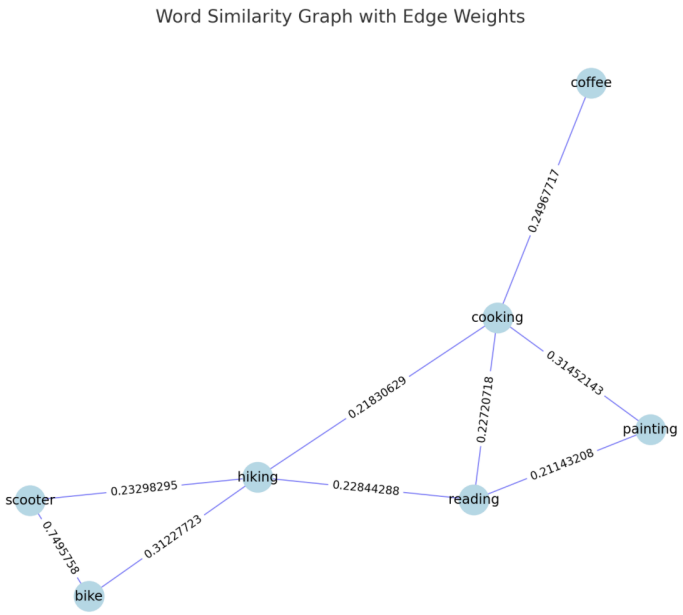
SEMANTIC RELATIONAL WEIGHT GENERATION SYSTEM APPLICATIONS FOR CONTENT METADATA

Amer K. Mriziq
INFO 202 Information Organization and Retrieval
School of Information, UC Berkeley Fall 2023

The proposed project involves developing a system for generating semantic relational weights between words, outputting a hierarchical cluster diagram and a network diagram with edge weights representing similarity. This system leverages a pretrained word2vec model for semantic analysis, where cosine similarity between word vectors is used to determine relational weights. Words with a similarity score above 0.2 undergo a linear transformation to fit a 0-1 scale, dictating the edge weights in the network diagram. The project particularly focuses on embeddings, retrieval, indexing, lexical relations, and information architecture.



The implementation involves first constructing a graph where nodes represent words and edges are added based on cosine similarity. The similarity threshold ensures that only meaningful relationships are visualized. The transformed edge weights on a 0-1 scale represent the strength of the semantic relationship, with values closer to 1 indicating stronger similarity. The resulting network diagram and hierarchical cluster diagram provide visual representations of the semantic relationships, aiding in understanding and analyzing the interconnectedness of words based on their usage and context in the word2vec model.

How to Use

1. Install the requirements
2. Run the script `python3 main.py`
3. Open the network html file found at `graph.html`

How it Works

The implementation involves three key components:

1. **Semantic Relational Graph**: I created a function, `create_similarity_graph_and_distance_matrix`, to construct a network graph and a distance matrix. In this graph, nodes represent words, and edges connect words with a cosine similarity above 0.3. I transformed the edge weights from the -1 to 1 cosine similarity scale to a more interpretable 0-1 scale.

2. **Hierarchical Cluster Diagram**: Using the `create_hierarchical_cluster_network` function, I leveraged the distance matrix to create a dendrogram that visually represents word groupings based on semantic distances.

3. **Network Visualization**: With the `visualize_graph` function, I utilized Pyvis to generate an interactive network diagram, showcasing words and their transformed similarity scores.

Hyperparameters

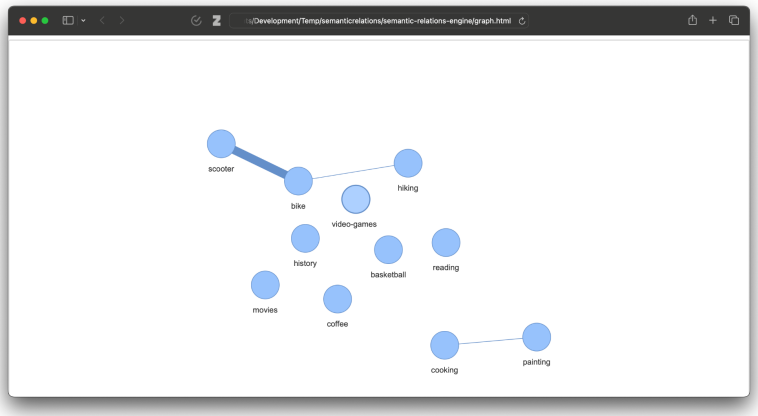


Figure 1: The network is using a high similarity threshold (>.3)

The similarity threshold is the hyperparameter in which the system operator can adjust. Adjusting the threshold significantly impacted the network's density. A lower threshold included more edges, leading to a denser graph, while a higher threshold resulted in a sparser, more

manageable network. The transformation of similarity into relational weights posed a challenge. I needed to decide whether the network should represent only positive relations or encompass all relations, including negative ones. This system's linear transformation and similarity threshold can be combined to change the underlying network to represent both positive and negative relationships.

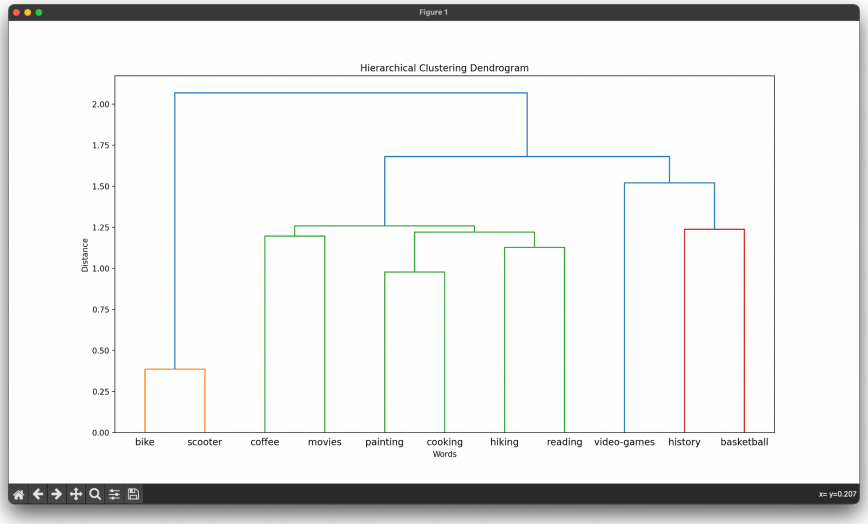


Figure 2: A hierarchical cluster dendrogram representing distance (1 - cosine similarity)

Ultimately, I found that a hierarchical cluster diagram is beneficial, especially when converted into a network diagram. This approach allowed for a more nuanced representation of word relationships, balancing network density with meaningful insights. There are certain pre-processing considerations like converting a string of words into one word, but semantic context is based on a window.

This system has potential applications in enhancing content metadata analysis, offering a nuanced understanding of word relationships.The approach of using a threshold and transforming similarity scores for visualization also contributes to the field by offering a novel way to interpret and represent word relationships graphically. This approach can be deployed into various settings for preference elicitation, such as pairwise comparison. Given a set of items, users are able to explicitly vote for individual items in a pairwise setting, while scoring underlying items they haven ot yet seen based on the relationship between their explicit preferences and the semantic relationship between those explicitly selected and those not seen yet. This is extremely useful for pair selection, or one-step look ahead technique where pairs are adaptively selected.