Pattern Recognition Letters 131 (2020) 390–397

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Flexible character accuracy measure for reading-order-independent evaluation

Christian Clausner*, Stefan Pletschacher, Apostolos Antonacopoulos

University of Salford, The Crescent, Salford, M5 4WT, United Kingdom

ARTICLE INFO

Article history:
Received 1 July 2019
Revised 4 December 2019
Accepted 1 February 2020
Available online 3 February 2020

Keywords:
Performance evaluation
OCR
Reading order
Accuracy

ABSTRACT

The extraction of textual information from scanned document pages is a fundamental stage in any digitisation effort and directly determines the success of the overall document analysis and understanding application scenarios. To evaluate and improve the performance of optical character recognition (OCR), it is necessary to measure the accuracy of that step alone, without the influence of the processing steps that precede it (e.g. text block segmentation and ordering). Current OCR performance evaluation measures (based on edit distance) are strongly subjective as they need to first serialise the entire text in the documents – a process influenced heavily by the specific reading order determined (often wrongly, especially in cases of multicolumn and complex layouts) by processing steps prior to OCR. This paper presents a new objective and practical edit-distance-based character recognition accuracy measure which overcomes those limitations. It achieves its independence from the reading order by comparing sub-strings of text in a flexible way (i.e. allowing for ordering variations). The precision of the flexible character accuracy measure enables the effective tuning of complete digitisation workflows (as OCR errors are isolated and other steps can be evaluated and optimised separately). For the same reason, it also enables a better estimation of post-OCR (manual) correction effort required. The proposed character accuracy measure has been systematically analysed and validated under lab conditions as well as successfully used in practice in a number of high-profile international competitions since 2017.

1. Introduction

Document Recognition systems, also known as Page Reading systems, play a crucial role in all digitisation efforts to extract and describe the information on scanned physical documents for further analysis and understanding. The accuracy of the information extracted at this fundamental stage of digitisation directly determines the success of all subsequent analysis stages which construct higher-level semantic representations of the information contained in the documents.

Starting with scanned pages as input, document recognition systems perform multiple processing steps, including layout analysis (region and text line segmentation) and optical character recognition (OCR).

Performance evaluation is used for assessing and benchmarking different systems or methods (e.g. to choose the best one for a certain document collection or use case) or, at a lower level, when adapting a specific method (improving the method, parameter tuning, or training). Although an overall black-box (system-level) performance measure is useful in some circumstances, more detailed measures that target specific steps are crucial for tailored tuning and identification of bottlenecks. The authors have worked on and proposed performance evaluation approaches for different layout analysis steps [1] which have been adopted by the research community. This paper focuses on the OCR step.

OCR accuracy is typically measured using the edit distance between two text strings: the serialised text output of an OCR method and the corresponding serialised ground truth (previously constructed absolutely accurate) text. One specific distance is the Levenshtein Distance [2] based on deletions, insertions, and substitutions required to transform one string into another (see also the Ukkonen algorithm [3] – an efficient implementation for calculating the edit distance).

The edit distance provides an absolute value of OCR errors (zero edit distance equals perfect OCR result). For easier comparison however, a relative value in the form of a percentage is preferable. Rice proposed such a measure (based on the edit distance), called *character accuracy* [4,5].

Given the need to serialise the entire document/page text, accuracy measures based on edit distance only work well for simple text block sequence comparisons (e.g. a single-column book

* Corresponding author.
E-mail address: c.clausner@salford.ac.uk (C. Clausner).

https://doi.org/10.1016/j.patrec.2020.02.003
0167-8655/

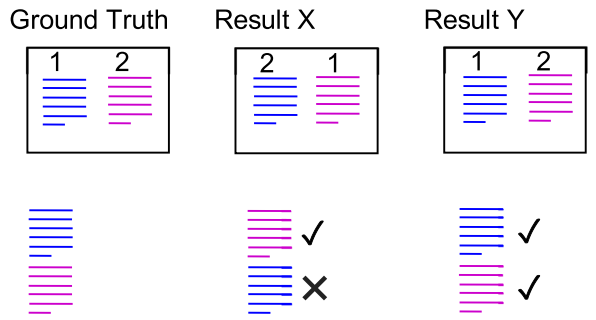

Fig. 1. Impact of reading order on Character Accuracy (top: paragraph order on page; bottom: serialised text). Given two paragraphs and the same system performance at the character level, the order in which the text is serialised is significant. OCR result *X* has the inverse paragraph order than the ground truth and will receive a low Character Accuracy score. OCR result *Y* has the same paragraph order as the ground truth and is likely to receive a higher Character Accuracy score.

page). For more complex page layouts (multi-column, containing marginalia etc.) such measures are heavily impacted by the performance of the page segmentation and text line detection steps.

Since the ordering of the page content blocks (i.e. reading order) identified by those preceding layout analysis steps is reflected in the input to the OCR step and the subsequent serialisation of the text recognised in those blocks, if the reading order is wrong the character accuracy measure can be very low, even if the actual recognition of each individual character is perfect.

Moreover, it should be noted that even in the case where the layout analysis steps do not introduce errors, there may not be a unique actual reading order on a page – e.g. there is no prescribed order in which to read the different articles on a newspaper page.

A simple example illustrating the problem of the reading order affecting the character accuracy is given in Fig. 1. Assuming all individual characters were recognised perfectly by two OCR systems *X* and *Y*, the character accuracy scores can differ significantly merely due to differences in the detected reading order of paragraphs. OCR result *X* has the inverse paragraph order than the ground truth and will receive a low character accuracy score. OCR result *Y* has the same paragraph order as the ground truth and will receive a perfect character accuracy score.

The same problem arises when text lines are merged across two columns. Although that is clearly an error in terms of reading order, it might not be of significant interest in the evaluation of OCR itself (for instance if the use scenario is keyword search) and at any rate should not be attributed to the employed character recognition method but rather to the preceding segmentation stage.

The root of these problems is that measures based on edit distance require as input the entire text in serialized form and then treat the text as an inseparable string of characters.

A more targeted and direct approach would be to consider the location (pixel coordinates) of every character when comparing OCR result and ground truth text. This, however, is most of the time not feasible for practical reasons. The list of potential impediments is long and includes problems such as systems/methods that do not provide coordinates, originals with overlapping regions, broken characters, ambiguities when dealing with special characters like ligatures, and most prominently the high costs associated with creating ground truth that contains character outlines and manually entered character codes for each one (which is orders of magnitude more expensive than entering text on block level).

An alternative is to use other, more broad performance metrics that disregard the sequence of the text entirely. One such measure is the Bag of Words measure (see [6]) which only considers the occurrence of words and their counts, not the context or location of a word. However, these metrics lack precision and are less meaningful than the character accuracy.

The most appropriate solution would be a practical measure similar to the character accuracy but less dependent on segmentation and reading order. The next section provides more context of OCR evaluation in general. Section 3 introduces the proposed measure, called Flex Character Accuracy. Section 4 contains the experimental validation. Finally, discussion and conclusions are provided in Sections 5 and 6.

2. Evaluation of OCR systems

From the numerous publications on how digitisation projects have attempted to evaluate the results of their OCR efforts it is clear that the particular approaches have evolved over time and that there are also certain constraints which are more relevant in real-world/large-scale projects than in experiments undertaken as part of research activities. In [7] it is, for instance, reported that digitisation projects in the past referred to OCR confidence values as a measure for the quality of recognised text. From a scientific point of view this appears very questionable as such values are calculated by the OCR engine itself (as a measure of internal decision process difficulty not accuracy of the decision itself) and can therefore deviate significantly from the true result. On the other hand, this approach also documents the problem these projects were facing – that there was no other viable option available to them. In this context, the viability of an evaluation approach would most likely have been determined by the associated costs.

More precise approaches were subsequently based on selecting representative examples and human “proof-reading” the OCR output with the goal of annotating/counting and potentially also correcting errors in a manual process. The logical continuation of this idea was to have the correct result ready from the start in the form of ground truth files. Ground truth also has to be created manually but there is the great benefit that it presents a one-off effort and can then be used repeatedly in automated evaluation experiments (for instance with a view to optimising certain OCR process parameters).

Evaluation based on ground truth is generally recognised as the most systematic and repeatable approach but there are differing views on what level of detail should be applied. This is again related to the associated costs for producing larger amounts of ground truth. The highest grade of ground truth records any information that could be relevant for a digital rendition of the digitised document such as region outlines and classification (type of content), logical reading order, text lines, words and potentially even the precise location and outlines of characters – all together with the respective Unicode [8] code points of the actual text or, if missing in the official standard (as is often the case for historical documents) following generally accepted recommendations on private use areas as for instance stipulated by the Medieval Unicode Font Initiative [9]. With such detailed ground truth it is possible to objectively evaluate OCR systems on all possible levels – from page segmentation down to recognition of individual characters. This also allows precise accuracy measurements for meaningful entities such as words, something that is for instance advocated in [10].

As a result of the prohibitive costs for producing highly detailed ground truth it is common that text is only entered at region (e.g. text block) level. This also means that automated evaluation approaches will have to deal with a level of uncertainty as the mapping of ground truth text blocks onto recognised characters and words (OCR result) might not be straight forward.

In [11] such constraints and limitations were taken into account and emphasis in performing the evaluation was placed also on context and on the intended use of the digitised material. To this end the evaluation of separate processing steps, such as region seg-

mentation and text recognition, was performed independently and then individual results were combined afterwards into an overall success measure related to specific scenarios. Use scenarios were expressed through a set of weights for the individual error measurements aiming to reflect the importance/unimportance of specific requirements (e.g. the correctness of reading order might be irrelevant but accurate word segmentation is very important in a "keyword search" scenario). In the absence of other methods, only the strict matching of text strings and the very relaxed bag of words approach were used for text evaluation in this context. Considering that the ground truth used in that project had text only available on region level it becomes clear that a more flexible character accuracy measure would have been highly beneficial for more precise definition of use scenarios.

There are a number of other general approaches that go in the direction of more independent character accuracy evaluation but these are subject to certain other limitations. For instance, the block edit distance (e.g. [12]) extends the standard edit distance by block move operations. Due to the computational cost involved, current implementations only approximate this by trying to find split points (heuristically). Also, this approach only works as expected if there are matching blocks of a certain size that can be moved (which might not necessarily be the case).

Another example is bipartite graph matching (see [13]) which is word-based and therefore limited in terms of precision of results (at the required character level). Accordingly, the origin of errors might not be possible to locate. Furthermore, over- or under-segmentation of words (i.e. different word splits in ground truth and result text) can lead to problems when using this measure.

3. Flex character accuracy evaluation measure

The core concept of the new measure proposed in this paper is to break down the texts (that are to be compared) into smaller chunks, perform partial edit distance measurements, and sum up the distances to obtain an overall character accuracy measure.

The algorithm can be summarised as the following steps:

1. Split the two input texts into text lines
2. Sort the ground truth text lines by length (in descending order)
3. For the first ground truth line, find the best matching OCR result line segment (by minimising a penalty that is partly based on string edit distance)
4. If full match (full length of line)
   a. Mark as done and remove line from list
   b. Else subdivide and add to respective list of text lines; resort
5. If any more lines available repeat step 3
6. Count non-matched lines / strings as insertions or deletions (depending on origin: ground truth or result)
7. Sum up all partial edit distances and calculate overall character accuracy

The sorting by length helps to match the longest chunks as one and not break them apart (step 4b) unless necessary.

When comparing chunks of different lengths, all possible positions of the shorter chunk relative to the longer chunk are considered. Fig. 2 shows the algorithm for calculating the edit distance used in step 3.

During the development of the algorithm it was observed that, when comparing a ground truth line with all OCR result lines, the partial edit distance is not the best match criterion. This is mainly because it does not differentiate between partial matches of the same distance at different substring positions (the position of the shorter chunk in relation to the longer chunk). Instead, a penalty score p was defined, using the return values of the CalculateEditDist function:

```
CalculateEditDist(t1, t2)
{
   minDist = MAXINT
   if (t1.length > t2.length)
      lengthDiff = t1.length - t2.length
      for (i=0; i<=lengthDiff)
      {
         dist = editDist(t2,t1.substr(i,t2.length))
         if (dist < minDist)
            minDist = dist
            minPos = i
      }
      subLength = t2.length
   else (t2.length > t1.length)
      //reverse to above
   else //equal length
      minDist = editDist(t2,t1)
      subLength = t1.length

   return [minDist,
           minPos,
           subLength,
           lengthDiff]
}
```

Fig. 2. Edit distance calculation for two chunks of text (pseudo code).

Table 1
Ranges of coefficients for penalty calculation.

Coefficient	Min	Max	Step
c_M	15	30	5
c_L	0	23	3
c_O	0	3	1
c_S	0	5	1

$$p = minDist * c_M + lengthDiff * c_L + offset * c_O - subLength * c_S \quad (1)$$

Where c_M, c_L, c_O, and c_S are variable coefficients (explained below) and offset is the distance of the shorter chunk from the left or right side of the longer chunk:

$$offset = lengthDiff/2 - |subPos - lengthDiff/2| \quad (2)$$

The coefficients represent weights that change how much each of the four components (edit distance, length difference, offset, and length of smaller chunk) impact the penalty score. Different values for the coefficients can lead to different overall character accuracy, depending on the complexity of the texts. This is mainly due to the fact that strings are subdivided at different positions.

Finding suitable values for the coefficients was a difficult challenge. Values that work well in one scenario might not work well in another scenario. However, regardless of the actual values, the resulting subdivisions of the text chunks are always valid. So instead of deciding on a fixed set of values for the coefficients, the algorithm uses multiple combinations of values and returns the maximum character accuracy that was achieved.

The value ranges of the coefficients (Table 1) were determined empirically using datasets from ICDAR competitions (see Section 4.4), trying to strike a balance between effectiveness and runtime performance. Using these boundaries, the flex character accuracy algorithm minimises the penalty p for each partial matching.

The splitting of strings/chunks (step 4 b) splits the longer of the compared two strings into two or three parts, depending on the position of the best match of the shorter string. If the best

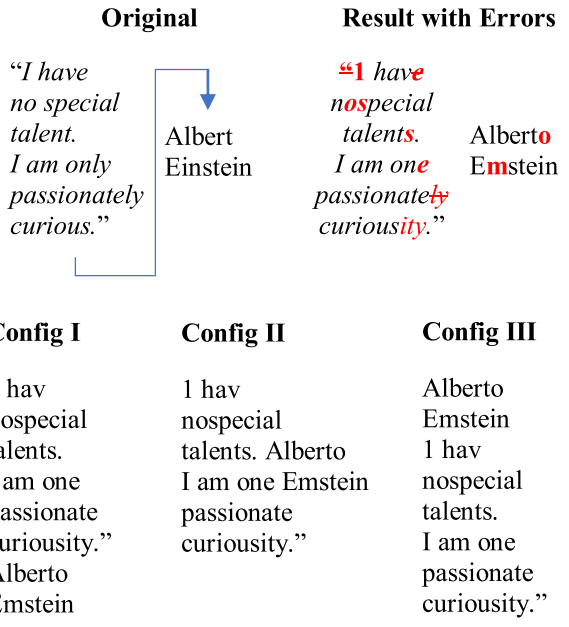

Fig. 3. Example with original text, result with errors, and different result text configurations. Character errors are indicated in red (substitutions in bold, deletions crossed out, insertions as plain red text).

Table 2
Evaluation results for different text configurations of basic example.

Configuration	Flex Character Accuracy	Character Accuracy	Bag of Words
I	77.9%	80.0%	24.0%
II	77.9%	46.7%	24.0%
III	77.9%	46.7%	24.0%

match is at the very left or very right, the long string is split into two strings (the matching part and the remaining part). If the best match is somewhere in the middle, the long string is split into left, middle, and right. The non-matching parts are put back into the processing queue.

The source code of the flex character accuracy measure is publicly available at github.com/PRImA-Research-Lab/prima-text. A library with other evaluation measures is also available at primaresearch.org.

4. Experiments and discussion

In this section the flex character accuracy measure is validated and compared to the traditional character accuracy measure. This was done in three ways:

- under controlled conditions with short example texts (see Sections 4.1 and 4.2),
- with a dataset of historical documents (see Section 4.3), and
- in the context of ICDAR (2019) and ICFHR (2018) competitions (see Section 4.4).

4.1. Basic experiment

To test whether the measure performs as intended, a short text (a quote and the author) was chosen (Fig. 3). Artificial character errors were introduced and three different configurations of how the two parts could be combined were created. Both character accuracy measures were calculated as well as a bag-of-words measure for comparison (Table 2).

For configuration I (baseline), where there are errors, but the OCR result has the same general order as the ground truth, both character accuracy measures return a success rate of 80%, confirming the equivalence of the two in this case. The other two configurations (same words but differing order) highlight the difference between the traditional and the flex character accuracy measure. The traditional measure produces rates that diverge significantly from the baseline (47% vs. 80% accuracy), whereas the flex accuracy measure is stable (same success rate for all configurations).

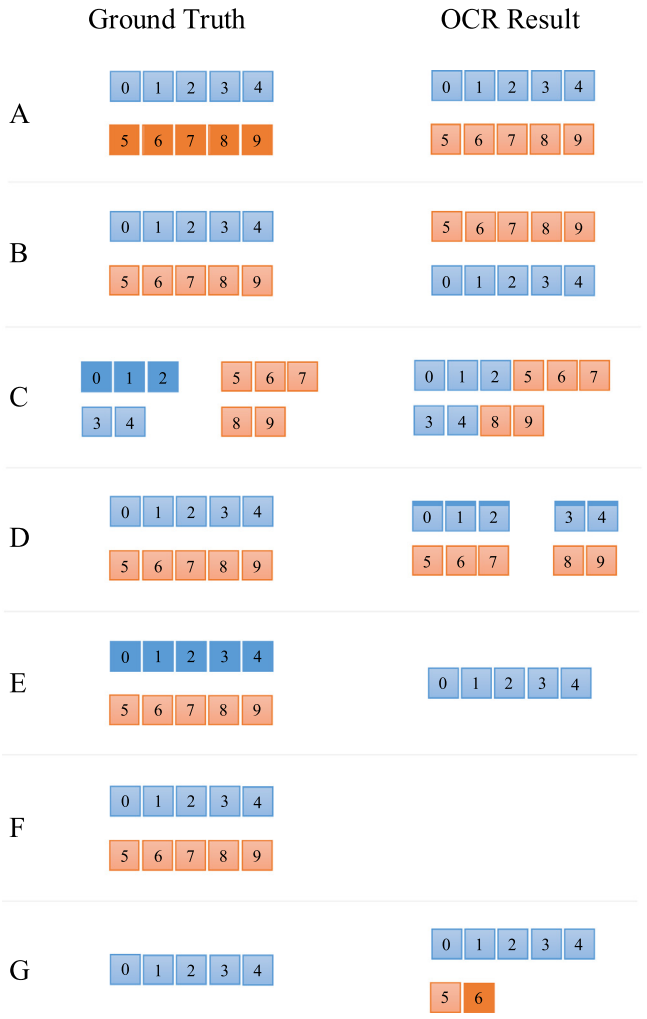

Fig. 4. Different ground truth and OCR configurations. Words of the example text are represented as numbered boxes (0 = "Eight", 1 = "happy", 2 = "frogs" etc.).

4.2. Extended experiments

For the extended experiments two short example text paragraphs were used, each containing five words of five characters each:

"Eight happy frogs scuba dived" and
"Jenny chick flaps white wings".

Taking into account whitespaces, this leads to 59 characters in total.

Fig. 4 shows different configurations of ground truth and OCR result for commonly occurring situations:

- A: No errors
- B: Different ordering of text blocks
- C: Merge across columns
- D: Over-segmentation
- E: Part missing
- F: All missing
- G: Added parts

Table 3
Evaluation results for different configurations of extended example.

Configuration	Flex Character Accuracy	Character Accuracy
A	100%	100%
B	100%	25.4%
C	96.4%	39.0%
D	96.6%	59.3%
E	50.0%	49.2%
F	0%	0%
G	62.1%	58.6%

Table 4
Evaluation results for different measures and different OCR engines.

Evaluation measure	Bag of words	Character accuracy	Flex character accuracy
Complete dataset (17 pages)			
Tesseract 4	92.7%	36.8%	94.9%
Google OCR	94.0%	28.3%	91.4%
FRE11	81.8%	35.7%	93.1%
First page (see Fig. 5), original ground truth			
Tesseract 4	93.0%	34.4%	96.8%
Google OCR	93.3%	29.3%	90.5%
FRE11	75.9%	34.8%	89.8%
First page, ground truth without reading order			
Tesseract 4	93.0%	75.3%	96.8%
Google OCR	93.3%	44.6%	90.5%
FRE11	75.9%	91.2%	91.1%

N.B. For simplicity in this example only word-level substitutions are considered. As illustrated before, and in contrast to other methods, the proposed method is perfectly capable of handling any character-level substitutions as well.

Table 3 contains the respective evaluation results using the flex character accuracy measure in comparison to the standard character accuracy measure.

Where all ten words are contained in the OCR result and none of the words is split internally, the new measure maintains accuracy values close to 100% (situations A to D). Small deviations can be explained by differences in whitespaces. In contrast, the traditional character accuracy measure drops to 25% in the worst case, despite all ground truth words being present in the OCR result in those examples. Such low values are not justifiable from a character accuracy point of view in that they do not reflect the actual recognition performance on character level. This demonstrates a severe limitation of the traditional measure and underlines the usefulness of the proposed measure if an in-depth analysis of the pure character recognition quality is required.

In situations where actual recognition errors occur (E to G), the proposed measure behaves similarly to the traditional measure. Again, the relatively small deviations are due to differences in whitespaces caused by splitting into text lines (line breaks are disregarded in the new measure but count as normal characters in the traditional measure).

4.3. Historical documents

As a real-world test, a dataset containing 17 pages of a Dutch legal document (in a two-column layout) from 1857 was processed by different OCR systems and the results evaluated. The originals are held by the Dutch National Library and images as well as ground truth are available as part of the IMPACT Digitisation collection [14]. Fig. 5 shows the first page of the dataset with overlays of ground truth regions and OCR result regions.

The dataset was processed with the following state-of-the-art OCR engines: Tesseract 4, Google Cloud Vision OCR, and ABBYY FineReader Engine 11 (FRE11). Table 4 and Fig. 6 show the evaluation results for the three measures under comparison.

Comparing the overall results (17 pages) for the three different measures, it can be seen that the flex character accuracy measure appears much more in line with the Bag of Words measure (which is independent of the text order). However, the two measures are not fully correlated. For example, the Google OCR has the highest word-level success rate but the lowest character-level success rate. This can be explained by the distribution of character errors (clustered within few words vs. being spread across a larger number of words).

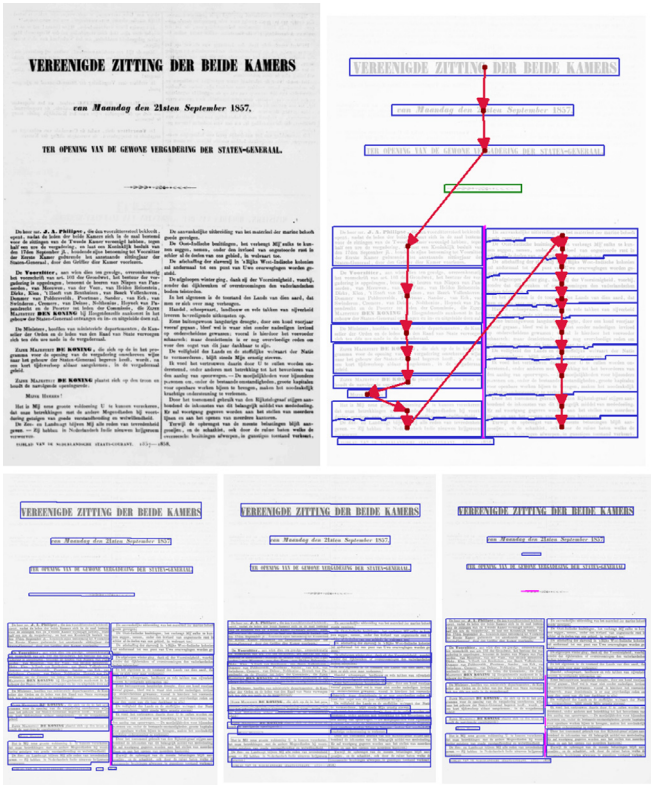

Fig. 5. First page of Dutch legal document from 1857 ("Handelingen der Staten-Generaal"), (top left: original image, top right: ground truth regions and reading order, bottom: OCR result regions - left: Tesseract 4, centre: Google OCR, right: FineReader Engine 11).

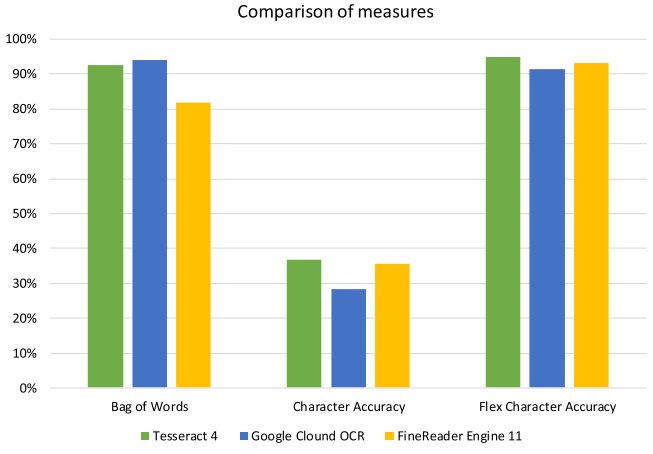

Fig. 6. Comparison of evaluation measures and OCR engines across all images in the example historical document dataset.

Fig. 7. Example pages with text regions highlighted in blue (left: RDCL2019, right: RASM2018).

Table 5
Evaluation results for RDCL2019 competition.

METHOD	BAG OF WORDS	CHAR ACC.	FLEX CHAR ACC.
BKZA	94.89%	47.98%	94.77%
LINGDIAR	91.32%	73.02%	89.81%
MHS	92.43%	61.33%	94.62%
FRE11	97.49%	48.24%	95.90%
FRE12	97.07%	80.39%	96.12%
TESSERACT4	95.68%	47.28%	95.27%

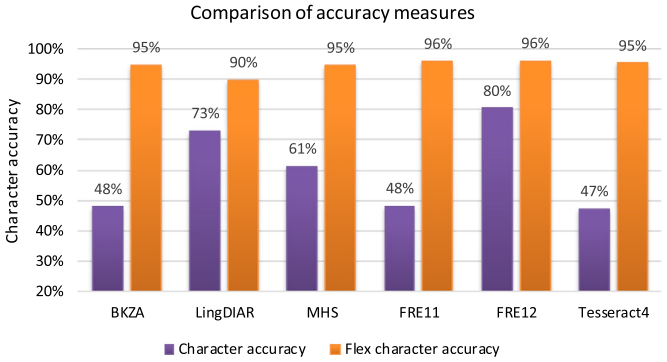

Fig. 8. Character accuracy and flex character accuracy for RDCL2019 submissions.

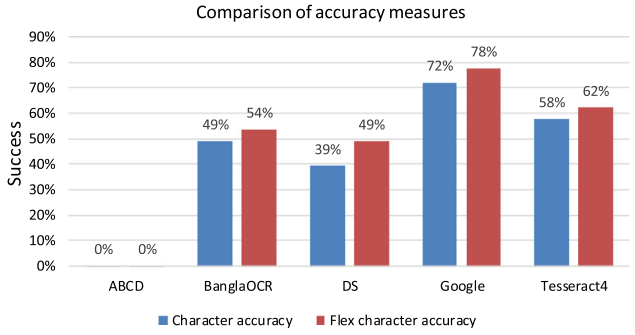

Fig. 9. Character accuracy and flex character accuracy for RASM2018 submissions.

For a better understanding of the results, a more detailed analysis was performed for the first page of the dataset. The bottom part of Table 5 shows the evaluation scores using the original ground truth and a version of the ground truth without reading order. If present, the reading order is used by the evaluation system to serialise the text content. If no explicit reading order is provided, the text is serialised top-to-bottom using region coordinates.

The results of the traditional character accuracy measure vary dramatically, confirming how much it depends on the text order of both ground truth and OCR result being in sync. Google's OCR scores very low, mostly because it merges the text across the two columns (see Fig. 5 bottom centre). FineReader scores very high, as its paragraph segmentation result is very close to the ground truth.

Looking at the results overall, it becomes clear that the traditional character accuracy measure does not necessarily provide reliable information on the actual character recognition performance for real-world documents. The flex character accuracy measure, on the other hand, is very stable and enables a more realistic and objective assessment of OCR engines.

4.4. Evaluation of competition results

Among other applications, the flex character accuracy measure was used for OCR result evaluation in context of an ICDAR competition for the recognition of documents with contemporary (complex) layouts (RDCL2019) [15] and an ICFHR competition for the recognition of historical Arabic scientific manuscripts (RASM2018) [17]. Both competitions involve the tasks of page segmentation and text recognition of complex documents, partly of low image quality (see Fig. 7 for two examples).

Fig. 8 and Table 5 show the evaluation results of the OCR outputs of the methods that participated in RDCL2019 [15,16]. As expected, the success rate produced by the flex character accuracy measure is always greater or equal to that produced by the traditional character accuracy measure as it strips away side effects caused by other processing steps (which should be measured/evaluated separately). Depending on the OCR method, the differences in accuracy range from 16% up to 48%. It can also be observed that the flex character accuracy rate is more in line with the Bag of Words success rate. This supports the claim that the proposed measure is less impacted by differences in reading order and segmentation (similar tendencies as the Bag of Words, which disregards reading order and context by definition) and therefore provides a more focused view on character accuracy.

Fig. 9 and Table 6 show the results of the RASM2018 competition [17] which also included segmentation evaluation on its own (although not all participants submitted results for all levels/challenges). The same general observations as described before can be made. The differences between the rates produced by the flex and the traditional character accuracy measures range from 0% to 9.5%. The first three methods suffer from very poor segmentation (below 50% success rate) which is then penalised again in a very low character accuracy. This can be concluded from the apparent correlation between segmentation performance (Challenges 1 and 2) and the observed differences between the success rates produced by the two different character accuracy measures and is most likely a consequence of under- and over-segmentation leading to different text arrangements in the OCR result when compared to the ground truth text.

4.5. Runtime performance

The runtime performance of the evaluation method in a real-world situation can be determined by looking at the RASM2018 competition, for example. On average (for each page) each ground truth text and OCR result had 846 characters. The total number of evaluated pages (across all participating OCR methods) was 680. The total execution time (unoptimized code) was 197 min, leading to an average of 17.4 s (on a Windows PC with a 3 GHz Intel Xeon processor).

Table 6
RASM2018 evaluation results using different measures (rates denote percent success).

	Tesseract 3	Tesseract 4	FRE11	Google	KFCN	RDI
OCR (Challenge 1)						
	Original text					
Character accuracy	13.0	18.3	11.0	60.4	N/A	78.1
Flex character accuracy	20.9	27.8	10.8	60.6	N/A	78.1
Bag of words	2.2	4.7	0.4	20.9	N/A	42.3
	Normalised text[1]					
Character accuracy	13.3	19.2	12.3	64.4	N/A	85.4
Flex character accuracy	20.9	30.5	12.2	64.8	N/A	85.4
Bag of words success rate	2.5	5.5	0.4	26.7	N/A	60.6
Page segmentation (Challenge 1)	48.4	54.4	40.9	70.6	87.9	N/A
Text line segmentation (Challenge2)	28.8	44.2	43.2	N/A	67.7	81.6

1: Rare characters with diacritics replaced by non-diacritic versions.

In contrast, the average execution time of the standard character accuracy (unoptimized code also) was 0.3 s per page. The large difference in performance is caused by mainly two factors. First, when comparing shorter text chunks with longer chunks, the algorithm calculates the edit distance for each possible offset. Second, the method tries multiple combinations of four coefficients to find the best character accuracy ($4*8*4*6 = 768$ combinations). Although there is a large difference in performance, the benefits of the new measure outweigh the additional resource requirements in most use scenarios.

5. Discussion

In general, character accuracy measures (traditional and proposed) work purely on textual input data. This fact limits the capability of such measures to pinpoint the precise origin of OCR errors. Only the context of the surrounding text can help to locate errors. If the locations of each character (or glyph) on the page were known a perfect character accuracy algorithm could be applied, checking each ground truth character for a local match from the OCR result. This, however, is typically not the case (due to a number of factors such as nature and quality of the originals, employed methods and costs involved in producing ground truth), as briefly mentioned earlier in this paper. Therefore, despite the limitations, text-based measures are usually the best option for benchmarking and process optimisation.

The proposed measure, while advantageous by flexibly matching and comparing partial text line strings, introduces a minor issue for consideration. It subdivides text lines if they are only partially matched. The unmatched parts are fed back into the list of chunks that are still to be processed. This, however, can lead to situations where very short chunks are created (e.g. a single punctuation mark). These small chunks are very indistinct and might be (wrongly) matched to parts occurring elsewhere in the text. Although this has no negative impact on the accuracy score, it might lessen the error information when analysing individual matches.

The experiments showed that the flex character accuracy measure is indeed less dependent on reading order and text object segmentation. A useful by-product is that the difference between the traditional and the proposed character accuracy can give an indirect insight into the performance of the segmentation and order detection steps of the OCR system under investigation.

6. Conclusion

A new measure for character accuracy has been presented that strongly reduces the impact of reading order variances in text blocks (as detected by the segmentation step prior to OCR), providing a more stable and precise representation of the actual character recognition performance. The algorithm was validated using both controlled examples and real-world evaluation scenarios (from ICDAR and ICFHR competitions). A more focused and independent character accuracy measure such as the proposed one, which still works on arbitrary serialised text is highly desirable as it enables the pinpointing of areas for improvement in OCR results and can aid training and fine-tuning of character recognition methods in OCR systems. If a combined measure of character accuracy including segmentation and reading order is required then this can be very easily obtained in the form of a potentially weighted overall benchmark based on the independent method-related measures.

The precision of the proposed flexible character edit distance measure makes it possible to evaluate and optimize complete workflows (e.g. from image pre-processing to segmentation and OCR) as it pinpoints the OCR performance only without being affected by other steps of the workflow (which can be evaluated and optimized separately). Furthermore, the proposed measure also enables a more accurate estimation of post-correction effort required in digitization projects, as the errors it indicates correspond precisely to the cases where text needs to be corrected and not to the accumulated errors from earlier workflow steps that influence the existing evaluation approaches based on the traditional character edit distance measure.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Christian Clausner: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing - review & editing. **Stefan Pletschacher:** Formal analysis, Investigation, Validation, Writing - review & editing. **Apostolos Antonacopoulos:** Supervision, Validation, Writing - review & editing.

References

[1] C. Clausner, S. Pletschacher, A. Antonacopoulos, Scenario Driven In-Depth Performance Evaluation of Document Layout Analysis Methods, in: Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011), Beijing, China, 2001, pp. 1404–1408. September 2011.
[2] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Phys. Dokl. 10 (8) (1966) 707–710 February 1966.
[3] E. Ukkonen, Algorithms for approximate string matching, Inf. Control 64 (1–3) (1985) 100–118, doi:10.1016/S0019-9958(85)80046-2.
[4] S.V. Rice, Measuring the Accuracy of Page-Reading Systems, PhD Thesis, University of Nevada, Las Vegas, 1996.
[5] S.V. Rice, F.R. Jenkins, T.A. Nartker, 2012. The fourth annual test of OCR accuracy. 2012.
[6] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, ICDAR2013 competition on historical book recognition – HBR2013, in: Proc. ICDAR2013, Washington DC, USA, 2013 Aug 2013.

[7] R. Holly, How good can it get? Analysing and improving ocr accuracy in large scale historic newspaper digitisation programs, D-Lib Mag. 15 (2009) Number ¾, March/April 2009.
[8] The Unicode Consortium. The unicode standard. http://www.unicode.org/versions/latest/.
[9] O.E. Haugen, 2015. MUFI character recommendation v. 4.0. Medieval unicode font initiative, 2015, http://hdl.handle.net/1956/10699.
[10] S. Tanner, T. Munoz, P. Hemy Ros, Measuring mass text digitization quality and usefulness: lessons learned from assessing the OCR accuracy of the British Library's 19th century online newspaper archive, Dlib Mag. 15 (78) (2009) 2009.
[11] S. Pletschacher, C. Clausner, A. Antonacopoulos, Europeana newspapers OCR workflow evaluation, in: Proceedings of the 2015 Workshop on Historical Document Imaging and Processing (HIP2015), Nancy, France, 2015, pp. 39–46. August 2015.
[12] M. Ganczorz, P. Gawrychowski, A. Jez, T. Kociumaka, Edit distance with block operations, In proceedings of 26th Annual European Symposium on Algorithms (ESA 2018), 2018.
[13] K. Riesen, H. Bunke, Approximate graph edit distance computation by means of bipartite graph matching, Image Cumput. Vis. 27 (7) (2009) 950–959 4 June 2009.
[14] IMPACT Digitisation Dataset, accessed 18/11/2019. https://www.primaresearch.org/datasets/IMPACT_Digitisation.
[15] C. Clausner, S. h, A. Antonacopoulos, ICDAR2019 competition on recognition of documents with complex layouts – RDCL2019, in: Accepted for publication in Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR2019), Sydney, Australia, 2019 September 2019.
[16] RDCL2019 Website, last visited 30/06/2019. https://www.primaresearch.org/RDCL2019.
[17] C. Clausner, A. Antonacopoulos, N. McGregor, ICFHR 2018 competition on recognition of historical arabic scientific manuscripts - RASM2018, in: Proceedings of the 17th International Workshop on Frontiers in Handwriting Recognition (ICFHR2018), Niagara Falls, USA, 2018, pp. 471–476. August 2018.