Suffix Tree Clustering

Definition

- A Suffix Tree: a data structure that keeps track of all n-grams of any length in a set of word strings
  - \( \text{Txt} = t_1 \ t_2 \ t_3 \ldots \ t_n \) \( \rightarrow \) string
  - \( \text{Ti} = t_i \ t_{i+1} \ldots \ t_n \) \( \rightarrow \) suffix, where \( i \geq 1 \)
  - \( \text{Ts} = t_s \ t_{s+1} \ldots \ t_{s+k} \) \( \rightarrow \) sub-string, where \( s \geq 1 \) and \( 0 \leq k \leq n - s \)
- Strength: it allows strings to be inserted incrementally in time linear to the number of words in each string.

Search
- Like a tree search

Suffix Tree

- Precise definition:
  - A suffix tree is a rooted, directed tree.
  - Each internal node has 2+ children.
  - Each edge is labeled with a non-empty sub-string of \( S \). The label of a node is defined to be the concatenation of the edge-labels on the path from the root to that node
  - No two edges out of the same node can have edge-labels that begin with the same word — compact.

Construction of a Suffix Tree

- Given a string
  - Generate all suffixes
  - Sort the suffixes
  - Split at the new character

\[
\begin{align*}
T_1 &= i \\
T_2 &= \text{ippi} \\
T_3 &= \text{issippi} \\
T_4 &= \text{mississippi} \\
T_5 &= \text{pi} \\
T_6 &= \text{ppi} \\
T_7 &= \text{sippi} \\
T_8 &= \text{si} \\
T_9 &= \text{sipp} \\
T_{10} &= \text{si} \\
T_{11} &= \text{ssipp} \\
T_{12} &= \text{ssippi}
\end{align*}
\]

mississippi
Suffix Tree Clustering (STC)

- STC is a linear time clustering algorithm that is based on a suffix tree which efficiently identifies sets of documents that share common phrases.
- STC satisfies the key requirements:
  - STC treats a document as a string, making use of proximity information between words.
  - STC is novel, incremental, and O(n) time algorithm.
  - STC succinctly summarizes clusters’ contents for users.
  - Quick because of working on smaller set of documents, incrementality
  - ...

Suffix tree clustering

- Step1- document phrasing
- Step2- phrase cluster identification
- Step3- phrase cluster merging
- Features of STC
  - Complexity
  - Incremental
  - Simple cluster definitions
  - Overlapping
  - Does not require to specify the number of clusters
Step 1 – Document Phrasing

- Each document is transformed into a sequence of words and phrase boundaries are identified
  - Perform stemming
    - Cleaning → clean, items → item
  - Mark sentence boundaries
    - Punctuation . and HTML tags
  - Maintain word ordering
    - Cat eat mouse

Step 2 – Phrase Cluster Identification

- The STC algorithm identifies all maximal phrase clusters
  1. Build a suffix tree
  2. The identification of phrases can be viewed as the creation of an inverted index of phrases
  3. The phrase clusters are scored

Suffix Tree Construction

- Each node represents a group of documents and their common phrase

<table>
<thead>
<tr>
<th>Node</th>
<th>Phrase</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>cat ate cheese</td>
<td>1,3</td>
</tr>
<tr>
<td>b</td>
<td>ate</td>
<td>1,2,3</td>
</tr>
<tr>
<td>c</td>
<td>cheese</td>
<td>1,2</td>
</tr>
<tr>
<td>d</td>
<td>mouse</td>
<td>2,3</td>
</tr>
<tr>
<td>e</td>
<td>too</td>
<td>2,3</td>
</tr>
<tr>
<td>f</td>
<td>ate cheese</td>
<td>1,2</td>
</tr>
</tbody>
</table>
Step 3 – Phrase cluster merging

- The clusters may overlap and may even be identical
- The STC algorithm merges phrase clusters with a high overlap in their document sets
- Measure the Similarity of phrase cluster
  - Given
    - $B_m$: phrase cluster; $|B_m|$: the number of the doc in $B_m$
    - $|B_m \cap B_n|$: the number of docs in $B_m$ and $B_n$
  - Define
    - Similar iff
      - $|B_m \cap B_n| / |B_m| > 0.6$ and $|B_m \cap B_n| / |B_n| > 0.6$
    - Dissimilar otherwise

Node Phrase Documents

- **Bm**: a cat ate 1,3
- **Bn**: b ate 1,2,3

| $B_m \cap B_n| / |B_m|$ | $|B_m \cap B_n| / |B_n|$ |
|-------------------------|-------------------------|
| 2/2=1                   | 2/3=0.67              |

Phrase cluster Graph

Two nodes are connected if and only if the two phrase clusters have a similarity of 1