16:55:06 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? https://simplea.com/Articles/what-is-content-as-a-service ??
[16:55:06 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? https://simplea.com/Articles/what-is-content-as-a-service ??
[16:55:07 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 22, ?? 5, ?? 0, ?? 17
[16:55:07 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? https://simplea.com/Articles/what-is-content-convergence ??
[16:55:08 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? https://simplea.com/Articles/what-is-content-convergence ??
[16:55:08 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? https://simplea.com/Articles/facing-content-supply-chain-problems
```


```
{
  "Paragraph 20": "Make basic, <a href=\"https://simplea.com/Articles/What-is-Website-and-Content-Taxonomy\" target=\"_b
  "Paragraph 19": "Even if only via a shared spreadsheet, we need to agree on terminology (what we call things) and how w
  "Paragraph 13": "Find ways to streamline content workflows, improving author experience.",
  "Paragraph 21": "Finally, look for all opportunities to build bridges. Bridge builders are the new power brokers.",
  "Paragraph 25": "[A]&amp;nbsp;is the Content Intelligence Service. In partnership with leading global enterprises,&amp;nbsp;[A
  "Paragraph 7": "Start to quantify the problem. Track what it actually takes to produce content in all stages.",
  "Paragraph 22": "Engineering Intelligence: An Interview with Cruce Saunders",
  "Paragraph 4": "Expanded personalization and adaptive experiences",
  "Paragraph 11": "Exemplars of that content",
  "Paragraph 5": "In summary, if we are to prepare for a more flexible future, which includes AR and conversational user
  "Paragraph 24": "Watch [A] founder, Cruce Saunders, as he explores how to streamline and automate content supply chains
  "Paragraph 6": "Find others that care about content and form a content club. Describe the problems in writing, brainsto
  "Paragraph 17": "Start content modeling.",
  "URL": "https://simplea.com/Articles/facing-content-supply-chain-problems",
  "Paragraph 15": "Identify content types and their relationships.",
  "Keywords": "content marketing, content supply chain, intelligent content, content modeling",
  "Paragraph 3": "Voice, <a href=\"https://simplea.com/Publications/Whitepapers/Chatbot-Inside-CMS\" target=\"_blank\">ch
  "Paragraph 9": "Expressions and renderings",
  "Title": "\r\n\tNow Is the Time to Face Our Content Supply Chain Problems\r\n",
  "Paragraph 12": "Customer types and segments",
  "Paragraph 10": "Customer experiences or journeys",
  "Paragraph 8": "Start inventories of content assets:",
  "Paragraph 23": "Culture and Technology Change: How to Get Organizational Buy-In",
  "Paragraph 1": "So, we already have a lot to manage. But now new significant content elements are emerging that will fu
  "Paragraph 2": "Augmented reality (AR) and virtual reality (VR)",
  "Paragraph 16": "Create an inventory.",
  "Summary": "The dynamics of content creation and supply chains are literally shifting under our feet. We face many cha
```

Program sequence

```
foreach (string line in file)
{
    article = JsonConvert.DeserializeObject<Article>(line);
    article.Url = article.Url.Trim();
    article.Keywords = article.Keywords.Trim();
    article.Title = article.Title.Trim();
    article.Summary = article.Summary.Trim();
    jFoo = JObject.Parse(line);
    paragraphs = new List<Paragraph>();
    foreach (JToken child in jFoo.Children())
    {
        var property = child as JProperty;
        if (property.Name.Contains("Paragraph"))
        {
            paragraph = new Paragraph();
            paragraph.Text = property.Value.ToString().Trim();
            paragraph.Url = article.Url;
            paragraphs.Add(paragraph);
        }
    }
    article.Paragraphs = paragraphs;
    articles.Add(article);
}
```

```
// 1. Get full article text by adding all paragraphs together
fullArticleText += paragraph.Text + "\n";
// 2. Split each paragraph into single words
List<string> words = paragraph.Text.Split(" ").ToList();
string stopwords = File.ReadAllText(_projectPath + "stopwords.txt");
foreach (string word in words.ToList())
{
    // 3. Remove whitespace and characters and convert to lower case for each word
    string normalizedWord = Regex.Replace(word.Trim().ToLower(), @"[^A-Za-z]+", "");
    words.Remove(word);
    words.Add(normalizedWord);
    // 4. Remove stopwords
    if (stopwords.Contains(normalizedWord))
    {
        words.Remove(normalizedWord);
    }
}
// 5. Find Bigram objects and add to list
List<Bigram> bigramObjects = new List<Bigram>();
try
{
    for (int i = 0; i < words.Count; i++)
    {
        string bigram = words[i] + " " + words[i + 1];
        if (fullArticleText.Contains(bigram))
        {
            Bigram newBigram = new Bigram { Label = bigram };
            bigramObjects.Add(newBigram);
        }
```

Program sequence

```
public bool AddWord(Word word)
{
    try
    {
        using (var session = Driver.Session())
        {
            return session.WriteTransaction(
                tx =>
                {
                    tx.Run("MERGE (w:Word { Label: $Label }) " +
                        "ON CREATE SET w.Label = $Label " +
                        "ON MATCH SET w.Label = $Label",
                        new { word.Label });
                    return true;
                });
        }
    }
    catch (ServiceUnavailableException)
    {
        return false;
    }
}
```

```
[16:57:12 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 51, ?? 2
[16:57:18 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:25 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:31 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:37 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:43 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:49 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:49 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ??
[16:57:49 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ?? 106, ?? 53, ?? 53, ?? 0
[16:57:50 INF] ?? b769ecbfe3754852a2b013841fcd6f0c ???????
[17:02:12 INF] ????? a8f9ada54ce6404895519f81b1c264ce ???????: 1
```

Future work

Improvements

- Dialogflow/chatbot synchronization
- GUI implementation so that there’s no more need for a console app, only one project where the crawling can be started by a user.
- “More frequent means more important”.
  → Synonyms
  → Semantics
- Stemming is missing, different versions of the same word.