Digging Through Garbage: Detection of 'Garbage' Words in Digitized Historical Documents

Mirjam Cuper[1], and Ethan den Boer[2]

[1] Research Department, KB, The National Library of The Netherlands, The Hague, The Netherlands
[2] Independent researcher, Rotterdam, The Netherlands

Abstract

Digitized historical heritage is widely available, and the need for high-quality material is increasing. However, there are still some unsolved Optical Character Recognition (OCR) problems. Approaches have arisen to correct such problems after digitization, with the use of automatic post-processing. But sometimes the OCR output is of such low quality that these post-processing methods are not an option. To detect this 'garbage' output, various 'garbage detection' methods have been developed, based on the English and German language. We expand upon these methods by developing a garbage detection method tailored to 17th century Dutch. Using a dataset of 6,245 17th century Dutch Newspapers for which we have both the original OCR output and the manually corrected transcriptions available, we compare various rule-based and machine learning methods. We developed a semi-automated method to create a labeled dataset for training, and we created a rule-based method tailored to 17th century Dutch. Our results show that various machine learning models outperform rule-based methods. We made our models publicly available for use by both researchers and heritage institutions.

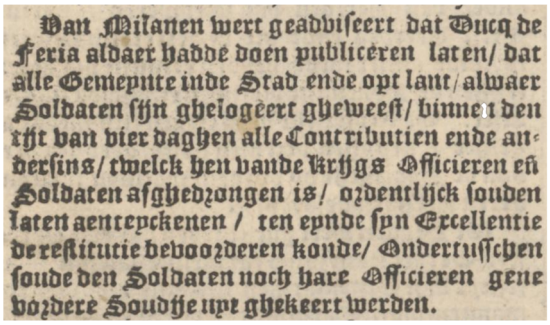
Keywords: Digital Heritage, OCR Quality, Garbage Detection

1 Introduction

Although more and more heritage is digitally accessible through, among others, the use of Optical Character Recognition (OCR), the provided OCR output of this material is of variable quality. Low OCR quality can cause problems for both information retrieval and computational text analysis [**stien**; **bazzo**]. In some cases, this can be mitigated by post-processing the digitized texts with computational methods or by manual correction. While the latter is more precise, it is a high-cost method and usually not feasible. Therefore, automatic post-processing methods are widely used [**nguyen**].

However, for some types of material the OCR output is of such a low quality that automatic post-correcting is not possible. In these cases (large) parts of the OCR output consists of illegible words. These types of OCR output can be considered as 'garbage'. The chance of obtaining garbage depends on certain characteristics of the source material. Firstly, certain fonts increase the chance of obtaining garbage OCR output. This is especially true for older OCR software, as these were generally trained on Latin fonts, leading to difficulties when used on other fonts, such as Gothic (see figure 1). Material that is OCR-ed with the use of this older software is still present in digital archives. Fortunately, for newer OCR software these types of fonts are less of a problem. Secondly, when the original source material is badly damaged due to, for example, stains, mold, or cracks, there is a higher chance of garbage output. Thirdly, graphical material, such as advertisements, is especially challenging for OCR software, which increases garbage output (see figure 2) [**Wevers**].

Mirjam Cuper, and Ethan den Boer. "Digging Through Garbage: Detection of 'Garbage' Words in Digitized Historical Documents." In: *Computational Humanities Research 2025*, ed. by Taylor Arnold, Margherita Fantoli, and Ruben Ros. Vol. 3. Anthology of Computers and the Humanities. 2025, 1473–1483. https://doi.org/10.63744/wD9bYr0WxUTa.



Van Milanen wert geadviſeert dat Ducq de Feria aldaer hadde doen publiceren laten/ dat alle Gemeynte inde Stad ende opt lant/ alwaer Soldaten ſijn gheloogeert gheweeſt/ binnen den tijt van vier daghen alle Contributien ende anderſins/ twelck hen vande Krijgs Officieren eñ Soldaten afgheeyzongen is/ ozdentlijck ſouden laten aentepckenen / ten eynde ſyn Excellentie de reſtitutie bevooꝛderen konde/ Ondertuſſchen ſoude den Soldaten noch hare Officieren gene bozdere Soudije upt ghekeert werden.

```
$an jfl9ttanen toert geabbifeert bat
©neg be jfetia albaer Ijaöbc boen
publicéren (aten /bat alle «öcmepittc
inbe jètab ciiücoptlaittaUuacr
jsolbaten fijn g&eiogeett gyciucefl /
binnen ben t#t ban bier bagljen
alleContributicn enbe an< Dcrfina/
ttoelck bcnbanbcteljga officieren
eft ^5>oIöatenafgi)eb|oitgen  is /
ojbcntltjck foubcn  latenaentepekenen
/ ten epnbc fpn <2jetellentie
bere|litHticbcbaa?bcrcn koitbe /
«önbcrtuffcijeit foube ben
j&olbatenitocf) ttare Officieren gene
toojbere .éoubfje tipt gljekeert
toerben.
```

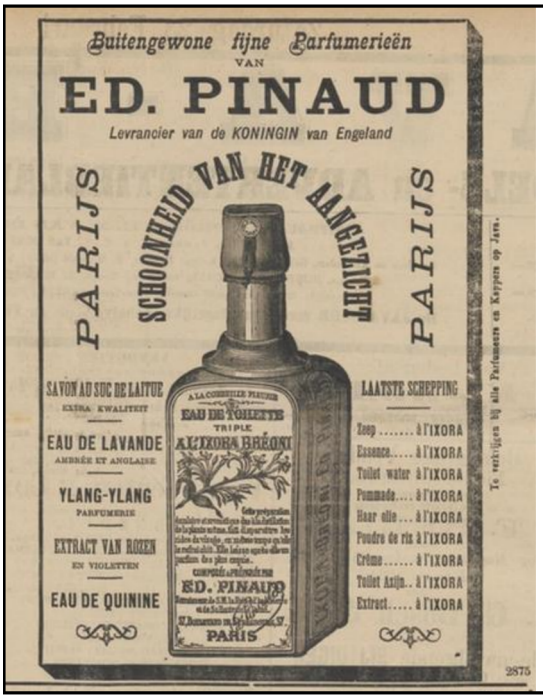
Figure 1: Example of garbage OCR output from a newspaper from 1626 with a non-latin font

To obtain an indication of OCR quality, lexicon comparison is often used. [**strange**; **stien**]. Although this method is quite good at distinguishing OCR output of high quality and average to low quality, it has a major drawback. When the output, for example, consists of 50% small OCR-errors that can easily be resolved with automatic post-processing, it is put in the same pile as output that consists for 50% of garbage words. Separating OCR output into garbage and non-garbage could be used to decide how to proceed with post-processing. When there is a lot of garbage, automatic post-processing is probably not an option, leaving manual correction or re-OCRing with newer or specialized software as the only solutions.

Previous research has shown that garbage in texts can be detected automatically. [**taghva**] pioneered a rule-based system that utilized generalized rules based on the target language, demonstrating the promise of simple rule-based systems. [**kulp**] expanded on this research, adding rules and adjusting some rules to better fit their corpus. [**wudtke**] introduced a novel method, using a Support Vector Machine to detect garbage. The definition of garbage used in this study was more strict, and non-severe errors were not seen as garbage. Because the method of labeling is not described, we assume that the classifier was trained on manually labeled data. The performance of the Support Vector Machine was compared to the two rule-based methods described by [**taghva**] and [**kulp**], and outperformed the rule-based methods for two out of three experiments. Rule-based garbage methods based on the aforementioned studies have recently been applied by two heritage institutions [**cuper**; **lib_luxembourg**] that are developing automatic ways to determine the quality of their digitized collections. This shows that there is an interest in garbage detection methods.

The methods developed by [**taghva**], and [**kulp**] were based on the English language, whereas the method by [**wudtke**] is based on German. We aim to create a classification method for the detection of garbage tailored specifically to 17th century Dutch. We chose this time period, because of the higher prevalence of Gothic fonts in newspapers and books, increasing the chance of garbage output.

Although a lot of recent research includes deep learning, neural networks, LLMS and generative AI, we deliberately did not use these methods in our research but rather focus on more traditional machine learning methods. We follow the general rule derived from Knox's [**knox2018machine**] translation of Occam's Razor to machine learning models: use an as simple model as possible for the task at hand if this model has good enough performance for that task. In addition, as computationally expensive models have a greater impact on the environment [**rozycki**], we also find it preferable to use less computationally expensive models. As our




```
(Buitengewone fijne parfumerieën g ED.
PINAUD I Levrancier van de KONINGIN
van Engeland S .# gsm* <&r s f h BB h
gj SiTOIIAN-MLilTDE^^ÜÜ^^^^n LAATSTE
SCHEPPING 1 \ ⊠xtba _._uip |
EsSStSBiiB ⊠⊠ 1 i pab™_ gZ_!^K_3__a
HH Haar 01ie... srixoßA 1 pvtdipt ïïih
Dn7i.Tï _to_-_--$.,-_.m_--__ï-qi-_-fl
lliHii I rOU_U_ BB HZ ül IXORA n.
tAltlAul ÏAll ItUAtll i_n_-_d___.__e__
..e»ïirils---)-- I I ($3 EN VIOLKTTEN
 |Cr6lllfl "HOR» I EAU DE QUININE
fmmmgm^W, WÊÈ Extract aMORA É
```

Figure 2: Example of garbage OCR output from an advertisement from 1884

study focuses on a simple two-class classification task, we believe that this can be performed by traditional machine learning algorithms with a lower environmental impact.

We use a dataset of 17th century Dutch news articles, for which we have both the original OCR and the manually corrected transcriptions. To determine the best performing garbage detection method, we used our dataset to tailor a rule-based method to 17th century Dutch and to train various traditional machine learning models. For a thorough comparison we also included the rule-based methods as designed by [**taghva**] and [**kulp**]. Our models and data are published on GitHub.[1]

2 Data

2.1 Corpus presentation

Our dataset consists of 17th century Dutch newspapers, ranging from 1615 till 1699, and contains a total of 34,808 news articles from 6,425 newspapers. The quality of the set varies, and much of the original material is of low quality with, for example, run-through ink and stains. In addition, due to the time period, there is a high percentage of Gothic fonts. For this dataset we have both the original, non-corrected, OCR available (digitized by the KB,The National Library of The Netherlands) as well as the manually corrected 'Ground Truth' by the Meertens Institute [**Meertens**].

2.2 Data cleaning

Even though the Ground Truth is created manually, it is not always faultless. Some textual errors persist or were introduced by accident and there are instances of unwanted annotation to indicate unreadable passages or special tokens. As these introduce noise in our Ground Truth, we cleaned the Ground Truth using the following steps:

- All occurrences of *&* are replaced with *&*;

[1] https://github.com/KBNLresearch/DiggingThroughGarbage

- All different variants of apostrophes are replaced with one utf-8 version;
- If the word ends with double punctuation marks, the last one is removed;
- Some words contain *[...]* as marker for unreadable text, these words were removed;
- All words that contain *{tab}*, an equal sign (=) or a plus symbol (+) are removed;
- All words that contain a random occurrence of a comma (,), stop (.), colon (:), or semicolon (;) are removed.

After these steps, we removed all starting and ending Dutch punctuation marks from both the Ground Truth and the OCR words (see table 1). Finally, we removed all words that were numeric.

	Punctuation
start	‘ ” ([
end	. ? ! , ; : - ” ’)]

Table 1: Removed punctuations.

OCR-ed word	**Corresponding GT word**	**Garbage**
gpepjefenteect	ghepresenteert	no
vacantiu:	vacantie	no
«ugcncii.Vaa	?	yes
W-,ntw!lß	?	yes

Table 2: Examples of applying labels.

2.3 Data annotation

We created a labeled dataset as input for our machine learning algorithms. For labeling words as either garbage or non-garbage, we adapted the definition of garbage used by [**wudtke**]. They define a word as garbage when it is an erroneous word for which the correct word cannot be predicted by examining the OCR-ed word in isolation. Following this definition, words that contain small OCR-errors are not considered as garbage. Table 2 gives an example of when we consider a word as garbage and when not.

Because our dataset was quite large, we developed a semi-automated labeling approach. Using a set of 1,197 words which were randomly extracted from our original OCR, two volunteers labeled these words as either garbage or non-garbage following the aforementioned criteria. As our dataset consists of historical news articles, which can contain spelling that deviates from modern spelling, we also provided the manually corrected transcription. This could be used to help determine if a word was garbage or a historical spelling variant.

For each manually labeled word, the normalized Levenshtein distance to each word from the corresponding Ground Truth article was calculated using the *edit_distance* function from the Natural Language Toolkit [**nltk**]. We normalized this distance by dividing it by the length of the largest word. The smaller this distance, the more the two words are alike. For every word, the smallest calculated Levenshtein distance was selected.

The labels obtained from the volunteers were combined into three categories: ’garbage’ or ’non-garbage’ where both volunteers agreed, and a ’doubt’ category where they had classified a

word differently. The categories were plotted against the Levenshtein distance obtained for each word (figure 3). 494 words were labeled as garbage and 540 words as non-garbage. The remaining 163 words made up the ‘doubt’ category. These three categories were then used to determine cut-off values for the Levenshtein distance to differentiate between garbage and non-garbage words in our training and test set. As the goal was to determine words that are definitively garbage or not, we omitted the ‘doubt’ category. To minimize overlap between categories, the cut-off points for our garbage and non-garbage categories were determined by taking the Levenshtein distance corresponding to the bottom and top 5% of the ’doubt’ category in the labeled dataset. This led to cut-off points of a Levenshtein distance lower than 0.127 for non-garbage and greater than 0.588 for garbage.

After determining the cut-off points, we used the aforementioned method to calculate the smallest Levenshtein distance for the remaining OCR words. This distance was then used to label words as garbage, non-garbage, or to omit the word. To ensure that our garbage labeling was strict enough, we checked a random sample of 100 words. None of these words were labeled erroneously.

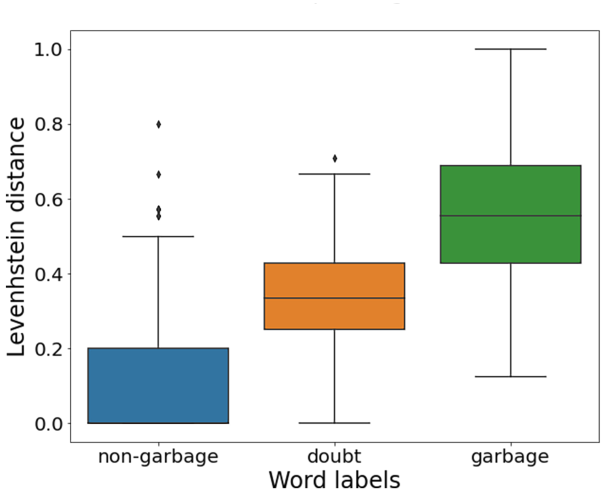


Figure 3: Labeled words plotted against minimal Levenshtein distance

Set	Garbage words	Non-garbage words
training set	148,357	174,036
test set	63,582	74,588
GT words	0	222,633

Table 3: Number of garbage and non-garbage words per set.

2.4 Dataset partitioning

After labeling, the dataset was divided into a dataset of unique Ground Truth words and a dataset of unique OCR-ed words by splitting the articles on space. The unique Ground Truth words were used to determine the rules for our own rule-based method. The OCR-ed words were divided into 70% for training the machine learning algorithms and 30% for testing the performance of both the classifiers and the rule-based methods. Table 3 shows the resulting amount of labeled words per category.

3 Garbage detection approaches

We compared various methods on their ability to classify words as garbage or not. These consisted of three rule-based approaches (section 3.1) and five machine learning algorithms (section 3.2). As the papers from [**taghva**], [**kulp**], and [**wudtke**] did not provide any code, we developed our own code based on their descriptions. The labeled dataset, the code for the rule-based methods and our trained models are publicly available on Github.[2]

3.1 Rule-based approach

Although the rule-based methods described by Taghva et al. [**taghva**] and Kulp & Kontostathis [**kulp**] are promising, they are based on the English language. As our dataset contains only 17th century Dutch articles we wanted to develop a rule-based method tailored to Dutch. We used the rules from both [**taghva**] and [**kulp**] as a starting point, but modified these to make them more suitable to Dutch. To achieve this, we performed a statistical analysis on the characteristics of the words in our unique Ground Truth dataset to determine the best cut-off values per rule. These were compared with the values used by [**taghva**] and [**kulp**]. Where these values differed, we used the value obtained from our dataset. We also added two new rules. A detailed overview of the rules for our Dutch rule-based method is included in Appendix A. To determine if a rule-based method tailored to a specific language works better, we tested our rule-based method against the methods of [**taghva**] and [**kulp**].

3.2 Machine Learning approach

We trained the following traditional machine learning algorithms: Decision tree, Support Vector machine (linear kernel), K-Nearest Neighbours, Naive Bayes and Random Forest. The SKLearn package [**scikit-learn**] version 1.6.1 was used for all algorithms.

3.3 Feature creation

For our study we used the features proposed by [**wudtke**] as a starting point. We then implemented some alterations to the features as well as adding some of our own. The alterations were partly to optimize the features to the Dutch language, and partly because some of the features by [**wudtke**] already implemented rules whereas we prefer to use purely descriptive features. We therefore altered features that included any form of decision making or scoring to merely descriptive features. With the exception of the length of the word and the number of consecutive characters, we chose to use only ratios and quotients for the occurrence of characters, as the absolute number would be strongly influenced by the length of the word.

For our Dutch character list, we included all vowels and consonants, including their diacritics, that can occur in Dutch words. Furthermore, we added some common punctuation, such as the dash (-), apostrophe (’) and slash (/). We treated the letter ‘y’ as a vowel, as in historical Dutch this is typically used instead of the modern ‘ij’. The complete list of features is shown in Appendix B.

4 Experimental setup

We used the MinMax scaler from the SKlearn package [**scikit-learn**] to ensure that all features are weighed similarly, preventing the possibility of uneven contribution during training. This scaler normalizes all features by transforming them to a scale ranging from 0 to 1. The MinMax scaler was selected because of its compatibility with the Naive Bayes algorithm. Each machine learning algorithm was trained on the train data using all features.

[2] https://github.com/KBNLresearch/DiggingThroughGarbage

The performance of all classifiers was measured on the same test set. Before applying the trained machine learning classifiers, the test set was scaled using the same fitted MinMax scaler as used on the train set. Performance was determined using precision, recall and F1-score.

5 Results

5.1 Performance

Table 4 shows an overview of the trained classifiers and their performance on the test set.

When looking at precision, there are four classifiers with a high score: the Decision Tree (0.952), the Random Forest (0.948), the Taghva rule-based method (0.942) and the Naive Bayes (0.933). However when looking at recall, we see that the recall for the Tagva rule-based method and the Naive Bayes is extremely low (0.386 and 0.582, respectively). The classifier that outperformed all other classifiers based on recall, is the K-Nearest Neigbours (0.971). However, its precision is only 0.506, which is equal to a random guess. To see which classifiers performed the best in general, we look at the F1-score. The two classifiers with the highest F1-score are the Random Forest (0.912) and the Decision Tree (0.907).

When comparing the three rule-based methods, we see that the F1-score of the Dutch rule-based method is the highest (0.792) but still did not perform well. The precision of this method is the lowest (0.737) but the recall is a lot higher (0.856) than that of the Taghva and Kulp rule-based methods (0.386 and 0.509, respectively).

Classifier	**Precision**	**Recall**	**F1-score**
Rules based on Taghva et al.	0.942	0.386	0.547
Rules based on Kulp & Kontostathis	0.894	0.509	0.649
Dutch rule-based methods	0.737	0.856	0.792
Naive Bayes	0.933	0.582	0.717
Decision Tree	**0.952**	0.867	0.907
K-Nearest Neighbours	0.506	**0.971**	0.666
Support Vector Machine	0.888	0.795	0.839
Random Forest	0.948	0.878	**0.912**

Table 4: Classifier scores.

6 Discussion

We compared the performance of machine learning classifiers against the performance of rule-based methods in detecting garbage in OCR output of Dutch 17th century news articles.

Our findings are in line with [**wudtke**] as our machine learning classifiers generally perform better than rule-based methods. We showed that this is even true when the rule-based method is specifically tailored to a specific language and time period. When looking at the rule-based methods, we see that the method tailored to Dutch has a higher recall, but the precision is much lower. This is remarkable as it would be expected that a method tailored to a specific language would perform better across all metrics.

Even though the precision of the two best performing machine learning classifiers is quite high, the recall is still relatively low, missing respectively 13% and 12% of the garbage words. This may be optimized using cross-validation or hyper-parameter optimization.

Our research resulted in various classifiers that are able to determine, with reasonable certainty, if an OCR-ed word is garbage. Although improvements may be possible, we think that our results

are already useful for distinguishing between OCR output with and without a lot of garbage. To be able to determine if OCR output is suitable for automatic post-processing or other types of processing, it is often enough to get an indication of the amount of garbage in the output. For this application it is not needed for all garbage words to be found, as long as the majority is detected.

We made our classifiers publicly available in order for researchers and heritage institutions to be able to use them to detect garbage in OCR-output from Dutch (in particular 17th century) material. If rule-based methods are currently in use, we advise to compare their performance with that of our machine learning classifiers as there is a good chance that they perform better.

However, as our machine learning classifiers were trained on our dataset, which consisted of 17th century newspapers, they need to be tested on other, and preferably, more modern datasets to see how generalizable our approach and the resulting classifiers are.

During our study, we found that the Ground Truth was not faultless. In addition to encoding errors, we found multiple human-made errors. We therefore recommend to stringently check Ground Truth that is used to train classifiers or create any sort of metric. Furthermore, it is good to keep in mind that while this method distinguishes garbage from non-garbage, a word labeled as non-garbage can still contain OCR errors. Finally, during the automatic labeling of our dataset we labeled words as garbage or not based on the closest Levenshtein distance in regards to the Ground Truth. Due to the nature of our automatic Ground Truth cleaning method, some errors may have been missed, possibly leading to wrongly labeled words in the training set. This is expected to be a negligible account.

During our experiments we ran across an interesting finding. We initially used an older version of SKlearn (1.0.2) to train our classifiers, and the obtained K-Nearest Neighbour classifier gave results similar to the other machine learning classifiers (precision: 0.936, recall: 0.88, F1: 0.907). However, when repeating this using an upgraded version of SKlearn (1.6.1), the K-Nearest Neighbour classifier performed considerably worse (precision: 0.506, recall: 0.971, F1: 0.666), whereas the other machine learning classifiers stayed close to their original performance. We were unable to determine the cause of this change and did not expect such a large difference in performance between SKlearn versions.

7 Conclusion

We developed a rule-based method and multiple machine learning classifiers trained on 17th century Dutch. These classifiers are made publicly available and can be used by both heritage institutions and researchers to detect OCR-ed output with a lot of 'garbage' words. Confirming previous research, we showed that machine learning models outperform rule-based methods. Although our method performs good enough for our intended purposes, it may be worthwhile to optimize the classifiers to see if the performance, especially on the recall part, can be improved.

A Dutch rule-based method

Adopted rules:

- If a word is more than 18 characters in length, it is garbage (adopted from [**taghva**; **kulp**] but altered the number of characters);
- If a word contains more than one punctuation character, it is garbage (adopted from [**taghva**; **kulp**] but we do not strip the first and last character as we already remove Dutch punctuation marks during the data cleaning);
- If there are three or more identical characters in a row in a word, it is garbage (adopted from [**taghva**; **kulp**] with the cut-off point as used by [**kulp**]);
- If all the characters in a word are alphabetic, and if the quotient of vowels to consonants is larger than 2, it is garbage (adopted from [**taghva**; **kulp**] but with another ratio);
- If all the characters in a word are alphabetic and the quotient of consonants to vowels is larger than 4 it is garbage (adopted from [**taghva**; **kulp**] but with another ratio);
- If a word has more than 3 consecutive vowels, it is garbage (adopted from [**kulp**], but we shortened the number of vowels);
- If a word has more than 5 consecutive consonants, it is garbage (adopted from [**kulp**]).

New rules:

- If a word has no vowels, it is garbage;
- If a word consists of less than 70% dutch letters, it is garbage