ber of patterns which are found in both dictionaries. We have $FCD(x, y) = 0$ iff all patterns in $D(x)$ are contained also in $D(y)$, and $FCD(x, y) = 1$ if no single pattern is shared between the two objects.

The FCD allows computing a compression-based distance between two objects in a faster way with respect to NCD (up to one order of magnitude), as the dictionary for each object must be extracted only once and computing the intersection between two dictionaries $D(x)$ and $D(y)$ is faster than compressing the concatenation of x appended to y (Cerra and Datcu, 2012). The FCD is also more accurate, as it overcomes drawbacks such as the limited size of the lookup tables, which are employed by real compressors for efficiency constraints: this allows exploiting all the patterns contained in a string. Furthermore, while the NCD is totally data-driven, the FCD enables a token-based analysis which allows preprocessing the data, by decomposing the objects into fragments which are semantically relevant for a given data type or application. This constitutes a great advantage in the case of plain texts, as the direct analysis of words contained in a document and their concatenations allows focusing on the relevant informational content. In plain English, this means that the matching of substrings in words which may have no semantic relation between them (e.g. 'butter' and 'butterfly') is prevented. Additional improvements can be made depending on the texts language. For the case of English texts, the subfix 's' can be removed from each token, while from documents in Italian it helps to remove the last vowel from each word: this avoids considering semantically different plurals and