@Author Vali Jafarov

InformationExtraction project can be used to extract fields from document.

Following classes have been developed for following uses:

1. FileOperations Class.

Reads and writes into file. Following public methods/services have been developed for these operations:

* boolean read (String filePath) method.

Can be used to read file and set as text attribute of object. Takes as input path of file as String. Returns true if reading is completed successfully. False in either cases.

* boolean exportToFile(String text, String outputPath) method.

Can be used to write/export any text into directory sent. Takes as input text to be overwritten to file and directory to save file. Returns true if text successfully extracted into file, false in either cases

1. RabinKarpSearch class.

Finds the occurrences of a pattern string in a text string. This implementation uses the Rabin-Karp algorithm. Following public methods/services have been developed for these operations:

* RabinKarpSearch(String text) constructor. RabinKarpSearch object is initialized by setting Text to be parsed to search patterns. Takes as input text to be parsed/searched in and sets as text attribute of object. R value used in hash value calculation is set as 256.
* RabinKarpSearch(String text, int R) constructor. Allows to customize R value in addition to RabinKarpSearch(String text) constructor.
* int findFromIndex (int beginIndex, String pat) method.

Can be used to find first occurrence of the pattern string in the text string from beginIndex. Uses Rabin-Karp algorithm. Makes 2 comparisons: Hash values of pattern with substrings in text and their equality. Takes as input beginIndex (as offset) to search in text and pattern. Returns index of first occurrence of pattern in the text starting from beginIndex.

* List<Integer> findAll(String pat) method.

Finds the list of all occurrences (indexes) of the pattern string in text. Calls findFromIndex (int beginIndex, String pat) method to find all occurrences of pattern. Takes pattern as input and returns list of indexes in text where pattern is found.

1. Extract class.

Extracts information from file. Following public methods/services have been developed for its operation:

* String boldFace(Map<String, List<Integer>> occurenceMap, String text) method. BoldFaces all fields in fieldMap indicating fields (adds <b></b> tag) starting from the indexes of their occurrences. Takes map of patterns x Index list of their occurrence, text of document as input. Returns text with all fields in occurenceMap bolded in html format.
* String extractFromFile(List<String> fields, String filePath) method. Extracts info from file in path to HTML file with same name in /output/ path. Extracted fields are boldfaced in file and saved to /output/ directory. Takes list of patterns to be extracted and filePath of document to be searched in. Following operations are implemented sequentially to achieve information extraction:

1. File is read and content is assigned to a text string via read method of FileOperations class.
2. findAll(String pat) method of RabinKarpSearch is called in iteration to find all occurrences of patterns in text and results are put to occurenceMap mapping each pattern with list of indexes of its occurrences.
3. Patterns are boldfaced in HTML format via boldFace(Map<String, List<Integer>> occurenceMap, String text) method.
4. Result text is exported to HTML file with same name and saved in /output/ directory.
5. Message about result of extraction and export to file is returned.