Home Depot

I am thinking!!

Marc de Groot [joining-the-data software developer]

Patrick Tan [baseline software developer]

Suzanne van den Bosch [GitHub manager and forum's scripts researcher]

Zhuoran Liu [SVR researcher]

Marta Parada Seguí [Liu's assistant and slides manager]

iamthinking@hellokitty.com

Thursday March 17

The data:

- Description for each product: *product_descriptions.csv*
- Additional information for some products: *attributes.csv*
- *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.

Levenshtein distance

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

Distance between *kitten* and *sitting* is 3:

- kitten $\rightarrow$ sitten (substitution of "s" for "k")
- sitten $\rightarrow$ sittin (substitution of "i" for "e")
- sittin $\rightarrow$ sitting (insertion of "g" at the end)

The Approach:

- Compare test/training search queries with Levenshtein.
- 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.

Current rank on Kaggle: 1464