1. Summation
\[ s = \sum_{i=0}^{n-1} i \]

2. Program of that summation

public class Sumup {
    int n=10;
    int s = 0;
    int i = 0;
    while (i <= (n-1)) {
        s = s+i;
        i = i + 1;
    }
}

Or

public class multup2 {
    int n=10;
    int s = 0;
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9,10,11};
    for (i = 1; i <= n; i++)
        s = s + (a[i] * a[i+1]);
}

3. Summation

\[ S = \sum_{i=1}^{n} a_i \cdot a_{i+1} \]

```java
public class multup {
    int n=10;
    int s = 0;
    int i = 1;
    int a[] = {0,1,2,3,4,5,6,7,8,9,10,11};
    while (i <= n) {
        s = s + (a[i] * a[i+1]);
        i = i + 1;
    }
}
```

or...

```java
public class multup2 {
    int n=10;
    int s = 0;
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9,10,11};
    for (i = 1; i <= n; i++)
        s = s + (a[i] * a[i+1]);
}
```
4. Simple tf * idf

\[ w_{ik} = tf_{ik} \times \log \left( \frac{N}{n_k} \right) \]

- \( T_k \) = term \( k \) in document \( D_i \)
- \( tf_{ik} \) = frequency of term \( T_k \) in document \( D_i \)
- \( idf_k \) = inverse document frequency of term \( T_k \) in \( C \)
- \( N \) = total number of documents in the collection \( C \)
- \( n_k \) = the number of documents in \( C \) that contain \( T_k \)

\[ idf_k = \log \left( \frac{N}{n_k} \right) \]

**Inverse Document Frequency**

For a collection of 10000 documents (\( N = 10000 \))

- \( \log \left( \frac{10000}{10000} \right) = 0 \)
- \( \log \left( \frac{10000}{5000} \right) = 0.301 \)
- \( \log \left( \frac{10000}{20} \right) = 2.698 \)
- \( \log \left( \frac{10000}{1} \right) = 4 \)

5. Similarity Measures

**Simple matching (coordination level match)**

\[ \frac{|Q \cap D|}{2 \times |Q| + |D|} \]

- **Dice’s Coefficient**

\[ \frac{|Q \cap D|}{|Q \cup D|} \]

- **Jaccard’s Coefficient**

\[ \frac{|Q \cap D|}{|Q \cup D|} \]

- **Cosine Coefficient**

\[ \frac{|Q \cap D|}{\sqrt{|Q|} \times \sqrt{|D|}} \]

- **Overlap Coefficient**

\[ \frac{|Q \cap D|}{\min(|Q|, |D|)} \]
6. tf*idf Normalization

\[ w_{ik} = \frac{(tf_{ik} \cdot \log(N / n_k))}{\sqrt{\sum_{k=1}^{t} ((tf_{ik})^2) \cdot ([\log(N / n_k)]^2)}} \]

7. Vector Space Similarity

\[ sim(D_i, D_j) = \sum_{k=1}^{t} w_{ik} \cdot w_{jk} \]

8. Vector Space Similarity Measure

\[ D_l = w_{d_{i1}}, w_{d_{i2}}, ..., w_{d_{il}} \]
\[ Q = w_{q1}, w_{q2}, ..., w_{qt} \]

If term weights are normalized: \( sim(Q, D_l) = \sum_{j=1}^{t} w_{qj} \cdot w_{dj} \)

Otherwise normalization and the similarity comparison are combined:

\[ sim(Q, D_l) = \frac{\sum_{j=1}^{t} w_{qj} \cdot w_{dj}}{\sqrt{\sum_{j=1}^{t} (w_{qj})^2 \cdot \sum_{j=1}^{t} (w_{dj})^2}} \]
9. Computing Similarity Scores

![Graph showing similarity scores and angles]

\[ D_1 = (0.8, 0.3) \]
\[ D_2 = (0.2, 0.7) \]
\[ Q = (0.4, 0.8) \]
\[ \cos \alpha_1 = 0.74 \]
\[ \cos \alpha_2 = 0.98 \]

What's Cosine Anyway?

![Cosine and sine graphs]

Cosine vs. Degrees

![Graph showing cosine values vs. degrees]
Computing a Similarity Score

Say we have query vector $Q = (0.4, 0.8)$
Also, document $D_2 = (0.2, 0.7)$
What does their similarity comparison yield?

$$
sim(Q, D_2) = \frac{(0.4 \cdot 0.2) + (0.8 \cdot 0.7)}{\sqrt{[(0.4)^2 + (0.8)^2][0.2]^2 + (0.7)^2]}} = \frac{0.64}{\sqrt{0.42}} = 0.98
$$

Vector Space Matching

$D_1 = (d_{11}, w_{d11}; d_{12}, w_{d12}; \ldots; d_{1n}, w_{d1n})$
$Q = (q_{11}, w_{q11}; q_{12}, w_{q12}; \ldots; q_{1m}, w_{q1m})$

$$
sim(Q, D_1) = \frac{\sum_{j=1}^{t} w_{qj}w_{dj}}{\sqrt{\sum_{j=1}^{t} (w_{qj})^2 \sum_{j=1}^{t} (w_{dj})^2}} = \frac{0.64}{\sqrt{0.42}} = 0.98
$$

$D_2 = (d_{21}, w_{d21}; d_{22}, w_{d22}; \ldots; d_{2n}, w_{d2n})$

$$
sim(Q, D_2) = \frac{(0.4 \cdot 0.2) + (0.8 \cdot 0.7)}{\sqrt{[(0.4)^2 + (0.8)^2][0.2]^2 + (0.7)^2]}} = \frac{0.64}{\sqrt{0.42}} = 0.98
$$

$$
sim(Q, D_1) = \frac{0.56}{\sqrt{0.58}} = 0.74
$$