Home Depot

I am thinking!!

Marc de Groot
Marta Parada Seguí
Patrick Tan
Suzanne van den Bosch
Zhuoran Liu

Thursday March 17

The data:

- *product_descriptions.csv* - description for each product
- *attributes.csv* - additional information for some products
- *train.csv* and *test.csv*

The goal:

- each test case consists of:
  - product_uid
  - product title
  - search query
- calculate relevance for each test case:
  - 1 - irrelevant.
  - 2 - Partially or somewhat relevant.
  - 3 - perfect match.

Introducing Levenshtein

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

kitten → sitting costs 3:

- kitten → sitten (substitution of “s” for “k”)
- sitten → sittin (substitution of “i” for “e”)
- sittin → sitting (insertion of “g” at the end).

The Approach:

- compare test/training search queries with Levehnstein
- use relevance of closest search query

The Future:

- include brand information
- implement SVM/SVR (“Do some actual machine learning”)

The Problem:

- lots of messy data (typos, inconsitency, etc)
- SVM/SVR requires numbers.