Evaluating Deep Learning Methods for Tokenization of Space-less texts in Old French and Latin

Thibault Clérice[1]

[1]École nationale des Chartes, France
[2]Université Lyon 3, 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 stands on the presence mostly of markers such as spaces and punctuation. However, when dealing with old sources like manuscripts written in *scripta continua*, antiquity 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 is released with a simple interface for tokenizing a corpus or generating a training set.

Keywords

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

I INTRODUCTION

Tokenization of space-less strings is a task that is specifically difficult for computers as compared to ”whathumanscando”. *Scripta continua* is a writing phenomenon in which words are separated by spaces that disappeared around the 8th century (see Zanna [1998]). Nevertheless, spacing can be somewhat erratic in later centuries writings as Stutzmann [2016] explains (*cf.* Figure 1) and it becomes an issue for OCR, as continuous bag of word is only interesting when those are not glued together. In the context of text mining of HTR or OCR output, lemmatization and tokenization of medieval western languages can be a pre-processing step for further research to sustain analyses such as authorship attribution or simply allow full-text search.

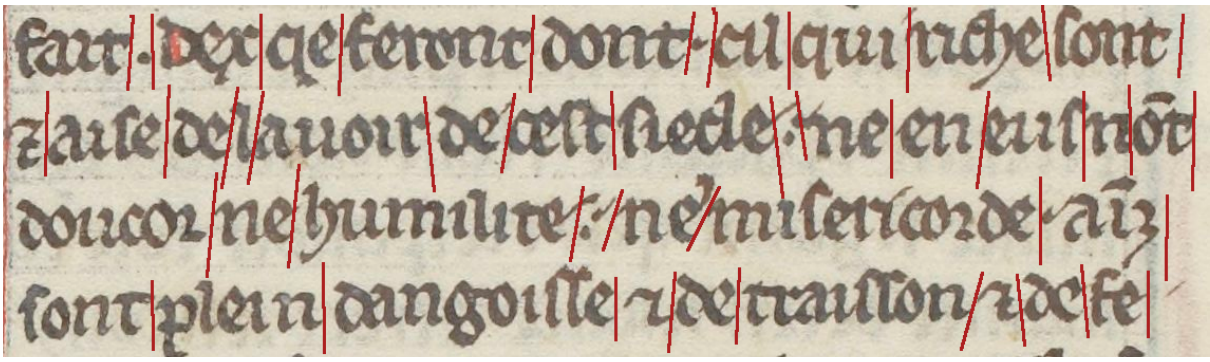

Figure 1: 4 lines from fol.103rb Manuscript fr. 412, Bibliothèque nationale de France. Red lines indicate word boundaries

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. Chinese word segmentation has lately been driven by deep learning methods, specifically ones based on *sequence to sequence translations*: Chen et al. [2015] defines a process based on LSTM model, while Yu et al. [2019] uses BiDirectional GRU and CRF. [1]

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 apparition counts span from 3907 to 1[2]. 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 is based on traditional text input encoding where each character is transcoded to 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 (fr. 2). On this point, a lot of issues were encountered with transliteration of medieval paelographic characters that were part of the original datasets, as they are poorly interpreted by the `unidecode` python package. Indeed, `unidecode` will simply remove characters it does not understand. A derivative package named `mufidecode` was built for this reason(T. [2019]): it takes precedent over `unidecode` equivalency tables when the data is known of the Medieval Unicode Font Initiative (MUFI, Initiative [2015]).

2.1.2 Model

Aside from normalizations of the input and output, three different model structures were tested. Every model is composed of one encoder, as described below, and one Linear Classifier which classifies into 5 classes : Start of Sentence (= SOS), End of Sentence (= EOS), Padding (= PAD), Masked Token (= Word Content), Space (= Word Boundary). For final scores, SOS, EOS and PAD were ignored.

The encoders are the following (configurations in parenthesis):

[1] Chu-Ren et al. [2008] actually gave us the denomination used here: word boundary (WB) and word content (WC).

[2] 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*

```
import mufidecode
import unidecode
"sot la gn̊t abstinence dess eintes uirges ele  ꝑla"
mufidecode.mufidecode(" sot la gn̊t abstinence dess eintes uirges ele  ꝑla")
# ' sot la gnat abstinence dess eintes uirges ele  pla'
mufidecode.mufidecode(" sot la gn̊t abstinence dess eintes uirges ele  ꝑla", 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 gn̊t abstinence dess eintes uirges ele  ꝑla")
# ' 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

- 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 Datasets

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

- 193,734 training examples;
- 23,581 validation examples;
- 25,512 test examples
- Number of classes in testing examples: 482,776 WC; 169,094 WB
- Number of classes in unknown examples: 26,393 WC; 10,193 WB

Examples were generated automatically. They are between 2 and 8 words-length. In order to recreate the condition of OCR noise, dots were added randomly (0.2) between words. In order to augment the dataset, words are randomly (0.1) passed over the next example[3]. 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 example however should not go beyond 100 characters. The results corpora should be varied in sizes as shown by Figure 3. The corpora contains 193 different characters when not normalized, in which some MUFI characters appears few hundred times (*cf.* Table 2).

	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

[3]This data augmentation was limited to one word per sample

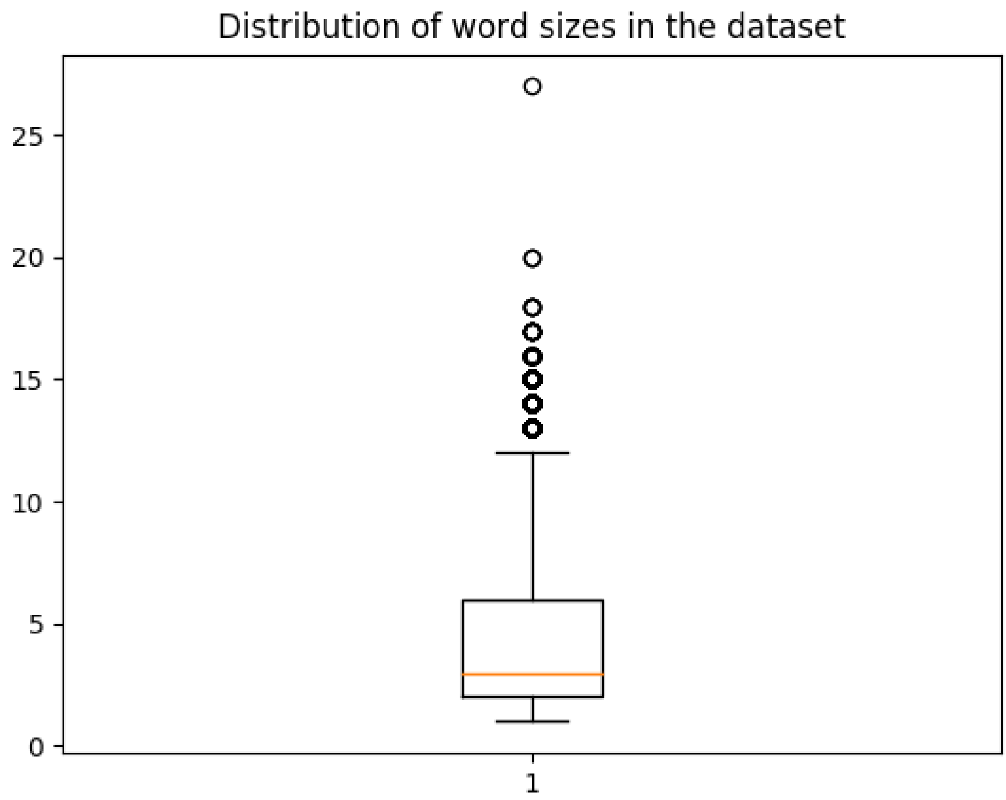

Figure 3: Distribution of word size over the train, dev and test corpora

2.2.2 *Results*

The training parameters were 0.00005 in learning rate for each CNN model, 0.001 for the LSTM model, and batch sizes of 64. Training reached a plateau fairly quickly for each model (*cf.* 4). Each model except LSTM reached a really low loss and a high accuracy on the test set (*cf.* 3). To compare the results, the `wordsegment` package G. [2018] was used as a baseline.

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	0.741	0.184	0.500	0.269	169094	0

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

2.2.3 *Unknown texts*

While all models using CNN show improvement over the baseline, the models definitely do not significantly outperform (<0.02 FScore). There is a reason for this: the baseline already

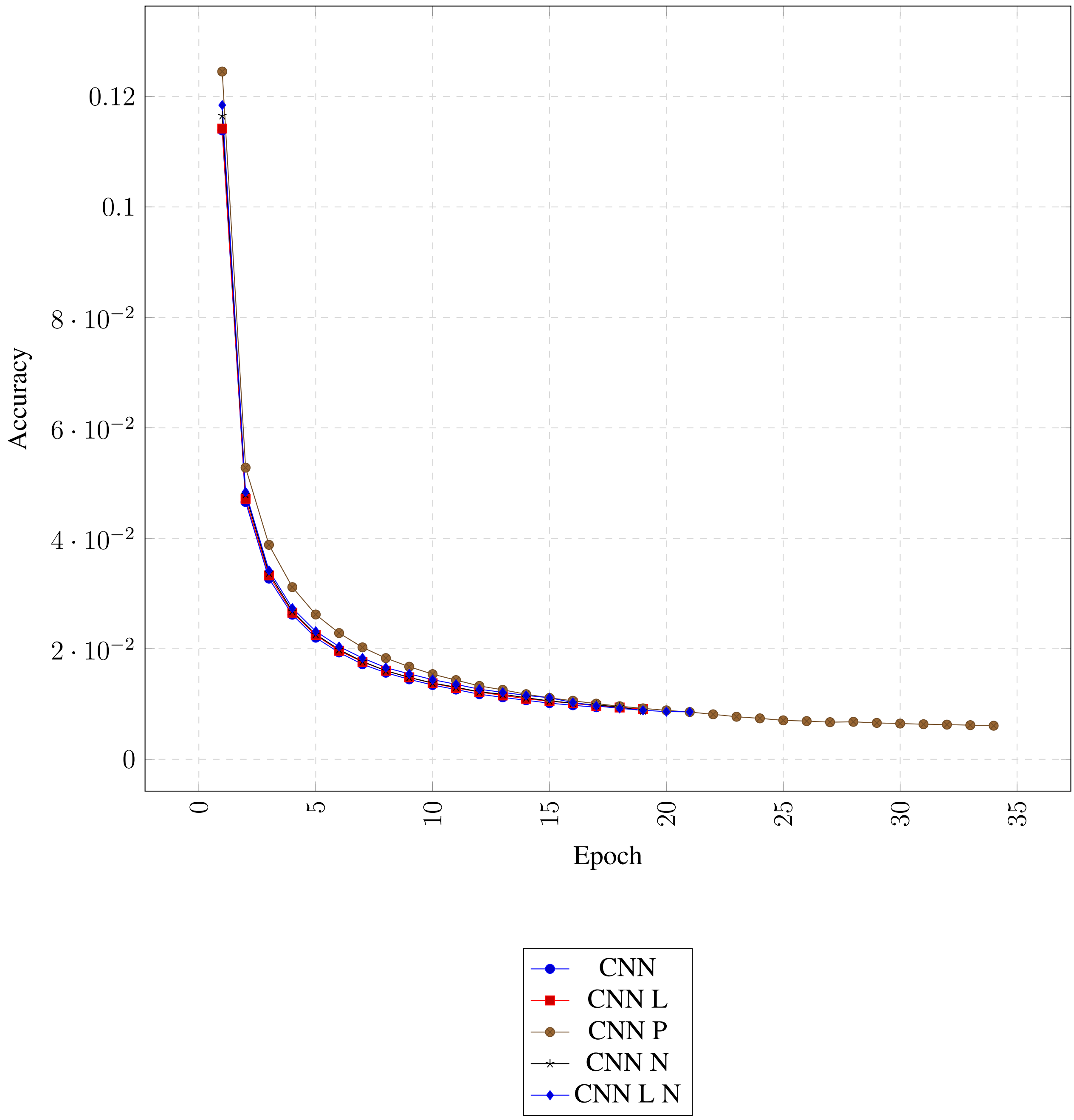

Figure 4: Training Loss (Cross-entropy) until plateau was reached. N = normalized, L = Lower, P = no position embedding. LSTM was removed as it did not go below 0.65

performs nearly perfectly on the test corpus. The dictionary attack using n-grams did actually perform well. Therefore, an additional evaluation method was constructed. The baseline and the best achieving deep-learning model (CNN P) were evaluated on a secondary test corpus composed of texts that were not used in training: indeed, the training, dev and test corpus share the same texts while not sharing the same source corpora and as such the same writing, the same vocabulary. This new corpus is composed by 4 texts and counts 742 examples : the diplomatic edition of the *Graal* (Marchello-Nizia et al. [2019]), a *Passion* and a *Vie de Saint Leger* (Sneddon [2019]), a *Vie de Saint Thibaut* (M.-G. [2019]). Neither noise characters nor random keeping of words were applied.

The results here were highly different (*cf.* Table 4): while it appears that the CNN is able to expand its ”comprehension” of the language to newer texts, the new words are more difficult to take into account for the baseline `wordsegment` n-gram approach, resulting in a respective drop to 0.945 and 0.838 FScore. WordSegment specifically performed badly with WB false positives : it had 3658 over a corpus containing 10,193 WB token (around 35 %).

	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 over the unknown dataset. FN = False Negative, FP = False Positive

2.2.4 *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**
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

2.3 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 by the same texts that the training text, and an unknown corpora composed by unseen texts.

2.3.1 *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	Unknown	0.933	0.897	0.890	0.893	1587	1409
CNN P	Unknown	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 more noisy than the Old French, as it was less curated than the digital edition provided for Old French. They are part of the Perseus corpus Crane et al. [2019] and were cut into passages in the context of my thesis. The training, evaluation and test corpora are built upon prose works from Cicero and Suetonius. The unknown corpus is built upon *Epigrammata* from Martial, from book 1 to book 2, as it should be fairly different in word order, vocabulary, etc. Both corpus 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 unknown examples: 35,910 WC; 8,828 WB

Example:

- Input : operecuperemdeberemqueprofecto
- Output : opere cuperem deberemque profecto

2.3.2 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	Unknown	0.929	0.900	0.865	0.881	14,382	27,019
CNN P	Unknown	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 corpora is based on the project Formulae - Litterae - Chartae's open data (Depreux et al. [2019]) for its training, evaluation and test sets; the unknown corpora is based on three texts from the Monumenta Germanica (K. [2019]) that are from early to late medieval period (Andreas Agnellus, Manegaldus, Theodoricus de Niem) and are drawn from the Corpus Corporum Project. Both corpus were generated without noise and word keeping, with a maximum sample size of 150 characters. The data presents some MUFI characters but still look like mostly 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 unknown examples: 472,655 WC; 113,004 WB

p

Example:

- Input : nonparvamremtibi
- Output : non parvam rem tibi

2.3.3 *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	Unknown	0.879	0.834	0.817	0.825	8693	11332
CNN P	Unknown	0.953	0.939	0.926	0.932	4689	3112
Baseline	Unknown Uppercase	0.879	0.834	0.817	0.825	8693	11332
CNN P	Unknown 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 epigraphic Latin corpora is based on the Epigraphic Database Heidelberg open data Depreux et al. [2019] for its training, evaluation and test sets (HD000001-HD010000 and HD010001-HD020000 from Witschel et al. [2019]) while the corpus of unknown is drawn from an automatic conversion of the Pompei Inscriptions (Clérice [2017]). Both the baseline and the model were evaluated on uppercase data, as it would normally be the state the text would be found in. Each of the corpora presents a high number of unresolved abbreviations (*ie.* one letter words). Both corpus were generated without noise and word keeping, with a maximum sample size of 150 characters. The data presents some polytonic Greek characters, some sample being 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 unknown examples: 127,268 WC; 38,055 WB

Example:

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

2.4 Discussion

Aside from a graphical challenge, word segmentation in OCR from manuscripts can actually be treated as a NLP task. Word segmentation for some text can be even difficult for humanist, as shown by the manuscript sample, and as such, it seems that the post-processing of OCR through tools like this one can be a better way to achieve 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 corpus, the CNN model definitely performed better on a completely unknown corpus. In this context, the proposed model actually shows its ability to carry over unknown corpora in a better way than classical n-gram approaches. In light of the high accuracy of the CNN model, the model should perform the same way independently of the language in Medieval Western Europe,.

2.5 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 manuscript. Given the results, studying the importance of normalization and lowering should probably be a further step, as it might be of high influence in smaller corpora.

2.6 Acknowledgements

Boudams has been made possible by two open-source repositories from which I learned and copied bits of 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.

References

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

T. Bluche, S. Hamel, C. Kermorvant, J. Puigcerver, D. Stutzmann, A. H. Toselli, and E. 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.

J.B. Camps, T. Clérice, M. Kestemont, and Manjavacas E. 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.

J.B. Camps, T. Clérice, and A. Pinche. Stylometry for noisy medieval data: Evaluating paul meyer's hagiographic hypothesis. In *DH2019*, July 2019a.

J.B. Camps, A. Cochet, L. Ing, and P. Levêque. Jean-Baptiste-Camps/Geste: Geste: un corpus de chansons de geste, 2016-..., April 2019b. URL https://doi.org/10.5281/zenodo.2630574.

X. Chen, X. Qiu, C. Zhu, P. Liu, and X. 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.

H. Chu-Ren, Y. Ting-Shuo, S. Petr, and H. Shu-Kai. A realistic and robust model for chinese word segmentation, 2008.

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

G. R. Crane, T. Clérice, L. Cerrato, B. Almas, N. Jovanović, A. Gessner, P. J. Burns, S. R. Dee, M. Munson, M. Jøhndal, Z. Himes, M. Foradi, M. Mernitz, M. Seydi, and T. Buckhingam. Perseusdl/canonical-latinlit 0.0.421, May 2019. URL https://doi.org/10.5281/zenodo.3236496.

P. Depreux, M. Munson, M. Pica, M. Faye, and ??? Formulae - litterae - chartae, June 2019. URL https://github.com/Formulae-Litterae-Chartae/formulae-open.

Jenks G. Wordsegment, July 2018. URL https://github.com/grantjenks/python-wordsegment.

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

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

Grossel M.-G. 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.

E. Manjavacas, C. Clérice, and M. Kestemont. emanjavacas/pie v0.2.3, April 2019. URL https://doi.org/10.5281/zenodo.2654987.

C. Marchello-Nizia, A. Lavrentiev, I. Vedrenne-Fajolles, and Heiden S. 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.

A. 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.

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

D. 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.

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

Master TNAH. Exercices TEI du master Technologies Numériques Appliquées à l'Histoire, 2019. URL https://github.com/Chartes-TNAH/digital-edition.

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

C. Witschel, G. Alföldy, J. M.S. Cowey, F. Feraudi-Gruénais, B. Gräf, F. Grieshaber (IT), R. Klar, and J. and Osnabrügge. Epigraphic Database Heidelberg, 2019. URL https://edh-www.adw.uni-heidelberg.de/.

C. Yu, S. Wang, and J. 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.

P. 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

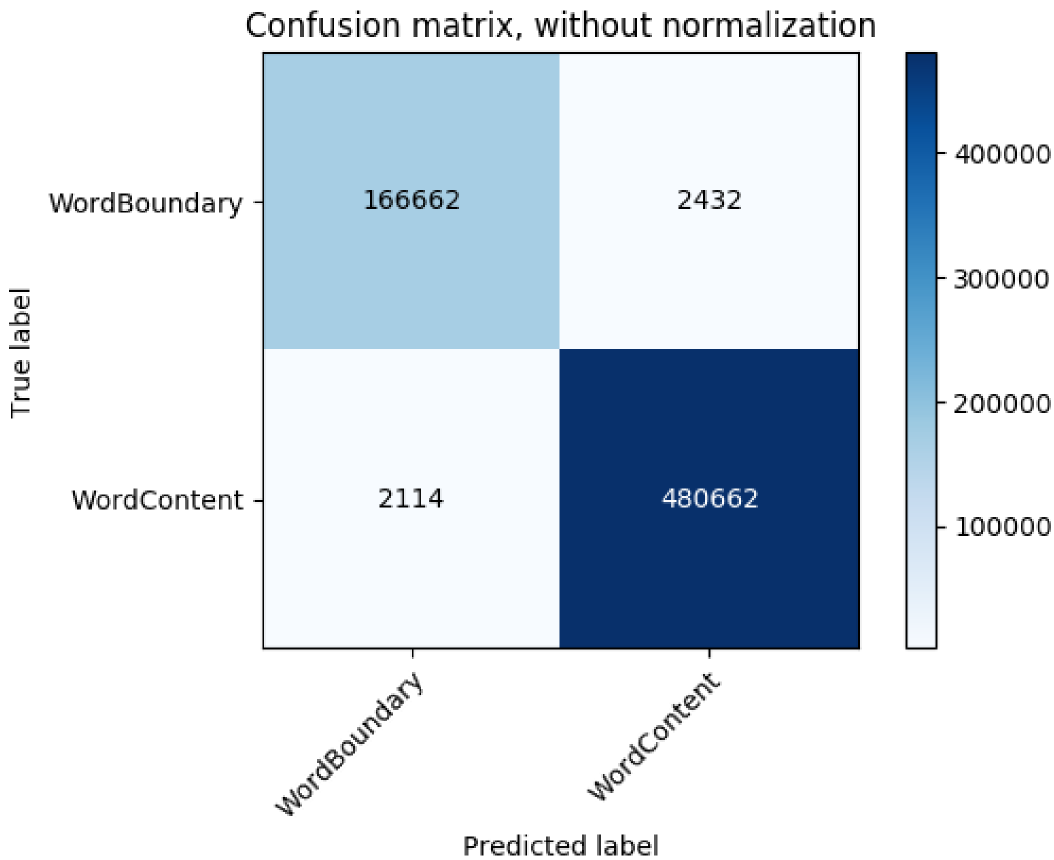

Figure 5: Confusion matrix of the CNN model without position embedding