JSON INPUT FILE

First Job: Preprocessing_job

Mapper: get_word_cat_pairs	Combiner Catcount_per_word	Reducer Catcount_reducer
Input: Amazon review json file **Output:** for each word found in the reviewtext that is not a stopword and has length > 1, the key-value pair (word: category) is yielded	**Input:** word keys with a list of categories the word appeared in (duplicates possible) **Output:** key value pairs with the word as key and a dictionary containing all categories the word appeared in as keys and the number of appearances as values.	**Input:** for each word key a list of dictionaries of category counts (words can appear more than once) **Output:** key value pairs with a word as key and a combined dictionary of total counts of appearance per category as value

Counter.txt	Intermediate.txt
Contains: ▪ **n_reviews** : counts number of mapper callse = total number of reviews ▪ **rev_per_cat:** counts number of mapper calls for each category	contains output from Preprocessing job

Second job: Chisquare_job

Mapper init: Mapper_initialize	Mapper: Chisquare_per_word	First reducer: Top_75_per_cat	Second reducer: Sort_alphabet
	Input: intermediate.txt file from the first job ,key-value pairs of the form (word: dictionary with number of appearances per category) **Output:** key-value pairs of the form (category: [word, chi^2 value])	**Input:** key-value pairs of the form (category: [word, chi^2 value]) from the mapper **Output:** key-value pairs of the form (None, [category,list of 75 words with highest chisquare in category])	**Input:** key-value pairs of the form (None, [category,list of 75 words with highest chisquare in category]) from the first reducer **Output:** for each category a string of the form: <category> firstWord: highest chi^2 secondWord: 2nd highest chi^2 …. Also a final line with all of the words in alphabetical order

Final_out.txt