Journal of Data Mining and Digital Humanities

Evaluating Deep Learning Methods for Word Segmentation of Scripta Continua Texts in Old French and Latin

Thibault Clérice[1]

[1]Centre Jean-Mabillon, École nationale des Chartes, PSL Research University, France
[2]Hisoma (UMR 5189), Université Lyon 3, Université de Lyon (UDL), France

Corresponding author: Thibault Clérice , thibault.clerice@chartes.psl.eu

Abstract

Tokenization of modern and old Western European languages seems to be fairly simple, as it relies on the presence of markers such as spaces and punctuation. However, when dealing with old sources like manuscripts written in *scripta continua*, ancient epigraphy or Middle Age manuscripts, (1) such markers are mostly absent, (2) spelling variation and rich morphology make dictionary based approaches difficult. Applying convolutional encoding to characters followed by linear categorization to word-boundary or in-word-sequence is shown to be effective at tokenizing such inputs. Additionally, the software created for this article (Boudams) is released with a simple interface for tokenizing a corpus or generating a training set [1].

Keywords

convolutional network; scripta continua; tokenization; Old French; word segmentation

I INTRODUCTION

Tokenization of spaceless strings is a task that is specifically difficult for computers as compared to "whathumanscando". *Scripta continua* is a western Latin writing phenomenon in which words are not separated by spaces. It disappeared around the 8th century (see Zanna [1998]), but, nevertheless, spacing remained erratic [2] in later centuries writing as Stutzmann [2016] explains (*cf.* Figure 1). The fluctuation of space width, or simply their presence becomes an issue for OCR. Indeed, in the context of text mining of HTR or OCR output, lemmatization and tokenization of medieval western languages is quite often a pre-processing step for further research to sustain analyses such as authorship attribution, corpus linguistics or simply to allow full-text search [3].

It must be stressed in this study that the difficulty inherent to segmentation is different for *scripta continua* than the one for languages such as Chinese, for which an already impressive amount of work has been done. Indeed, the dimensionality alone of the Chinese character set is

[1]The software has ongoing development on https://github.com/ponteineptique/boudams . The software at the time of the article is also available (Clérice [2019a])

[2]From a modern point of view. What we call here *scripta continua* ranges from traditional *scripta continua* to variable spacing writing.

[3]We have found no previous study of *scripta continua* and the likes as a natural language processing issue, only as an HTR/OCR one, such as Wahlberg et al. [2014] and Bluche et al. [2017]

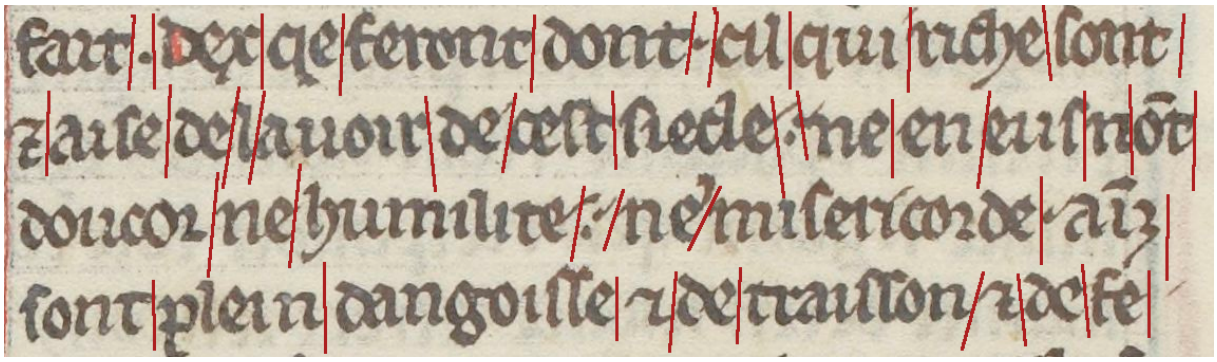

Figure 1: 4 lines from fol.103rb Manuscript fr. 412, Bibliothèque nationale de France. Red lines indicate word boundaries

different from Latin alphabets [4], and the important presence of compound words is definitely an issue for segmentation [5]. Chinese word segmentation has lately been driven by deep learning methods: Chen et al. [2015] defines a process based on LSTM model, while Yu et al. [2019] uses bi-directional GRU and CRF. [6]

Indeed, while the issue with Chinese seems to lie in the decomposition of relatively fixed characters, Old French or Medieval Latin present heavy variation of spelling. In Camps et al. [2017], Camps notes, in the same corpus, the existence of not less than 29 spellings of the word "*cheval*" (horse in Old and Modern French) whose occurrence counts range from 3907 to 1[7]. This makes a dictionary-based approach rather difficult as it would rely on a high number of different spellings, making the computation highly complex.

II DESCRIPTION AND EVALUATION

2.1 Architecture

2.1.1 Encoding of input and decoding

The model used in this study is based on traditional text input encoding where each character is represented as an index. Output of the model is a mask that needs to be applied to the input: in the mask, characters are classified either as word boundary or word content (*cf.* Table 1.

	Sample
Input String	`Ladamehaitees'enparti`
Mask String	`xSxxxSxxxxxSxxxSxxxxS`
Output String	`La dame haitee s'en parti`

Table 1: Input, mask and human-readable output generated by the model. x are WC and S are WB

For evaluation purposes, and to reduce the number of input classes, two options for data transcoding were used: a lower-case normalization and a "reduction to the ASCII character set" feature

[4] a range of twenty to a few hundred characters if we take into account all diacritic combination in Latin scripts, at least few thousands in Chinese, depending on the period Campbell and Moseley [2012]

[5] According to Tse et al. [2017], *"about 73.6 % of modern Chinese words are two-character compound words"*.

[6] Huang et al. [2008] actually gave us the denomination used here: word boundary (WB) and word content (WC).

[7] These are *cheval*, *chevaus*, *cheual*, *ceval*, *chevals*, *cevaus*, *chival*, *ceual*, *cheuaus*, *cevals*, *chaval*, *chivaus*, *chiual*, *chevas*, *cheuals*, *chiuaus*, *ceuaus*, *chevaul*, *chiuau*, *chivals*, *chevau*, *kevaus*, *chavaus*, *cheuas*, *keval*, *cheua*, *cheuau*, *cheva*, *chiuals*

(fr. 2). On this latter point, several issues were encountered with the transliteration of medieval paelographic characters that were part of the original datasets, as they are poorly interpreted by the `unidecode` python package. Indeed, `unidecode` simply removed characters it did not understand. A derivative package named `mufidecode` was built for this reason(Clérice [2019b]): it takes precedent over `unidecode` equivalency tables when the character is a known entity in the Medieval Unicode Font Initiative (MUFI, Initiative [2015]).

```
import mufidecode
import unidecode
"sot la gn̓t abstinence dess eintes uirges ele  p̛la"
mufidecode.mufidecode(" sot la gn̓t abstinence dess eintes uirges ele  p̛la")
# ' sot la gnat abstinence dess eintes uirges ele  pla'
mufidecode.mufidecode(" sot la gn̓t abstinence dess eintes uirges ele  p̛la", join=False
# (' ',  's',  'o',  't',  ' ',  'l',  'a',  ' ',  'g',  'n',  'a',  't',  ' ',  'a',  'b',  's',  't',  'i',
'n',  'e',  'n',  'c',  'e',  ' ',  'd',  'e',  's',  's',  ' ',  'e',  'i',  'n',  't',  'e',  's',  ' ',
'u',  'i',  'r',  'g',  'e',  's',  ' ',  'e',  'l',  'e',  ' ',  ' ',  'p',  'l',  'a')
unidecode.unidecode(" sot la gn̓t abstinence dess eintes uirges ele  p̛la")
# ' sot la gnat abstinence dess eintes uirges ele  la'
```

Figure 2: Different possibilities of pre-processing. The option with join=False was kept, as it keeps abbreviation marked as single characters. Note how `unidecode` loses the P WITH BAR

2.1.2 Model

Aside from normalization of the input and output, three different model structures were tested. Every model was composed of one encoder, as described below, and one Linear Classifier which classified characters into 5 classes : Start of Sentence (= SOS), End of Sentence (= EOS), Padding (= PAD), Masked Token (= Word Content), Space (= Word Boundary) [8].

The following encoders were used (configurations in parentheses):

- LSTM encoder with hidden cell (Embedding (512), Dropout(0.5), Hidden Dimension (512), Layers(10))
- Convolutional (CNN) encoder with position embeddings (Embedding (256), Embedding(Maximum Sentence Size=150), Kernel Size (5), Dropout(0.25), Layers (10))
- Convolutional (CNN) encoder without position embeddings (Embedding (256), Kernel Size (5), Dropout(0.25), Layers (10))

2.2 Evaluation

2.2.1 Evaluation on Old French Data

2.2.1.1 Main Dataset

The dataset is composed of transcriptions (from different projects) of manuscripts with unresolved abbreviations. The **Old French** is based on Bluche et al. [2017], Pinche [2017], Camps et al. [2019b], Lavrentiev [2019], and Pinche et al. [2019]. It contains

- 193,734 training examples (group of words);
- 23,581 validation examples;
- 25,512 test examples
- Number of classes in testing examples: 482,776 WC; 169,094 WB

[8] For final scores, SOS, EOS and PAD were ignored.

Examples were generated automatically. They are between 2 and 8 words in length. In order to recreate the condition of OCR noise, full stop characters were added randomly (20% chance) between words. In order to augment the dataset, words were randomly (10% chance) copied from one sample to the next [9]. If a minimum size of 7 characters was not met in the input sample, another word would be added to the chain, independently of the maximum number of words. The examples, however, could not contain more than 100 characters. The word lengths in the results corpora were expected to vary as shown by Figure 3. The corpora contained 193 different characters when not normalized, in which certain MUFI characters appeared a few hundred times (*cf.* Table 2).

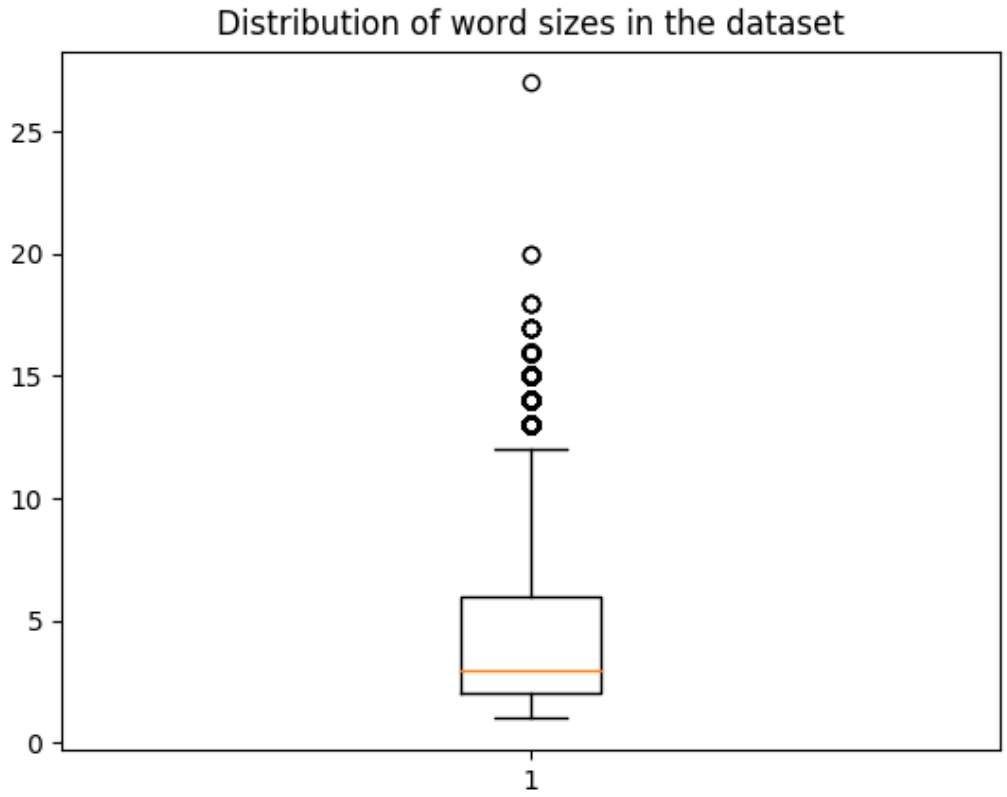

Figure 3: Distribution of word size over the train, dev and test corpora

	Train dataset	Dev dataset	Test dataset
TIRONIAN SIGN ET	4367	541	539
CON	508	70	76
P WITH STROKE THROUGH DESCENDER	580	69	84

Table 2: Examples of some MUFI characters distributions

[9]This data augmentation was limited to one word per sample. *e.g.* If the phrase "I have lived here for a long time" were broken into "I have lived here" and "for a long time", the word "here" might be copied to the second sample, thus producing "I have lived here" and "here for a long time."

2.2.1.2 Results

The training parameters were 0.00005 in learning rate for each CNN model (LeCun et al. [1998]), 0.001 for the LSTM model (Hochreiter and Schmidhuber [1997]), and batch sizes of 64. Training reached a plateau fairly quickly for each model (*cf.* 4). Each model except LSTM achieved a relatively low loss and a high accuracy on the test set (*cf.* 3). To compare the results, the `wordsegment` package Jenks [2018] was used as a baseline. For this purpose, UDPipe (Straka and Straková [2017]) was evaluated but scores were lower than this baseline: our LSTM and GRU implementations show however the same difficulties while sharing the same apparent architecture ([10].

Model	Accuracy	Precision	Recall	FScore	WB FN	WB FP
Baseline	0.989	0.986	0.984	0.985	4031	3229
CNN	0.991	0.985	0.990	0.987	2137	3860
CNN L	0.991	0.979	0.990	0.985	**2117**	3750
CNN P	**0.993**	**0.990**	**0.991**	**0.990**	2432	**2114**
CNN N	0.991	0.987	0.988	0.988	2756	3312
CNN L N	0.992	0.988	0.989	0.988	2500	3567
LSTM	0.939	0.637	0.918	0.720	21174	18662
GRU	0.933	0.645	0.645	0.910	23706	19427

Table 3: Scores over the test dataset.
For models: N = normalized, L = Lower, P = no position embedding.
In headers, FN = False Negative, FP = False Positive

2.2.1.3 Out-of-domain (OUD) texts

While all models using CNN showed improvement over the baseline, none of them significantly outperformed it, with a maximum improvement of 0.005 FScore. There is a reason for this: the baseline already performs nearly perfectly on the test corpus. Therefore, an additional evaluation method was constructed. The baseline (which is dictionary based) and the best achieving deep-learning model (CNN P) were evaluated on a secondary test corpus composed of texts of a different domain. This new corpus is composed of 4 texts and counts 742 examples : a diplomatic edition of the *Graal* (Marchello-Nizia et al. [2019]), a *Passion* and a *Vie de Saint Leger* (Sneddon [2019]), and a *Vie de Saint Thibaut* (Grossel [2019]). Neither noise characters nor random keeping of words were applied. The resulting corpus contains 26,393 WC and 10,193 WB.

The results here were significantly different (*cf.* Table 4): while the CNN was able to expand its "comprehension" of the language to newer texts, the baseline `wordsegment` n-gram approach had difficulty dealing with the new vocabulary. This resulted in a drop in the FScore to 0.945 for CNN and 0.838 for the baseline. WordSegment performed particularly poorly with WB false negatives : it had 3658 over a corpus containing 10,193 WB token (to put it simply, around 35 % of the spaces were not identified).

[10] An issue regarding parameters or implementations is not to be excluded.

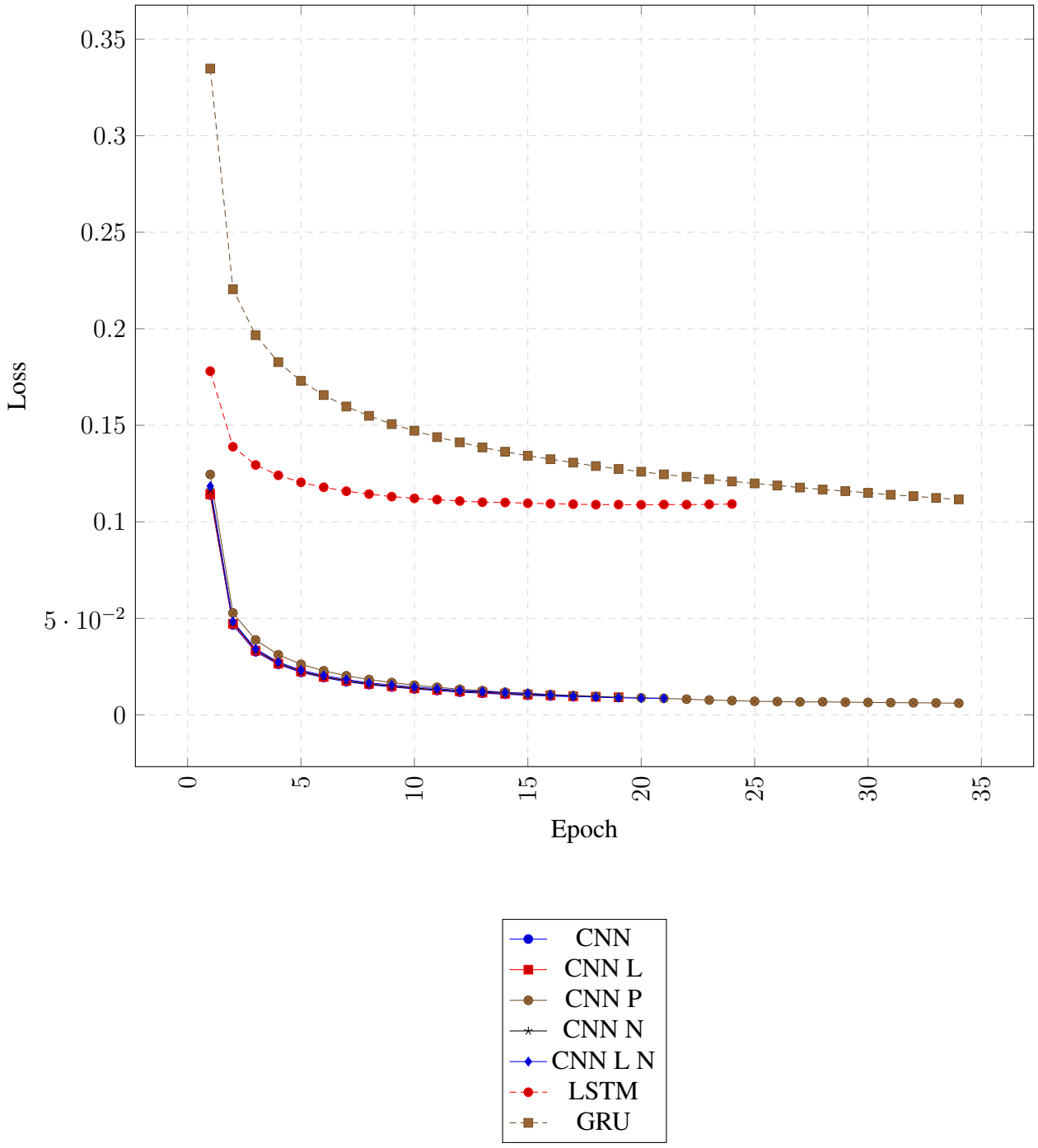

Figure 4: Training Loss (Cross-entropy) until plateau was reached. N = normalized, L = Lower, P = no position embedding.

	Accuracy	Precision	Recall	FScore	WB FN	WB FP
Baseline	0.882	0.893	0.808	0.838	3658	**644**
CNN P	**0.957**	**0.948**	**0.944**	**0.945**	**854**	723

Table 4: Scores (Macro-average) over the out-of-domain dataset. FN = False Negative, FP = False Positive.

2.2.1.4 Example of Outputs

The following inputs have been tagged with the CNN P model. Batches are constructed around the regular expression `\W` with package `regex`. This explains why inputs such as `".i."` are

automatically tagged as " . i . " by the tool. The input was stripped of its spaces before tagging, only the ground truth is shown for readability.

Ground truth	**Tokenized output (CNN P)**
Aies joie et leesce en ton cuer car tu auras une fille qui aura .i. fil qui sera de molt grant merite devant Dieu et de grant los entre les homes.Conforte toi et soies liee car tu portes en ton ventre .i. fil qui son lieu aura devant Dieu et qui grant honnor fera a toz ses parenz.	Aies joie et leesce en ton cuer car tu auras une fille qui aura . i . fil qui sera de molt grant merite devant Dieu et de grant los entre les homes . Confort e toi et soies liee car tu portes en ton ventre . i . fil qui son lieu aura devant Dieu et qui grant honnor fera a toz ses parenz .

Table 5: Output examples on a text from outside the dataset

2.2.2 Evaluation on Latin data

For the following evaluations, the same process was deployed: CNN without Position was evaluated against the baseline on both a test set composed of excerpts from the texts of the training set, and an out-of-domain corpus composed of unseen texts. Evaluation has been done on three different categories of Latin texts (edited, classical Latin (1); medieval Latin of charters (2); epigraphic Latin (3)) as they show different levels of difficulty: they always present rich morphology, but medieval Latin displays spelling variations while epigraphic Latin displays both spelling variation and a high number of abbreviations.

2.2.2.1 Latin Prose and Poetic Corpora

	Corpus	Accuracy	Precision	Recall	FScore	WB FN	WB FP
Baseline	Test	0.978	0.961	0.974	0.968	886	1893
CNN P	Test	**0.992**	**0.987**	**0.989**	**0.988**	**439**	**584**
Baseline	OUD	0.933	0.897	0.890	0.893	1587	1409
CNN P	OUD	0.970	0.952	0.956	0.954	600	709

Table 6: Scores over the Latin classical datasets. FN = False Negative, FP = False Positive

The Latin data is much noisier than the Old French, as it was less curated than the digital editions provided for Old French. They are part of the Perseus corpus Crane et al. [2019]. The training, evaluation and test corpora contain prose works from Cicero and Suetonius. The out-of-domain corpus comes from the *Epigrammata* from Martial, from book 1 to book 2, which should be fairly different from the test corpus in word order, vocabulary, etc. Both corpora were generated without noise and word keeping, with a maximum sample size of 150 characters.

Statistics:

- Number of training examples: 30725
- Number of evaluation examples: 3558
- Number of testing examples: 4406
- Number of classes in testing examples: 105,915 WC; 26,404 WB
- Number of classes in OUD examples: 35,910 WC; 8,828 WB

Example:

- Input : operecuperemdeberemqueprofecto
- Output : opere cuperem deberemque profecto

2.2.2.2 Medieval Latin corpora

	Corpus	Accuracy	Precision	Recall	FScore	WB FN	WB FP
Baseline	Test	0.989	0.981	0.986	0.982	1036	933
CNN P	Test	**0.997**	**0.995**	**0.995**	**0.995**	**251**	**298**
Baseline	OUD	0.929	0.900	0.865	0.881	14,382	27,019
CNN P	OUD	0.976	0.960	0.963	0.962	6509	7444

Table 7: Scores over the Latin medieval datasets. FN = False Negative, FP = False Positive

The medieval Latin corpus is based on the project Formulae - Litterae - Chartae's open data (Depreux et al. [2019]) for its training, evaluation and test sets; the out-of-domain corpora are based on three texts from the Monumenta Germanica (Ceynowa [2019]) that are from early to late medieval period (Andreas Agnellus, Manegaldus, Theodoricus de Niem) and are drawn from the Corpus Corporum Project. Both corpora were generated without noise and word keeping, with a maximum sample size of 150 characters. The data presents some MUFI characters but otherwise mostly resembles normalized editions, unlike the Old French data.

Statistics:

- Number of training examples: 36814
- Number of evaluation examples: 4098
- Number of testing examples: 5612
- Number of classes in testing examples: 137,465 WC; 34,053 WB
- Number of classes in OUD examples: 472,655 WC; 113,004 WB

Example:

- Input : nonparvamremtibi
- Output : non parvam rem tibi

2.2.2.3 Latin epigraphic corpora

	Corpus	Accuracy	Precision	Recall	FScore	WB FN	WB FP
Baseline	Test	0.956	0.935	0.943	0.939	2646	3547
CNN P	Test	**0.987**	**0.983**	**0.979**	**0.981**	**1149**	**722**
Baseline	Test Uppercase	0.956	0.935	0.942	0.938	2664	3457
CNN P	Test Uppercase	**0.979**	**0.972**	**0.967**	**0.969**	**1715**	**1275**
Baseline	OUD	0.879	0.834	0.817	0.825	8693	11332
CNN P	OUD	**0.953**	**0.939**	**0.926**	**0.932**	**4689**	**3112**
Baseline	OUD Uppercase	0.879	0.834	0.817	0.825	8693	11332
CNN P	OUD Uppercase	**0.936**	**0.914**	**0.902**	**0.908**	**6152**	**4464**

Table 8: Scores over the Latin epigraphic datasets. FN = False Negative, FP = False Positive

The Latin epigraphic corpus is based on the Epigraphic Database Heidelberg open data for its training, evaluation and test sets (HD000001-HD010000 and HD010001-HD020000 from Witschel et al. [2019]) while the out-of-domain corpus is drawn from an automatic conversion of the Pompei Inscriptions (Clérice [2017]). Both the baseline and the model were evaluated on uppercase data, as the texts would normally be found in this state. Each of the corpora presents

a high number of unresolved abbreviations (*ie.* one letter words). Both corpora were generated without noise and word keeping, with a maximum sample size of 150 characters. The data present some polytonic Greek characters, since some sample were only in Greek.

Statistics:

- Number of training examples: 46,423
- Number of evaluation examples: 5,802
- Number of testing examples: 5,804
- Number of classes in testing examples: 107,963 WC; 31,900 WB
- Number of classes in OUD examples: 127,268 WC; 38,055 WB

Example:

- Input : DnFlClIuliani
- Output : D n Fl Cl Iuliani

2.3 Discussion

As opposed to being a graphical challenge, word segmentation in OCR from manuscripts can actually be treated as an NLP task. Word segmentation for some texts can even be difficult for humanists, as shown by the manuscript sample, and as such, it seems that the post-processing of OCR or HTR through tools like this one can enhance data-mining of raw datasets.

The negligible effects of the different normalization methods (lower-casing; ASCII reduction; both) were surprising. The presence of certain MUFI characters might provide enough information about segmentation and be of sufficient quantity for them not to impact the network weights.

While the baseline performed unexpectedly well on the test corpora, the CNN model definitely performed better on the out-of-domain corpora. In this context, the proposed model deals better with unknown corpora classical n-gram approaches. In light of the high accuracy of the CNN model on the different corpora, the model should perform equally well no matter to which Medieval Western European language it is applied.

2.4 Conclusion

Achieving 0.99 accuracy on word segmentation with a corpus as large as 25,000 test samples seems to be the first step for a more thorough data mining of OCRed manuscripts. Given the results, studying the importance of normalization and lowering should be the next step, as it will probably show greater influence in smaller corpora.

2.5 Acknowledgements

Boudams has been made possible by two open-source repositories from which I learned and copied bits of the implementation of certain modules and without which none of this paper would have been possible: Manjavacas et al. [2019] and Trevett [2019]. This tool was originally intended for post-processing OCR for the presentation Camps et al. [2019a] at DH2019 in Utrecht. The software was developed using PyTorch (Paszke et al. [2019]), Numpy (Oliphant [2006–]) and SK-Learn (Pedregosa et al. [2011]).

References

Théodore Bluche, Sebastien Hamel, Christopher Kermorvant, Joan Puigcerver, Dominique Stutzmann, Alejandro H. Toselli, and Enrique Vidal. Preparatory kws experiments for large-scale indexing of a vast medieval manuscript collection in the himanis project. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 311–316, Nov 2017. doi: 10.1109/ICDAR.2017.59.

George L Campbell and Christopher Moseley. *The Routledge handbook of scripts and alphabets*. Routledge, 2012.

Jean-Baptiste Camps, Thibault Clérice, Mike Kestemont, and Enrique Manjavacas. Pandora, a (language independent) tagger lemmatizer for latin and the vernacular. Atelier COSME, November 2017. URL https://www.academia.edu/35076560/Pandora_A_language_independent_Tagger_Lemmatizer_for_Latin_and_the_Vernacular.

Jean-Baptiste Camps, Thibault Clérice, and Ariane Pinche. Stylometry for noisy medieval data: Evaluating paul meyer's hagiographic hypothesis. In *DH2019*, July 2019a.

Jean-Baptiste Camps, Alice Cochet, Elena Albarran, Lucence Ing, and Pauline Levêque. Jean-Baptiste-Camps/Geste: Geste: un corpus de chansons de geste, 2016-. . . , April 2019b. URL https://doi.org/10.5281/zenodo.2630574.

Klaus Ceynowa. Monumenta Germanica Historica, 2019. URL http://www.mgh.de/.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, 2015.

Thibault Clérice. Pompei inscriptions, 2017. URL https://github.com/lascivaroma/pompei-inscriptions.

Thibault Clérice. Ponteineptique/boudams: Article release, June 2019a. URL https://doi.org/10.5281/zenodo.3241754.

Thibault Clérice. Ponteineptique/mufidecode: v0.1.0, June 2019b. URL https://doi.org/10.5281/zenodo.3237731.

Gregory R. Crane, Thibault Clérice, Lisa Cerrato, Briget Almas, Neven Jovanović, Annette Gessner, Patrick J. Burns, Stella R. Dee, Matt Munson, Maryam Foradi, Masoumeh Seydi, and Tim Buckhingam. Perseusdl/canonical-latinlit 0.0.421, May 2019. URL https://doi.org/10.5281/zenodo.3236496.

Philippe Depreux, Matt Munson, Morgane Pica, and Corentin Faye. Formulae - litterae - chartae, June 2019. URL https://github.com/Formulae-Litterae-Chartae/formulae-open.

Marie-Geneviève Grossel. Vie en prose romane de saint thibaut, d'après le manuscrit fr. 23686 de la bibliothèque nationale de france, 2019. URL http://www.theobaldus.org/histoire-spiritualite-8/vies-romanes/16-vie-en-prose-romane-de-saint-thibaut.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Chu-Ren Huang, Ting-Shuo Yo, Petr Šimon, and Shu-Kai Hsieh. A realistic and robust model for chinese word segmentation, 2008.

Medieval Unicode Font Initiative. Medieval Unicode Font Initiative V4.0, dec 2015.

Grant Jenks. Wordsegment (1.3.1), July 2018. URL https://github.com/grantjenks/python-wordsegment.

Alexei Lavrentiev. Corpus BFMMSS, 2019. URL http://txm.bfm-corpus.org/.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Enrique Manjavacas, Thibault Clérice, and Mike Kestemont. emanjavacas/pie v0.2.3, April 2019. URL https://doi.org/10.5281/zenodo.2654987.

Christiane Marchello-Nizia, Alexei Lavrentiev, Isabelle Vedrenne-Fajolles, and Serge Heiden. Queste du graal d'après bibliothèque municipale de lyon, ms. arts 77, June 2019. URL http://bfm.ens-lyon.fr/IMG/html/qgraal77_dipl.html.

Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–. URL http://www.numpy.org/. [Online; accessed ¡today¿].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Ariane Pinche. Édition nativement numérique des oeuvres hagiographiques 'Li Seint Confessor' de Wauchier de Denain, d'après le manuscrit 412 de la bibliothèque nationale de France. 40 ans du laboratoire du CIHAM et de la création du pôle de Lyon de l'EHESS, October 2017. URL https://hal.archives-ouvertes.fr/hal-01628533. Poster.

Ariane Pinche, Clément Andrieux, Lucie Vieillon, Marie Morillon, Marie-Caroline Schmied, Olivier Jacquot, and Ségolène Albouy. Exercices TEI du master Technologies Numériques Appliquées à l'Histoire, 2019. URL https://github.com/Chartes-TNAH/digital-edition.

Clive R. Sneddon. Old french corpus, 2019. URL http://purl.ox.ac.uk/ota/0176.

Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/K/K17/K17-3009.pdf.

Dominique Stutzmann. Words as graphic and linguistic structures: word spacing in psalm 101 domine exaudi orationem meam (11th-15th c.). In *13e symposium annuel de la Société Internationale des Médiévistes*, June 2016.

Ben Trevett. Pytorch seq2seq, April 2019. URL https://github.com/bentrevett/pytorch-seq2seq.

Chi-Shing Tse, Melvin J Yap, Yuen-Lai Chan, Wei Ping Sze, Cyrus Shaoul, and Dan Lin. The chinese lexicon project: A megastudy of lexical decision performance for 25,000+ traditional chinese two-character compound words. *Behavior Research Methods*, 49(4):1503–1519, 2017.

Fredrik Wahlberg, Mats Dahllöf, Lasse Mårtensson, and Anders Brun. Spotting words in medieval manuscripts. *Studia Neophilologica*, 86(sup1):171–186, 2014.

Christian Witschel, Géza Alföldy, James M.S. Cowey, Francisca Feraudi-Gruénais, Brigitte Gräf, Frank Grieshaber, Regine Klar, and Jonas Osnabrügge. Epigraphic Database Heidelberg, 2019. URL https://edh-www.adw.uni-heidelberg.de/.

Chenghai Yu, Shupei Wang, and Jiajun Guo. Learning chinese word segmentation based on bidirectional gru-crf and cnn network model. *International Journal of Technology and Human Interaction (IJTHI)*, 15(3):47–62, 2019.

Paolo Zanna. Lecture, écriture et morphologie latines en irlande aux viiè et viiiè siècles. *Archivum Latinitatis Medii Aevi-Bulletin du Cange (ALMA)*, 1998.

A ANNEX 1 : CONFUSION OF CNN WITHOUT POSITION EMBEDDINGS

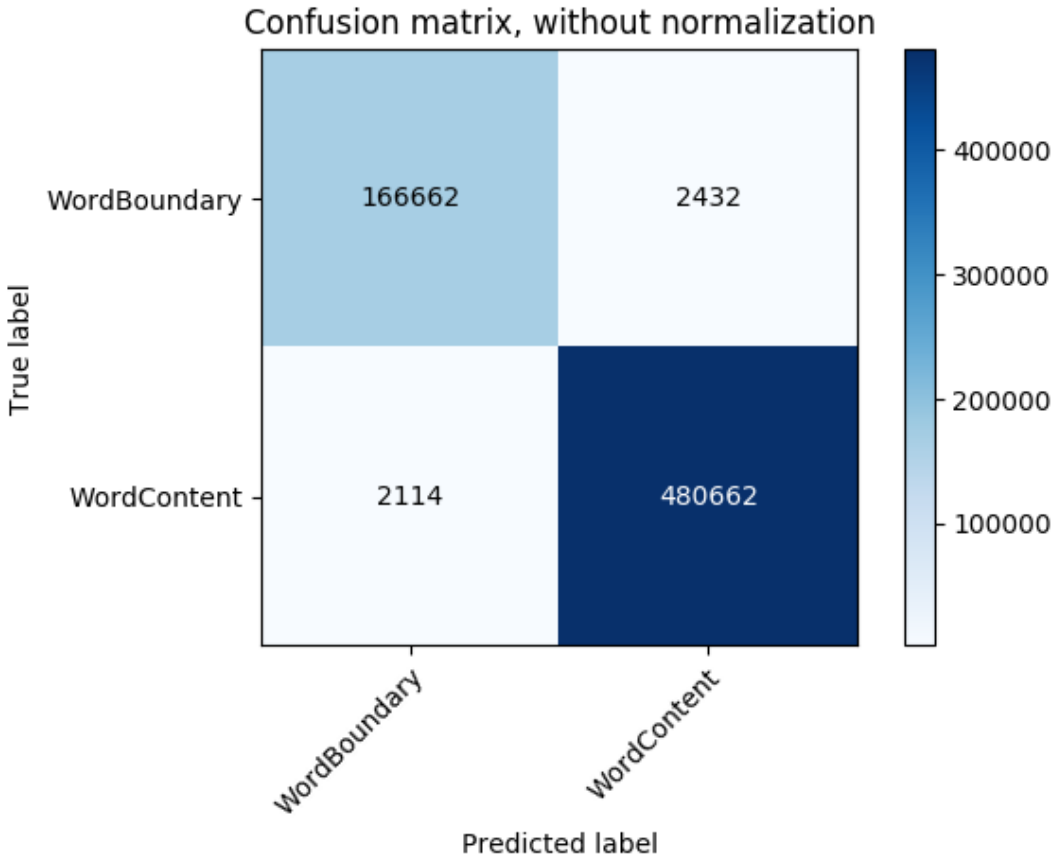

Figure 5: Confusion matrix of the CNN model without position embedding