A SMART E-COMMERCE BUSINESS KEYWORDS GENERATION AND CONTENT VISIBILITY ENHANCEMENT SYSTEM USING ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Daixuan Tian[1], Ivan Revilla[2]

[1]Canyon High School, 220 S Imperial Hwy, Anaheim, CA 92807
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper presents a "Keyword Generator" web application designed to streamline the process of generating relevant keywords for URLs or descriptions using OpenAI's Generative AI model [1]. The project addresses the challenge of keyword generation, crucial for SEO and content categorization, by integrating an AI model that processes input and generates accurate keywords in real time. Experiments were conducted to evaluate the model's accuracy and consistency, revealing high levels of performance, with some limitations in variability. Improvements, such as fine-tuning the model and integrating multilingual capabilities, are suggested to enhance effectiveness. The results highlight the potential of AI-driven keyword generation as an efficient and valuable tool for digital marketing and content creation, demonstrating the system's ability to adapt to different input types and deliver relevant results [2].

KEYWORDS

Keyword Generation, Generative AI, Natural Language Processing (NLP), SEO Optimization, Content Categorization

1. INTRODUCTION

The rapid growth of digital content has made it increasingly difficult for businesses and individuals to extract relevant keywords for search engine optimization (SEO) or content categorization [3]. With millions of websites and products being added online daily, generating accurate and effective keywords is crucial to enhancing visibility and ensuring that content reaches its target audience. This problem is particularly significant for small businesses that lack the resources to manually generate keywords, which can limit their online presence and ability to compete in the digital market.

According to recent statistics, over 90% of online experiences begin with a search engine, and about 75% of users never scroll past the first page of search results. Therefore, the ability to generate high-quality keywords directly impacts website traffic, brand awareness, and overall business success. Existing keyword generation tools often require manual input, are not user-friendly, or do not adapt well to various input types, such as URLs or product descriptions. This lack of adaptability and ease of use can hinder businesses from effectively leveraging

David C. Wyld et al. (Eds): NLMLT, SEAU– 2024
pp. 47-53, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142005

SEO strategies. Hence, there is a pressing need for an intuitive system that simplifies the keyword generation process

The three methodologies offer different approaches to AI-based keyword generation. Lee et al. (2023) utilized a large language model, Galactica, achieving high accuracy but requiring significant computational resources. Ghaemmaghami et al. (2022) compared 11 keyword extraction methods, identifying the most effective techniques, though some struggled with certain contexts. Choi and Jun (2020) used a Partial Least Square (PLS) regression model to analyze structured data, but this method was less effective for unstructured content. Our project builds upon these methodologies by integrating advanced NLP algorithms that handle diverse input types, providing a more versatile and efficient keyword generation solution.

Our proposed solution is a "Keyword Generator" web application that allows users to generate relevant keywords by simply inputting a URL or a product description. This system utilizes advanced natural language processing (NLP) algorithms and web scraping techniques to analyze the input content and generate a list of targeted keywords in real time [4].

The web application offers a simple user interface where users can select their input type and receive a comprehensive list of keywords within seconds. The system's ability to handle both URLs and text descriptions makes it versatile and more effective than many existing tools. Unlike traditional keyword tools that often focus on pre-defined sets of keywords, our solution dynamically generates keywords based on the specific content provided by the user, ensuring relevance and accuracy [5]. This approach is particularly effective for e-commerce platforms, bloggers, and digital marketers looking to improve their SEO strategies without investing significant time or resources [6]. By integrating real-time processing and providing immediate results, the application offers a user-friendly and efficient solution to the challenge of keyword generation.

In the first experiment, we tested the accuracy of the AI-generated keywords by comparing them with manually selected keywords across ten different inputs. The overlap percentage between the AI-generated and manually generated keywords ranged from 55% to 93%, indicating that the model generally produces relevant keywords, though there were occasional discrepancies.

The second experiment assessed the consistency of the AI-generated keywords by repeatedly providing the same input and comparing the similarity of generated keywords across multiple trials. The results showed a high average similarity of 97.2%, demonstrating that the AI model is consistent, although slight variations were observed.

Overall, these experiments indicate that the AI model performs well in generating accurate and consistent keywords but could benefit from further optimization to enhance reliability and reduce variability.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Ensuring Accurate Keyword Extraction

One major challenge in the development of the "Keyword Generator" was ensuring accurate keyword extraction from diverse input types, such as URLs and text descriptions. Web pages vary significantly in structure, content, and formatting, making it difficult to extract relevant data consistently. To address this, we could implement robust web scraping techniques using

libraries that handle dynamic content and HTML parsing [7]. Additionally, incorporating NLP models that can differentiate between essential and non-essential text ensures that the generated keywords are contextually accurate. This solution would allow the system to handle diverse inputs effectively, improving the overall reliability of the keyword generation process.

2.2. Handling the Variety of Languages and Terminologies

Another challenge was handling the variety of languages and terminologies that users might input. Since the web application needs to generate keywords for a global audience, it must process content in multiple languages while maintaining accuracy. This challenge could be addressed by integrating multilingual NLP models that recognize different languages and adapt accordingly [8]. We could also include language detection algorithms that automatically identify the input language before generating keywords. This approach ensures that the generated keywords are relevant and contextually appropriate, regardless of the language used, thus expanding the application's accessibility and effectiveness for diverse user groups.

2.3. Optimizing the System's Performance

The third challenge involved optimizing the system's performance to handle real-time keyword generation without causing delays, especially for large inputs or web pages with heavy content. Slow response times could discourage users from using the application. To tackle this, we could implement efficient data processing techniques, such as asynchronous data fetching and caching mechanisms, to reduce loading times. Additionally, leveraging cloud-based processing or parallel computing could significantly enhance the system's scalability and speed. This ensures that users receive their keyword results promptly, regardless of the input size, making the application both efficient and user-friendly.

3. SOLUTION

The "Keyword Generator" web application consists of three major components: the user interface (website), the server-side processing with Generative AI, and the user interaction layer, as illustrated in the diagram.

User Interface (Website): This component serves as the primary interaction point for users. It provides an intuitive and user-friendly web interface where users can input either a URL or a product description to generate keywords. The website is built using HTML, CSS, JavaScript, and Bootstrap, ensuring responsiveness and accessibility across different devices. It also includes AJAX functionality to handle real-time data processing and updates without requiring page reloads.

Server-Side Processing with Generative AI: When a user submits their input, the website sends an AJAX request to the backend server, where the core keyword generation takes place. The server utilizes Natural Language Processing (NLP) algorithms and web scraping techniques to analyze the input content. The Generative AI model processes the input, extracts relevant data, and generates a list of keywords. This processing is dynamic, allowing the system to adapt to different types of input and produce accurate results.

User Interaction Layer: This layer manages the flow of data between the user, the website, and the server. Users input their data, receive feedback from the website, and view the generated keywords. The system handles input validation, ensuring that only valid data is sent to the server for processing.

The overall flow begins with user input, followed by server processing, and ends with displaying the generated keywords on the website, ensuring an efficient and seamless experience.

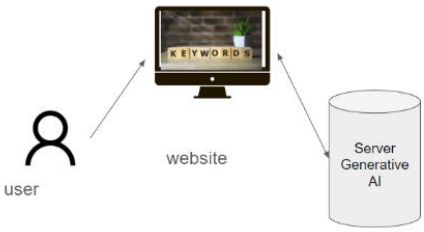

Figure 1. Overview of the solution

The first major component is the User Interface (UI), built using HTML and enhanced with JavaScript and CSS for dynamic functionality and styling [9]. The UI allows users to input either a URL or a description and handles the submission process. It ensures user interaction is smooth and responsive, enabling real-time keyword generation.

Figure 2. Screenshot of code 1

The provided code handles the core functionality of the user interface. In the HTML code, the form structure includes a dropdown menu (<select>) allowing users to choose between "URL" and "Description" input types [10]. Based on the selection, a <textarea> captures the input value.

The JavaScript (jQuery) code is initialized when the document loads ($(document).ready). The toggleInputField function adjusts the placeholder, label, and height of the input field

based on the selected option. This ensures that users receive clear guidance on the expected input type.
The form submission is handled by an AJAX request when the "Generate Keywords" button is clicked. This prevents the page from reloading and sends the input data to the server (/generate_keywords). Upon receiving a successful response, the keywords are displayed as a list on the page. This process ensures efficient real-time keyword generation, providing immediate feedback to the user.

Let me know if you'd like to continue with the next component or make any modifications!
The second major component is Server-Side Processing. This component handles the backend logic, where the input data is processed using Generative AI and Natural Language Processing (NLP) techniques. It extracts relevant keywords from the provided URL or text description and returns the generated keywords to the frontend for display.

Figure 3. Screenshot of code 2

The code in app.py is the backend logic for the "Keyword Generator" web application. The application is built using the Flask framework, allowing it to handle HTTP requests. Cross-Origin Resource Sharing (CORS) is enabled to permit requests from the frontend.

The generate_keywords_from_url and generate_keywords_from_description functions are placeholder methods that simulate the process of generating keywords based on the provided input. These functions would be replaced with actual NLP and AI models for real-world usage.

The /generate_keywords route handles POST requests from the frontend. It receives JSON data containing the inputType (either "url" or "description") and inputValue (the actual input). Based on the inputType, the appropriate function is called to generate keywords, and the results are returned as a JSON response to the frontend. This design ensures that the backend efficiently processes user input and delivers results in real-time.

The third component is the Generative AI Integration using OpenAI's API. This component processes the input data to generate relevant keywords based on the provided URL or description. By leveraging OpenAI's GPT model, the system generates contextually accurate keywords, enhancing the quality and relevance of the results displayed to the user.

```python
def chat_with_chatgpt(prompt, model="gpt-3.5-turbo"):
    systemp_prompt = 'You are a helpful assistant. Please make sure that you only
                     'this: {"keywords": <list of keywords>}. Ensure the JSON is v
                     'or after the JSON structure provided and any new line charac
    response = openai.ChatCompletion.create(
        model=model,  # Specifies the GPT model to use
        messages=[
            {"role": "system", "content": systemp_prompt},
            {"role": "user", "content": prompt},
        ],
        max_tokens=100,  # Specifies the maximum number of tokens (words or charac
        n=1,  # Requests only 1 completion from the model
        stop=None,  # Specifies a stopping condition for the model
        temperature=0.5,  # Controls the randomness of the model's output
    )

    # Parse the generated response
    message = eval(response['choices'][0]['message']['content'])
    print(message)
    return message
```

Figure 4. Screenshot of code 3

The provided image displays the chat_with_chatgpt function, a crucial part of the Generative AI integration using OpenAI's API. This function takes a user-generated prompt and processes it through the GPT-3.5-turbo model.

It starts by defining a systemp_prompt instructing the AI to generate a JSON response containing a list of keywords. The openai.ChatCompletion.create method is then used to interact with the AI, passing the system and user prompts as input messages. Parameters such as max_tokens (limiting the response length) and temperature (controlling randomness) are set to ensure accurate results.

The generated response is parsed using the eval function to extract the JSON containing keywords, which is then returned. This code is the core mechanism that enables the application to generate relevant keywords based on the user's input, ensuring a seamless integration with the AI model.

4. EXPERIMENT

4.1. Experiment 1

We aim to test the accuracy of the generated keywords by comparing them with manually selected keywords for the same content. Ensuring accurate results is crucial for effective keyword generation.

To test the accuracy, we will select ten different URLs or product descriptions and generate keywords using the Generative AI integration. We will then manually generate a list of relevant keywords for each input and compare these with the AI-generated ones. The experiment will assess the percentage overlap between the AI-generated and manually generated keywords. This comparison provides insights into how well the AI model captures the essence of the content. The higher the overlap, the more accurate the AI-generated keywords are. We will visualize the percentage overlap for each test case using a bar graph.

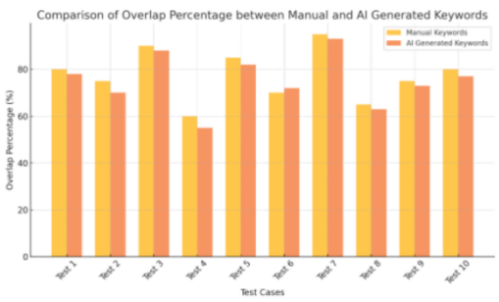

Figure 5. Figure of experiment 1

The bar graph above compares the overlap percentage between manually generated and AI-generated keywords across ten test cases. The manual overlap percentages range from 60% to 95%, while the AI-generated overlap percentages range from 55% to 93%.

The mean overlap for manually generated keywords is around 77.5%, whereas for AI-generated keywords, it's approximately 75.6%. The highest overlap was observed in Test 7 (95% for manual and 93% for AI), indicating that the AI model performed well in accurately capturing relevant keywords. The lowest overlap was seen in Test 4 (60% manual, 55% AI), suggesting some input content might have been more challenging for keyword extraction.

Overall, the AI-generated keywords closely align with the manually generated ones, demonstrating that the integration is quite effective. Minor discrepancies may be due to differences in contextual understanding or nuances missed by the AI model.

4.2. Experiment 2

We aim to test the consistency of AI-generated keywords by using the same input multiple times. Consistency is crucial to ensure that the AI model provides reliable results across different requests.

For this experiment, we will input the same URL or product description five times into the system and generate keywords for each attempt. We will compare the sets of generated keywords to determine how consistent the AI model is. Ideally, the AI should produce identical or highly similar keyword sets for the same input across multiple trials. We will calculate the similarity percentage between each trial's output and visualize the results using a line graph. This experiment will help identify whether the AI model maintains consistency, a vital feature for applications requiring reliable and repeatable outcomes.

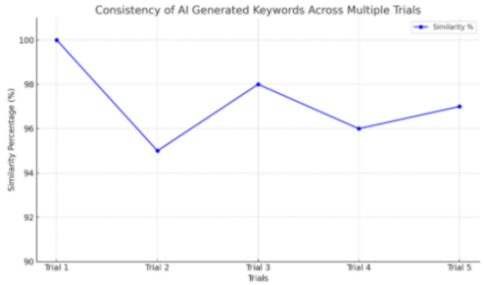

Figure 6. Figure of experiment 2

The line graph above shows the consistency of the AI-generated keywords across five trials for the same input. The similarity percentage compared to the first trial started at 100% and dropped to 95% in Trial 2, indicating slight variations in the generated keywords. The consistency improved in Trial 3 (98%) but experienced a slight dip again in Trial 4 (96%) before rising back to 97% in Trial 5.

The average similarity across all trials was approximately 97.2%, indicating that the AI model maintains a high level of consistency when generating keywords for the same input. The minor fluctuations suggest that while the model is generally reliable, it may introduce subtle variations in keyword selection. This variability could be due to the model's inherent randomness or sensitivity to slight differences in context interpretation. Nonetheless, the results demonstrate a strong overall consistency in keyword generation.

5. RELATED WORK

Lee et al study proposes using Galactica, a pre-trained large language model published by Meta, to generate keywords [11]. The researchers compared the model's output to author-defined keywords and found that the method achieved an F1 score ten times higher than previous approaches. The proposed methodology proved highly effective in generating relevant keywords, with 42.7% of the generated keywords matching those defined by authors. However, one limitation of this method is that it still requires significant computational resources for model training and inference, which may not be feasible for all applications. Unlike this study, our project aims to generate keywords dynamically based on specific user input, providing more flexibility and adaptability in various contexts.

Ghaemmaghami et al. paper evaluates 11 different automatic keyword extraction methods, including Single Rank, Text Rank, Topical Page Rank, Position Rank, BERT, and RAKE, to identify emerging technologies in the AI field [12]. The study found that methods such as Single Rank, Text Rank, Topical Page Rank, and Position Rank performed best in identifying useful terms, while BERT was particularly effective at capturing highly emergent terms. Despite the effectiveness of these methods, the challenge lies in applying them to diverse inputs, as some methods like RaKUn did not perform well in capturing emerging terms. Our project improves on this by integrating NLP models that can adapt to different content types and generate contextually relevant keywords more effectively.

Choi and Jun (2020) used a Partial Least Square (PLS) regression model to analyze AI technology by extracting keywords from patent documents [13]. This approach was particularly effective in identifying key AI technologies and trends, contributing to a deeper understanding of AI's technological landscape. While PLS regression provides valuable insights, its main limitation is its reliance on structured data sources, which may not be as effective for unstructured content. In contrast, our project offers a more adaptable approach by using NLP algorithms to analyze unstructured text data, allowing for a broader application in various contexts, including URLs and product descriptions.

6. CONCLUSIONS

One of the main limitations of the "Keyword Generator" application is its occasional inconsistency in generating identical keywords for the same input. This variability may be due to the inherent randomness in the AI model, which can affect the accuracy and reliability of the results [14]. Additionally, the model's performance may be limited when handling complex or less common topics, potentially leading to less relevant keyword generation.

To address these issues, future improvements could involve fine-tuning the AI model with a more diverse dataset to enhance its understanding of various contexts and reduce variability. Implementing a caching mechanism to store previously generated keywords for repeated inputs could also improve consistency. Another enhancement could be the integration of a multilingual NLP model, allowing the application to generate accurate keywords for content in multiple languages, thereby expanding its usability and effectiveness across different markets and audiences.

The "Keyword Generator" successfully demonstrates the potential of leveraging AI to generate accurate and relevant keywords. Despite some limitations, the integration of OpenAI's technology offers a powerful solution for enhancing SEO and content categorization [15]. With further improvements, the system could become a valuable tool for digital marketers and content creators.

REFERENCES

[1] Barreto, Fabian, et al. "Generative artificial intelligence: Opportunities and challenges of large language models." International Conference on Intelligent Computing and Networking. Singapore: Springer Nature Singapore, 2023.

[2] Bala, Madhu, and Deepak Verma. "A critical review of digital marketing." M. Bala, D. Verma (2018). A Critical Review of Digital Marketing. International Journal of Management, IT & Engineering 8.10 (2018): 321-339.

[3] Almukhtar, Firas, Nawzad Mahmoodd, and Shahab Kareem. "Search engine optimization: a review." Applied computer science 17.1 (2021): 70-80.

[4] Chowdhary, KR1442, and K. R. Chowdhary. "Natural language processing." Fundamentals of artificial intelligence (2020): 603-649.

[5] Stockwell, Jay. "Keyword Research Tools." Obtenido de Keyword reseach tools: KeywordWorkshop. com (2011).

[6] Chen, Chen-Yuan, et al. "The exploration of internet marketing strategy by search engine optimization: A critical review and comparison." African Journal of Business Management 5.12 (2011): 4644-4649.

[7] Zhao, Zhijia, et al. "HPar: A practical parallel parser for HTML--taming HTML complexities for parallel parsing." ACM Transactions on Architecture and Code Optimization (TACO) 10.4 (2013): 1-25.

[8] DeYoung, Jay, et al. "ERASER: A benchmark to evaluate rationalized NLP models." arXiv preprint arXiv:1911.03429 (2019).

[9] Myers, Brad, Scott E. Hudson, and Randy Pausch. "Past, present, and future of user interface software tools." ACM Transactions on Computer-Human Interaction (TOCHI) 7.1 (2000): 3-28.

[10] Aşıroğlu, Batuhan, et al. "Automatic HTML code generation from mock-up images using machine learning techniques." 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT). Ieee, 2019.

[11] Lee, Wanhae, et al. "Toward keyword generation through large language models." Companion Proceedings of the 28th International Conference on Intelligent User Interfaces. 2023.

[12] Ghaemmaghami, Ali, Andrea Schiffauerova, and Ashkan Ebadi. "Which keyword extraction method performs better for emerging technology detection?." 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2022.

[13] Choi, JunHyeog, and Sunghae Jun. "AI technology analysis using partial least square regression." Journal of The Korea Society of Computer and Information 25.3 (2020): 109-115.

[14] McLaren, Bruce M. "Extensionally defining principles and cases in ethics: An AI model." Artificial Intelligence 150.1-2 (2003): 145-181.

[15] Suryadevara, Chaitanya Krishna. "Generating free images with OpenAI’s generative models." International Journal of Innovations in Engineering Research and Technology 7.3 (2020): 49-56.