Sofia Kyriazi
s1790986

The file example_matches20.txt contains twenty rows separated in two columns. The first column represents the customer description and the second one is the catalogue description.

One important obstacle in understanding the matching of the queries to the responses, was that luck of experience with the labeling of those products. The same conditions apply to the algorithm the search engine is using and also the lack of training as to which combinations and order of words should be matched. The searches are then more general (matching numbers, names, codes) and not restricted to a specific set of logical rules, thus making the search fail in certain matchings. That general set of rules that the search engine is binded to follow, produces errors. Given those two first observations the things that I detected looking at the 20 different matches are the following:

There exist many cases where the results don't match to the query, for different reasons. For example maybe the two columns matched because a common word was found, or a word was found approximately matching the letters(sub-word), but decline to a letter and then a combination of that word with a code and a prefix resulted in the query which was not what was requested. On the other hand there are some queries that match due to the sequence of words being correct and the codes of numbers matching in the right order, or the search algorithm mining the query to extract info, such as dimensions from complete description of the dimensions.

Some suggestions for improvements:

- Match lower with uppercase letters, or combine capital letters to create acronyms.
- Follow the order of numbers because they usually represent codes, but split the string to symbol or spaces separators, then add some conditions like if the prefix is the same and the numbers match look for more physical characteristics in the query.
- Having a dictionary with the terminology that concerns keywords such as brands, shapes, special code prefixes and have minimum to zero flexibility on those terms, when matching, thus to be strict and under the circumstances that the rest of the query is matching 100%.
- Matching words individually shouldn't be the case except if the codes match absolutely, or pair of brand and prefix match. Avoid to search for words in the order given, but keep the order of prefix-code for example.
- Require the id code to be present in the results, if there exists one in the query.
- Require the pairs to match as a set, prefix-code, brand-prefix. Another way, to separate the query to subqueries, set weights to the sets(ranking by importance), search for those combinations in a document, then select the most likely to match.

Interface Improvement: Maybe there is also a way to have a button in the front end communicating when the results was correct and then through a feedback mechanism improving the priority and combination of techniques that give better results.