Stephen Xu

July 19, 2023

Network Topology Report

**Task Description**

The goal of this task is to create a mapping between buses in two different Power Flow Cases. Currently, the Python Scripts under the ‘Network Topology’ folder have moderate success at mapping 500 KV nodes between the WECC Planning Case and the CRR model. At a high level, buses are compared and scored based on three main criteria:

1. Similarity of their Bus Name (‘MIDWAY10’)
2. Similarity of the Bus ID (30051)
3. Similarity of the Network Topology

For further details, refer to the section below.

**Solution Design**

In this section, we will delve into the technical intricacies of the bus comparison process. As there are a wide variety of ways to compare buses within these electrical systems, detailed explanations are essential for fostering understanding and enabling future improvements.

The following helper functions are utilized heavily:

* We custom define a function that takes in two strings and and returns their Levenshtein Distance. This metric determines the closeness of two strings in terms of the number of operations (insertion, deletion, substitution) that it takes to convert one string to another. We use this to compare Bus Names and Bus IDs. To improve performance, vowels, spaces, and numbers and filtered out of each Bus Name.
* We borrow a function from Python’s SciPy library that implements the*Hungarian Algorithm*. This is a polynomial-time algorithm that solves the *Assignment Problem*, a task where an optimal mapping is desired between two sets in order to maximize a certain objective. This function takes two sets and and returns the optimal matching between the two sets, along with a score from 0 to 1 that represents the closeness of the matching. We use this to find the best possible mapping between the sets of buses.
* Finally, functions and find the set of all buses and edges (lines) in the neighborhood of a node at a depth . These are found through a graph Breadth-First Search Algorithm.

The first milestone is to implement a function that takes in two Buses from different power flow cases, a depth, and outputs a score in representing the similarity between Buses and . A similarity score close to indicates a strong match, while a score barely above suggests otherwise.

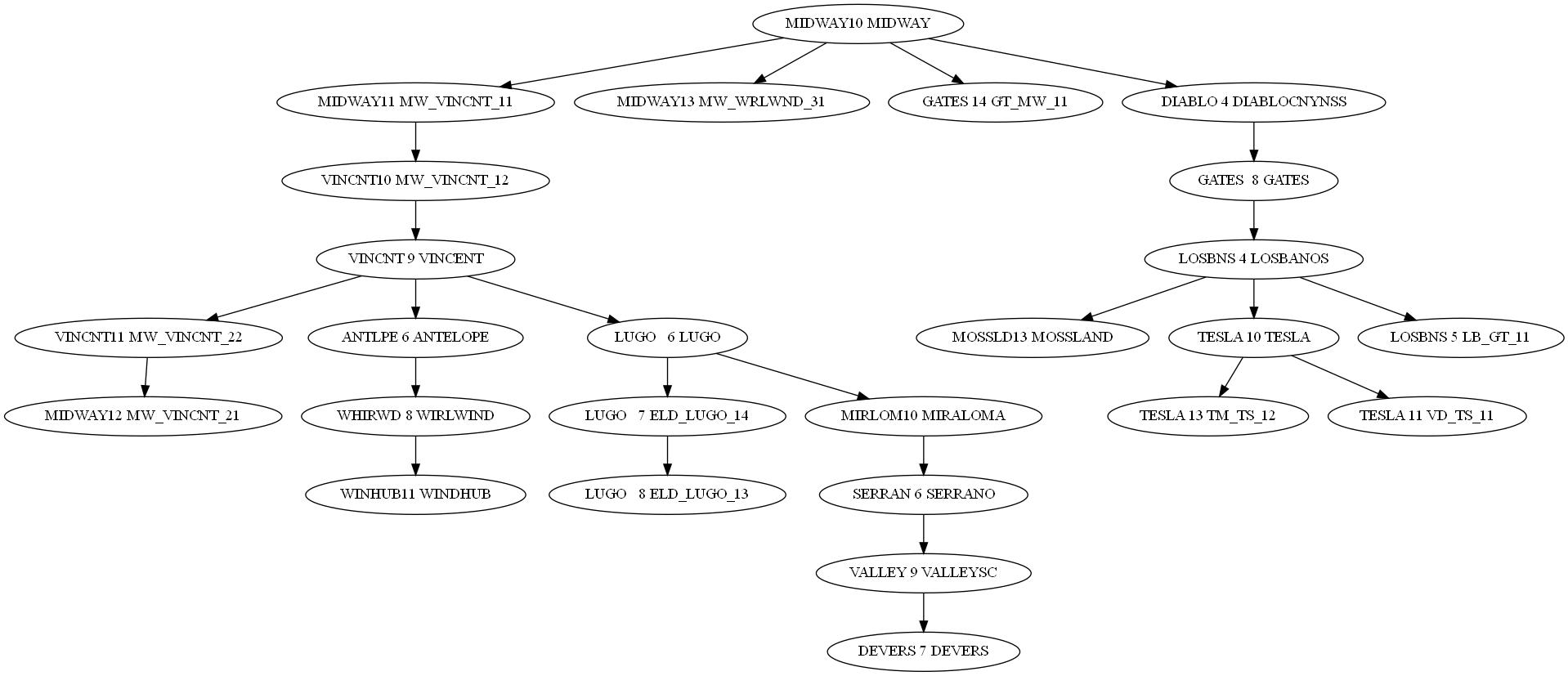
Letting and represent the bus number and bus ID of buses and, respectively, the similarity function is calculated as follows:

where represents the topology comparator function:

and are various weights that represent the relative importance of each similarity metric. For specifics on the exact values of these weights, please refer to the code. Observe that the similarity function makes a modification if , the comparison of Bus ID’s, is below a certain threshold. The reasoning for this was that distinct nodes can still be matches even if their bus IDs are completely different, so we are giving these nodes a ‘second chance’ with adjusted weight parameters.

Let and be the set of buses between the two power flow cases. In the end, the final mapping between the sets of Buses is determined by the matching given by

**Sample Output**

****