Dependency Graph

This parameter presents a graphical representation of dependencies between topic that is being searched by the user and its related topics from the dataset.

Factors involved in constructing graph from data sets:

- **Graph Depth (D)(Level of neighbors to consider):** The depth of the graph represented for a particular topic is fixed. The depth here corresponds to the level of relevance according to the dataset (Ex: nodes at depth 1 has higher relevance to the topic than nodes at depth 2 and so on).
  This also defines the number of priority levels of tags.

- **Tag Frequency(F)(Degree of each node/Centrality):** The frequency of occurrence of each tag from all the data sets combined. Count of the number of times each tag is repeated across all queries of data sets gives its frequency.

- **Tag priority (P)(Based on degree):** Based on the graph depth D, tags are grouped into various priority levels. The total range of F values is split into D subsets.

  Ie. Size of frequency range for each priority = (F_Max – F_Min)/D

  The tags which occur most frequently (whose frequency falls in the first set) has priority 1. (Primary topics/Nodes)
  The tags with frequencies in the next range has priority 2. (Secondary nodes) and so on till D priority levels.

- **Defining the term “Dependency”** - “A tag is said to share a dependency of 1 with another tag if both the tags appear in the same query“. The dependency weight increases as the number of occurrences of both tags in same query increases.

- **Dependency Matrix:** (D-1) number of dependency matrices represent dependencies between tags of various priorities. (Ex: P1 → P2 , P2 → P3, P3 → P4 etc).
  D number of dependency matrices within each priority (Ex: P1 → P1, P2 →

P2 etc).

- **Populating dependency matrices:** For each post in the data set we identify P for each tag. Accordingly we assign dependencies in all relevant matrices.

Steps to display graph for each user query:

- **Tag identification:** Analyze user query word by word to pick only those words that match our list of tags in the graph data(that we created in previous step). We also keep track of P for each tag we pick.

  For each tag identified in the query, the graph represents only relevant dependencies from matrices, starting from the current tag going until we reach D dependency in the graph. (If there is no P1 tag in the query, graph can still represent P1 tags that have dependency on P2 tag from the query – this way we can stretch graph in both directions of dependency)

- Look and feel of a node in the graph changes according to its centrality and degree.