# Evaluation of Named Entity Recognition in Latin using an Unsupervised Model

**Gabriel Cristian Circiu**
IT University of Copenhagen
Copenhagen, Denmark
gaci@itu.dk

**Mykyta Taranov**
IT University of Copenhagen
Copenhagen, Denmark
myta@itu.dk

**Wenzel Keil**
IT University of Copenhagen
Copenhagen, Denmark
weke@itu.dk

## Abstract

Tackling the challenges of Natural Language Processing (NLP) in Latin using an unsupervied model has lead to the observation that certain tokenizers such as BERT are better suited for such tasks and that the choice of the model itself plays a significant role in the final results. Clustering has proved to be the most straightforward method to group entities together but a more intricate pipeline lead to a significantly better result, achieving an F1 score of (???).

## 1 Introduction

As international students, the probability of sharing a common language outside of English is not high, however, during our studies we came to the realization that some of usshare knowledge in one such language, Latin. Albeit we are nowhere near fluent, it has sparked curiosity to work with it. We chose to head in the direction of our shared curiosity of Latin, and to poke at it within the Named Entity Recognition (NER) sphere.

As such we chose to build upon a past research, titled *Challenges and Solutions for Latin Named Entity Recognition*[1] (Erdmann et al., 2016), which tackles the problem of NER in Latin using a supervised and semi-supervised model. Our dataset is based on the one used in the original paper, following the same structure, with some additional specifications for clarity.

## 2 Dataset

The dataset from the original paper named 3 works of literature, Caesar's *De Bello Gallico (BG)*, Pliny the Younger's *Epistulae (EP)*, and Ovid's *Ars Amatoria (AA)*. We have had the great fortune of receiving the full dataset in its entirety, as large text files, in IOB format. While all the literature was scrambled in each file, having it in the right format has

[1]Original paper: https://aclanthology.org/W16-4012/

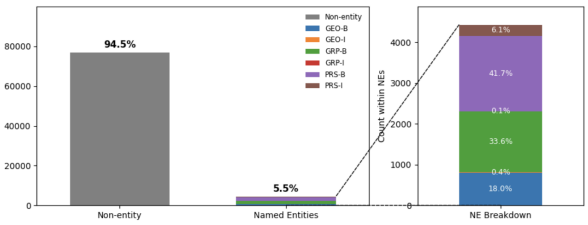


Figure 1: Distribution of data

made it all the more easier to process, and do some Exploratory Data Analysis (EDA). After thorough investigation, we have found that there is a slight inconsistency in the original paper regarding the dataset, and that of which we have received. As mentioned, the entirety of *BG*, and parts of *EP* and *AA* were annotated, However, parts of Caesar's *De Bello Civili (BC)* were also annotated within the dataset, but left unmentioned in the original paper.

## 3 Notes for myself

Distribution of data First tokenize on full words, but then used BERT tokenizer with Multilanguage. Capitalization is notable. Bert split words embedding not great, no benefit from using it. Combining back to words is good since we cluster on word level Cluster 1 stage, then 2 stage, and then NN Fpan F1 not a good idea to even make even if it's the main metric, because our model already does not perform well. NE vs Non NE is really good.

## 4 Embeddings

Our initial approach was to vectorize tokens using basic Word2Vec. However, this produced unacceptably low results — the model wasn't able to separate entities from non-entities in a meaningful way. So we moved on to a more powerful method: using contextual word embeddings from a pretrained BERT model, specifically bert-base-multilingual-cased. This model was trained on a large corpus of multilingual text, including Latin, and is capable of

generating high-quality embeddings for words in context. We used the Hugging Face Transformers library to load the model and generate embeddings for our tokens. We extacted the embeddings from the last hidden layer of the model. Since BERT works on subword level, we combined the embeddings of the subwords that make up a token by averaging subwords embeddings. This approach has been shown to work well in practice and allows us to obtain a single embedding for each token. We discusse the options of using subwords embeddings for clustering and switching to the word level later on, but this would led to unnecessary complications of the pipeline, for example, by introducing the necessaty of handling scenarios when embeddings of one word end up in different clusters.

## 5 Clustering

Once we had the word level embeddings, we applied Kmeans clustering to automatically group the tokens into clusters. Kmeans was the most straightforward choice for the task. One of the challenges we faced was the choice of K - number of clusters. We tried several methods to determine the optimal number of clusters, including the elbow method and silhouette score. The elbow method suggested that big values of K are better, up to the point that number of clusters was close to the number of tokens. Silhouette score also was not very helpful, since bigger score didn't correspond to better evaluation scores afterwards. So we boiled down the problem to the simplest approach: trial and error. We end up with K=!!!INSERT K!!! We used the KMeans implementation from the scikit-learn library, which provides a simple and efficient way to perform KMeans clustering. After clustering we named each cluster by taking the most frequent token in the cluster. We had a short discussion on whether this makes the model semi-supervised, but concluded that it doesn't: the label data is only used after clustering, and doesn't influence how the clusters are formed. Then we evaluated the clusters as a baseline for our model. We used the F1 score to measure the perfomance. F1 = !!!INSERT F1!!1

## 6 NN

Next we reconstructed the model using a neural network. We used a lightweight neural network consisting of two linear layers with GELU activation and LayerNorm, mapping contextual BERT embeddings to entity class logits. The network is trained using pseudo-labels derived from clustering, which makes inherently supervised neural network unsupervised. We used Pytorch to implement and train the model. More than 30 !!!!! epochs didn't seem to improve the results, so we stopped at 30.

## 7 Results

.

## 8 Future Work

.

.

## 9 Introduction (Original)

These instructions are for authors submitting papers to *ACL conferences using LaTeX. They are not self-contained. All authors must follow the general instructions for *ACL proceedings,[2] and this document contains additional instructions for the LaTeX style files.

The templates include the LaTeX source of this document (`acl_latex.tex`), the LaTeX style file used to format it (`acl.sty`), an ACL bibliography style (`acl_natbib.bst`), an example bibliography (`custom.bib`), and the bibliography for the ACL Anthology (`anthology.bib`).

## 10 Engines

To produce a PDF file, pdfLaTeX is strongly recommended (over original LaTeX plus dvips+ps2pdf or dvipdf). The style file `acl.sty` can also be used with lualaTeX and XeLaTeX, which are especially suitable for text in non-Latin scripts. The file `acl_lualatex.tex` in this repository provides an example of how to use `acl.sty` with either lualaTeX or XeLaTeX.

## 11 Preamble

The first line of the file must be

```
\documentclass[11pt]{article}
```

To load the style file in the review version:

```
\usepackage[review]{acl}
```

For the final version, omit the `review` option:

```
\usepackage{acl}
```

To use Times Roman, put the following in the preamble:

```
\usepackage{times}
```

(Alternatives like txfonts or newtx are also acceptable.)

Please see the LaTeX source of this document for comments on other packages that may be useful.

Set the title and author using `\title` and `\author`. Within the author list, format multiple authors using `\and` and `\And` and `\AND`; please see the LaTeX source for examples.

By default, the box containing the title and author names is set to the minimum of 5 cm. If you need more space, include the following in the preamble:

[2] http://acl-org.github.io/ACLPUB/formatting.html

| Command | Output | Command | Output |
|---|---|---|---|
| {\"a} | ä | {\c c} | ç |
| {\^e} | ê | {\u g} | ğ |
| {\`i} | ì | {\l} | ł |
| {\.I} | İ | {\~n} | ñ |
| {\o} | ø | {\H o} | ő |
| {\'u} | ú | {\v r} | ř |
| {\aa} | å | {\ss} | ß |

Table 1: Example commands for accented characters, to be used in, *e.g.*, BibTeX entries.

```
\setlength\titlebox{<dim>}
```

where <dim> is replaced with a length. Do not set this length smaller than 5 cm.

## 12 Document Body

### 12.1 Footnotes

Footnotes are inserted with the `\footnote` command.[3]

### 12.2 Tables and figures

See Table 1 for an example of a table and its caption. **Do not override the default caption sizes.**

As much as possible, fonts in figures should conform to the document fonts. See Figure 2 for an example of a figure and its caption.

Using the `graphicx` package graphics files can be included within figure environment at an appropriate point within the text. The `graphicx` package supports various optional arguments to control the appearance of the figure. You must include it explicitly in the LaTeX preamble (after the `\documentclass` declaration and before `\begin{document}`) using `\usepackage{graphicx}`.

### 12.3 Hyperlinks

Users of older versions of LaTeX may encounter the following error during compilation:

> `\pdfendlink` ended up in different nesting level than `\pdfstartlink`.

This happens when pdfLaTeX is used and a citation splits across a page boundary. The best way to fix this is to upgrade LaTeX to 2018-12-01 or later.

### 12.4 Citations

Table 2 shows the syntax supported by the style files. We encourage you to use the natbib styles.

[3]This is a footnote.

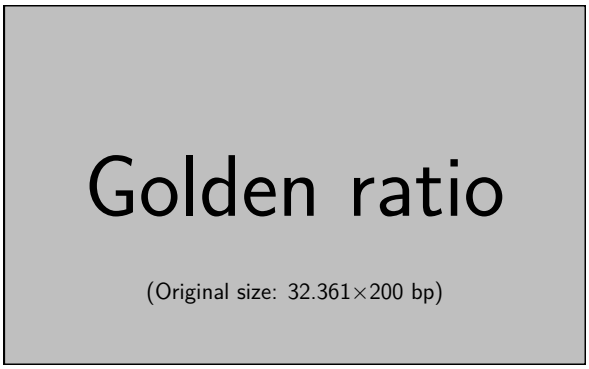


Figure 2: A figure with a caption that runs for more than one line. Example image is usually available through the `mwe` package without even mentioning it in the preamble.

You can use the command `\citet` (cite in text) to get "author (year)" citations, like this citation to a paper by Gusfield (1997). You can use the command `\citep` (cite in parentheses) to get "(author, year)" citations (Gusfield, 1997). You can use the command `\citealp` (alternative cite without parentheses) to get "author, year" citations, which is useful for using citations within parentheses (e.g. Gusfield, 1997).

A possessive citation can be made with the command `\citeposs`. This is not a standard natbib command, so it is generally not compatible with other style files.

### 12.5 References

The LaTeX and BibTeX style files provided roughly follow the American Psychological Association format. If your own bib file is named `custom.bib`, then placing the following before any appendices in your LaTeX file will generate the references section for you:

```
\bibliography{custom}
```

You can obtain the complete ACL Anthology as a BibTeX file from https://aclweb.org/anthology/anthology.bib.gz. To include both the Anthology and your own .bib file, use the following instead of the above.

```
\bibliography{anthology,custom}
```

Please see Section 13 for information on preparing BibTeX files.

### 12.6 Equations

An example equation is shown below:

$$A = \pi r^2 \tag{1}$$

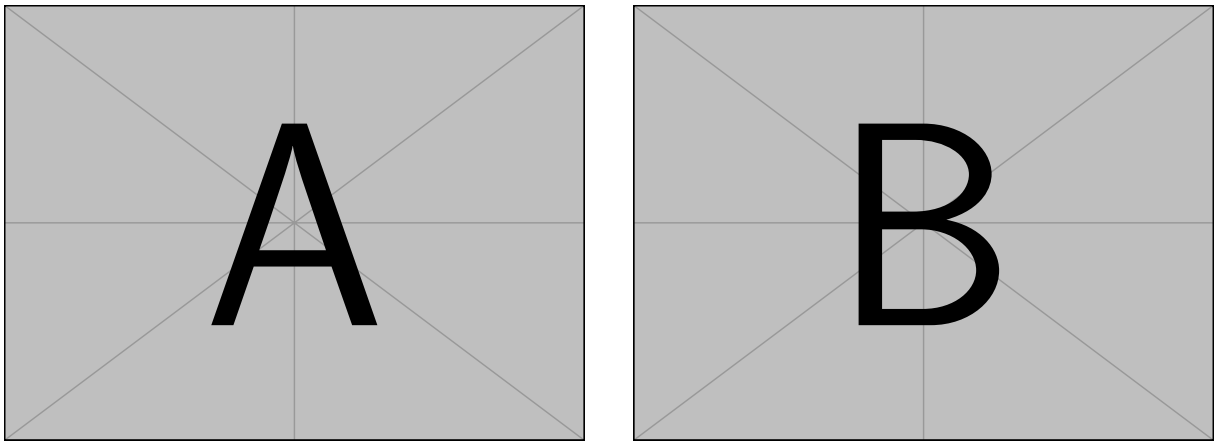


Figure 3: A minimal working example to demonstrate how to place two images side-by-side.

| Output | natbib command | ACL only command |
|---|---|---|
| (Gusfield, 1997) | \citep | |
| Gusfield, 1997 | \citealp | |
| Gusfield (1997) | \citet | |
| (1997) | \citeyearpar | |
| Gusfield's (1997) | | \citeposs |

Table 2: Citation commands supported by the style file. The style is based on the natbib package and supports all natbib citation commands. It also supports commands defined in previous ACL style files for compatibility.

Labels for equation numbers, sections, subsections, figures and tables are all defined with the `\label{label}` command and cross references to them are made with the `\ref{label}` command.

This an example cross-reference to Equation 1.

### 12.7 Appendices

Use `\appendix` before any appendix section to switch the section numbering over to letters. See Appendix A for an example.

## 13 BibTeX Files

Unicode cannot be used in BibTeX entries, and some ways of typing special characters can disrupt BibTeX's alphabetization. The recommended way of typing special characters is shown in Table 1.

Please ensure that BibTeX records contain DOIs or URLs when possible, and for all the ACL materials that you reference. Use the `doi` field for DOIs and the `url` field for URLs. If a BibTeX entry has a URL or DOI field, the paper title in the references section will appear as a hyperlink to the paper, using the hyperref LaTeX package.

## Limitations

Since December 2023, a "Limitations" section has been required for all papers submitted to ACL Rolling Review (ARR). This section should be placed at the end of the paper, before the references. The "Limitations" section (along with, optionally, a section for ethical considerations) may be up to one page and will not count toward the final page limit. Note that these files may be used by venues that do not rely on ARR so it is recommended to verify the requirement of a "Limitations" section and other criteria with the venue in question.

## Acknowledgments

This document has been adapted by Steven Bethard, Ryan Cotterell and Rui Yan from the instructions for earlier ACL and NAACL proceedings, including those for ACL 2019 by Douwe Kiela and Ivan Vulić, NAACL 2019 by Stephanie Lukin and Alla Roskovskaya, ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu, NAACL 2018 by Margaret Mitchell and Stephanie Lukin, BibTeX suggestions for (NA)ACL 2017/2018 from Jason Eisner, ACL 2017 by Dan Gildea and Min-Yen Kan, NAACL 2017 by Margaret Mitchell, ACL 2012 by Maggie Li and Michael White, ACL 2010 by Jing-Shin Chang and Philipp Koehn, ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, ACL 2005 by Hwee Tou Ng and Kemal Oflazer, ACL 2002 by Eugene Charniak and

Dekang Lin, and earlier ACL and EACL formats written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

## References


Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Alexander Erdmann, Christopher Brown, Brian Joseph, Mark Janse, Petra Ajaka, Micha Elsner, and Marie-Catherine de Marneffe. 2016. Challenges and solutions for Latin named entity recognition. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 85–93, Osaka, Japan. The COLING 2016 Organizing Committee.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.


## A Example Appendix

This is an appendix.