Examining a Multi Layered Approach for Classification of OCR Quality without Ground Truth

Mirjam Cuper

KB, national library of the Netherlands

While the digital availability of heritage text collections is increasing, there is a lack of reliable methods to assess the OCR quality of these texts. There are several possible measures, but each has its disadvantages. We examined if, instead of a single measure, a *combination* of measures would give a more accurate indication of OCR quality. We therefore built a first version of a multi-layered approach for the classification of OCR quality, named QuPipe. We tested QuPipe on a set of sentences from 17th century Dutch newspapers and found that using QuPipe led to an increase in correctly classifying the quality. However, although these results are positive, QuPipe needs to be developed and tested further to become feasible as an instrument to classify the OCR quality.

Keywords: Optical Character Recognition (OCR); OCR quality; digital heritage; digital humanities; digitized texts

1 Introduction

In the past few decades, more and more heritage institutions have made their collections digitally available. At the same time, the accessibility and amount of computer driven research tasks is increasing. With this combination, large data sets can be analysed in a fairly short time, something which would not be possible by hand. However, there is a pitfall; the Optical Character Recognition (OCR) quality of digitised text is not always high enough. This leads to several possible problems which can cause bias in the research results, both on the information retrieval and the analysis level (Nguyen, 2020). Although most researchers are aware of the presence of OCR errors, often they are not able to quantify these errors or the impact on their research. This leads to uncertainty about whether results can be published or not (Traub et al., 2015).

A measure for the (relative) quality of OCR would be very beneficial for the field of Digital Humanities. Furthermore, digital heritage institutions can use such a measure to improve their digitised collections. To get an indication of the OCR quality of a collection, two methods are most prevalent. The first method consist of extracting a sample from a batch of digitised text and manually inspecting the quality of this

sample. The results of this quality control are then extrapolated to the rest of the batch. The second method is based on the existence of a 'Ground Truth'-set. A Ground Truth set consist of digitised texts that are manually corrected by humans. Therefore, these digitised texts are of high quality and contain very few or no errors. They can be used to determine the quality of the corresponding OCR output. The results can then be extrapolated to the rest of the collection. However, the creation of a Ground Truth set is time consuming and expensive (Holley, 2009), which results in only a small number of available Ground Truth sets. The extrapolations of these methods can only be considered as rough estimates and are not very reliable for a variety of reasons, such as: the variety in quality of the original material, the used font types, the text direction, and how the original material is stored and bound. This can even differ between various issues of, for example, the same newspaper.

Due to the unreliability of above methods and the scarcity of Ground Truth sets, various other methods have been developed to determine the quality of OCR-ed text without the existence of a Ground Truth. Previous research has mentioned methods such as a dictionary lookup (Strange et al., 2014; Van Strien. et al., 2020), garbage detection (Kulp and Kontostathis, 2007; Taghva et al., 2001), and confidence values from the OCR engine (Holley, 2009; Springmann et al., 2016). Some of these measures are more accurate than others, but they all have problems with their accuracy due to the nature of language and problems in digitisation. In addition, some measures are not always available, such as a historical dictionary or confidence values.

In an attempt to overcome these problems, we developed a first version of a multi-layered approach that combines measures: a quality pipeline named QuPipe. We included statistical measures for word length and sentence length. Furthermore, we added more complex measures: language detection, garbage detection, trigrams and two different approaches of dictionary lookup. Since every language and time period has its own characteristics, we created a set with reference values which we evaluated the measures against.

We tested our first version of QuPipe on a controlled set of sentences from 17th century newspaper articles. For this set, both the original OCR and the Ground Truth are available. The Ground Truth was used to calculate the Character Error Rate (CER) for every sentence. This CER was used to calculate the precision and recall of our experiments. Historical dictionaries are not always available for historical languages, therefore, we tested our approach with and without a dictionary. We also examined the minimum amount of Ground Truth needed to create useful reference values. With our experiments, we test the following hypothesis:

"A combination of measures is more accurate for the prediction of OCR quality than just a single measure."

This paper is structured as follows. In section 2, we examine related work in which various quality measures are described in more detail. Section 3 explains our method, including the used data, implemented measures, creation of the reference sets, and the QuPipe scores calculations. Section 4 shows the results of our experiments. Section 5 contains our conclusion, in which we reflect on our hypothesis and the advantages and disadvantages of our study. This section also contains our plans for future work.

2 Related work

A commonly used method for measuring OCR quality is a dictionary lookup, where every word in a text is matched against a dictionary (Strange et al., 2014; Van Strien.

et al., 2020). Then, the total amount of found words is divided by the total amount of words to get an indication of the quality of the OCR-ed text. The higher the result, the better the quality. Various research has been conducted to determine what can be considered as the minimum dictionary lookup value to indicate a high enough OCR quality. Strange et al. (2014) performed a sample task of finding words. Their findings were that correcting the accuracy of OCR from 80% to a higher accuracy is not essential for such a task. Van Strien. et al. (2020) performed an analysis on several Natural Language Processing tasks and concluded that a dictionary lookup score of at least 80% is preferable for these tasks. We conducted a small experiment and also concluded that a dictionary lookup is overall quite reliable, see Figure 1. However, we detected that several problems can occur, such as false positive prediction due to incomplete OCR or garbage in texts (Cuper, 2021). Strange et al. (2014) mention that problems can be caused by correct words which are not present in a dictionary, and with incorrect words in the digitised text which are correct words in real life. An example of such a real world error is the word 'grass' that has been interpreted by the OCR software as 'glass'. Van Strien. et al. (2020) mention that how languages change over time can possibly lead to difficulties with historical texts. Also, especially for older languages, a historical dictionary is not always available.

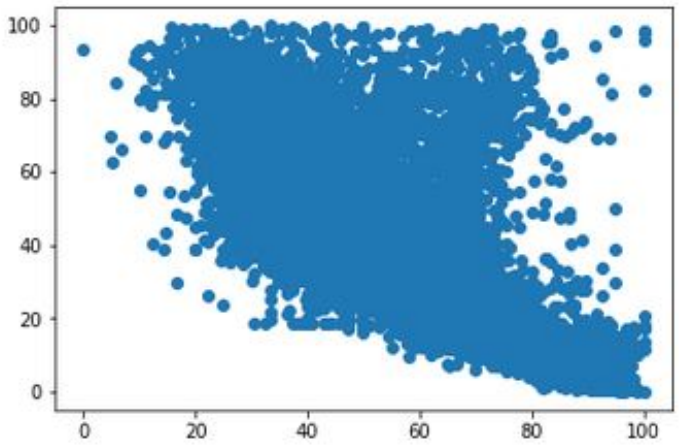

Figure 1: Correlation between CER and dictionary lookup.

A Python tool called 'Language Identifier' (Lui, 2011) was used by Baumann (2015) to measure OCR quality. He used the tool line by line and used the provided confidence score of the tool as OCR quality measure.

Holley (2009) proposed a method for the development of an accuracy algorithm based on the correlation between manually determined quality and the confidence values from the OCR engine itself. However, they did not test the algorithm themselves. Springmann et al. (2016) used character confidence values as one of the measures for OCR accuracy. A downside of the use of confidence scores is that they are not always available to researchers.

Various studies focused on the detection of errors instead of measuring OCR quality. However, if one can detect errors, this information can be used to determine the quality of a text.

Taghva et al. (2001) came across the problem of garbage strings in OCR-ed texts that were complicated for information systems. They developed a rule-based approach

to detect such garbage in texts, consisting of six rules. Kulp and Kontostathis (2007) adapted this idea, altered the rules and added two new rules. Their method is more strict, likely leading to more words that are classified as garbage. Wudtke et al. (2011) used the idea of garbage detection and implemented this idea as a support vector machine, leading to better results than the rule-based approach.

In multiple studies, different types of ngrams were used for the detection and correction of errors in texts. All these studies suggest that the use of ngrams is effective (Ahmed et al., 2009; Atawy and ElGhany, 2018; Robertson and Willett, 1998; Wu et al., 2013). Although we could not find any studies about the use of ngrams for measuring OCR quality, the use of ngrams for error detection and corrections suggests that ngrams can be a useful addition for the determination of OCR quality.

3 Method

Based on the related work and practical considerations, we created a list of five measures for our first implementation of QuPipe: language detection, garbage detection, trigrams and two different approaches of a dictionary lookup. Since every language and time period had its own characteristics, we also included the statistical measures average word length and sentence length. A detailed description of every measure is provided in section 3.2. We used part of the available Ground Truth data to create a set of reference values. Except for the language detection measure, every measure uses a reference value to determine whether the measure is in the expected range or not. Section 3.3 provides a description of the creation of these reference values.

We tested QuPipe on a dataset of 17th century newspapers. We used a controlled dataset with pairs of sentences. Each pair consisted of a Ground Truth sentence and the corresponding OCR sentence. A benefit of such a controlled dataset is that you can closely monitor the effect of changes in approach. Section 3.1 describes the preparation of the data.

Since we had the Ground Truth available, we were able to calculate the Character Error Rate (CER) of every OCR sentence. This was done with the ocrevalUAtion tool (IMPACT Centre of Competence, 2019). Holley (2009) described an accuracy of below 90% as low. Based on this, we classified our CER output as 'good' and 'not good', with a CER equal to or lower than 10 classified as 'good', and a CER higher than 10 classified as 'not good'.

The primary output of QuPipe is a binary value per measure, but for our experiments we needed a single output from QuPipe. We tested three different ways to calculate this single output: the normal calculation, the 'smart quality' calculation and the 'smart quantity' calculation. Section 3.4 describes these calculations in more detail. After calculation, QuPipe has classified every OCR sentence as either 'good' or 'not good'.

To test the performance of QuPipe, we compare the outcome of QuPipe with the calculated CER. This comparison leads to a precision and a recall for every calculation type. The precision indicates how much of the sentences that were classified as 'good' indeed had a CER score of 10 or higher, whereas the recall indicates how many of the sentences with a CER score of 10 or higher were correctly classified as 'good'.

Since historical dictionaries are not always available, we performed experiments with and without a dictionary. For the experiments with a dictionary, we compared the performances of a modern dictionary, a historical dictionary, and a combined dictionary. Furthermore, we compared QuPipe outcomes with a 'token' dictionary

lookup with a cutoff point of 80%, based on the minimum required OCR quality according to the literature (Strange et al., 2014; Van Strien. et al., 2020).

3.1 Collecting and preparing data

For our experiments, we used a dataset containing 34.808 articles from 6.425 newspapers from the 17th century (Colavizza and Cuper, 2021). For this dataset, both the original OCR and the manually corrected Ground Truth were available. The OCR of these articles has a high variance in quality. Sometimes parts of articles are missing, which can lead to misleading results if we compare the CER with QuPipe outcomes. To decrease the chance of misleading results due to unmatched texts, we decided to create a controlled set based on matched sentences.

Per article, we automatically extracted sentences from the Ground Truth and matched these with sentences from the OCR. To split the text in sentences, we arbitrarily used the indicators '.', '?' and '!'. We used only sentences that contained at least 7 words. The Python package SequenceMatcher (Python Software Foundation, 2022) was used to match the sentences from the Ground Truth with the corresponding sentences from the OCR. The basic idea of SequenceMatcher is to find the longest contiguous matching subsequence. Sentences were selected as a pair when the match was equal to, or higher than 75%. The cut-off point of 75% was based on a random sample draw, in which we determined the lower limit to prevent false positive matches.

After matching, 146.216 sentences were returned. A disadvantage of SequenceMatcher is that it can match sentences with a difference in length. This means, that it can match a complete sentence to a partial sentence, which can lead to misleading results in the experiments (see table 1). We therefore decided to remove all sentences with a larger than 10% difference in word count. This led to a final dataset of 94.471 matched sentences.

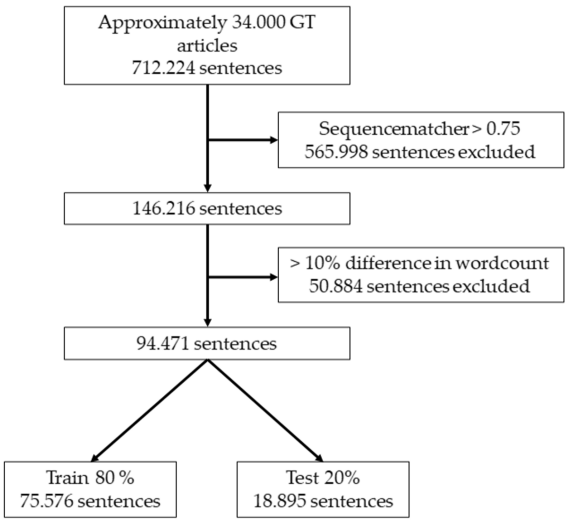

Figure 2: Schematic overview of the data collection process .

Table 1: An example of noise in sentence matching

Type	Sentence
GT	een wijdt-loopighe memorie heeft overgelevert raeckende eenighe geestelijcke goederen
OCR	mog een wijdt-loopighe memorie heeft overgelev raeckende eenighe geeftelijcke goede ridders van malt ha ondert staet leggendec

We divided the dataset in a training and test set with a ratio of 80/20, leading to a training set of 75.576 sentences and a test set of 18.895 sentences. The training set was used for the creation of the trigram model and the reference values. The test set was used for evaluating QuPipe. Figure 2 shows a schematic overview of the data pre-processing. Figure 3 shows the distribution of the test set among the various newspaper publishers.

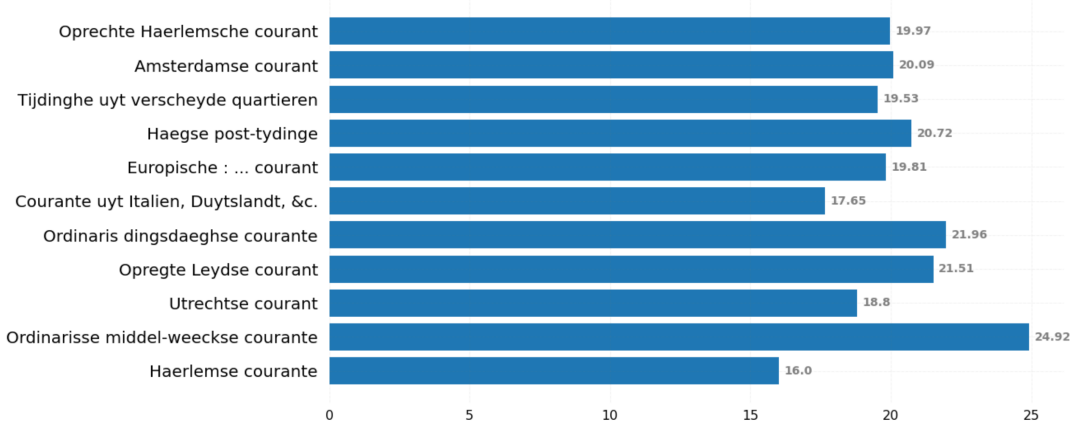

Figure 3: Distribution (%) of the test set across the newspaper publishers.

We calculated the Character Error Rate (CER) for the test set using the ocrevalUAtion tool (IMPACT Centre of Competence, 2019). The results were then split into 'good' (CER equal to or less than 10) or 'not good' (CER above 10). The distribution between these two categories in the test set is shown in Table 2.

Table 2: Distribution of sentences between CER ≤ 10 and CER > 10

CER	absolute	percentage of total
CER ≤ 10	10330	54.7%
CER > 10	8565	45.3%
Total	18895	100%

3.2 Implementation of the measures

3.2.1 Dictionary lookup 'token'

We performed a 'token' dictionary lookup with a modern dictionary, a historical dictionary, and a combination of both. The modern dictionary consists of a word list from Open Taal (Open Taal, 2020) and the historical dictionary consists of two word lists from INT (Instituut voor de Nederlandse taal, 2012). The combined dictionary is a combination of all these word lists. For the 'token' dictionary lookup, all words were used, regardless of the number of time they appeared in the text.

For all dictionary lookups, we followed these preparation steps:

- All punctuation marks, apart from the hyphen, are removed from the text;
- The text is lemmatised with the Python package SpaCy;
- All characters are set to lower case.

After these steps, every word of the text is checked against the selected dictionary. This results in a number of words that are found in the dictionary. This number is divided by the total amount of words in the text to get a percentage of correct words. The value of the dictionary lookup 'token' is 1 if the percentage is equal or above a the reference value (see section 3.3) and 0 if otherwise.

3.2.2 Dictionary lookup 'type'

As with the dictionary lookup 'token', the dictionary look up 'type' was performed on a modern, a historical and a combined dictionary. Furthermore, the preparation steps are the same. However, where the dictionary lookup 'token' is performed on all words, the dictionary lookup 'type' is performed on the set of words. This means that words that occur multiple times in a text, only count once for the dictionary lookup 'type'. When a text has a high number of repeated words, a 'type' lookup can give a more accurate representation of the quality for those words that are important for the essence of a text. An example of the differences between a 'token' and a 'type' representation of a sentence is shown in Table 3.

Table 3: 'Token' versus 'type' sentence representation

Variant	Sentence representation
Token	(de, alexander, is, er, waarschijnlijk, de, merlijn, is, er, zeker, geschreven)
Type	(de, alexander, is, er, waarschijnlijk, merlijn, zeker, geschreven)

Every word in the set is checked against the selected dictionary. This results in a number of words that are found in the dictionary. This number is divided by the total amount of words in the set to get a percentage of correct words. The value of the dictionary lookup 'type' is 1 if the percentage is equal or above the reference value (see section 3.3) and 0 if otherwise.

3.2.3 Average word length

For the average word length, the length of every word is calculated first. Then, we calculated both the mean and the median word length of all words in a sentence. Both

the mean and median are compared with a reference set (see section 3.3). If both the mean and median are in the range of the reference value, the average word length value is set to 1. If only one or neither of the values are in this range, the average word length value is set to 0.

3.2.4 Sentence length

The sentence length is calculated by counting the amount of words in the sentence. This value is compared with the reference set for sentence length (see section 3.3). The sentence length value is set to 1 if the sentence length is in the range of the reference value and 0 if otherwise.

3.2.5 Language detection

Lui (2011) developed a Python package which can identify the language of a given text. The output of the tool is the identified language and the non-normalized probability estimate for the language. We ran some sample tests to test how this tool performed on historical Dutch texts and on texts with OCR errors. We found that the tool was capable of recognizing historical Dutch as the a Dutch language. However, when there are a lot of OCR-errors in the text, it predicts another language.

Since we are testing on Dutch historical news articles, the value is set to 1 if the identified language is 'nl' (Dutch) and 0 if otherwise. For this first version of QuPipe, we did not take the probability estimates into account.

3.2.6 Garbage detection

For our garbage detection measure, we used some of the rules introduced by Taghva et al. (2001) and Kulp and Kontostathis (2007). For this first version of QuPipe, we implemented the following selection of rules:

- A word with more than 20 characters is considered as garbage.
- A word with more than 3 identical characters in a row is considered as garbage.
- If a word has only vowels and consonants, and if the number of vowels is 8 times greater than the number of consonants or the other way around, the word is considered as garbage.
- If we strip the first and last letter of a word, and there are more than two punctuation characters in the word, it is considered as garbage.
- If a word start and ends with a lower-case letter, and one of the remaining characters is upper-case, it is considered as garbage.

Furthermore, we added our own rules, based on Dutch language characteristics:

- If a word contains letters or punctuation that are not part of the Dutch languages, it is considered as garbage.
- If a word starts or ends with more than 4 consonants, it is considered as garbage.

For every rule, the percentage of words in a sentence that meets this rule is calculated, leading to a set of percentages. The garbage value is set to 0 ('garbage') if the percentage of at least one of these rules is equal to or above reference value (see section 3.3) and 1 if otherwise.

3.2.7 Trigram

A trigram is a contiguous sequence of three items. These items are, for example, words or characters. Trigrams can be extracted from a corpus of text. A trigram model is a probabilistic model that can be used to predict if a certain sequence of three items is expected or not. We use the perplexity score for this prediction. The perplexity is the inverse probability, normalized by the number of words. It can be interpreted at how 'perplex' the model is to see a certain sequence. With a perplexity score of zero, the model is not perplexed at all. This indicates a high chance of a existing sequence. With a high perplexity score, the model is 'perplex' about the score, indicating that it is less likely to be a correct sequence. An example of perplexity scores on word level is shown in Table 4.

Table 4: Example of perplexity

Word	Perplexity	Existing word
fish	0	yes
flurp	105	no

To be able to use the trigram measure, we needed a trigram model with probabilities per trigram. We created this model based on our training set. Since we have a historical data set that may contain words that occur very seldom, we decided to use a character level trigram instead of a word level trigram. This means that our trigram sequences consist of a sequence of three contiguous characters. We used our trigram model to calculate the perplexity per word. If the perplexity was equal to or lower than the reference value (see section 3.3), the trigram value was set to 1. If otherwise, the value was set to 0.

3.3 Creation of reference sets

Every language and time period has its own characteristics for written texts. Therefore, it is important to have reference values specifically geared towards these characteristics. These reference values can be used as cutoff or boundaries to determine whether the outcomes of the QuPipe measures are within the expected range.

Since the Ground Truth consists of nearly error-less texts, we used the Ground Truth sentences from the training set to obtain our reference values. First, we calculated the outcome values of every measure for every sentence of this set. The resulting values were ordered and the cutoff and boundaries were chosen to include 90% of the data, as detailed below.

For the **dictionary lookup**, both 'token' and 'type' measures, we used the values based on the combined dictionary. The reference value was selected as the value corresponding to the cutoff point at 10% of the data. This means that all values above this value are considered as desired. The reference values for the words statistics **mean** and **median** were selected by taking the values corresponding to 0.05% and 0.95% of the distribution. The values in between are considered as the desired values. Since the **sentence length** was artificial set to a minimum of 7 words, this is considered as the lowest reference value. The highest value was then selected as value for the cutoff point at 90% of the data. The values in between are considered as the desired values. For both the **garbage** and **trigram**, the reference value was selected at the cutoff point at 90% of the data. This means that all data beneath this value is considered as desired.

3.4 Calculation of the total score of QuPipe

There are various ways in which QuPipe can be used to determine the total score. Which method to choose, depends on the needs of the user. In this section, we describe the three approaches we used for our study: the total calculation, the 'smart quality' calculation and the 'smart quantity' calculation. As this is a first exploratory study on the impact of various methods for determine the OCR quality, the proposed methods are arbitrarily chosen and based on an inspection of a small manual sample. Both smart calculations need to be pre-trained on Ground Truth data. For this study, we focused our calculations on the division between 'good' OCR quality and 'not good' OCR quality.

The normal calculation is the most basic calculation and uses all of the current implemented measures. As described in section 3.2, every measure results in a score of 0 or 1. A score of 1 means the measured value was in the expected range, whereas a value of zero indicates it was not in the expected range. For the normal calculation, these scores are added up to one final score. Only sentences from which the final score is equal to the maximum achievable score is considered as 'good', whereas the rest is considered as 'not good'.

We introduce two other methods to explore if modification of the selected measures changes the output of the calculation. We introduce the 'smart quality' measure that focuses on precision, and the 'smart quantity' measure with a focus an recall. Due to the variety of dictionary measures and the fact that dictionaries are not always available, we exclude these measures during this calculation.

For the 'smart quality' calculation, the most restrictive approach was used to classify a sentence as 'good' or 'not good', meaning that sentences were only classified as 'good' if the final score is equal to the maximum achievable score. This maximum achievable score is based on the number of used measures with every step. This method will lead to the highest achievable precision. To calculate the optimal combination of the 'smart quality' calculation, we started with calculating the precision for all isolated measures. We select the measure with the highest precision. Then, we calculate the precision of this measure with every single other measure. From this, we selected the two measures with the highest precision, but only when the precision is equal to or higher than the current precision. We continued these steps until there was no further increase in precision. An illustration of this process is shown in Figure 4.

The calculation for the best combination of measures for the 'smart quantity' calculation differs slightly from the 'smart quality' measure. For this measure, an sentence was classified as 'good' when its final score was equal to half of the maximum achievable score. For odd numbers, the maximum achievable score was rounded down. With this method, there is more recall while still providing a high precision. This method also starts with the single measure, after which one by one other measures are added. Furthermore, instead of only looking at precision, we chose the measure with the highest recall having a precision that was equal to or above the current precision. We continued these steps until there was no further increase in recall without decreasing the precision. An illustration of this process is shown in Figure 5.

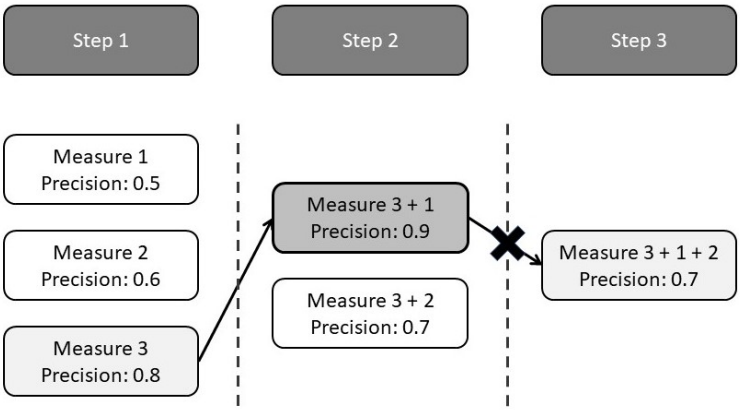

Figure 4: Illustration of the process for the 'smart quality' measures calculation: the optimal score per step (light grey) leads to the total optimal score (dark grey)

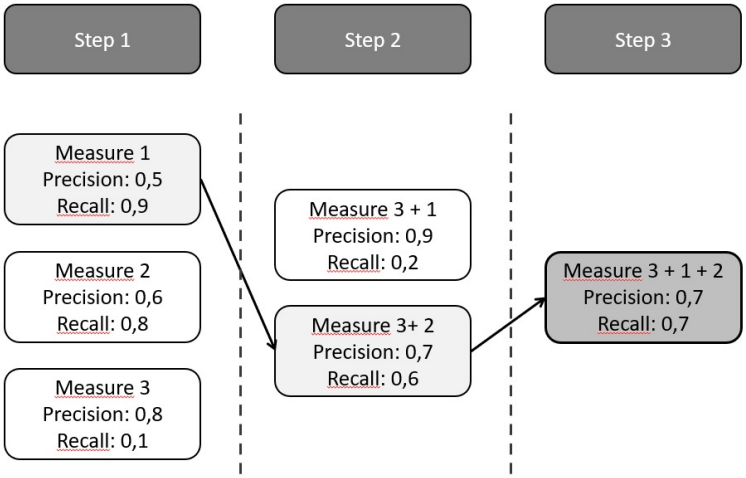

Figure 5: Illustration of the process for the 'smart quantity' measures calculation: the optimal score per step (light grey) leads to the total optimal score (dark grey)

4 Results

After preparing the data and implementing the measures in QuPipe, we performed a series of experiments to evaluate the performance of QuPipe. The first experiment compares the results of a 'token' dictionary lookup (which is the most common used dictionary lookup) with the results of the QuPipe calculations. We used a modern dictionary, a historical dictionary and a combined dictionary (see section 3.2). The second experiment focuses on classifying the OCR quality without a dictionary. Since historical dictionaries are not always available, it is valuable to know if usable predictions can be done without one. For this experiment, we compare two isolated measures with the results of QuPipe. The reference values that were used for the QuPipe measures are shown in Table 5.

Table 5: Reference values

Measure	Reference value
Dictionary lookup token	84.6%
Dictionary lookup type	83.3%
Word length median	between 3 and 6
word length mean	between 4.15 and 6.25
Sentence length	between 7 and 31
Garbage	0.08
Tri-gram	46

For the QuPipe total score, all measures were used. To obtain the best combination of measures for the 'smart quality' calculation, we calculated the combination of measures with the highest precision. For the 'smart quantity' calculation, we calculated the combination of measures that represents the best trade-off between recall and precision. Both were determined using a sample of the training set, as described in section 3.4. Based on these calculations, we excluded the sentence length measure from the smart calculations, as it had a negative influence on the precision. All other measures were kept. For section 4.1 we added the various dictionary measures to the smart calculations.

We calculated the precision and recall of the classification from QuPipe based on the CER classification (see section 3.1).The precision and recall were used to compare the various measures and calculations.

We conclude the results with a comparison of reference values with various amounts of Ground Truth data.

4.1 Results with dictionary lookup

Since historical dictionaries are not always available, we compare the use of a modern dictionary, a historical dictionary, and a 'combined' dictionary (with both historical and modern words). Furthermore, we compared the QuPipe outcomes with the outcomes of a 'token' dictionary lookup with the cutoff point at 80%, based on the suggested minimal accuracy in literature (Strange et al., 2014; Van Strien. et al., 2020). The results are shown in Table 6.

For all three dictionaries, the QuPipe 'smart quality' calculation returned the highest precision with a precision above 0.9, which means that more than 90% of the sentences were correctly classified as 'good'. However, for all three dictionary types, the recall is low. For the modern dictionary, the recall is only 0.008, meaning that only 0.8% of all 'good' sentences were classified as good. The highest recall is for the combined dictionary, where 14.2% of all 'good' sentences were classified as good. When looking at the recall for the modern dictionary, the QuPipe 'smart quantity' calculation gives the best result in terms of both precision and recall compared to the 'token' dictionary lookup. For both the historical and the combined dictionary, the 'token' dictionary lookup provides the highest recall in comparison to the QuPipe results.

When using a modern dictionary, there is only a small difference between the QuPipe total calculation and the 'smart quality' calculation. Both have a high precision, but a very low recall. If we compare this to the 'smart quantity' calculation, we see that while the precision decreases around 0.095, the recall increases with 0.255.

For the historical dictionary, the QuPipe 'smart quality' calculation has a 0.01 higher

precision and a 0.021 higher recall than the QuPipe total calculation. If the 'smart quality' calculation is compared with the 'smart quantity' calculation, the precision decreases with 0.074. However, the recall increases from 0.139 tot 0.414. This means that from all 'good' sentences, the correct classified sentences went from 13.9% to 41.4%. For the combined dictionary, the QuPipe 'smart quality' calculation has a higher precision and a higher recall than the total calculation. If the 'smart quality' calculation is compared with the 'smart quantity' calculation, the precision decreases with 0.075. However, the recall increases from 0.142 to 0.418.

If we compare the three 'token' dictionary lookup methods, the modern dictionary performs the best on precision. However, it performs the worst on recall, with a recall of only 0.050, which means that from all sentences classified as 'good', only 5% was predicted correct. When comparing the historical and the combined dictionary, the historical dictionary has a slightly higher precision (+0.003), whereas the combined dictionary has a slightly higher recall (+0.008).

Table 6: Dictionary lookup versus QuPipe

Type	Outcome	80%	QuPipe total	'smart quality'	'smart quantity'
Modern dictionary	Precision	0.837	0.933	**0.935**	0.840
	Recall	0.050	0.008	0.008	**0.263**
Historical dictionary	Precision	0.821	0.897	**0.907**	0.833
	Recall	**0.462**	0.118	0.139	0.414
Combined dictionary	Precision	0.818	0.896	**0.906**	0.831
	Recall	**0.470**	0.119	0.142	0.418

4.2 Results without dictionary lookup

Since dictionaries are not always available, especially for older texts, we also performed an experiment to see how the pipeline performs without dictionary lookup. We started by calculating the precision and recall for every isolated measure, as shown in Figure 6.

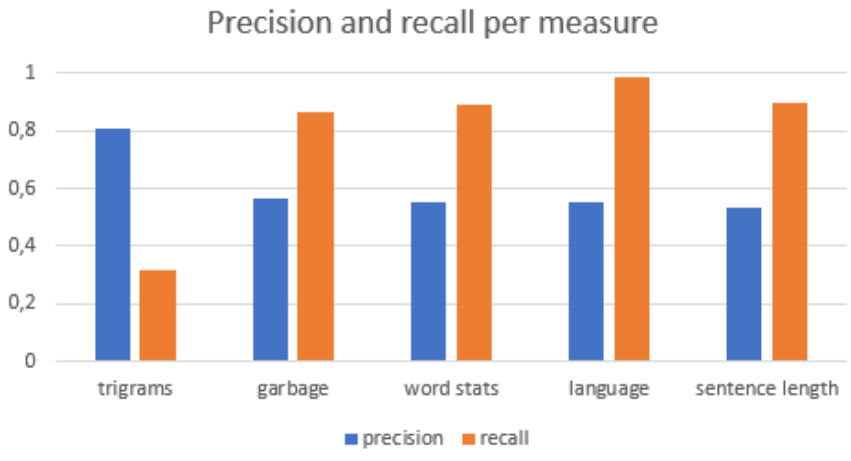

Figure 6: Precision and recall per measure

We first compare QuPipe's calculations with the measure with the highest precision, and then with the measure with the highest recall.

4.2.1 Quality before quantity

The measure with the highest precision is the trigram measure, with a precision of 0.810, which indicates that 81% of the data is correctly classified as 'good'. However, the recall is 0.317, which means that only 31.7% of all 'good' sentences were classified as good. Table 7 shows the precision and recall of the trigram measure, the QuPipe total calculation and the QuPipe 'smart quality' calculation.

With the QuPipe total calculation, we see that the precision increases from 0.810 to 0.823 (+0.013). This means that more data is correctly classified as 'good'. However, the recall decreases from 0.317 to 0.209 (-0.108), which means that less 'good' sentences were classified as good in total. Looking at the 'smart quality' calculation, we see that the precision increases further, leading to a difference of 0.029 in precision between the trigram measure and the 'smart quality' calculation. Also, there is a smaller decrease in the recall, with a difference of 0.071 compared to the trigram measure. When looking purely at quality, the QuPipe 'smart quality' calculation performs best.

Table 7: Highest precision measure versus QuPipe

Outcome	Trigram measure	QuPipe total	QuPipe 'smart quality'
Precision	0.810	0.823	**0.839**
Recall	0.317	0.209	0.246

4.2.2 Quantity before quality

If we look at quantity before quality, the measure with the highest recall is the language detection measure, with a recall of 0.982. This means, that 98.2% of all 'good' sentences were classified as good. However, this measure has a precision of only 0.552, which means that almost 50% of the data is incorrectly classified as good. Therefore, this measure is not much more effective than just using all the data without classifying. Table 8 shows the precision and recall of the language detection measure, the QuPipe total calculation and the QuPipe 'smart quantity' calculation.

When we use QuPipe total calculation, we see that the precision strongly increases from 0.552 to 0.823 (+0.271), however, the recall strongly decreases from 0.982 to 0.209 (-0.773). When we look at the QuPipe 'smart quantity' calculation in comparison to the language detection measure, the precision increases with 0.034 and the recall decreases with 0.149. Looking at as high as possible quality while also taking quantity into account, the QuPipe 'smart quantity' calculation performs slightly better than the language detection measure.

Table 8: Highest recall measure versus QuPipe

Outcome	Language detection measure	QuPipe total	QuPipe 'smart quantity'
Precision	0.552	0.823	0.586
Recall	**0.982**	0.209	0.833

4.3 References and lack of Ground Truth

Since Ground Truth data is sparse and expensive to obtain, we calculated and compared reference values based on various amounts of Ground Truth data. We started with

our 'base' reference values, which were calculated on the whole training set. Then, we calculated reference values with 25% of the Ground Truth and with 2.5% of the Ground Truth. Table 9 shows the difference between the values. The table shows that the reference values for the 'type' dictionary lookup, the word length median, the sentence length, and the garbage measure stay the same although the amount of Ground Truth is reduced. The reference values for dictionary lookup 'token', word length mean, and trigram show a small shift when they are based on a smaller amount of Ground Truth data.

Table 9: Reference values for different amounts of Ground Truth data

Reference	Full data 75576 sentences	25% of data 18894 sentences	2.5% of data 1890 sentences
Dictionary lookup 'token'	84.6%	84.6%	84.4%
Dictionary lookup 'type'	83.3%	83.3%	83.3%
Word length median	between 3 and 6	between 3 and 6	between 3 and 6
Word length mean	between 4.15 and 6.25	between 4.17 and 6.25	between 4.17 and 6.21
Sentence length	between 7 and 31	between 7 and 31	between 7 and 31
Garbage	0.08	0.08	0.08
Trigram	46	45.5	45.2

5 Conclusion and future work

Our study showed that when quality is the most important aspect, the QuPipe 'smart quality' calculation results in the highest precision when measuring OCR quality. This confirms our hypothesis that a combination of measures gives a more accurate prediction of OCR quality than a single measure, although sometimes the difference with the other methods is small.

Although the 'smart quality' measure has the highest precision, in all cases this is at the expense of recall. When only a high quality is important this is no issue. However, in most cases, quantity will also be taken into consideration as most analyses depend on having a big enough dataset. In such cases, one can opt to use the 'smart quantity' calculation, which has a lower precision but a higher recall. Even though it focuses on getting the highest quantity, in all cases the precision is still higher than when using a single measure. When compared to the historical and combined dictionary, the difference in precision and recall is small. However, when only a modern dictionary is available, QuPipe performs clearly better on both precision and recall. We also noticed that although the performance is not quite so good with the QuPipe total calculation, the precision of this calculation is still better compared to using a single measure.

A disadvantage of QuPipe is the need for a Ground Truth set to be able to create the reference sets. However, this only has to be done once for each language and/or time period. From then on, the created reference set can be used on all data with the same characteristics. Additionally our small experiment suggests that reliable reference sets can be obtained from only a small amount of Ground Truth, but further research is needed to confirm this. The same disadvantage is present for the QuPipe smart calculations. These also need a (small) Ground Truth set to optimise the combination of used measures. However, given the improved performance compared to the total calculation, and the fact that there is also Ground Truth needed for the reference sets, it seems to be worth the investment. Especially because some measures can unintentionally have a negative effect on the total result.

Since these first experiments with QuPipe seem positive and validate the idea that a combination of measures works better than using just one measure, we will continue improving QuPipe. Our next steps will be to expand and improve some of the measures, such as the garbage measure and the language detection. Then, we plan to examine new measures to be added to the pipeline, such as letter ratio and word embedding techniques. In addition, we want to add more variation in outcome instead of just 'good' or 'not good'. Furthermore, we want to investigate if reference values based on a small amount of Ground Truth are as reliable as those based on larger amount of Ground Truth. We also want to look at possible other smart calculation options and improve our current methods, thereby creating a more robust approach based on more extensive analyses and data. For both the references and calculation, we aim to explore if the type of material has an influence on the outcomes. Lastly, we want to expand our experiments to other data sets, with other types of text, different time periods and, if possible, other languages.

References

Farag Ahmed, Ernesto De Luca, and Andreas Nürnberger. Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness. *Polibits*, 40: 39–48, 12 2009. doi: 10.17562/PB-40-6. http://dx.doi.org/10.17562/PB-40-6.

S. El Atawy and A. Abd ElGhany. Automatic spelling correction based on n-gram model. *International Journal of Computer Applications*, 182:5–9, 08 2018. 10.5120/ijca2018917724.

Ryan Baumann. Automatic evaluation of OCR quality. https://ryanfb.github.io/etc/2015/03/16/automatic_evaluation_of_ocr_quality.html, 2015.

Giovanni Colavizza and Mirjam Cuper. Is your OCR good enough? A comprehensive assessment of the impact of OCR quality on downstream tasks [Data set]. http://doi.org/10.5281/zenodo.4498186, 2021.

Mirjam Cuper. Examining a multi layered approach for classification of OCR quality without Ground Truth [presentation]. https://zenodo.org/record/4621253#.YWXrFBxcKUk, 2021.

Rose Holley. How good can it get? analysing and improving ocr accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine: The Magazine of the Digital Library Forum*, 15, 01 2009.

IMPACT Centre of Competence. ocrevalUAtion. https://github.com/impactcentre/ocrevalUAtion, 2019.

Instituut voor de Nederlandse taal. INT Historische Woordenlijst. https://taalmaterialen.ivdnt.org/download/tstc-int-historische-woordenlijst/, 2012.

Scott Kulp and April Kontostathis. On retrieving legal files: Shortening documents and weeding out garbage. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007. URL `http://trec.nist.gov/pubs/trec16/papers/ursinus.legal.final.pdf`.

Marco Lui. Language Identifier. https://github.com/saffsd/langid.py, 2011.

Thi Tuyet Hai Nguyen. *Facilitating Access to Historical Documents by Improving Digitisation Results*. Theses, La Rochelle Université, April 2020. URL `https://hal.archives-ouvertes.fr/tel-03058611`.

Open Taal. Nederlandse woordenlijst. https://github.com/OpenTaal/opentaal-wordlist, 2020.

Python Software Foundation. difflib — Helpers for computing deltas. https://docs.python.org/3/library/difflib.html, 2022.

Alexander M. Robertson and Peter Willett. Applications of n-grams in textual information systems. *J. Documentation*, 54:48–67, 1998. https://doi.org/10.1108/EUM0000000007161.

U. Springmann, Florian Fink, and K. Schulz. Automatic quality evaluation and (semi-) automatic improvement of mixed models for ocr on historical documents. *CoRR*, abs/1606.05157, 06 2016.

Carolyn Strange, Daniel McNamara, Josh Wodak, and Ian Wood. Mining for the meanings of a murder: The impact of ocr quality on the use of digitized historical newspapers. *Digital Humanities Quarterly*, 8:1–17, 01 2014.

Kazem Taghva, Tom Nartker, Allen Condit, and Julie Borsack. Automatic removal of "garbage strings" in ocr text: An implementation. *The 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, 01 2001.

Myriam C. Traub, Jacco van Ossenbruggen, and Lynda Hardman. Impact analysis of ocr quality on research tasks in digital archives. In Sarantos Kapidakis, Cezary Mazurek, and Marcin Werla, editors, *Research and Advanced Technology for Digital Libraries*, pages 252–263, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24592-8. https://doi.org/10.1007/978-3-319-24592-8_19.

Daniel Van Strien., Kaspar Beelen., Mariona Ardanuy., Kasra Hosseini., Barbara McGillivray., and Giovanni Colavizza. Assessing the impact of ocr quality on downstream nlp tasks. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: ARTIDIGH,*, pages 484–496. INSTICC, SciTePress, 2020. ISBN 978-989-758-395-7. doi: 10.5220/0009169004840496.

Jian-Cheng Wu, Jim Chang, and Jason J. S. Chang. Correcting serial grammatical errors based on n-grams and syntax. *Int. J. Comput. Linguistics Chin. Lang. Process.*, 18, 2013.

Richard Wudtke, Christoph Ringlstetter, and Klaus U. Schulz. Recognizing garbage in ocr output on historical documents. In *MOCR-AND '11: Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, MOCR_AND '11, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306850. doi: 10.1145/2034617.2034626. https://doi.org/10.1145/2034617.2034626.