Low-resource OCR error detection and correction in French Clinical Texts

Eva D'hondt
LIMSI, CNRS
Université Paris-Saclay
F-91405 Orsay
eva.dhondt@limsi.fr

Cyril Grouin
LIMSI, CNRS
Université Paris-Saclay
F-91405 Orsay
cyril.grouin@limsi.fr

Brigitte Grau
LIMSI, CNRS, ENSIIE
Université Paris-Saclay
F-91405 Orsay
bg@limsi.fr

Abstract

In this paper we present a simple yet effective approach to automatic OCR error detection and correction on a corpus of French clinical reports of variable OCR quality within the domain of foetopathology. While traditional OCR error detection and correction systems rely heavily on external information such as domain-specific lexicons, OCR process information or manually corrected training material, these are not always available given the constraints placed on using medical corpora. We therefore propose a novel method that only needs a representative corpus of acceptable OCR quality in order to train models. Our method uses recurrent neural networks (RNNs) to model sequential information on character level for a given medical text corpus. By inserting noise during the training process we can simultaneously learn the underlying (character-level) language model and as well as learning to detect and eliminate random noise from the textual input. The resulting models are robust to the variability of OCR quality but do not require additional, external information such as lexicons. We compare two different ways of injecting noise into the training process and evaluate our models on a manually corrected data set. We find that the best performing system achieves a 73% accuracy.

1 Introduction

While most of the contemporary medical documents are created in electronic form, many of the older patient files are kept in paper version only. These files represent an invaluable source of information and experience for medical investigations, especially in domains with low-frequency diseases such as foetopathology, the medical domain which specializes in the treatment and diagnosis of illnesses in unborn children. Over the last two decades, Optical Character Recognition (OCR) technology has improved substantially which has allowed for a massive institutional digitization of textual resources such as books, newspaper articles, ancient handwritten documents, etc. (Romero et al., 2011).

In recent years, hospitals and medical centers have taken to processing older, paper-based resources into digital form in order to construct knowledge bases and resources that can be consulted by medical staff and students. When it comes to documents containing patient information, however, the process of digitization or the use of the resulting text corpus are not as straightforward as they may seem on first sight. Firstly, medical corpora are much less accessible than other general-purpose text corpora since the confidentiality of patients is a first priority. This results in limited access of researchers to original files which in turns directly limits the quantity of files that can be digitized. Secondly, text corpora that contain medical information can only be distributed (even internally in hospitals or research centers) when they are de-identified, that is, when all patient-specific information is identified and removed from the OCRed text (Richards, 2009). This additional processing step can have a significant impact on the quality of the resulting text corpus when information is incorrectly identified as patient-specific information and consequently trans-

Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 61–68,
Austin, TX, November 5, 2016.

formed or removed, e.g. 'Parkinson' in the phrase 'Parkinson's disease'. A side-effect of the obligation of de-identification is that OCR process information is often not available to the researcher using the text corpus afterwards, since it could potentially be used to reconstruct the original information in the paper version. Thirdly, medical files in hospitals are generated over many years. Consequently, the variations in paper, printing techniques or differences in structuring the text (e.g., one-column versus two-column paper formats) can impact the OCR process, and the quality of OCRed files can vary substantially from one year to another (Evershed and Fitch, 2014).

With the increased use of OCR to digitize paper corpora, the problem of OCR error detection and correction has received considerable attention from the research community, especially as regards to its impact on information retrieval and information extraction tasks (Ruch et al., 2002; Magdy and Darwish, 2010). The majority of the current OCR error correction systems use the same three-step approach: (1) OCR error detection; (2) candidate generation; (3) candidate ranking. In the first step, a potential OCR error is detected using either a lookup in a domain-specific lexicon (Kissos and Dershowitz, 2016) or unigram language model (Bassil and Alwani, 2012), and/or by consulting information from the OCR process, i.e., the confidence scores of the recognized characters. The second step, candidate generation, also heavily depends on external resources, either by generating potential candidate replacements for the erroneous words from a lexicon (Piasecki and Godlewski, 2006) or by learning and using a mapping of characters that were often interchanged during the OCR process to generate potential candidates with string distance metrics (Kukich, 1992). Such mappings are known as 'character confusions' but need to be learned over a training corpus of a considerable size before they can become effective (Evershed and Fitch, 2014). The lack of external information such as OCR process information or domain (and hospital)-specific lexicons and the high variability of OCR quality render these systems useless for OCR error detection in medical text corpora.

Unlike the current state-of-the-art systems, the method proposed in this article requires only a sample of (relatively) clean domain-specific text, and no other external information. It uses recurrent neural networks (RNNs) to train character-level language models. By artificially inserting noise into the training data, the system learns to filter out random noise, while learning the domain-specific language model that underlies the documents in the corpus. Since the models do not depend on external resources the method can also be applied to domain-specific text corpora outside the medical domain, on the condition that the documents in the training corpus are not too heterogeneous.

2 Background

OCR and orthography error detection and correction have received interest from the NLP community since the seventies. A good survey of the early work on this problem can be found in Kukich (1992). While most of the traditional OCR error detection systems focused on the construction of so-called 'confusion matrices' of character (pairs) to detect corruptions of existing words into non-words, more recent systems find that using information on the language context in which the error appears improves accuracy (Evershed and Fitch, 2014). A good example of the latter is the system proposed by Bassil et al. (2012) who use extensive n-gram word and 2-character models from the Google Web 1T 5-Gram data set to identify OCR errors and generate and identify the most plausible replacements. Kissos et al. (2016) studied the relative impact of different information sources by combining features from language models constructed over the training corpus, OCR process information and document context information. They found that bigrams, i.e., localized context information was the most useful feature in OCR correction.

To the best of our knowledge, the only existing OCR error detection and correction systems for medical texts focus on either OCR correction for historical text with adapted language models (Thompson et al., 2015) or OCR recognition of handwritten notes by doctors, which is not surprising given the absence of large OCRed text corpora in this domain. Notable work in this area was carried out by Piasecki et al. (2006) who examined the construction of word-level language models to improve OCR correction of Polish handwritten medical notes. They

found that the repetitive character sequences and recurrent structure of medical notes greatly aided the construction of language models but that this positive effect is domain-specific and does not carry over the similar corpora in a different medical subdomain. Like the more generic OCR error detection and correction systems, they also depend on external resources, in this case, an extensive domain-specific lexicon for the detection of errors and generation of candidates.

'Automatic misspelling detection and correction', a subtask related to OCR error detection and correction, has received a lot of attention over the last few years with the increased use of Electronic Health Records (EHRs) in the medical domain. While these tasks have a similar goal, the underlying assumptions are quite different: Character confusions in misspellings are often regular, either due to phonetic misspelling, or due to the proximity of certain letters on a keyboard. OCR errors, however, are often more random and can occur more frequently (Kumar, 2016). Notable work in this domain include Lai et al. (2015) who combine a noisy channel spelling correction approach with an extensive domain-specific dictionary to generate probable misspelling-correction pairs, and Mykowiecka et al. (2006) who use bigram language models to estimate the probability of a misspelling in a given word.

3 Corpus

3.1 Corpus construction

We train and evaluate our system on a data set of French patient notes from the domain of foetopathology. This corpus was assembled and digitized within the context of the Accordys project, and spans a total of 22 years.[1] In total, the corpus contains the files from 2476 individual patients which amounts to 16,573 paper documents. The files were processed with a custom-trained commercial OCR engine, and later de-identified with an in-house de-identification tool (Grouin and Zweigenbaum, 2013). All identifying data were replaced by generic tags with a numerical identifier for all occurrences of the same information in order to maintain the original distribution of tokens along the corpus (e.g., the tag "DATE-8734" was used for all occurrences of "May 21st, 2016"). There is a substantial amount of redundancy in the corpus: For some documents, several (nearly-identical[2]) copies were added to the patient's folder. It should be noted that the patient notes in the corpus are very similar with regards to their contents: the vast majority of the patient files are either reports of the pathologic examination of fetus and placenta or results of genetic tests. While the style and structure of these reports change over time in the corpus, their content—and consequently much of the terminology used—remain stable.

3.2 OCR quality in corpus

Since the model of the OCR engine which was used to convert the entire corpus was trained on a subset of documents of more recent years (implying good paper quality, clear font, no ink problem, etc.), the OCR quality of the OCRed documents decreases substantially for the older documents. In a test set of 100 randomly selected documents from the corpus, we found that 16.4% of the words[3] did not appear in the Unified Medical Lexicon for French (Zweigenbaum et al., 2005), a word list with specific technical terms. Of these 16.4%, 3.8% pertained to words that were domain-specific terms that has been correctly identified in the OCR process but which did not feature in the UMLF, and 10.8% were words which contained at least one OCR error. The remaining out-of-vocabulary[4] (OOV) words were not classifiable. Table 1 shows a representative example of an OCRed document of mediocre quality in the corpus.

3.3 Training set

For the purposes of training the neural network described in section 4, we needed to provide the model with relatively clean data to learn a reliable language model. We used the proportion of OOV words with regards to the number of words in the document as

[1] The files range from 1983 to 2005.

[2] While the original paper documents might be identical, the process of OCR and de-identification has introduced enough noise that very few identical files remain.

[3] We performed simple whitespace tokenization with removal of punctuation to obtain the set of words.

[4] The vocabulary was made up of Unified Medical Lexicon for French and a list of domain-specific terms extracted from a comprehensive French handbook of foetopathology (Bouvier et al., 2008)

I. EXAMEN MACROSCOPIQUE - f**æ**tus de sexe mascu**1**in - état frais - macération absente - poids 440 gr - **menurations** VT 2**/** cm VC 19 cm PC 19 cm Pied 3,5 - ces paramètres sont compatibles avec un âge gestationnel de **21'22S,A** ... La dissection des viscères met en évidence : - **hvpoplasie** du cœur gauche ... Les clichés ne montrent pas **danomalie** osseuse autre que **faOsence Oe** la 12ème paire de côtes.

Table 1: Feto-placental report sample with fake data and realistic digitization errors. Incorrectly digitized tokens are in bold.

a simple heuristic to determine the OCR quality of the document. Using this metric we divided the corpus into four categories, as shown in Table 2. The right column shows the cut-off rates that were used to distinguish between the different categories. The lower the document score, the fewer OOV words were found which indicates a good OCR quality. We would like to stress that although we use external resources to classify the training corpus into categories, this information is not used during the training of the neural networks.

OCR quality	# of documents	score cut-off
Excellent	1,088 (6.6%)	x <= 0.1
Good	7,694 (46.4%)	0.1 > x <= 0.25
Mediocre	3,595 (21.7%)	0.25 > x <= 0.50
Unusable	4,196 (25.3%)	x > 0.50

Table 2: Distribution of OCR quality categories in the training corpus

3.4 Evaluation set

All evaluations in this paper were carried out on a set of 53 files, randomly selected from the Excellent and Good quality subsets, which had been annotated manually by one annotator in two passes. These annotations were later verified by a second annotator.[5]

In total, the evaluation contains 473 errors. Table 3 shows the distribution of the four main types of OCR errors in the evaluation set. For each error the annotator provided a corrected string. Consequently, for each document in the evaluation set we had an original version with OCR errors, and a corrected version as the Gold Standard.

error type	#	OCR error ex.	Gold Standard
insertion	38	nuquer	nuque
deletion	69	maroscopique	macro-scopique
substitution	349	extrei/iities	extremities
other	17	e};,.ez ,J2	e};,.ez ,J2[7]

Table 3: Distribution of OCR error types in the evaluation set

4 Model

4.1 Character-based LSTM model

Our model consists of a many-to-many character sequence learning network using Long Short Term Memory nodes (LSTMs). The main idea is that the input sequence, in this case a string of characters, is mapped to a vector which is fed into a recurrent neural network (RNN) to generate the output sequence conditioned on the encoding vector. We use LSTMs[8] (Graves, 2013) as the basic RNN unit since this has shown improved performance on various NLP tasks such as text generation. In our model, we stack two LSTM layers on top of each other: the first level is an encoder that reads the source character sequence and the other is a decoder that functions as a language model and generates the output. We also added a drop-out layer since this has been shown to improve performance (Srivastava et al., 2014). The model was implemented in Keras (Chollet, 2015), a python library for deep learning. Figure 1 shows the network hierarchy.

[5]The role of the second annotator was to check that the existing annotations were correct and consistent. Ergo the annotations were not done independently.

[6]Since the annotators did not have access to the original PDF files to check the original text, it was not possible to generate corrected text for some badly corrupted strings.

[7]Since the annotators did not have access to the original PDF files to check the original text, it was not possible to generate corrected text for some badly corrupted strings.

[8]An excellent low-level introduction to RNNs and LSTMs can be found at `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`.

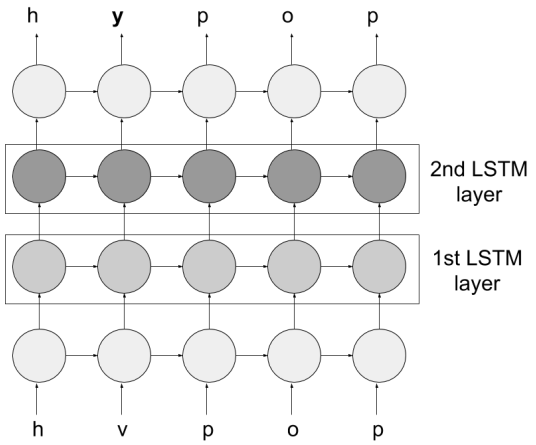

Figure 1: Hierarchy of 2-layer many-to-many sequence learning network; 'hvpop' taken as input, 'hypop' as expected output

In order to learn a robust language model, we fed the neural network with randomly corrupted input strings and provided the original (non-corrupted) strings as output labels. This way the NN learns both a character level language model that is domain-specific but it also learns to detect and eliminate random noise. We created corrupted strings by deleting, inserting and substituting one or two characters for a given string. Since a string could be submitted to multiple corrupting edits this resulted in both mono-error as well as multi-error words in the corrupted string. We heuristically determined the rate of noise so as to resemble the level of corruption, i.e., number of OCR errors of the actual test data. Table 4 shows an example of the generated training input with label output. We used windows of 20 characters from the initial text but since the length of the corrupted text strings varied due insertions and deletions, the network was fed (padded) sequences of 23 characters. The network was trained on data from the 'Excellent' OCR quality subset.

original text (reference)	'après l'expulsion de'
corrupted text (input)	'arpèS1'exVlsion e'
model output	'après 1'exulsion de'

Table 4: Example of input, output and reference in the training process

We experimented with two different string corruption settings:

1. *Random generation (randomNoise)* in which we used a random number generator to determine if and which edit options were selected. Character substitutions were performed at random with characters from the character set;

2. *Insertion of character confusions (confusionPair)*: In this setting we want to examine if injecting information on possible character confusions in the corpus, i.e. teaching the model that character x is likely to be replaced with character y, leads to a faster convergence of the trained models. While we do not have annotated training material to learn character confusions, we can exploit the natural redundancy in the corpus: Using a string alignment algorithm we identified near-duplicates in the subset of documents with 'Good' OCR quality. We then extracted confusion pairs, i.e. 1:1, 2:2, 1:2 and 2:1 character pairs that occurred in the same contexts, and had a relatively high frequency in the corpus. Table 5 shows the top 5 of the most frequent confusion pairs extracted from the corpus. This information was added to the randomization module so that instead of a substitution of a character by a random character, the only substitutions allowed were chosen from this list. We should stress that, since we do not use annotated training material, the extracted list might not be complete.

string to be replaced	replacement string	character pair type
l	I	1:1
I	l	1:1
!	l	1:1
W	VT	1:2
rr	m	2:1

Table 5: Most frequent character confusions from the subset of the corpus with 'Good' OCR quality

4.2 Baseline model

In order to evaluate the relative improvement of our character-based model, we also implemented a traditional word-based OCR error detection and correction system. Our implementation follows the basic structure of such systems which were presented in

section 1. The algorithm consists of the following steps:

1. Tokenization of the text into token sequences;
2. OCR error detection by vocabulary[9] look-up. We allowed up to a minimal edit distance of three[10] transformations of a given token, and the combination of the given token glued to the subsequent token in the token sequence[11] to find a suitable entry in the lexicon.
3. The candidates were then ranked and the highest-ranking candidate was used the replace the original token in the text. We experimented with different weighting schemes and finally opted for a ranking by the number of edits, in which substitution edits that used the confusion pairs (presented in section 4.1) had a lower weight than edits which were not significantly present in the training corpus.

5 Experiments

The character-based models were trained for 4 iterations with 20 epochs[12] per iteration. The randomNoise and confusionPair models achieved 73% and 71% accuracy respectively while the baseline model achieved an accuracy of 51%. Inspection of the intermediate scores shows that the randomNoise model achieves convergence fairly quickly, while the confusionPair model has a slower learning rate. This indicates that corrupting the strings in a more 'consistent' manner, i.e. using information on likely confusion pairs extracted from the corpus, leads to more erroneous assumptions during training. While the randomNoise model is trained to robustly deal with random noise, the confusionPair model's focus on a subset of the possible errors does not train the model well enough to detect other kinds of errors.

A close analysis of the corrections and errors of the randomNoise model on the test set shows that the model is good at detecting 'close' substitutions of characters when they appear in a relatively clean environment, e.g. a substitution between 'e' and 'é' in the string 'theorique', or a switch between lowercase and uppercase, such as in 'develoPpement' . We find that when the original input string contains multiple OCR errors close together (and as such is no longer a 'clean' environment for a character substitution), the model cannot adequately decide which characters to replace. This suggests that either gradually increasing the ratio of noise or slowly extending the context window during training might have a positive impact on performance accuracy. Table 6 shows the proportions of OCR errors in the manually annotated evaluation set that were corrected by the two character-based approaches and the word-based baseline model.

OCR error type	randomNoise	confusionPair
insertion	0.0	0.1
deletion	24.5	23.6
substitution	75.5	76.3

Table 6: Proportions of corrections for different OCR error types in the evaluation set

We see that most substitution errors are most easily spotted by the models but that the detection of insertion errors proves very difficult. This is because most of the insertion errors are random insertions of whitespace in words. Since whitespace is used abundantly in structuring the documents, the model generally predicts this character with a high probability, and thus fails to detect it as an error. The addition of character confusion information in the creation of corrupted input data (column 2 in Table 6) has a slight positive impact on substitution errors but not as much as was expected.

When examining the cases in which the model failed to spot an error or generated corrections where none were needed, we find that text written in uppercase presents a great difficulty for the models. Only a small part of the documents are written in uppercase, i.e. the headers with de-identified personal information and the titles of the individual sections. The models clearly do not have enough training data to learn an adequate language model. In a follow-up study, we should either provide the model with more data, or add a lowercasing step to the preprocessing pipeline. Another interesting but infrequent

[9] Checks were performed using the same lexicon as for the calculation of the proportion of OOV words in section 3.3. We extended the lexicon by creating new entries which consisted of two original words of the lexicon glued together, in order to catch whitespace deletion errors.

[10] In our implementation insertion, deletion and substitution steps all had the same cost, i.e. 1.

[11] In order to find whitespace insertion errors

[12] The number of epochs was empirically determined.

error are the cases where the language model has clearly learned the character-based language models but uses it incorrectly given the wider context, for example, by changing 'facile' *(easy)* into 'faciale' *(facial)* in 'Ponction de trophoblaste facile' *(easy puncture of the trophoblasts)*. These types of error could be avoided by fitting a larger language model on top of the character-based LSTM model.

6 Conclusion

In this paper we presented a method for the detection and correction of OCR errors in French patient files. Our method consists of a many-to-many sequence learner using LSTMs which is robustly trained on artificially corrupted good-quality training data in order to learn both the underlying character level language model, as well as to detect and eliminate noise in the input string. The relatively fast convergence of the models is likely due to the natural redundancy in the medical corpus. We experimented with two different methods of adding noise to the input and found that injecting information on likely character confusion pairs extracted from the training corpus had no positive impact on accuracy. Interestingly, the models are not good at detecting insertion errors, i.e. the detection of word boundaries. In future work, we would like to extend the model by combining the output of the character level with information on word level through an embedding layer in order to improve the overall accuracy.

Acknowledgments

This work was supported by the French National Agency for Research under the grant Accordys[13] ANR-12-CORD-0007.

References

Youssef Bassil and Mohammad Alwani. 2012. OCR context-sensitive error correction based on google web 1t 5-gram data set. *American Journal of Scientific Research*.

Raymonde Bouvier, Dominique Carles, Marie-Christine Dauge, Pierre Déchelotte, Anne-Lise Delézoide, Bernard Foliguet, Dominique Gaillard, Bernard Gasser, Marie Gonzalès, and Férechté Encha-Razavi. 2008. *Pathologie fœtale et placentaire pratique*. Sauramps Médical, Montpellier.

François Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

John Evershed and Kent Fitch. 2014. Correcting noisy OCR: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 45–51. ACM.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Cyril Grouin and Pierre Zweigenbaum. 2013. Automatic de-identification of French clinical records: Comparison of rule-based and machine-learning approches. In *Stud Health Technol Inform*, volume 192, pages 476–80, Copenhagen, Denmark. MEDINFO.

Ido Kissos and Nachum Dershowitz. 2016. OCR error correction using character correction and feature-based word classification. In *Proceedings of the 12th IAPR International Workshop on Document Analysis Systems (DAS2016)*, Santorini, Greece.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4):377–439.

Atul Kumar. 2016. A survey on various OCR errors. *International Journal of Computer Applications*, 143(4):8–10.

Kenneth H Lai, Maxim Topaz, Foster R Goss, and Li Zhou. 2015. Automated misspelling detection and correction in clinical free-text records. *Journal of biomedical informatics*, 55:188–195.

Walid Magdy and Kareem Darwish. 2010. Omni font OCR error correction with effect on retrieval. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications*, pages 415–420. IEEE.

Agnieszka Mykowiecka and Małgorzata Marciniak. 2006. Domain–driven automatic spelling correction for mammography reports. In *Proceedings of the Intelligent Information Processing and Web Mining*, pages 521–530, Ustrón, Poland.

Maciej Piasecki and Grzegorz Godlewski. 2006. Language modelling for the needs of OCR of medical texts. In *Proceedings of International Symposium on Biological and Medical Data Analysis*, pages 273–284, Thessaloniki, Greece.

Margaret M Richards. 2009. Electronic medical records: Confidentiality issues in the time of HIPAA. *Professional Psychology: Research and Practice*, 40(6):550.

[13] Agrégation de Contenus et de COnnaissances pour Raisonner à partir de cas dans la DYSmorphologie foetale

Verónica Romero, Nicolás Serrano, Alejandro H. Toselli, Joan Andreu Sánchez, and Enrique Vidal. 2011. Handwritten text recognition for historical documents. In *Proceedings of the Language Technologies for Digital Humanities and Cultural Heritage Workshop*, pages 90–96, Hissar, Bulgaria.

Patrick Ruch, Robert Baud, and Antoine Geissbhler. 2002. Evaluating and reducing the effect of data corruption when applying bag of words approaches to medical records. *International Journal of Medical Informatics*, 67(13):75 – 83.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Paul Thompson, John McNaught, and Sophia Ananiadou. 2015. Customised ocr correction for historical medical text. In *2015 Digital Heritage*, volume 1, pages 35–42. IEEE.

Pierre Zweigenbaum, Robert Baud, Anita Burgun, Fiammetta Namer, Éric Jarrousse, Natalia Grabar, Patrick Ruch, Franck Le Duff, Jean-François Forget, Magaly Douyere, et al. 2005. UMLF: a unified medical lexicon for French. *International Journal of Medical Informatics*, 74(2):119–124.