List two differences between indexing the Web and indexing a well-controlled collection.

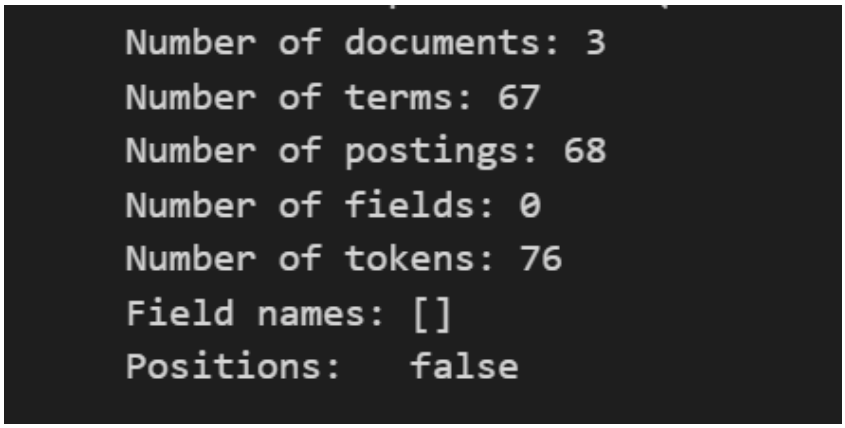
1. A well-controlled collection has a well-organized and consistent format and structure, while the Web has a lot of content with varying structures.
2. A well-controlled collection also has minimal duplications of values, whereas the Web can have a lot of duplication of documents.

Index the following three 'documents' and report the index statistics using PyTerrier.

"Some years ago - never mind how long precisely - having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world."

"It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair."

"When he was nearly thirteen, my brother Jem got his arm badly broken at the elbow. When it healed, and Jem's fears of never being able to play football were assuaged, he was seldom self-conscious about his injury. His left arm was somewhat shorter than his right; when he stood or walked, the back of his hand was at right angles to his body, his thumb parallel to his thigh. He couldn't have cared less, so long as he could pass and punt."



Explain intuition behind TFIDF in your own words. Start by stating formulations for TF and IDF and their individual purposes. How does BM25 connect to TFIDF? Respond in half a page to a full page.

TF = term frequency is **how often a term appears in an individual document**. The formula is $TF = tf/(k1 + tf)$, where TF is the frequency of the term in the document and k1 is a tuning parameter that controls how much weight is given to term frequency. The idea behind it is Term frequency for a document is the number of times the term appears in a **specific document**

divided by the number of terms in the document and to determine how relevant the term is to the document.

.

IDF = **how common a term is in a collection of documents**. The formula is $IDF = log((N + n + 0.5)/(n + 0.5))$, where N is the total number of documents in the collection and n is the number of documents containing the term. The idea is to narrow down and to put more weight on rare terms, so that terms that are vital to a document hold more weight.

TF-IDF combines the ideas of TF and IDF to find a balance for the importance of an individual term in a document and to distinguish between documents. Essentially, it tries to find a balance of **how important a term is within a document (TF)** and **how informative or unique it is across documents (IDF)**. A good example would be if I searched for Pizza, I would get a collection of results which have the word pizza because of the IDF idea, and it would be listed by the frequency of the term pizza in a document. So the document with the highest frequency of the word pizza would be at the top.

BM25 is an improvement on top of the TF-IDF idea. It adds 3 more thoughts for how a document is distinguished from another.

1. Accounts for the fact that mentioning a term more times in a document doesn't automatically make that document better than another.
2. Penalizes longer documents so as to not overlook shorter and more compact documents.
3. Having more flexibility in the calculation to allow for adjustments within a collection of documents.

TF-IDF, from my understanding is simpler and more effective for shorter retrieval tasks, but BM25 is better for working with large collections of diverse documents.

Index the set of documents (given in two files) from New York Times. Report statistics about this index. Run a query 'explosive' using TFIDF and BM25 and report the top 10 results for both.

data for BM25:

	qid	docid	docno	rank	score	query
0	q1	195	NYT20000131.0004	0	9.548229	explosive
1	q1	192	NYT20000131.0001	1	8.214721	explosive
2	q1	441	NYT20000131.0259	2	6.730696	explosive
3	q1	714	NYT20000131.0551	3	5.952960	explosive
4	q1	145	NYT20000130.0162	4	5.700672	explosive
5	q1	164	NYT20000130.0185	5	5.029963	explosive
6	q1	104	NYT20000130.0119	6	4.989011	explosive
7	q1	53	NYT20000130.0059	7	4.944284	explosive
8	q1	351	NYT20000131.0166	8	4.781394	explosive
9	q1	358	NYT20000131.0173	9	4.781394	explosive

data for TFIDF:

C:\Users\Fortu\AppData\Local\Temp\ipykernel_26680\206708198

```
tf_idf = pt.BatchRetrieve(index, wmodel="TF_IDF")
```

	qid	docid	docno	rank	score	query
0	q1	195	NYT20000131.0004	0	5.367714	explosive
1	q1	192	NYT20000131.0001	1	4.618058	explosive
2	q1	441	NYT20000131.0259	2	3.783786	explosive
3	q1	714	NYT20000131.0551	3	3.346567	explosive
4	q1	145	NYT20000130.0162	4	3.204738	explosive
5	q1	164	NYT20000130.0185	5	2.827687	explosive
6	q1	104	NYT20000130.0119	6	2.804665	explosive
7	q1	53	NYT20000130.0059	7	2.779521	explosive
8	q1	351	NYT20000131.0166	8	2.687949	explosive
9	q1	358	NYT20000131.0173	9	2.687949	explosive