The code is divided into 3 files:

1. Aspect.py
2. Clustering.py
3. Similarity.py

Aspect.py: This file generates a dictionary of probable aspects for the product. Each key is associated with a list whose first entry is the # of times that key(aspect) has been reviewed as positive and second entry is the #3 of times that key(aspect) has been reviewed as negative.
The probable aspects are nothing but nouns/noun-phrases filtered out from the reviews that have some sentiment associated with them(i.e, "the sushi was good") which is coreferenced using Spacy's dependency parser. A number of hand-crafted rules have been used for this sentiment classification.

Similarity.py: This is a helper file containing all the functions to calculate the distances between the words using the defined Model-1, Model-2 or Model-3.
getDistance1 gives distance as per Model-2(Wordnet)
getDistance2 gives distance as per Model-1(Statistical Assoc.)
getDistance3 gives distance as per Model-3(Both)

Clustering.py: This is the main file to be run. It calls a method cluster that takes the pickled dictionary(created by running the aspects.py file). I have created an array cluster where each index stores the index of the parent of the cluster to which it belongs. Also, I have maintained an array of rank to compute rank-based unions to favor the word from the bigger cluster.

The function select and select2 simply find the words with min distance and compute a union in the function itself. The function doesexist() checks if such a mergeable pair of words even exists or not.