We are used k modes clustering (Huang, 1997), which is an extension of k means for categorical data:

1. a simple matching dissimilarity measure is used for categorical variables
2. the means of clusters are replaced by modes
3. a frequency based method is used to find the modes

The main steps in the algorithm:

(Huang, Extensions to the k-means for clustering large data sets with categorical values, 1998):

1. k initial modes are selected, one for each cluster – we used the method proposed by Huang in (Huang, 1997)
2. an object is allocated to the cluster closest to it by the dissimilarity measure
3. After all objects have been allocated (as in 2), each object’s dissimilarity measure with respect to each mode is re-tested, and objects are re-allocated, if necessary
4. We repeat 3 until the clusters have stabilized – indicated by no reallocation in a complete testing cycle.

We used the Python implementation of the k-modes algorithm (package *kmodes*)

We clustered Documents based on the following categorical attributes (all anonymised by Outbrain in the training data set). These were selected based on a trade-off between computation cost and improvement in the performance of the model

1. source\_id (the host of a web page)
2. publisher\_id (the publisher of the content)
3. category\_id (the broadest classification of the subject matter of the Document’s contents)
4. topic\_id (the next level classification of the Document’s contents)

We ran the k modes algorithm for various values of k (5, 10, 20) and found significant performance improvement with k = 5 (as indicated by our model’s performance on kaggle.com)