Web Search Engines Problem Set 4
Himaja Rachakonda (hr970)
18th April 2016

1.A

Table 5.1 (Page 80)
This table is an example of Horizontal Listing. Here the table is used to compare various filtering techniques over terms, posting and tokens. Here the filter techniques are the predicates and columns are the subjects.

Table 5.3 (Page 88)
This table also looks like a horizontal listing as we are comparing between the predicates/rows “the” “computer” and “arachnocentric”.

However the structural aspect of this table slightly differs from that of the assumptions made in the paper. Here, we have a sub structure of subjects (docIds and gaps) within each predicate.

Table 9.8 (Page 176)
This table looks like a vertical listing where the columns are the attributes – word,nearest neighbors. The rows belong to instances of each of these attributes.

Table 12.3 (Page 221)
This looks like an attribute/value table where each word is an attribute and its corresponding probability is its value. Each word has a probability value.

The structural aspect of this table is that it has information within two inner attribute/value tables.

Table 14.3 (Page 276)
This looks like a matrix table where all the cells denote the time taken by the algorithm. The cells denote the time taken by kNN with and without preprocessing of data.

The information is not represented as a matrix in this case, though it could have been depicted that way.

Table 16.1 (Page 323)
This table is a vertical listing with 4 attributes and 4 instances.

Table 16.3 (Page 341)
This table is an attribute/value table type, where docId is an attribute and the text associated with it is its value.

The structure has a 4 column layout unlike the attribute/value tables mentioned in the paper.

1B

B1> Google Web tables functionality takes keyword based search query, similar to the one given in web document search engines. It fetches the pages and returns extracted relations in the URL order returned by the search engine relevant to this search query.

There are a number of ranking algorithms tht Google web tables implements to do this match, some depend on existing search engine to retrieve pages first and then extract relations from them; Others use relation specific features to score each extracted relation based on schema coherency and correlation between attributes

Challenges for extracting these tables –

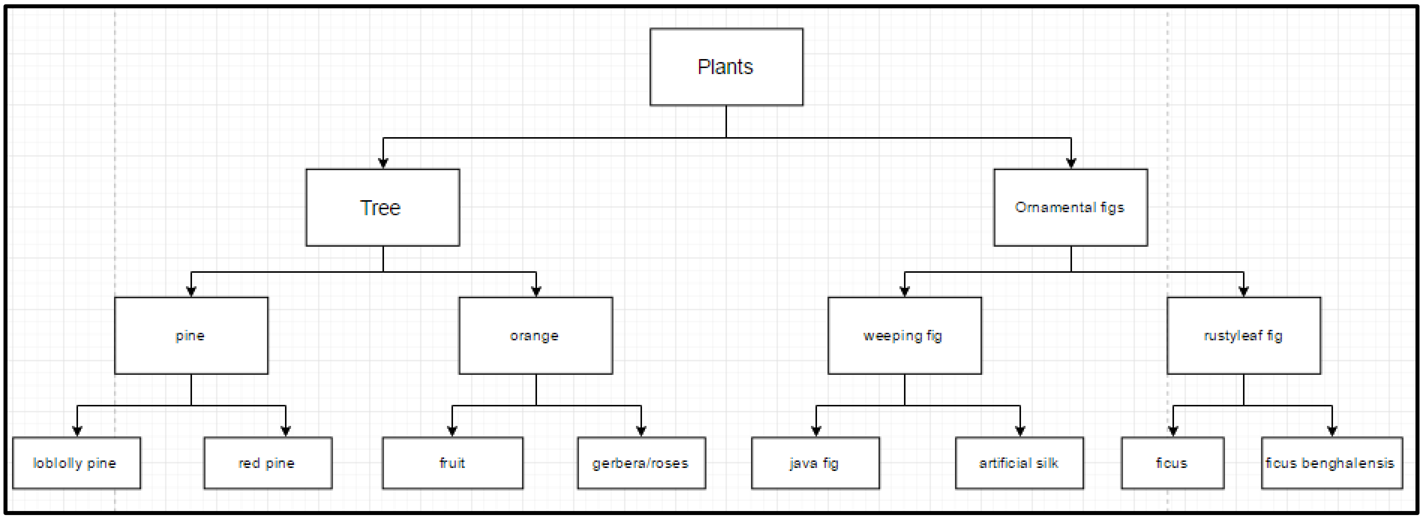
- Relations do not exist in domain-specific schema
- It is not clear which table in a page is described by a frequent word – ***(leads to false positives)***
- Attribute labels are extremely important, even if they appear infrequently
- A high quality page may contain tables of varying quality ***(leads to false positives)***
- Relations have special features that must be taken into consideration – schema elements, presence of keys, size of relation, # of NULLs

False positives can be avoided by building statistcal information about the corpus. This has shown better results when there is a large amount of data collected according to the paper.

2. A

Starting term : plants

Rule: Plural/Singular noun phrases of the form "Noun", "Noun noun" or "Adjective Noun". No proper nouns, no gerunds



Plant → tree
Plants→ornamental figs
Tree →pine
Tree →orange
Ornamental figs → weeping fig
Ornamental figs → rustyleaf fig
Pine → lolblolly pine
Pine → red pine
Orange → fruit
Orange → gerbera/roses
Weeping Fig → java fig
Weeping fig → Artificial Silk
Rustyleaf fig → ficus benghalensis
Rustyleaffig → ficus

2.B

As the depth increases we see that plants such as ornamental figs belong to the family “ficus” of plants. On going down a level further, we may conclude the “rustyfig is a ficus” belonging to that class in the plant family