Mind Masters (Sittichai Chaikamol, Andrew Kuruvilla, Oren Moreno, Auny Nazmul)

ITAI 2376

May 8, 2024

Autonomous Agent Creation Report

**Integration of AI Agent Design:**

The AI agent that we created for this project tackles textual data using advanced NLP techniques. Its core function is to identify and compare keywords across various categories within the 20 newsgroups dataset. Here's a summary of the integration process:

* **Expanded Dataset**: Initially limited to the 'sci.electronics' category, the agent now analyzes all categories, enriching its analytical capabilities.
* **Refined Keyword Extraction:** We transitioned from predefined keywords to a dynamic approach. The "get\_top\_tfidf\_word" function identifies the most relevant word (highest TF-IDF score) in each category, leading to more accurate and insightful keyword extraction.
* **Automated Processing:** The agent utilizes an automated loop, efficiently processing each category. It dynamically extracts the top keyword based on TF-IDF scores and compiles a comprehensive list with keywords, scores, and corresponding categories. This automation streamlines the analysis, eliminating the need for manual intervention.

**Documentation:** The first change that was made to the original Vectorization Adventure notebook was expanding the downloaded database to include all categories from the 20 newsgroups dataset rather than just the ‘sci.electronics’ category. This was done so that the agent could access all of the categories to look for keywords. Since the goal of this agent is to find the word with the highest TF-IDF score in each category rather than finding the scores for given words, we also had to change the “extract\_keywords” function, which we renamed as “get\_top\_tfidf\_word.” The changes to this function were made after it creates the DataFrame and included changing the statements that check for the keywords in the DataFrame to instead locate the word with the highest TF-IDF score in the DataFrame then outputting that word with its TF-IDF score. This was done by first locating the index position of the word with the highest TF-IDF score in the category, then getting the word at that position and its associated TF-IDF score. After defining this function, we added a section that creates a loop that calls the function for each category in the dataset, so that each category’s keyword is output to a list along with that word’s TF-IDF score and the category that it belongs to. We then altered the section that compares keyword similarity. We made the goal of our agent to loop through the list of keywords and find the most similar keyword from a different category for each keyword. To accomplish this, the agent first checks to make sure the words do not belong to the same category, then it calculates the similarity between the current category’s keyword and every other category’s keyword, storing the highest similarity value and associated keyword.

**Team Collaboration:** Our team collaborated regularly to implement the AI agent, each member bringing their unique perspectives and knowledge to the table. Oren lead the execution and implementation phases with precision, ensuring that every aspect of the AI agent's functionality was meticulously crafted. He was also responsible for the documentation section, recording our progress and insights for future reference. Auny completed the “Integrating the AI Agent Design” section, explaining how the various components of the agent work to create a cohesive and efficient system. Sittichai authored the reflection section, providing valuable insights and analysis on our process and outcomes, helping us learn and grow from our experiences. Andrew's contribution was the team collaboration section, which explains the team members’ work as individuals and as a team. Together, we collaborated and assisted each other, dividing tasks based on our strengths and integrating them into a unified, successful implementation of the AI agent.

**Reflection:** During this project, we faced some hard challenges that tested our skills in managing and analyzing a lot of data using advanced techniques. The fetch\_20newsgroups dataset we used was big and diverse, which made it hard to handle at first. We used TF-IDF vectorization to find important words in the texts, but it took some time to learn how to use this technique effectively on such a varied dataset.

We also used the Word2Vec model to see how keywords were related to each other. This was hard because the relationships between words can be complex and hard to understand. We spent a lot of time experimenting with this model, trying to understand the output it gave us and the semantic connections it showed. It was often hard to tell if we were making progress, and we sometimes felt like we didn’t fully understand the concepts.

This project really tested our persistence and problem-solving skills. We often felt unsure and overwhelmed, and had a hard time applying what we learned in a practical way. But despite these difficulties, we kept going, learning from each setback and gradually improving our approach. This experience not only made us better at technical skills but also taught us a lot about how to tackle complex problems and keep going through challenges. We’re still learning, but this project has prepared us well for future work in AI and natural language processing

**Resources:**

<http://qwone.com/~jason/20Newsgroups/>

<https://numpy.org/doc/stable/reference/generated/numpy.argmax.html>

<https://huggingface.co/fse/word2vec-google-news-300>

<https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html>

<https://radimrehurek.com/gensim/models/word2vec.html>