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

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. Although, when dealing with old sources like manuscript written in *scripta continua* or later manuscripts, (1) such markers are mostly absent, (2) spelling variation and rich morphology makes dictionary based approaches difficult. We show that applying convolutional encoding to characters followed by linear categorization to word-boundary or in-word-sequence can be used to tokenize such inputs. Additionally, we release a software with a rather simple interface for tokenizing one's corpus.

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 computer when compared to "whathumancando". *Scripta continua* is a writing phenomenon where words would not be separated by spaces and it appears to have disappeared around the 8th century (see Zanna [1998]). Never the less, spacing can be somewhat erratic in later centuries writings, as show by Figure 1, a document from the 13th century. 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 **CITE JBCAMPS ?**.

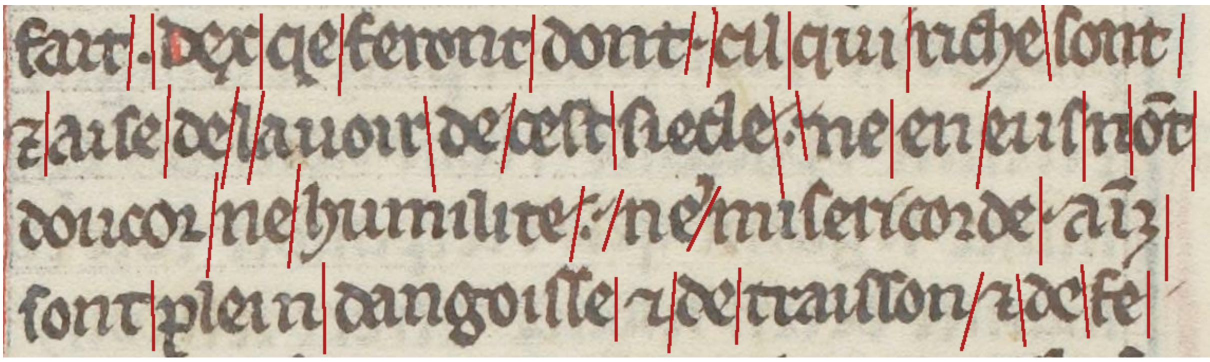

Figure 1: 4 lines from fol.103rb Manuscript fr. 412, Bibliothèque nationale de France. Red lines indicate word boundaries

We must stress in this study that the difficulty that we face is different for *scripta continua* than the ones researchers face languages such as Chinese for which an already impressive amount of work has been done as it. Indeed, 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. Actually, meanwhile redacting this article and producing the code-base, Huang et al. [2019] took the same approach of encoding to linear classification to both word boundary (WB) and word content (WC) for Chinese word segmentation.

Indeed, while Chinese's issue seems to lie in the decomposition of relatively fix characters, Old French or medieval latin present heavy variation of spelling. In Jean-Baptiste Camps [2017], Camps notes, in the same corpus, the existence of not less than 29 spelling of the word *cheval* (horse in Old and Modern French) whose apparition counts span from 3907 to 1[1]. This makes a dictionary approach rather difficult as it would rely on a high number of different spelling and makes 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, we propose two options for data transcoding: a lower-case normalization and a "reduction to the ASCII character set" feature (fr. 2). On this point, a lot of issues were found with transliteration of medieval paelographic characters that were part of the original datasets, as they are badly interpreted by the `unidecode` python package. Indeed, `unidecode` will simply remove characters it does not understand. I built a secondary package named `mufidecode` (Clérice [2019]) which precedes 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 structure of models were tested. Every model is composed by one encoder 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] 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 gnͣt abstinence dess eintes uirges ele  ꝑla"
mufidecode.mufidecode(" sot la gnͣt abstinence dess eintes uirges ele  ꝑla")
# ' sot la gnat abstinence dess eintes uirges ele  pla'
mufidecode.mufidecode(" sot la gnͣt 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 gnͣt 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*

Datasets are transcription from manuscripts with unresolved abbreviation coming from different projects. The **Old French** is based on Bluche et al. [2017], Pinche [2017], Jean-Baptiste-Camps et al. [2019], Lavrentiev [2019], and TNAH [2019]. It contains

- 193,734 training examples;
- 23,581 validation examples;
- 25,512 test examples

The input was generated by grouping at least 2 words and a maximum of 8 words together per sample. On a probability of 0.2, noise character could be added (noise character was set to DOT ('.')) and some words were kept randomly from a sample to another on a probability of 0.3 and a maximum number of word kept of 1. If a minimum size of 7 characters was not met in the input sample, another word would be added to the chain. A maximum input size of 100 was kept. The results corpora should be varied in sizes as shown by 3. The corpora is composed by 193 different characters when not normalized, in which some MUFI characters appears few hundred times 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

2.2.2 *Results*

The training parameters was 0.00005 in learning rate for each CNN model and 0.001 for the LSTM one, and 64 in batch sizes. 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, we used the `wordsegment` package Jenks [2018] as a baseline.

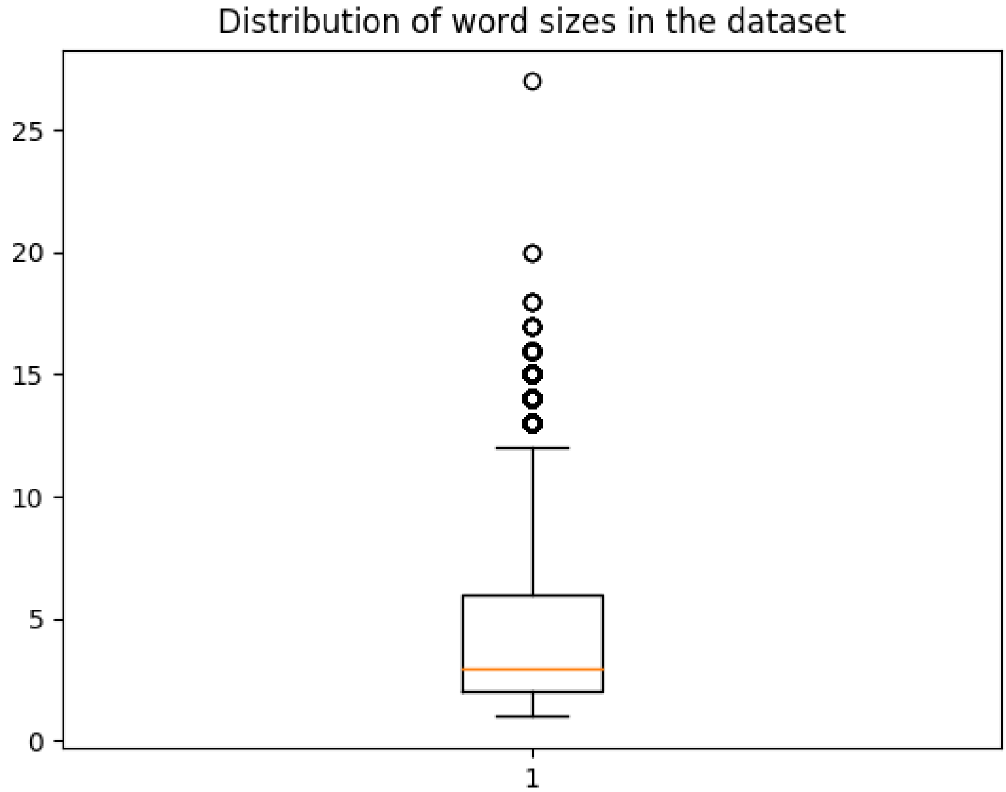

Figure 3: Distribution of word size over the train, dev and test corpora

Model	Accuracy	Precision	Recall	FScore
Baseline	0.989	0.986	0.984	0.985
CNN	0.991	0.985	0.990	0.987
CNN L	0.991	0.979	0.990	0.985
CNN P	**0.993**	**0.990**	**0.991**	**0.990**
CNN N	0.991	0.987	0.988	0.988
CNN L N	0.992	0.988	0.989	0.988
LSTM	0.741	0.184	0.500	0.269

Table 3: Scores over the test dataset. N = normalized, L = Lower, P = no position embedding.

2.2.3 Unknown texts

While all model using CNN shows improvement over the baseline, the model definitely does not outperform it by a huge margin (<0.02 FScore). And for a reason : the baseline already performs nearly perfectly on the test corpus. The dictionary attack using n-grams did actually perform well. As a result, we wanted to compare how both models would perform on a secondary test corpus composed by texts that were not used in training: indeed, the training, dev and test corpus share the same texts while not sharing the same inputs. As a result, we created a new corpus with 4 texts and 742 samples : the diplomatic edition of the Graal Cristiane Marchello-Nizia [2019], a Passion and a Vie de Saint Leger Sneddon [2019], a Vie de Saint Thibaut Grossel [2019]. No noise characters and no random keeping of words were applied.

The results here were highly different (*cf.* Table 4): while it appears that the CNN is able to

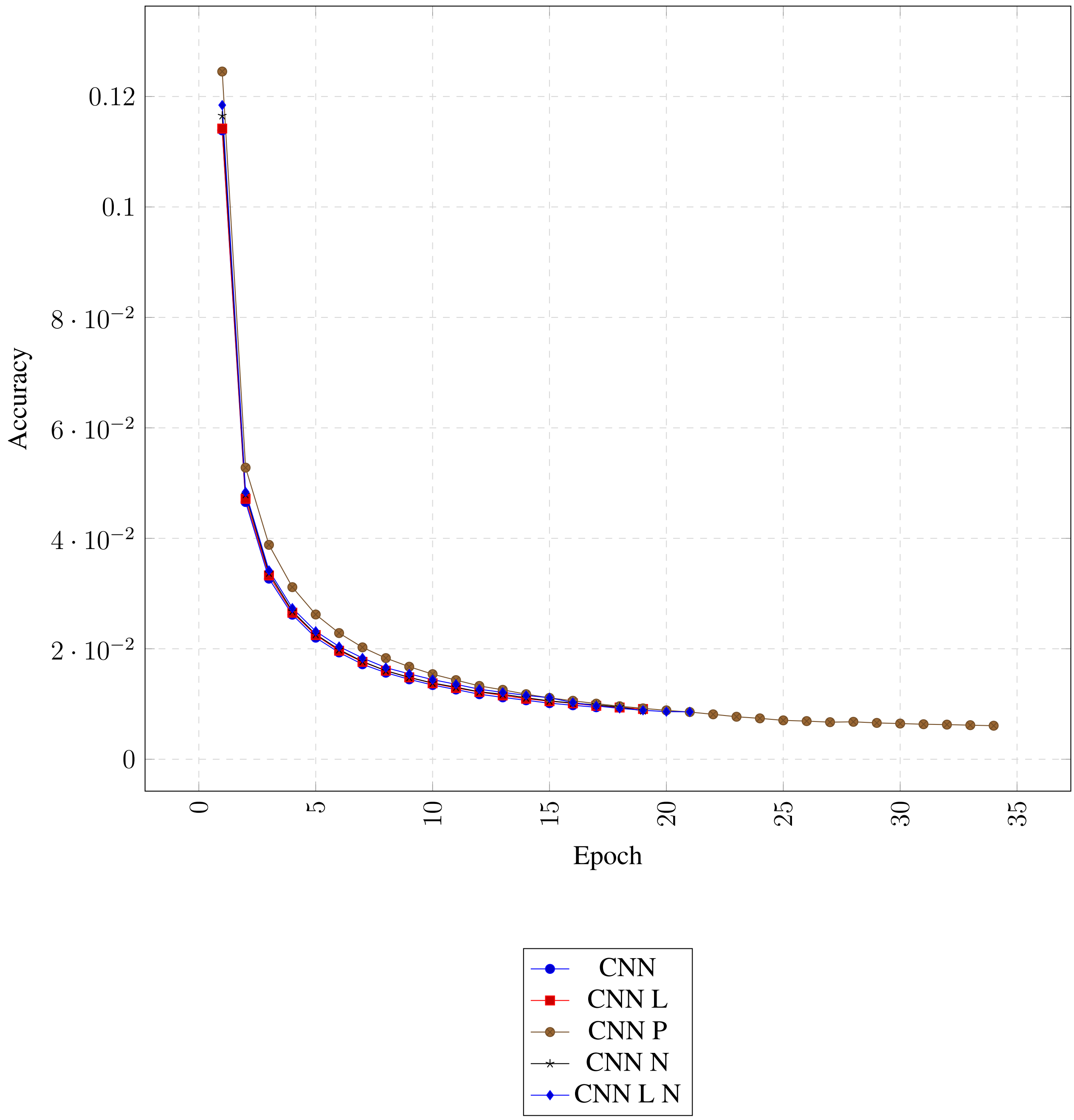

Figure 4: Training Loss (Cross-entropy) until plateau was reached. N = normalized, L = Lower, P = no position embedding. LSTM was removed as it did not go below 0.65

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 as WC	WC as WB
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. WB as WC and WC as WB are taken from the confusion matrices

2.2.4 *Example of outputs*

The following inputs has been tagged with the CNN P model. Batch 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, we only show the ground truth by commodity.

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 Discussion

We believe that, aside from a graphical challenge, word segmentation in OCR from manuscripts can actually be treated from a text point of view and as a NLP task. Word segmentation for some text can be even difficult for humanist, and as such, we believe that post-processing of OCR through tools like Boudams can be a better way to achieve data-mining of the dataset. In light of the high accuracy of the model, we believe the model should perform the same way independently from the language in Medieval Western Europe.

We were surprised by the negligible effects of the different normalization methods (lowercasing; ASCII reduction; both). The presence of certain MUFI characters might provide enough information about segmentation and be in enough numbers for them not to impact the network weights.

While the baseline surprised us by performing this well on the test corpus, it definitely performed less well than the CNN 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 ngram approaches.

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 important data mining of OCRed manuscript. In aftermath, we wonder if the importance of normalization and lowering should be higher depending on the size of the corpora and its content.

2.5 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. [2019] at DH2019 in Utrecht.

References

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.

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

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

Isabelle Vedrenne-Fajolles Serge Heiden Cristiane Marchello-Nizia, Alexei Lavrentiev. 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.

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.

Chu-Ren Huang, Ting-Shuo Yo, Petr Simon, and Shu-Kai Hsieh. A realistic and robust model for chinese word segmentation, 2019.

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

Jean-Baptiste-Camps, LAKME-ENC, AliceCochet, LucenceIng, and Paulinelvq. Jean-Baptiste-Camps/Geste: Geste: un corpus de chansons de geste, 2016-..., April 2019. URL https://doi.org/10.5281/zenodo.2630574.

Mike Kestemont-Enrique Manjavacas Jean-Baptiste Camps, Thibault Clérice. 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.

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

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

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

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.

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

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

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

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

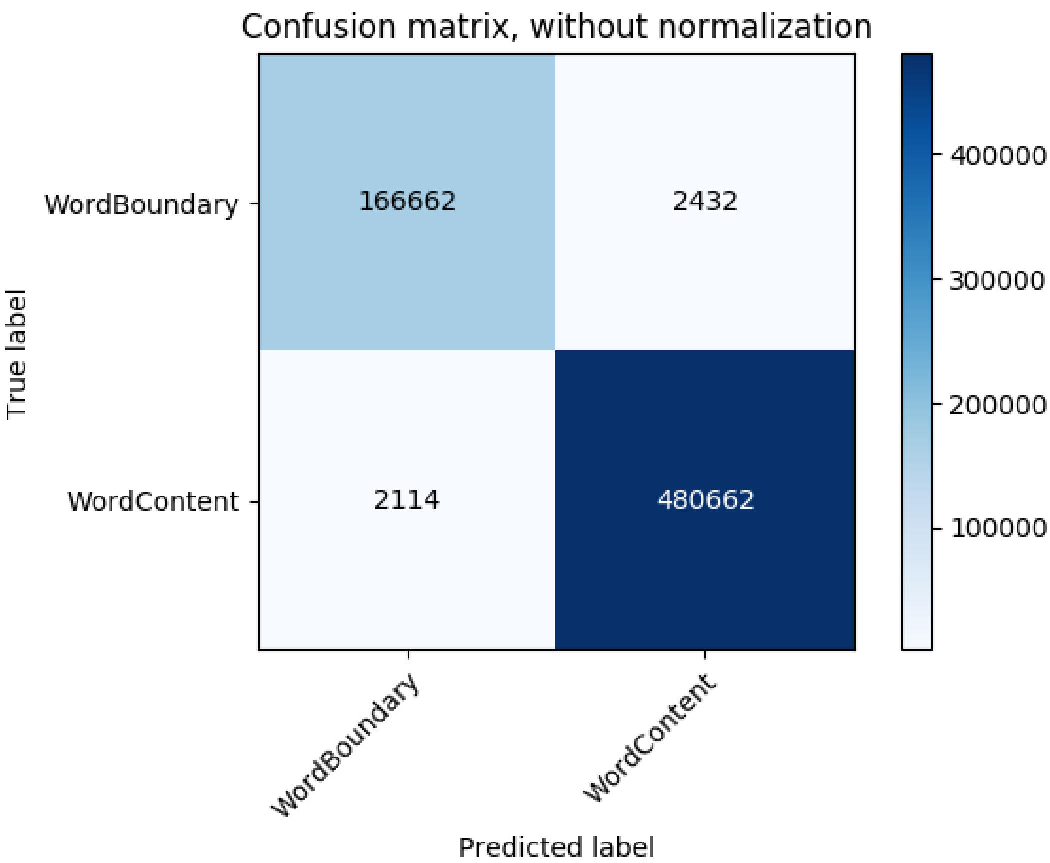

Figure 5: Confusion matrix of the CNN model without position embedding