Calculating a **TF-IDF**

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

	Doc 1	Doc 2	...	Doc *n*
Term(s) 1	12	2	...	1
Term(s) 2	0	1	...	0
...	...	...	...	
Term(s) *n*	0	6	...	3

- **TF**(n-gram ***w***) = (Number of times n-gram ***w*** appears in a document) / (Total number of n-grams in the document).

- **IDF**(n-gram ***w***) = $\log_e$[(Total number of documents in the corpus) / (Number of documents with n-gram **w** in it)]

Calculating a **TF-IDF**

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

	Doc 1	Doc 2	...	Doc *n*
Term(s) 1	12	2	...	1
Term(s) 2	0	1	...	0
...	...	...	...	
Term(s) *n*	0	6	...	3

- Let's consider an example corpus of just **unigrams**.

- Consider some document (e.g., a tweet!) containing 100 unigrams, where the unigram ***cat*** appears 3 times.
  - The term frequency **TF** for ***cat*** is then (3 / 100) = 0.03.
  - Now, assume we have 10 million documents in our **corpus**.
  - The unigram ***cat*** appears in one thousand of these documents.
  - Then, the inverse document frequency **IDF** is calculated as $\log_e(10{,}000{,}000 / 1{,}000) = 4$.
  - Thus, the **TF-IDF** is the product of these quantities: 0.03 * 4 = 0.12.

- Let's consider an example corpus of just **unigrams**.

- Consider some document (e.g., a tweet!) containing 100 unigrams, where the unigram **cat** appears 3 times.
  - The term frequency **TF** for **cat** is then (3 / 100) = 0.03.
  - Now, assume we have 10 million documents in our **corpus**.
  - The unigram **cat** appears in one thousand of these documents.
  - Then, the inverse document frequency **IDF** is calculated as $\log_e(10{,}000{,}000 / 1{,}000) = 4$.
  - Thus, the **TF-IDF** is the product of these quantities: 0.03 * 4 = 0.12.