1) Data Model

- Table articles (a)
  - Row key: Article title
  - Column families: Words (w) ⇒ one qualifier per word, which is updated with the current word count

```
{
    ”article1” : {
        ”w” {
            ”word1” {
                timestamp5 : 2,
                timestamp1 : 1,
            },
            ”word2” {
                timestamp2 : 1,
            }
        }
    },
    ”article2” : {
        ”w” {
            ”word3” {
                timestamp6 : 3,
                timestamp4 : 2,
                timestamp3 : 1,
            },
            ”word1” {
                timestamp7 : 1,
            }
        }
    }
}
```

4) Extended Data Model

The positions of a word in an article could be saved as timestamp. As we go through the text of the article the word counter is updated and the additional position of the word could be added as timestamp, which is then higher than the previous one.
If we want to get a list of positions of the word, we have to get all entries of the word including the timestamps. During the application of the program, we would have to be careful with the maximum number of versions per entry, which has to be increased if necessary.

Another possibility would be to use indices. For example we could use 3 column families: The first one maps words to indices, the second maps these indices to word count and the third maps indices to a list of positions of the word corresponding to the index.

The advantage of working with indices like that is that an arbitrary amount of characteristics to each word can be stored.