3.2 Building the Coordinate Matrix

For each class, only one sample is required for training. We accompany a Comma

Separated Value (CSV) \_les with each XML \_le, which contains key-value pair

for all entities within the document.

Example 3. Statement A contains the term `Account Number' followed by the

account number `061234-12345678', the term `Account Holder' with the value

`John Doe', and the term `Account Type' with the value `Savings Account'.

Whereas, in statement B, the relevant term `Account No.' followed by the number

`064321-87654321', and the term `Account Name' with the value `Jane Smith'

Coordinate Matrix Machine 5

is recorded. So in the CSV \_le for statement A, we have the entries \Account

Number,061234-12345678", \Account Holder, John Doe" and \Account Type,

Savings Account"; and for statement B, we have \Account No.,064321-

87654321" and \Account Name,Jane Smith" recorded.

We then search for the keys (e.g “Account Name”, “Date”)in the XML and construct a matrix with top, left, bottom and right position of the labels for each

key for each document as shown in Tab. 1. The first column in Tab.1 comes from the training data, the second column comes from the csv file accompanied by each xml file, and the last four column results from searching from the labels in the xml files.

Table 1. Example of the Coordinate Matrix

Statement A Account No. 254 1231 259 1261

Statement A Account Holder 261 1231 266 1327

Statement A Account Type 269 1231 274 1290

Statement B Account No. 1123 231 1130 255

Statement B Account Name 100 359 107 365

3.3 Classifying New Documents

When a new document comes in, we run OCR on the documents. On the XML

produced by the OCR, we look for all the coordinates available in the training

data, and extract all the words crossing those coordinates. We do this to ensure

the approach stays robust against rotation or shift of the coordinates that can

occur during the document scanning process.

We then match the extracted words against all the keys on the training data.

Once we extract the matched keys, we then form a coordinate matrix for the

test cases comprising the top, left, bottom and right position.

Example 4. Test case contains the term `Account No.' followed by the account

number `061111-11111111', and the term `Account Name' with the value `Jane

Doe'. So the coordinate matrix for the test case would be as shown in Tab. 2. Table 2. Coordinate Matrix for the Test Case

Document ID Key Top Left Bottom Right

Test case Account No 1120 230 1125 252

Test case Account Holder - - - -

Test case Account Type - - - -

Test case Account No. 1120 230 1125 252

Test case Account Name 101 360 105 365

We then calculate the distance for each key between the test data vs every

training sample. To calculate the distance between the keys, Manhattan distance

has been used. We use Manhattan distance because it gives less value to the horizontal and vertical shifts compared to the diagonal shifts. The horizontal and vertical shifts is more likely in a document due to extra empty line or space. 6 Sadri, Wang, and Hossain

We have introduced only one parameter in this algorithm.

De\_nition 1. Maximum Penalty is the maximum distance allowed between

two keys. If the distance between the same keys from two documents is more than

this threshold, then the actual distance is substituted by this value. Furthermore, if a key is not founded in a document the distance for that key is set to Maximum Penalty. The distance between a key and In other word, if the distance between the founded key and its counterpart is more than Maximum Penalty, we assume that key is not founded in the document. We define Maximum penalty because of two reasons:1- we should have a distance value when we cannot find a key. 2- we want to make the algorithm robust. Otherwise, if a key in two documents is very far, or if one key is picked by mistake, the defined distance will be high.

We define Maximum penalty to ensure that a few mismatched keys, either due to the poor

extraction quality of the OCR or being a variant of the trained template, between

documents do not have any greater effect on the overall similarity calculation. Up to this stage, for each key of each training sample, we have calculated the distance.

Finally, we calculate the mean distance for all keys of each training sample to get a similarity score between the

test case and that training sample. The sample that has the minimum distance to the test case identifies the class.

Example 5 demonstrates the calculation.

Example 5. Let us assume that the maximum penalty is 200. Table xx shows the distance between each key and the test case.

|  |  |  |  |
| --- | --- | --- | --- |
| Document ID | Key | calculation | distance |
| Statement A | Account No. | |254 – 1120| + |1231 – 230| | 200 |
| Statement A | Account Holder | Not found | 200 |
| Statement A | Account Type | Not found | 200 |
| Statement B | Account No. | |1123 - 1120| + |230 - 231| | 4 |
| Statement B | Account Name | |101 – 100| +|360 – 359| | 2 |

The distance between “Account No.” in Statement A is 200 because the Manhattan distance exceeds 200 pixels. For “Account Holder” and “Account type” the distances are 200 because they are not found in the test case. As a result the distance between the test case and Statement A is 200 while this value is the average of 4 and 2 for Statement B. Therefore, the test case belongs to Statement B class with a similarity score of 3.