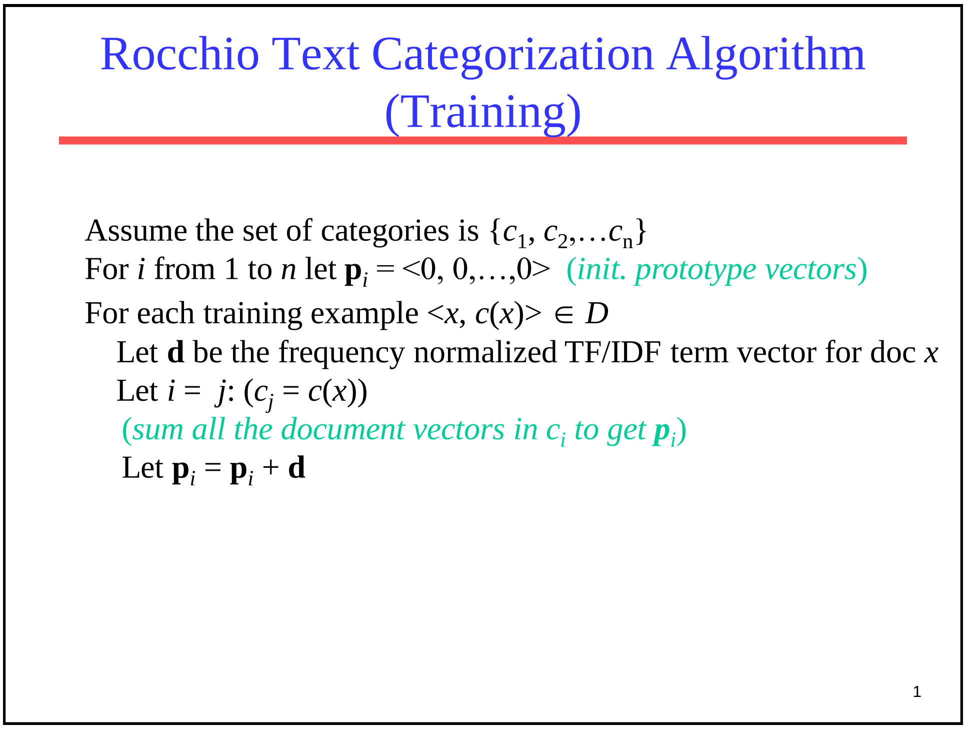
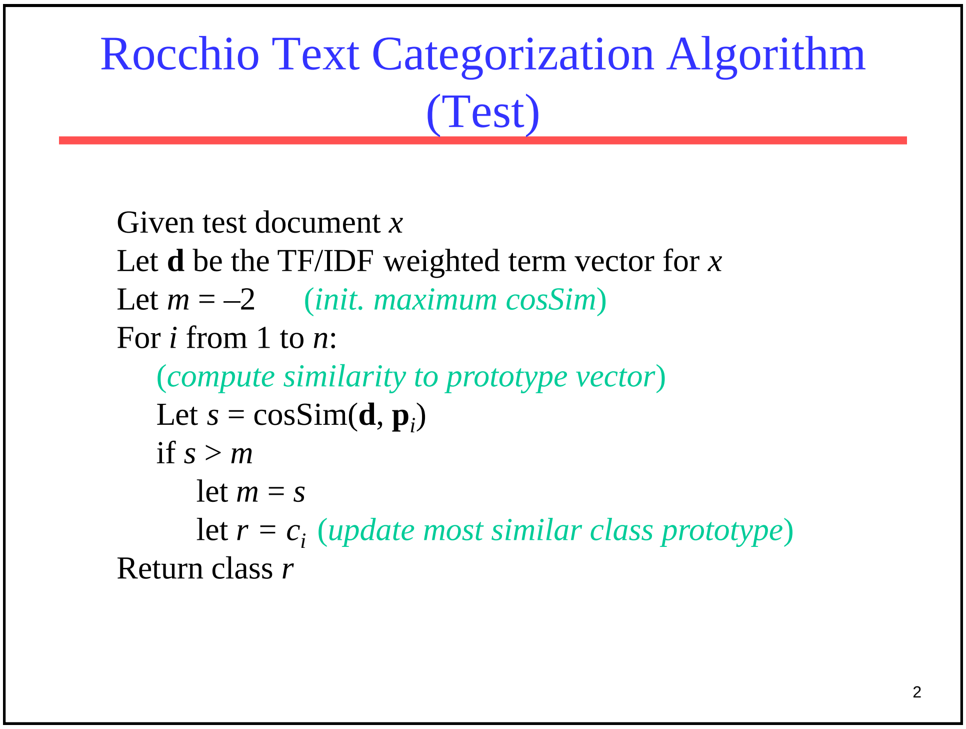


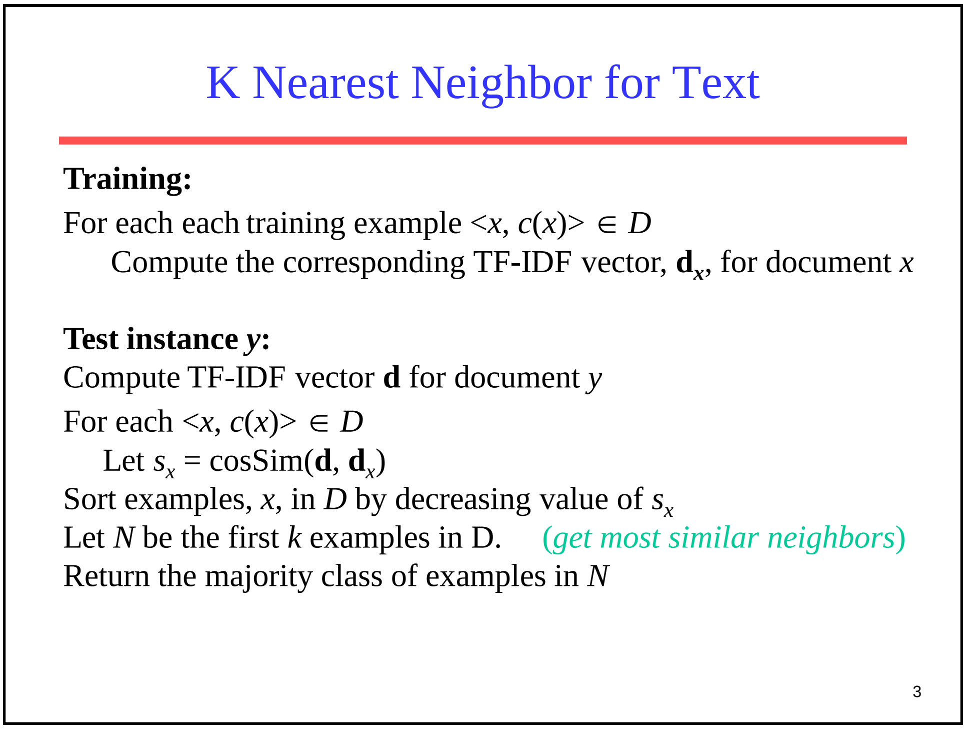
Rocchio Text Categorization Algorithm (Training)

```
Assume the set of categories is $\{c_1, c_2, \ldots c_n\}$
For $i$ from 1 to $n$ let $\mathbf{p}_i = <0, 0,\ldots,0>$  (init. prototype vectors)
For each training example $<x, c(x)> \in D$
    Let $\mathbf{d}$ be the frequency normalized TF/IDF term vector for doc $x$
    Let $i = j: (c_j = c(x))$
    (sum all the document vectors in $c_i$ to get $\mathbf{p}_i$)
    Let $\mathbf{p}_i = \mathbf{p}_i + \mathbf{d}$
```



Rocchio Text Categorization Algorithm (Test)

```
Given test document $x$
Let $\mathbf{d}$ be the TF/IDF weighted term vector for $x$
Let $m = -2$      (init. maximum cosSim)
For $i$ from 1 to $n$:
    (compute similarity to prototype vector)
    Let $s = \text{cosSim}(\mathbf{d}, \mathbf{p}_i)$
    if $s > m$
        let $m = s$
        let $r = c_i$ (update most similar class prototype)
Return class $r$
```



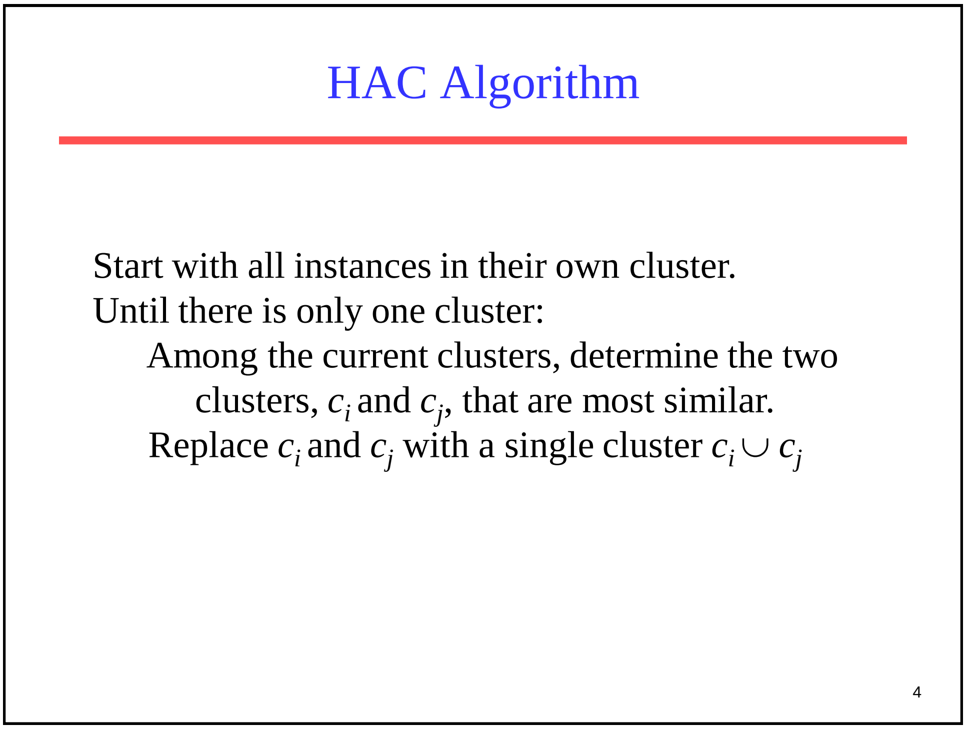
K Nearest Neighbor for Text

Training:

```
For each each training example $<x, c(x)> \in D$
    Compute the corresponding TF-IDF vector, $\mathbf{d}_x$, for document $x$
```

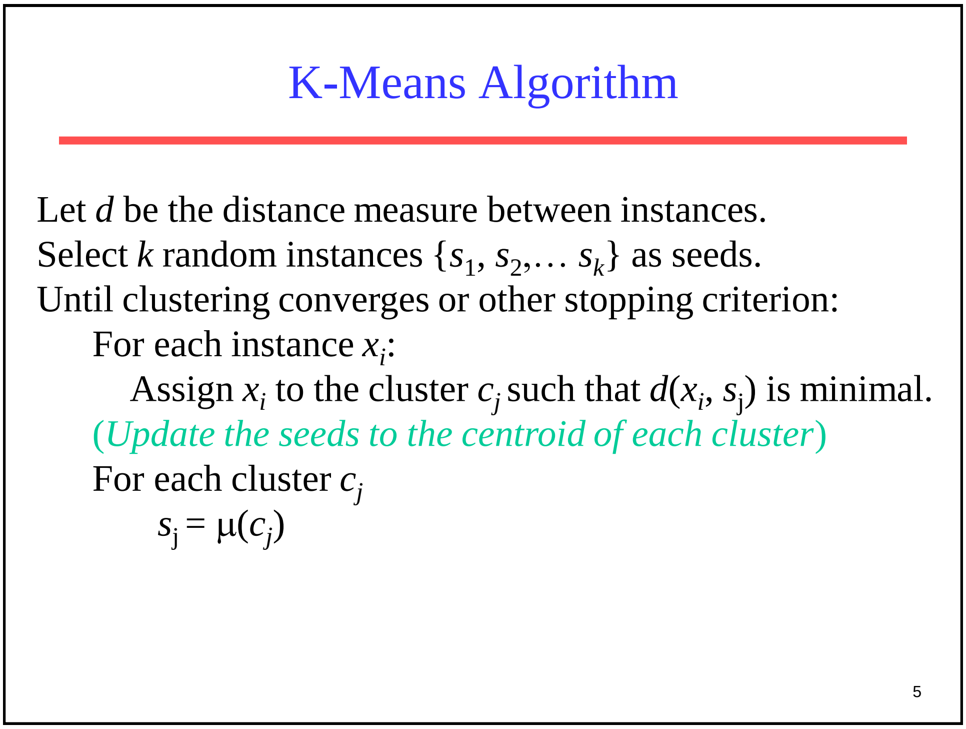
Test instance *y*:

```
Compute TF-IDF vector $\mathbf{d}$ for document $y$
For each $<x, c(x)> \in D$
    Let $s_x$ = cosSim($\mathbf{d}$, $\mathbf{d}_x$)
Sort examples, $x$, in $D$ by decreasing value of $s_x$
Let $N$ be the first $k$ examples in D.     (get most similar neighbors)
Return the majority class of examples in $N$
```



HAC Algorithm

```
Start with all instances in their own cluster.
Until there is only one cluster:
    Among the current clusters, determine the two
        clusters, $c_i$ and $c_j$, that are most similar.
    Replace $c_i$ and $c_j$ with a single cluster $c_i \cup c_j$
```



K-Means Algorithm

```
Let $d$ be the distance measure between instances.
Select $k$ random instances $\{s_1, s_2, \ldots s_k\}$ as seeds.
Until clustering converges or other stopping criterion:
    For each instance $x_i$:
        Assign $x_i$ to the cluster $c_j$ such that $d(x_i, s_j)$ is minimal.
    (Update the seeds to the centroid of each cluster)
    For each cluster $c_j$
        $s_j = \mu(c_j)$
```