Here are the finding details for all three types of document vectors:

1) Raw Count: 78.46% is correct
2) TF-IDF: 78.46% is correct
3) New Variant: 83.08% is correct

The percentage comes from the fact that if the Euclidean Distance between two documents is little then the documents are said to be similar than they are to other documents with higher Euclidean Distance. The classification algorithm used in this problem is K-Nearest-Neighbors (KNN), with a weight defined at 5.

Raw Frequency: With vocabulary more than 10,000, count of terms based on frequency becomes less useful, specifically for more frequent terms such as 'the' and 'to'. These terms cannot provide contexts shared by documents. Moreover, the most useful words that can potentially be used to differentiate the documents becomes trivial as their numbers will be mostly negligible compared to more frequent words. So, we find raw frequency skewed and less discriminative.

Count (t, d)

TF-IDF: To get rid of terms that are less important such as 'it' and 'to', we use TF-IDF weighting. For TF part in TF-IDF, we take the $\log_{10}$ of the count of terms in document and add 1. The intuition is that 100 times occurrence of terms such as 'it' and 'to' doesn't make these words 100 times more important.

To emphasize less frequent terms, we rely on IDF part in TF-IDF by taking log of (N/df_t), where N is the total number of documents and df_t is the document frequency that provides the number of documents t occurs in. Thus, the fewer documents in which term t occurs, the height its weight.

Overall, the product of TF-IDF weights is certainly better than other variants of document vectors because it chose to ignore the most frequent terms and emphasize more on less frequent terms.

log (count (t, d) + 1) * log (N/df_t + 1)

New Variant of TF-IDF:

In the new variant of TF-IDF, an importance of term becomes insignificant if it appears more than 70% of the time in the document. The intuition is more frequent terms in a document are trivial and hold less significance compared to terms that appears little.

log (count (t, d) + 1) * log (N/ (important term? = [either 1 or 0] + 1)

However, this new variant will fail if the more frequent terms in a document are important. For instance, a document about infants would contain a term 'baby' or 'infant' more than 70% of the time, and in such scenarios, this approach would fail to provide us an estimation on document similarity.