Program Parameters and Usage Instructions

Overview of Program Calls and Parameters

Program Call	Description
`python3 __init__.py graph_save_calculate plain_file encoded_file threshold <options>`	Program call for the complete process (graph creation + evaluation).
`python3 __init__.py graph_save plain_file encoded_file threshold <options>`	Program call for similarity graph creation.
`python3 __init__.py graph_load pickle_file <options>`	Program call for calculating precision values from the graph.

Positional Arguments

Argument	Type	Description
`plain_file`	String	Path to the plaintext CSV file.
`pickle_file`	String	Path to the created pickle graph file.
`threshold`	Float	Threshold for calculating the similarity graph.

Optional Arguments (for `graph_load` and `graph_save_calculate`)

Argument	Type	Description
`--init_comp_size`	Int	Minimum number of component sizes. Default: 0 (`graph_load`), otherwise 3.
`--results_path`	String	Path to save the results.
`--lsh_size_node_matching`	Int	Vector size for Hamming-LSH during node matching.
`--lsh_count_node_matching`	Int	Number of vectors for Hamming-LSH during node matching.
`--node_matching_tech`	String	Technique for node matching (possible values: `asm`, `ssm`, `mwm`).

`--weight_list`	List< *Float* >	Weights (for NF) for calculating embedding similarity between node features and embeddings. Default: `0.9, 0.8, ..., 0.1`.
`--graphwave_sg_lib`	Boolean	If set, the GraphWave implementation without edge weights is used.
`--hp_config_file`	String	Filename (without `.py`) for the configuration file in the `config` folder for hyperparameter tuning.
`--scaler`	String	Scaling technique for node features and embeddings (`minmaxscaler` or `standardscaler`).
`--num_top_pairs`	List< *Int* >	Sets of top matches to be considered for precision calculation.
`--node_matching_threshold`	Float	Threshold for cosine similarity in the bipartite graph during the node matching step.
`--vidanage_weights`	List< *Float* >	Weights for recalculating the final similarity in the bipartite graph for cosine similarity, similarity, and degree efficiency (`0.6, 0.3, 0.1`).

Optional Arguments (for `graph_save` and `graph_save_calculate`)

Argument	**Type**	**Description**
`--graph_path`	String	Path to save the pickle file with the calculated StellarGraph and the true matches.
`--remove_frac_plain`	Float	Relative proportion of records removed from the plaintext set.
`--remove_frac_encoded`	Float	Relative proportion of records removed from the encoded set.
`--record_count`	Int	Number of records considered from the dataset.
`--node_features`	String	Configuration regarding the node features to be used (`fast`, `egoneti`, `egoneti2`, `all`).
`--node_count`	Boolean	If set, the node count is used.
`--nf_scaled`	String	If set (`standardscaler` or `minmaxscaler`), the node features (for node embedding techniques) of both graphs are scaled together.
`--padding`	Boolean	If set, it is assumed that the encoded data is calculated based on padding.

`--lsh_size_blocking`	Int	Vector size for Hamming-LSH during blocking for the similarity graph.
`--lsh_count_blocking`	Int	Number of vectors for Hamming-LSH during blocking for the similarity graph.
`--ngram_attributes`	List$< String >$	Column names of the attributes for which Q-grams are calculated.
`--encoded_attr`	String	Column name for the attribute containing the encoded Bloom filter.
`--init_comp_size`	Int	Minimum number of component sizes. Default: 3.