Option A:

1. Vectorization: N-gram or K-shingle, (maybe also bag of words or TF-IDF)
2. Dimension reduction: min-hashing, (maybe also Principle Component Analysis or Autoencoder)
3. Finding candidate pairs: Locality –sensitive –hashing (LSH)
4. Similarity computing methods: cosine similarity or Jaccard Similarity.

Properties:

1. The input of the Locality Sensitive Hashing is binary array if using K-shingle methond, so the vectorization method will be limited to the algorithms that can produce binary vectors.

But if using different hash function, it seems this method can also process real valued vectors.

1. The speed of LSH is seems to be O(log(n)).
2. Can adapt to vector with large dimension. (>100?)
3. May miss some nearest vectors (can control).

Option B:

1. Vectorization: bag of words or TF-IDF
2. Dimension reduction: Principle Component Analysis or Autoencoder
3. Nearest Similarity Search: KD-tree search

Properties:

1. The input can be real numbers.
2. The speed of KD-tree search seems to be O(log(n)), but needs to be verified.
3. Can find similar vectors within a range using KD-tree search.
4. Vector dimensions cannot be large (n >> 2k). K as the dimension. So, for 30000 documents, the dimension should be less than 14.
5. May miss some nearest vectors (can control).

Videos:

1. LSH:
2. <https://www.youtube.com/watch?v=dgH0NP8Qxa8>
3. <https://www.youtube.com/watch?v=Arni-zkqMBA>
4. <https://www.youtube.com/watch?v=h21irtHDsBw>
5. KD-Tree:
6. <https://www.youtube.com/watch?v=Y4ZgLlDfKDg>
7. <https://www.youtube.com/watch?v=Z4dNLvno-EY>