SinNer@CLEF-HIPE2020: Sinful Adaptation of SotA models for Named Entity Recognition in Historical French and German Newspapers

Pedro Javier Ortiz Suárez[1,3][0000−0003−0343−8852], Yoann Dupont[2], Gaël Lejeune[2], and Tian Tian[2]

[1] ALMAnaCH, Inria, Paris, France
{firstname.lastname}@inria.fr
[2] STIH, Sorbonne Université, Paris, France
{firstname.lastname}@sorbonne-universite.fr
[3] Sorbonne Université, Paris, France

Abstract. In this article we present the approaches developed by the Sorbonne-INRIA for NER (SinNer) team for the CLEF-HIPE 2020 challenge on Named Entity Processing on old newspapers. The challenge proposed various tasks for three languages, among them we focused on Named Entity Recognition coarse-grained in French and German texts. The best system we proposed ranked third for these two languages, it uses FastText embeddings and Elmo language models (FrELMo and German ELMo). We combine several word representations in order to enhance the quality of the results for all NE types. We show that reconstruction of sentence segments has an important impact on the results.

Keywords: Named Entity Recognition · Historical Texts · German · French · ELMo · CRFs. · Sentence Segmentation

1 Introduction

Among the aspects for which Natural Language Processing (NLP) can be useful for Digital Humanities (DH) figures prominently Named Entity Recognition. This task interests researchers for numerous reasons since the application can be pretty wide. We can cite genealogy or history for which finding mentions of persons and places in texts is very useful. Researchers in digital literature have shown a great interest in NER since it can help for instance to highlight the path of different characters in a book or in a book series. There can be cross-fertilization between NER and DH since some researchers showed that some particular properties of literature can help to build better NER systems

[1]. Apart from literature, NER can also be used more generally to help refine queries to assist browsing in newspaper collections [20]. Like other NLP tasks, NER quality will suffer from different problems related to variations in the input data: variation in languages (multilinguality), variation in the quality of the data (OCR errors mainly) and specificity of the application domain (literature vs. epidemic surveillance for instance). These difficulties can be connected with the challenges for low-level NLP tasks highlighted by Dale *et al.* [3]. In CLEF-HIPE shared task [6], the variation in language and in text quality will be the main problems even if the specificity of the application can be of great interest.

NER in old documents represent an interesting challenge for NLP since it is usually necessary to process documents that show different kind of variations as compared to the particular laboratory conditions on which NER systems are trained. Most NER systems are usually designed to process clean data. Additionally, there is the multilingual issue since NER systems have been designed primarily for English, with assumptions on the availability of data on the one hand and on the universal nature of some linguistic properties on the other hand.

The fact that the texts processed in Digital Humanities are usually not born-digital is very important since, even after OCR post-correction, it is very likely that some noise would be found in the text. Other difficulties will arise as well in those type of documents. The variation in language is one of them since contemporary English will clearly not be the most frequent language. It is interesting for researchers to check how much diachronic variation has an influence on NER systems [5]. It makes it even more important to work on multilingual NER and to build architectures that need less training data [26]. More generally, NER in ancient texts represents a great opportunity for NLP to compare to main approaches to handle variation in texts: adapting the texts to an existing architecture via modernization or normalization [17] or adapting the pipeline to non standard data (OCR noise, language variants. . .) via domain adaptation or data augmentation techniques [9].

In Section 2 we present a brief state-of-the-art for Named Entity Recognition with a focus on digitized documents. Section 3 and 4 are respectively devoted to the description of the dataset of CLEF-HIPE 2020 shared task and the methods we developed to extract NE for French and German. The results of our systems are described in Section 5 and in Section 6 we give some conclusions and perspectives for this work.

2 Related Work on Named Entity Recognition

Named Entity Recognition came into light as a prerequisite for designing robust Information Extraction (IE) systems in the MUC conferences [11]. This task soon began to be treated independently from IE since it can serve multiple purposes, like Information retrieval or Media Monitoring for instance [34]. As such, shared task specifically dedicated to NER started to rise like the CoNLL 2003 shared task [32]. Two main paths were followed by the community: (i) since NER was at first used for general purposes, domain extension start to gain interest [7]; (ii) since the majority of NER systems were designed for English, the extension to novel languages (including low resource languages) became of importance [28].

One can say that NER followed the different trends in NLP. The first approaches were based on gazeeters and handcrafted rules. Initially NER was considered to be solved by a patient process involving careful syntactic analysis [12]. Supervised learning approaches came to fashion with the increase of available data and the rise of shared tasks on NER. Decision trees and Markov models were soon outperformed by Condition Random Fields (CRF). Thanks to its ability to model dependencies and to take advantage of the sequentiality of textual data, CRF helped to set new state-of-the-art results in the domain [8]. Since supervised learning results were bound by the size of training data, lighter approaches were tested in the beginning of the 2000's, among them we can cite weakly supervision [33] and active learning [29].

During a time, most of promising approaches involved an addition to improve CRFs : word embeddings [24], (bi-)LSTMs [15] or contextual embeddings [25]. More recently, the improvements in contextual word embeddings made the CRFs disappear as standalone models for systems reaching state-of-the-art results, see [30] for a review on the subject and a very interesting discussion on the limits attained by state-of-the-art systems, the *Glass Ceiling*.

3 Dataset for the CLEF-HIPE shared task

The dataset of the CLEF-HIPE shared task contains newspaper articles of 17th-20th century. The text is an output of an OCR software, then tokenised and annotated with labels corresponding to each sub-task. This pecularity of historical documents will be detailed later in this section. The corpus provided for French and German both contained training data (train) and development data (dev) whereas, for English only development data was provided for the shared task. For this reason, we chose to work on French and German only. Table 1 shows some statistics of this dataset. The size of the train dataset was twice higher for French than for German whereas the development sets have roughly the same size. As usual in NER, persons (Pers) and locations (Loc) are the most frequent entity types.

	Tokens	Documents	Segments	Labeled named entities				
				Pers	Loc	Org	Time	Prod
Train Fr	166217	158	19183	3067	2513	833	273	198
Dev Fr	37592	43	4423	771	677	158	69	48
Train De	86960	104	10353	1747	1170	358	118	112
Dev De	36175	40	4186	664	428	172	73	53

Table 1. Statistics on the training and development data in French and German

Table 2 shows an excerpt of the train dataset (CoNLL format). For each document, general information were provided. Among them, newspaper and date may have been features useful for recognising entities but we did not take advantage of it. Each document was composed of segments, starting with "# segment . . . " corresponding to lines in the original documents. Each segment is tokenized in order to correspond to the CoNLL format with one token per line. These two notions, segments and tokens, are very important since they do not always match the type of unit usually processed in NLP pipelines. Segments seldom correspond to sentences so that there is a need to concatenate the segments to get the raw text and then segment it into sentences. This is very interesting since it gets us close to real-world conditions rather than laboratory conditions, and we show in Section 5.2 that this segment vs. sentence question has an important influence on the results. Regarding tokens, the tokenization is obviously not perfect. We can see that there are non-standard words and bad tokenization due to the OCR output (in red in Table 2). If we concatenate the tokens we get the sequence "Su. _sss allemands" instead of "Suisse allemande". These non-standard words make the Named Entity Recognition task more complicated and, again, more realistic.

TOKEN	NE-COARSE		NE-FINE			NE-NESTED	NEL		MISC
	LIT	METO	LIT	METO	COMP		LIT	METO	
# language = fr									
# newspaper = EXP									
# date = 1918-04-22									
# document_id = EXP-1918-04-22-a-i0077									
# segment_iiif_link = https://iiif.dhlab.epfl.ch/iiif_impresso. . .									
Lettre	O	O	O	O	O	O	_	_	_
de	O	O	O	O	O	O	_	_	_
la	O	O	O	O	O	O	_	_	_
Su	B-loc	O	B-loc.adm.reg	O	O	B-loc.adm.nat	Q689055	_	NoSpaceAfter
.	I-loc	O	I-loc.adm.reg	O	O	I-loc.adm.nat	Q689055	_	_
_	I-loc	O	I-loc.adm.reg	O	O	I-loc.adm.nat	Q689055	_	NoSpaceAfter
sss	I-loc	O	I-loc.adm.reg	O	O	I-loc.adm.nat	Q689055	_	_
allemands	I-loc	O	I-loc.adm.reg	O	O	O	Q689055	_	EndOfLine
# segment_iiif_link = https://iiif.dhlab.epfl.ch/iiif_impresso. . .									
(	O	O	O	O	O	O	_	_	NoSpaceAfter
Nous	O	O	O	O	O	O	_	_	_
serons	O	O	O	O	O	O	_	_	_
heureux	O	O	O	O	O	O	_	_	_
de	O	O	O	O	O	O	_	_	_
publier	O	O	O	O	O	O	_	_	_
. . .									

Table 2. Example extracted from the French training dataset

4 CRFs and Contextualized Word Embeddings for NER

4.1 CRF model (run3)

SEM (Segmenteur-Étiqueteur Markovien)[4][5] [4] is a free NLP tool that relies on linear-chain CRFs [14] to perform tagging. SEM uses Wapiti [16] v1.5.0[6] as linear-chain CRFs implementation. For this particular NER task, SEM uses the following features:

[4] available at: https://github.com/YoannDupont/SEM

[5] translates to: Markovian Tokenizer-Tagger (MTT).

[6] available at: https://github.com/Jekub/Wapiti

- token, prefix/suffix from 1 to 5 and a Boolean isDigit features in a [-2, 2] window;
- previous/next common noun in sentence;
- 10 gazetteers (including NE lists and trigger words for NEs) applied with some priority rules in a [-2, 2] window;
- a "fill-in-the-gaps" gazetteers feature where tokens not found in any gazetteer are replaced by their POS, as described in [27]. This feature used token unigrams and token bigrams in a [-2, 2] a window.
- tag unigrams and bigrams.

We trained a CLEF HIPE specific model by optimizing L1 and L2 penalties on the development set. The metric used to estimate convergence of the model is the error on the development set ($1 - accuracy$). For French, our optimal L1 and L2 penalties were 0.5 and 0.0001 respectively (default Wapiti parameters). For German, our optimal L1 and L2 penalties were 1.0 and 0.0001 respectively.

One interest of SEM is that it has a built-in sentence tokenizer for French using a rule-based approach. By default, CLEF-HIPE provides a newline segmentation that is the output of the OCR. As a result, some NE mentions span across multiple segments, making it very hard to identify them correctly. It is to be expected that models trained (and labelling on) sentences would yield better performances than those trained (and labelling on) segments. SEM makes it simple to switch between different sequence segmentations, which allowed us to label sentences and output segments. SEM's sentence segmentation engine works using mainly local rules to determine whether a token is the last of a sequence (eg: is a dot preceded by a known title abbreviation?). It also uses non-local rules to remember whether a token is between parentheses or French quotes to not segment automatically within them. Since we work at token level, we had to adapt some rules to fit CLEF-HIPE tokenization. For example, SEM decides at tokenization stage whether a dot is a strong punctuation or part of a larger token, as for abbreviations. This has the advantage of making sentence segmentation easier. CLEF-HIPE tokenization systematically separates dots, so we adapted some sentence segmentation rules, for example: we decided not to consider a dot as a sentence terminator if the previous token was in a lexica of titles or functions. No specific handling of OCR errors were done. Another interest is that SEM has an NE mention broadcasting process. Mentions found at least once in a document are used as a gazetteer to tag unlabeled mentions within said document. When a new mention overlaps and is strictly longer than an already found mention, the new mention will replace the previous one in the document.

4.2 Contextualized word embeddings

Embeddings from Language Models (ELMo) [25] is a Language Model, i.e, a model that given a sequence of N tokens, $(t_1, t_2, ..., t_N)$, computes the probability of the sequence by modeling the probability of token t_k given the history

$(t_1, ..., t_{k-1})$:

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1}).$$

However, ELMo in particular uses a bidirectional language model (biLM) consisting of L LSTM layers, that is, it combines both a forward and a backward language model jointly maximizing the log likelihood of the forward and backward directions:

$$\begin{aligned}\sum_{k=1}^{N} (\ &\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) \\ &+ \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)\ )\,.\end{aligned}$$

where at each position k, each LSTM layer l outputs a context-dependent representation $\overrightarrow{\mathbf{h}}_{k,l}^{LM}$ with $l = 1, \ldots, L$ for a forward LSTM, and $\overleftarrow{\mathbf{h}}_{k,l}^{LM}$ of t_k given $(t_{k+1}, \ldots, t_N)$ for a backward LSTM.

ELMo also computes a context-independent token representation $\mathbf{x}_k^{LM}$ via token embeddings or via a CNN over characters. ELMo then ties the parameters for the token representation (Θ_x) and Softmax layer (Θ_s) in the forward and backward direction while maintaining separate parameters for the LSTMs in each direction.

ELMo is a task specific combination of the intermediate layer representations in the biLM, that is, for each token t_k, a L-layer biLM computes a set of $2L+1$ representations

$$\begin{aligned}R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,l}^{LM}, \overleftarrow{\mathbf{h}}_{k,l}^{LM} \mid l = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,l}^{LM} \mid l = 0, \ldots, L\},\end{aligned}$$

where $\mathbf{h}_{k,0}^{LM}$ is the token layer and

$$\mathbf{h}_{k,l}^{LM} = [\overrightarrow{\mathbf{h}}_{k,l}^{LM}; \overleftarrow{\mathbf{h}}_{k,l}^{LM}],$$

for each biLSTM layer.

When included in a downstream model, as it is the case in this paper, ELMo collapses all L layers in R into a single vector $\mathbf{ELMo}_k = E(R_k; \mathbf{\Theta}_e)$, generally computing a task specific weighting of all biLM layers:

$$\begin{aligned}\mathbf{ELMo}_k^{task} &= E(R_k; \Theta^{task}) \\ &= \gamma^{task} \sum_{l=0}^{L} s_l^{task} \mathbf{h}_{k,l}^{LM}\,.\end{aligned}$$

applying layer normalization to each biLM layer before weighting.

Following [25], we use in this paper ELMo models where $L = 2$, i.e., the ELMo architecture involves a character-level CNN layer followed by a 2-layer biLSTM.

4.3 ELMo-LSTM-CRF (run1 and run2)

The LSTM-CRF is a model originally proposed by Lample et al. [15] it consists of a Bi-LSTM encoder pre-appended by both character level word embeddings and pre-trained word embeddings, and a CRF decoder layer. For our experiments, we follow the same approach as Ortiz Suárez et al. [21] by using the Bi-LSTM-CRF implementation of Straková et al. [31] which is open source and readily available[7], and pre-appending contextualized word-embeddings to the model. For French we pre-append the FrELMo model [21], which is the standard ELMo [25] implementation[8] trained on the French OSCAR[9] corpus [22] [23]. For German we pre-append the German ELMo [19], which is again the standard ELMo implementation but trained on the German Wikipedia.

Contrary to the approach of Ortiz Suárez et al. [21], we do not use the CamemBERT model [18] for French or the German BERT [2]. Both of these models are BERT-based and as such they are limited to a 512-token contextualized window. Moreover, they both use SentencePiece [13] meaning that tokens are actually subwords, which considerably increases the number of tokens per sentence, specially for the longer ones, thus decreasing the contextual windows of both CamemBERT and the German BERT. SentencePiece also introduces the problem of a fixed-size vocabulary, which in the case of this shared task might negatively impact the performance of said models, as they could struggle handling OCR problems or just non-standard vocabulary. Since our main goal was to reconstruct the sentences and use long contextualized sequences we opted to use ELMo which can easily handle longer sequences with it's standard implementation and actually has a dynamic vocabulary thanks to the CNN character embedding layer, thus it might be better equipped to handle non-standard orthography and OCR problems.

For the fixed word embeddings we used the Common Crawl-based FastText embeddings [10] originally trained by Facebook as opposed to the embeddings provided by the HIPE shared task, as we obtained better dev scores using the original FastText embeddings for both French and German.

We used the standard hyperparameters originally[10] used by Straková et al. [31]. Namely a batch size of 8, a dropout of 0.5, a learning rate of 0.001 and 10 epochs. The difference between run 1 and 2, is that run 1 uses the data as is, while run 2 uses the reconstructed sentences.

5 Results and Discussion

5.1 Official shared task results

The results of our 3 runs compared to the best run on the NERC-coarse shared-task for French and German are given in Table 3 (strict scenario). For both

[7] Available at: `https://github.com/ufal/acl2019_nested_ner`.

[8] Available at: `https://github.com/allenai/bilm-tf`

[9] Available at: `https://oscar-corpus.com`

[10] `https://github.com/ufal/acl2019_nested_ner/blob/master/tagger.py#L484`.

tasks, we are the third best ranking team. We only did very minimal adaptation of existing systems. We did not modify tokenization for any language. The most notable change was to use custom sentence segmentation instead of given segments for French and using some additional lexica as features for our CRF model in German (for French, we only used existing SEM lexica). Other than that, we only optimized hyper-parameters on the dev set. This clearly illustrates the power of contextual embeddings and today's neural network architectures. This is encouraging in terms of usability of SotA models on real-world data.

run	French			German		
	P	R	F1	P	R	F1
winner	83.1	84.9	84.0	79.0	80.5	79.7
run 1	77.8	79.4	78.6	63.1	**66.6**	64.8
run 2	**78.8**	**80.2**	**79.5**	**65.8**	65.8	**65.8**
run 3	70.2	57.9	63.5	64.4	43.8	52.1
average	70.2	66.7	67.6	63.8	58.1	60.0
median	71.5	68.6	68.6	66.8	57.7	64.5

Table 3. Strict results for our systems compared to the winning system (micro measures)

5.2 Study of sequence segmentation

In this section, we evaluate the influence of sequence segmentation on system performances. This evaluation is done for French only, as we used SEM to provide sentence segmentation and SEM could only provide a proper sentence segmentation for that language. As can be seen in table 4, sentence segmentation allows to improve results by 3.5 F1 points. This is due to the fact that some entities were split across multiple segments in the original data. Using a custom sentence segmentation allows to have entities in a single sequence. This segmentation is applied both with training data and evaluation data, so that our systems can access a more proper context for named entities. The cost of using another segmentation is relatively cheap, as SEM can process nearly 1GB of raw text per hour.

A per entity comparison is also available in Table 4. One can see that the improvement of sentence segmentation is not very significant for locations (Loc). It is due to two facts : (i) locations are usually small in number of tokens and therefore less prone to be separated in two segments and (ii) there was less room from improvement since they were the easiest entity type to detect (86.35% F1-score). To the contrary, entities of type "product" (Prod), usually longer in tokens, were very hard to predict with only 48.57% F1-measure and benefited the most from segmentation in sentences (+16 percentage points in F1-measure).

Type	P		R		F1	
	Segments	Sentences	Segments	Sentences	Segments	Sentences
Loc	85.21	87.73 (+2.52)	87.52	87.08 (-0.44)	86.35	87.41 (+1.06)
Org	70.62	71.33 (+0.71)	62.78	65.64 (+2.86)	66.47	68.37 (+1.90)
Pers	80.24	84.64 (+4.40)	76.88	82.09 (+5.21)	78.52	83.35 (+4.83)
Prod	62.96	75.86 (+12.90)	39.53	56.41 (+16.88)	48.57	64.71 (+16.14)
Time	86.21	90.91 (+4.70)	78.12	87.72 (+9.60)	81.97	89.29 (+7.32)
Global	81.03	84.46 (+3.43)	81.61	84.46 (+2.85)	79.52	83.01 (+3.49)

Table 4. Comparison between segments and sentences on French dev dataset (run 1), strict scenario

5.3 To dev or not to dev?

In Table 5 we show the results that could have been obtained by training the Bi-LSTM model on both train and dev dataset. We used the same hyperparameters as we did for our official run. Despite the fact that it does not ensure the robustness of the system, the added-value seem to be quite disappointing[11]. In German the gain may be a bit more significant, probably due to the smaller size of the training dataset.

metric	french		german	
	not to dev	to dev	not to dev	to dev
P	78.8	**79.5** (+0.7)	65.8	**68.2** (+2.4)
R	80.2	**80.7** (+0.5)	65.8	**66.1** (+0.3)
F1	79.5	**80.1** (+0.6)	65.8	**67.1** (+1.3)

Table 5. Results obtained on the test set (strict metric) with only the train set (not to dev) and with train+dev sets (to dev) with our best system (run 2)

6 Conclusion

In this article we presented three methods developed for the Named Entity Recognition task in French and German historical newspapers. The first method relied on linear-chain CRFs while the other two methods use a Bidirectional LSTM and a bidirectional Language Model (ELMo). The later outperformed the CRF model and achieved rank 3 on the NER task in both French and German. We also showed that the type of sequences used has a significant influence on the results. When we segment in sentences rather than using the segments of

[11] In particular, if we consider that it would not have given us a better ranking on any language.

the dataset as it is the results are systematically much better, with an exception for locations where the gain is marginal. This proves that sentence segmentation remains a key component of efficient NLP architectures, in particular for models taking advantage of the context.

As a future work it would be interesting to assess the importance of noise in the data. For instance, by comparing the results of NER on texts obtained via different OCR tools. The influence of the qualitative jumps in the data, which is common in Digital Humanities, is an important aspect to evaluate the robustness of the system in real-world conditions rather than laboratory conditions. We also plan to provide an in-depth analysis of the impact of word embeddings and neural architecture, as we only provided our best results in this paper.

References

1. Brooke, J., Hammond, A., Baldwin, T.: Bootstrapped text-level named entity recognition for literature. In: Proc. of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 344–350. Berlin, Germany (2016)
2. Chan, B., Möller, T., Pietsch, M., Soni, T., Yeung, C.M.: German bert. `https://deepset.ai/german-bert` (2019)
3. Dale, R., Somers, H.L., Moisl, H.: Handbook of Natural Language Processing. Marcel Dekker, Inc., USA (2000)
4. Dupont, Y.: Exploration de traits pour la reconnaissance d'entités nommées du français par apprentissage automatique. In: 24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). p. 42 (2017)
5. Ehrmann, M., Colavizza, G., Rochat, Y., Kaplan, F.: Diachronic evaluation of NER systems on old newspapers. Proc. of the 13th Conference on Natural Language Processing (KONVENS 2016) pp. 97–107 (2016)
6. Ehrmann, M., Romanello, M., Flückiger, A., Clematide, S.: Extended Overview of CLEF HIPE 2020: Named Entity Processing on Historical Newspapers. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum. CEUR-WS (2020)
7. Evans, R.: A framework for named entity recognition in the open domain. In: Proc. of the Recent Advances in Natural Language Processing (RANLP). pp. 137–144 (2003)
8. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proc. of the 43rd Annual Meeting on Association for Computational Linguistics. p. 363–370. USA (2005)
9. Ghannay, S., Caubrière, A., Estève, Y., Camelin, N., Simonnet, E., Laurent, A., Morin, E.: End-to-end named entity and semantic concept extraction from speech. In: IEEE Spoken Language Technology Workshop. Athens, Greece (Dec 2018)
10. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proc. of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018)
11. Grishman, R., Sundheim, B.: Design of the MUC-6 evaluation. In: Proc. of the 6th Conference on Message Understanding. p. 1–11. MUC6 '95, Association for Computational Linguistics, USA (1995)

12. Hobbs, J.R.: The generic information extraction system. In: Proc. of the 5th Conference on Message Understanding. p. 87–91. MUC5 '93, Association for Computational Linguistics, USA (1993)
13. Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71. Association for Computational Linguistics, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/D18-2012, `https://www.aclweb.org/anthology/D18-2012`
14. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. of the Eighteenth International Conference on Machine Learning (ICML) 2001, Williams College, Williamstown, MA, USA. pp. 282–289 (2001)
15. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270. San Diego, California (Jun 2016)
16. Lavergne, T., Cappé, O., Yvon, F.: Practical very large scale CRFs. In: Proc. of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 504–513. Association for Computational Linguistics (2010)
17. Leaman, R., Lu, Z.: TaggerOne: joint named entity recognition and normalization with semi-Markov Models. Bioinformatics **32**(18), 2839–2846 (06 2016)
18. Martin, L., Muller, B., Ortiz Suárez, P.J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., Sagot, B.: CamemBERT: a tasty French language model. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7203–7219. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.645, `https://www.aclweb.org/anthology/2020.acl-main.645`
19. May, P.: German ELMo Model (2019), `https://github.com/t-systems-on-site-services-gmbh/german-elmo-model`
20. Neudecker, C., Wilms, L., Faber, W.J., van Veen, T.: Large-scale refinement of digital historic newspapers with named entity recognition. In: Proc. of IFLA 2014 (2014)
21. Ortiz Suárez, P.J., Dupont, Y., Muller, B., Romary, L., Sagot, B.: Establishing a new state-of-the-art for French named entity recognition. In: Proc. of The 12th Language Resources and Evaluation Conference. pp. 4631–4638. European Language Resources Association, Marseille, France (May 2020)
22. Ortiz Suárez, P.J., Romary, L., Sagot, B.: A monolingual approach to contextualized word embeddings for mid-resource languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1703–1714. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.156, `https://www.aclweb.org/anthology/2020.acl-main.156`
23. Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In: Bański, P., Barbaresi, A., Biber, H., Breiteneder, E., Clematide, S., Kupietz, M., Lüngen, H., Iliadi, C. (eds.) 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7). pp. 9 – 16. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, Leibniz-Institut für Deutsche Sprache, Mannheim (2019). https://doi.org/10.14618/ids-pub-9021, `http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215`

24. Passos, A., Kumar, V., McCallum, A.: Lexicon infused phrase embeddings for named entity resolution. In: Proc. of the Eighteenth Conference on Computational Natural Language Learning. pp. 78–86. Ann Arbor, Michigan (Jun 2014)
25. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics. pp. 2227–2237. New Orleans, USA (Jun 2018)
26. Rahimi, A., Li, Y., Cohn, T.: Massively multilingual transfer for NER. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 151–164. Association for Computational Linguistics, Florence, Italy (Jul 2019)
27. Raymond, C., Fayolle, J.: Reconnaissance robuste d'entités nommées sur de la parole transcrite automatiquement. In: TALN'10 (2010)
28. Rössler, M.: Adapting an NER-system for German to the biomedical domain. In: Proc. of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications. pp. 95–98. Geneva, Switzerland (2004)
29. Shen, D., Zhang, J., Su, J., Zhou, G., Tan, C.L.: Multi-criteria-based active learning for named entity recognition. In: Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics. pp. 589–596. Barcelona, Spain (2004)
30. Stanislawek, T., Wróblewska, A., Wójcicka, A., Ziembicki, D., Biecek, P.: Named entity recognition - is there a glass ceiling? In: Proc. of the 23rd Conference on Computational Natural Language Learning. pp. 624–633. Hong Kong (2019)
31. Straková, J., Straka, M., Hajic, J.: Neural architectures for nested NER through linearization. In: Proc. of the 57th Conference of the Association for Computational Linguistics, ACL, Florence, Italy. pp. 5326–5331 (2019)
32. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proc. of the Seventh Conference on Natural Language Learning. p. 142–147. CONLL '03, USA (2003)
33. Yangarber, R.: Counter-training in discovery of semantic patterns. In: Proc. of the 41st Annual Meeting of the Association for Computational Linguistics. pp. 343–350. Association for Computational Linguistics, Sapporo, Japan (Jul 2003)
34. Yangarber, R., Lin, W., Grishman, R.: Unsupervised learning of generalized names. In: In Proc. of the International Conference on Computational Linguistics (ICCL). pp. 1–7 (2002)