To implement a Jaccard similarity algorithm, a mapper was written which output all possible pairs of movie IDs as its key, and one set of ratings for that movie pair as its value, e.g. a mapper output of 134, 456| 3.0, 3.5 means that the movie whos ID was 134 received a rating of 3 from one specific user, and the movie who’s ID was 456 received a rating of 3.5 from that same user.

The reducer then combined all the ratings for each pair and created two arrays of ratings, one for each movie in the pair, e.g.

Movie 1 – [3.0, 3.5, 3.5, 2.0, 5.0, 5.0, 3.0]  
Movie 2 – [4.0, 2.0, 5.0, 1.5, 1.0, 1.5, 5.0]

It was then possible to calculate the intersection of the two ratings arrays, i.e. which ratings appear in both arrays. For the example given above this would be:  
  
Intersection – [2.0, 5.0, 5.0]

From this, the Jaccard similarity of the movie pair can be calculated using the formula:

Which, for the current example, gives:

It was decided that this is not the most effective similarity metric for comparing movies based on ratings, since two movies of different genres (i.e. one comedy and one horror) which are both widely considered very good and therefore both receive a lot of (e.g.) 5.0 ratings will have a high Jaccard similarity, despite the movies not being very similar at all. Instead, a Jaccard similarity metric would be better suited to comparing different features of movies, such as genres, or comparing how many times the same words (“horror”, “thriller” etc.) appear in the descriptions of two movies.

It could also be useful for user-based similarity comparisons, e.g. finding how many movies two users have both rated, and from this finding a similar *user* to the target user, and then recommending movies to the target user which the similar user has rated but the target user has not. Since this project is focused on *item* similarity, rather than user, the cosine similarity metric will be used in favour of Jaccard going forward.