My weighting scheme (kari-kari) provides better results than the (tfidf-tfidf) weighting scheme. For every macro-averaged precision and recall point (10, 50, 100, 500) kari-kari had higher precision and recall. This is likely due to some advantages of the changes I made to the weighting scheme.

For the document weighting scheme, I updated the term frequency weighting scheme to be augmented normalized. Tfidf uses a raw term frequency. In contrast, kari uses an augmented normalized term frequency which ensures that the term frequency is between [0.5 and 1] for present terms. This provides many advantages. For the augmented normalized term frequency, a less dramatic weighting is given to terms with high frequency in a document. A term will recieve the highest rating (1) if it used the most times in this document; otherwise the term will recieve a rating of 0.5 (the presence weight) + 0.5 \* occurrences/max\_occurences. This prevents terms that are used commonly in the corpus from having extremely high weights. For example, if the term frequency weight was purely raw frequency and two documents had the word 'cat' 8 times, the term weight for 'cat' would be 8 times higher than a word that only occurs once.

I also updated the documented weighting scheme to use a normalization factor that normalizes the term weights based on the document vector length. Tfidf is not normalized in anyway for document vector length. Longer documents have a higher probability of containing a term and therefore have higher probability of being returned than a short document, however this is not desired. Theoretically, a short document could be just as valuable as a long document. To counteract this effect, each term weight in the document is divided by the Euclidean vector length. (In implementation, I divided the total similarity score by the Euclidean vector length, since this is a less expensive operation and the two results are identical). This gives less overall weight to longer documents. This is particularly important to apply for cases such as the Cranfield set where the document lengths vary greatly.

For the query weighting scheme, I did not update the term frequency factor because repeated terms in these queries is fairly rare. Leaving the term frequency factor as raw term frequency allows for large increases with terms that are repeated. In a query, I think a repeated term is more indicative of the term being significantly more important.

However, I did choose to update the collection frequency component to be the probabilistic inverse collection frequency factor. Tfidf uses the standard inverse collection frequency factor. Using a probabilistic inverse collection frequency factor gives more weight to how rare a search term is. This will give importance to the terms in a query that are more likely to be unique identifiers.

I think accounting for high term frequency and varying lengths in documents and the rarity of search terms in the query is what accounts for the drastic improvement in search results between tfidf-tfidf and kari-kari.