Smart Crawler things to look for

Sites have
- hot deals
- daily sale
- friday sale
- bargain bin
- weekly special
- daily ad
- specials

- coupons

go to these sites

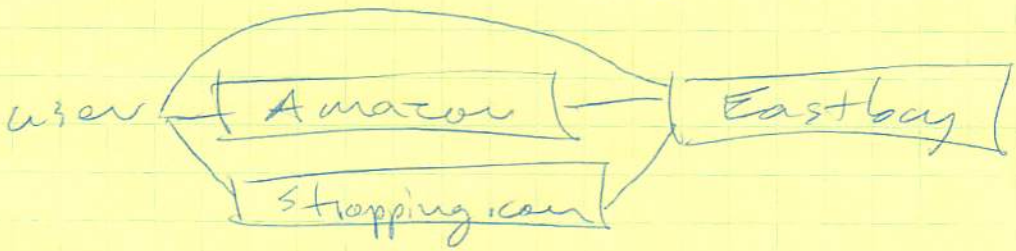
OCR the gif
or
read the url's
update each item in the db

create one page w/ all of the deals

~~Agent Site~~ Duplicater

Example Amazon has storefronts

Each storefront is a different store



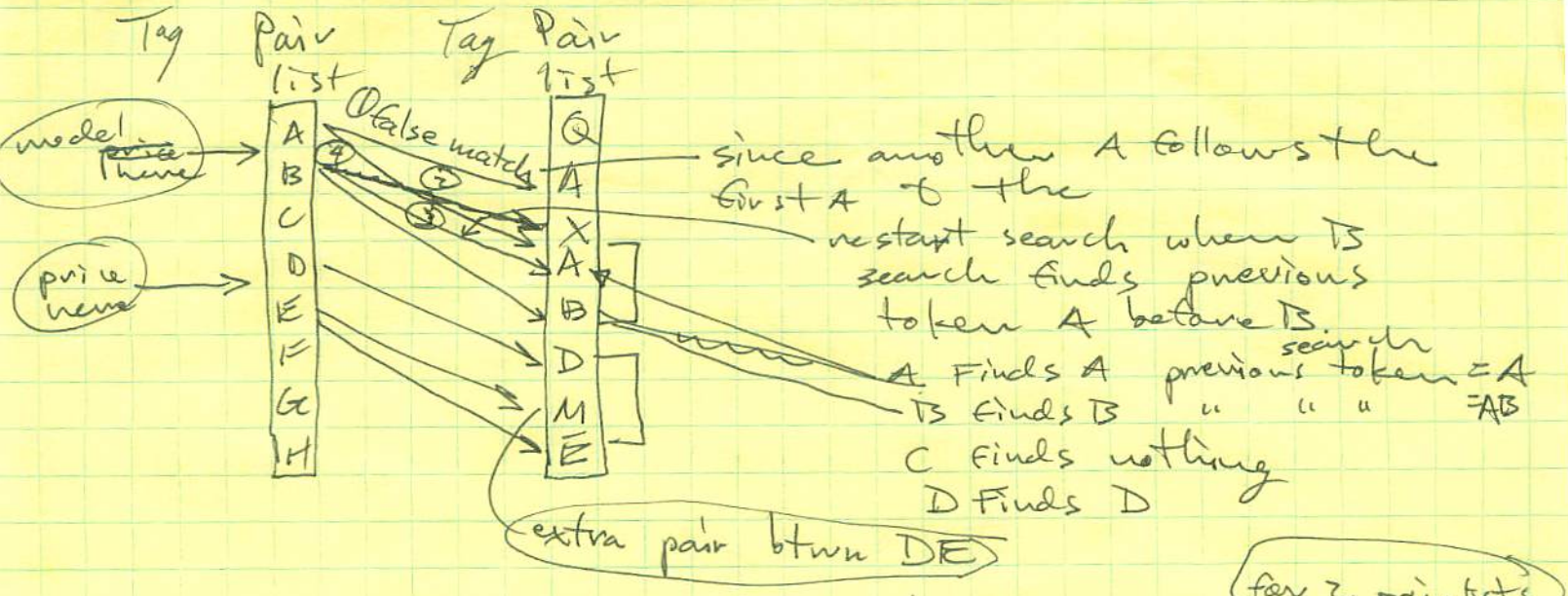
HTML "Anchor"

an " " is an HTML code fragment (report db template) which appears on each page on a site. (Our shopping site) The anchor can be one of the following:

- "click to buy"
- "add to shopping cart"
- "add to " " bag"
- "view shopping cart"
- "update cart"
- drop down menu w/ colors/size

The anchor can appear in the following formats in the HTML page

- a gif (need to OCR the gif to recognize the letters in the image)
- a script (need to interpret the script to determine the words associated w/ the script).



Identify candidate lists & eliminate pages
Create tag lists & tag pair lists
Figure out the structure of the page

E finds E
signature is (A, B, D, E)
for 2 pair lists is

Figure out which tag pairs occur in all pages

" " " " " enclose data that is being searched for.

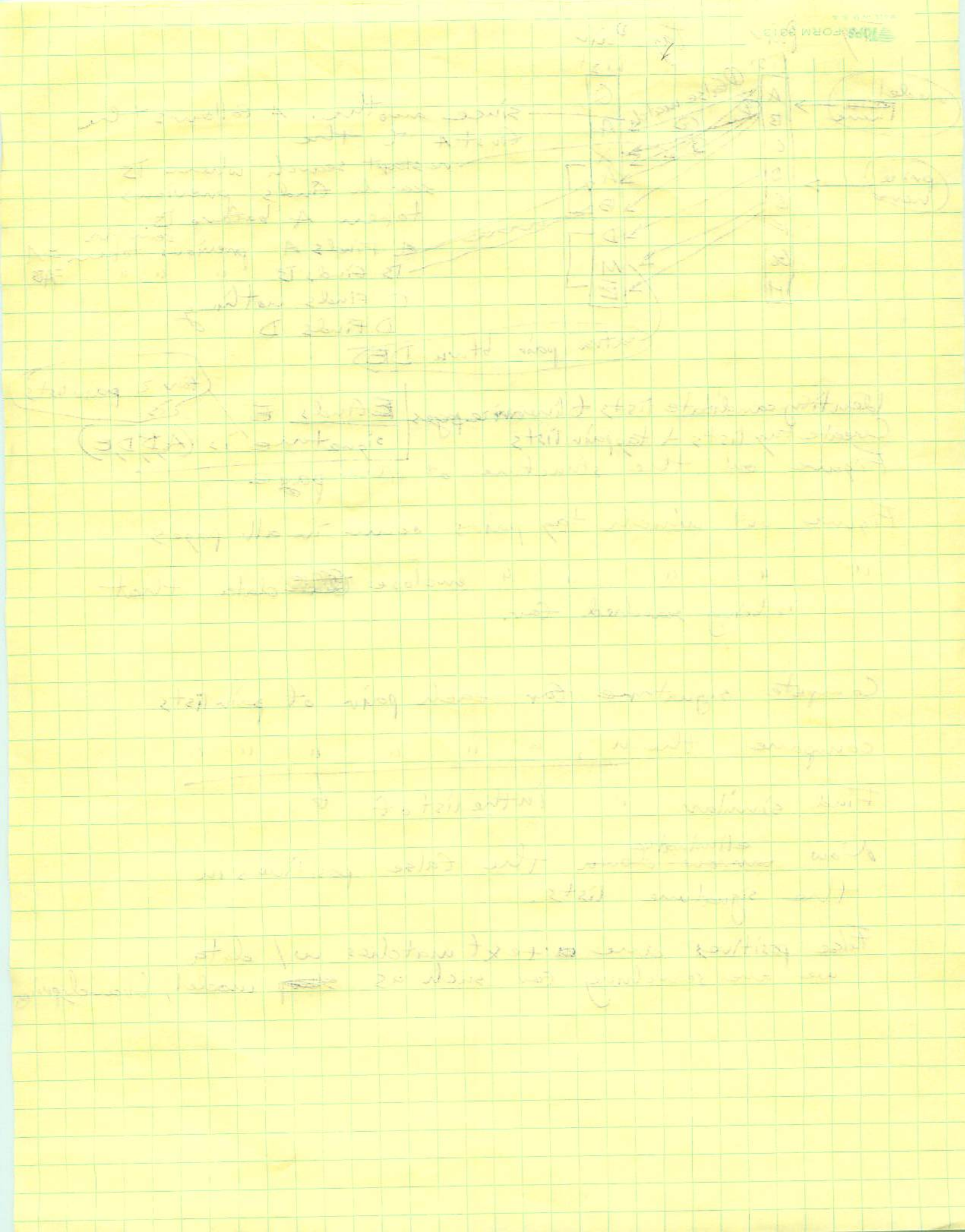
Compute signature for each pair of pair lists

compare the " 's " " " " " "

Find similar " in the list of

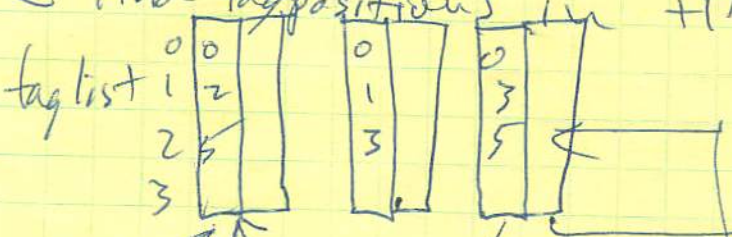
Now eliminate narrow down the false positives in the signature lists.

False positives are text matches w/ data we are searching for such as model, brand, price



(1) ~~Linearize~~ linearize HTML

(2) find tag positions in HTML



tag type
tag position in HTML matrix

(3) match tags in lists

stop tag position
start " "
) tag pair

(4) foreach tag pair match

positions in tag position vector

~~match the look a~~

examine the contents of the tag pair

if the tag pair contains

(1) currency

(2) price

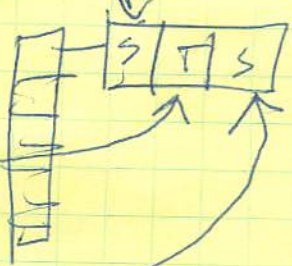
(3) description

(4) string that matches model name.

record ~~int~~ positions ↓ type

this is the signature

vector start tag position
type (s)
stop tag position



linearize the HTML

Match the tags

create a new list of indices of tags into the html list

T = tag
x = text
s = string
a = attribute

0 T
1 x
2 /T
3 T
4 a
5 a
6 /T
7 T
8 s
9 /T

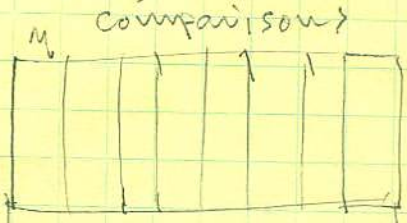
Tag list

Tag
Tag

repeat for each candidate pge

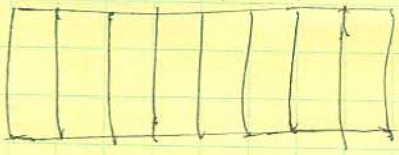
master is compared against all other lists

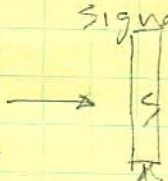
comparisons



contains 0's/1's

common tag locations





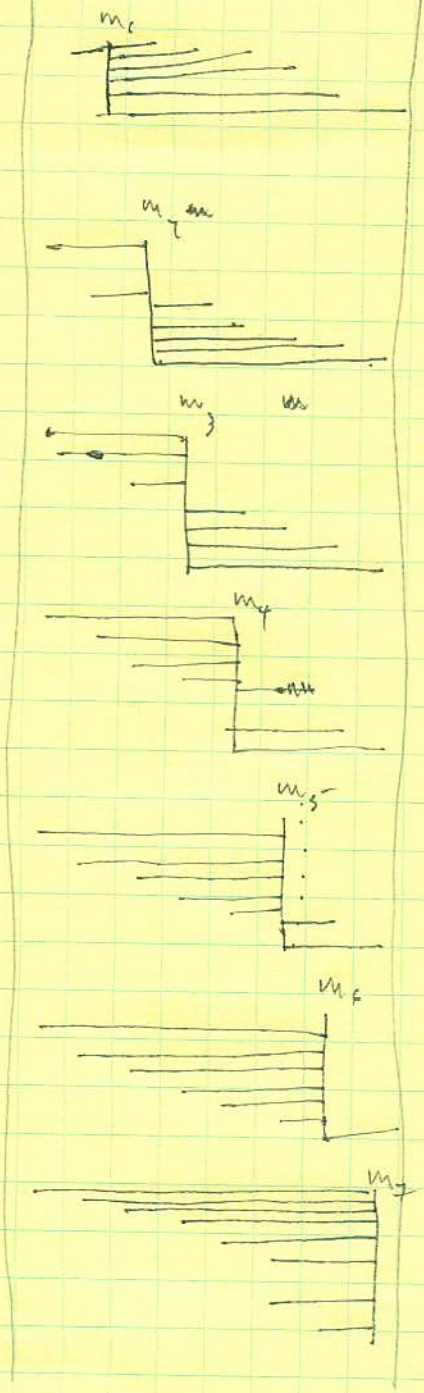
signature

list of tags

now break down the contents of the strings

now compare the extracted HTML sigs

do they match



This wants for inter merge compare (tables) an inter page compare

compare common tag locations merge the results 2 find differences?

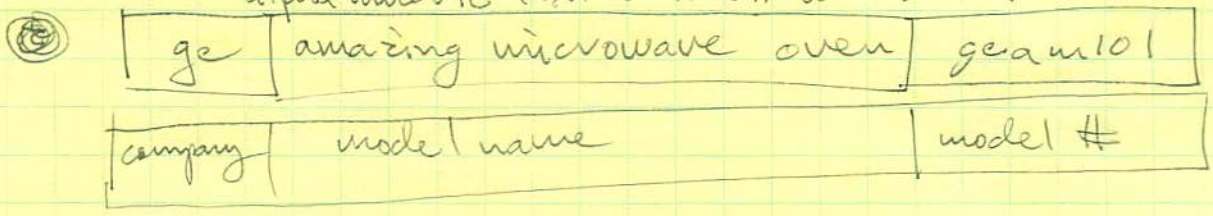
extraction of {db fieldnames and/or db values from (e.g. english) sentences / fragments / phrases
natural language

sentences / frags / phrases ~~are~~ containing

The must be separated in the

The separation process:

① find boundaries between
can be
- punctuation (#) or parens or comma or colon...
- db field names
- known db field values (company name)
- alphanumeric text which is a model #



To analyze the above string
① search for alphanumeric text
if found search in master model # dictionary
this is a short cut compare the comp. name and the model name in the db to the text string fragments to other strings on the page
② if # fails search for company name.

links

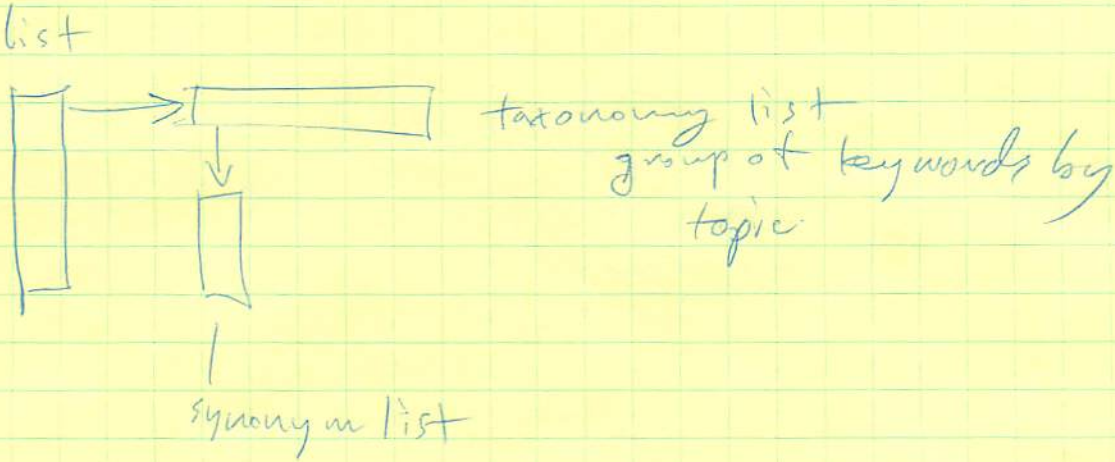
Time/Date identification
there are many formats
search for ~~the~~ " strings in the page

currency identification
there are many formats
search for the " strings in the page

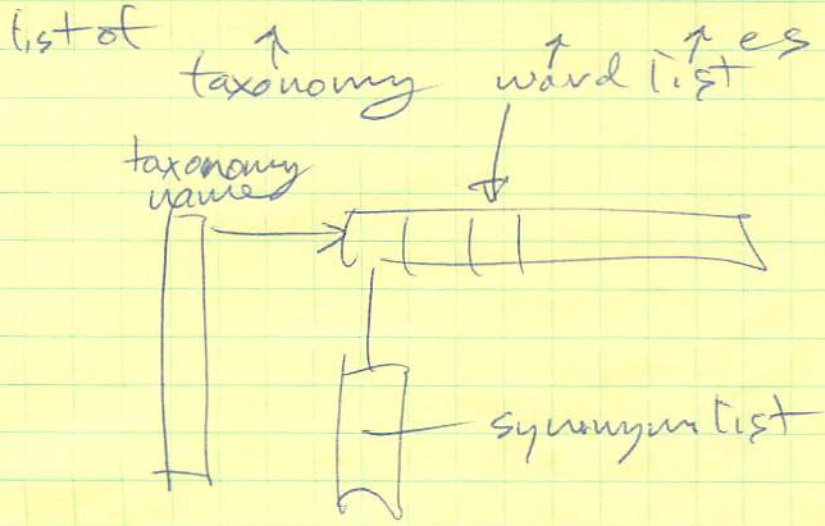
institutions ident (schools, univ's)
look up strings in institution list

Linear search of HTML

read each token in the HTML page and look it up in a taxonomy list



Patricia Trie for fast look up?



There are several different taxonomy lists

list of data base template synonym lists
(each field in the record has a field name and a "value") the field name can be many different names which is stored in a synonym list

example

fieldname
price

The field ฿100.00 ← either a formated field (currency date or time format etc.,.) or a word from a word list (e.g. product names model numbers).