Choosing the Right Tool for You: Informed Evaluation of Text Analysis Tools

Angel Daza
Netherlands eScience Center
j.daza@esciencecenter.nl

Antske Fokkens
Vrije Universiteit Amsterdam
antske.fokkens@vu.nl

Abstract

Natural Language Processing (NLP) research showcases many promising tools and methods for text analysis. Researchers from diverse fields who want to use NLP for their research are confronted with a wide availability of ready-to-use models that claim excellent performance on standard benchmarks. Consequently, choosing an appropriate tool has become a task on its own. Our goal is to exemplify a methodology that stimulates critical evaluation and detailed analysis of automatic outputs of NLP tools. Particularly, we analyze the case of choosing the best Named Entity Recognition (NER) tool for a corpus of Dutch biographies. Our use case is an example of how to make informed decisions by considering different aspects of custom datasets at the instance and aggregated levels, improving the outcomes of the original research question.

1 Introduction

Recent developments in NLP have seen great progress towards one of CLARIN's primary goals: providing (relatively) easy-to-use language technology. This has led to an increase in the use of these technologies in various domains, such as Digital Humanities (DH) (Colavizza et al., 2021; Ehrmann et al., 2023; Schweter et al., 2022). However, the same rapid advancement of NLP has left the area with a weak spot regarding detailed and careful evaluation. Standard benchmarks and metrics often do not provide insight at the right level of detail for users to establish which tool works best for their specific use case, or whether a tool is appropriate for their methodological set-up at all (Fokkens et al., 2014). In our view, choosing an appropriate tool has become a task on its own, and supporting users in this task is an essential part of CLARIN's mission of making tools available to researchers. Therefore, in this paper we exemplify a methodology that stimulates critical evaluation and detailed analysis of automatic outputs of NLP tools, which can help external researchers select the right tool for their specific needs. Specifically, we propose a visually aided assessment of the strengths and weaknesses of using different models for a specific use case.

Although the methods proposed here can be applied directly to any span-based NLP tasks, for clarity, we describe a specific Digital Humanities application of a classic NLP task. Specifically, we assume the perspective of a historian who wants to perform NER to build networks of people (`PER`), organizations (`ORG`), and places (`LOC`) from digitized biographies. To achieve this, they must first select an NER model for their corpus, a dataset of thousands of biographies written in Dutch between the 18th and the 21st centuries. We propose combining a *distant evaluation* (global metrics) with a *close evaluation* of NER outputs (instance-level inspection) to make more informed choices when selecting a tool and to gain insight into how such tools will behave when we apply them to our data. This way, the user can ensure they behave as desired on the instances that matter the most for answering the original research question.

In the remainder of the paper, we start by briefly describing related work on NER classifiers and evaluation (Section 2), then we describe the dataset that we use to illustrate why careful evaluation is needed when picking a tool for a custom use case (Section 3.1). Next, we list four different state-of-the-art NER systems for Dutch (ready to be used out of the box) that we apply to our data to compare their

performance (Section 3.2). In Section 4 we describe metrics that allow us to visually explore each of the models' predictions on specific documents, spotting edge cases where major disagreement exists. This analysis can be performed even if we do not have gold data available, which is often the case for non-NLP practitioners. Section 5 then dives into a scenario where we do have a human-labeled dataset; however, in this case, too, different ways of evaluating (on the same dataset) can lead to different outcomes, showing the need for grounding the model selection in the research goals that have been established before the start of the technical implementation. Finally, Section 6 discusses our main findings, and Section 7 shows some of the limitations of the approach that we took here.

2 Related Work

The techniques for identifying named entities in texts have evolved with the general state of NLP and have become less interpretable with time. Initially, gazetteers were used to identify entity occurrences in a text directly. Later, statistical learning and Machine Learning classifiers were used to identify sequences of labels as a token classification task with some loose notion of what instances each entity category should comprehend (Nadeau & Sekine, 2007). After that, neural networks offered the option of including sentence context to predict a label for each token (Lample et al., 2016), using LSTMs with a classifier on top (Panchendrarajan & Amaresan, 2018) or using wider contexts by fine-tuning pre-trained language models such as BERT (Devlin et al., 2019) to perform structured prediction (Akbik, Bergmann, & Vollgraf, 2019; Yamada et al., 2020), this includes the classifiers used by Stanza, Flair and XLM-R presented in this paper (for more examples see Yadav and Bethard, 2018). Lastly, with the advent of LLMs, NER has been re-framed to use prompt engineering where a model such as GPT-x could be queried to generate as a response the series of entities found in a text, such as Ashok and Lipton, 2023 and Wang et al., 2023. In this paper, we also test GPT3.5's capabilities for spotting entities and trying to provide the spans where they were initially found. We argue that the fact that NER is being constantly re-framed to work with different techniques, such as prompt engineering, calls for a closer look at how evaluation is done, since comparison across models keeps getting more difficult given the variety of conceivable experimental setups. This constant evolution of LLMs related to NER can be seen in even more recent publications such as (Jiang et al., 2024; Kim et al., 2024; Picco et al., 2023; Tong et al., 2025).

Recently, more attention has been paid to expanding NER into other specific domains (De Los Reyes et al., 2021; Li et al., 2022). This includes a wide variety of annotated datasets to train task-specific classifiers (Arnoult et al., 2021), pre-trained language models with historic corpora (Manjavacas Arevalo & Fonteyn, 2022; Schweter et al., 2022) and general techniques to identify entities beyond the standard core labels and resources (Luthra et al., 2023; Tedeschi & Navigli, 2022).

There is also important work on evaluation for NER beyond just applying scores. For example, Ushio and Camacho-Collados (2021) show a tool for directly comparing different flavors of transformer models fine-tuned in several well-known benchmark datasets. Closer to our approach, Fu et al. (2020) present a tool that shows a behavioral overview of model outputs by bucketing entities according to their attributes. Their tool generates an HTML report that helps users interpret at the corpus level where the NER models fail the most. In contrast, our approach focuses on instance-level inspection of errors, where a researcher can explore errors based on edge cases or instances of particular interest.

3 Methodology

3.1 Case Study: NER for Dutch Biographies

The Biography Portal of the Netherlands[1] (BPN) is an online collection containing several biographical dictionaries written in the country through the years. It includes 25 different existing collections with biographical information on the inhabitants of the Netherlands, with more than 75,000 biographical entries. We are interested in applying NER to the BPN texts, which comprise biographies written between the 18th and 21st centuries; thus, they can be partly seen as a type of historical text (Romary, 2014). This domain poses its own set of challenges, such as: Language variety and dynamic changes through the cen-

[1] http://www.biografischportaal.nl

turies, a mixture of record typologies (each collection contains a specific style and idiosyncrasies when describing people), significant divergence in biography length, a prominent presence of abbreviations and rare entities that do not necessarily exist nowadays, to mention a few.

We created a human-labeled dataset containing a subset of 346 biographies of various lengths (some biographies have only dozens of tokens, and others have thousands). This subset was generated by stratified sampling to keep the original dataset's source distribution, ensuring we have examples from each collection. Annotators were asked to follow the guidelines for labeling three core entities: Persons (`PER`), Locations (`LOC`), and Organizations (`ORG`), as well as Dates (`TIME`), works of art (`WOA`), and miscellaneous (`MISC`). For this paper, we will focus only on the three core entities. Of the 346 biographies, 50 were triply annotated, for which we obtained an inter-annotator agreement of 78.3 Krippendorff's Alpha,[2] had a round of discussions, and proceeded to annotate the rest with a partial overlap to ensure agreement remained high. We also asked annotators to manually correct tokenization errors and re-split or merge sentences when necessary to end up with a clean pre-tokenized dataset in conll format, as is customary for standard NER evaluation. We describe the annotated subset in Table 1, where we also include the mean, median, standard deviation and maximum of entities annotated in the documents to showcase the diversity of the documents.

Category	Count	Mean	Median	Max	StdDev
PER	5,743	17	10	92	18.7
LOC	3,879	11	8	77	12.0
ORG	2,196	6	2	58	9.8
ALL	11,818	34	21	164	35.8
Tokens	189,507	548	245	3,126	613.8
Sents	8,210	23	21	210	13.5
Docs	346	-	-	-	-

Table 1: General statistics of our manually annotated corpus. We include the average, maximum, and median of occurrences per document to illustrate the heterogeneity of the dataset at the document level.

3.2 NER Models

We consider four candidate models that deliver (at least) the three desired NER labels, `PER`, `LOC`, and `ORG`. We focus only on those three because their definition is less controversial than other entity categories, yet disagreement still emerges across model (Nadeau & Sekine, 2007). Three of the four models have a similar architecture with only minor variants on the top layers of the classifiers, which could make us think there will not be much difference in the final results. Importantly, all models are readily available to use out-of-the-box, which is very attractive for a non-NLP researcher, primarily if the models are published with a high F1 score (around 90 points) associated with them. Commonly, the reported scores are in one of the most famous benchmarks for NER, in our case, the Dutch portion of conll-02 Shared Task (Tjong Kim Sang, 2002). The systems we will benchmark here are:

- **Flair NLP:** This open-source NLP framework (Akbik, Bergmann, Blythe, et al., 2019) contains very strong models for the NER task. The latest model is based on FLERT (Schweter & Akbik, 2020), an optimized method for NER in different languages, including Dutch. The basic architecture is the cross-lingual version of RoBERTa (Y. Liu et al., 2019), with a CRF layer on top to improve the structured prediction. This model was fine-tuned to account for document-level features using the Conll-02 dataset and reports a global F1 score of 94.5 on the Dutch test set.

- **Stanza:** The stanza NER model for Dutch is based on a concatenation of BERT (Devlin et al., 2019) pre-trained vectors and a character-based bidirectional LSTM with a CRF layer on top for the structured prediction. The corpus used for training the Dutch model is from Nothman et al. (2013).

[2] We use the NLTK implementation with binary distance to compute the agreement of labels.

	B	W	Y	Z	AA	AB	AC	AH	AI	AJ	AK
1		PERSON			LOCATION			MODEL STANDARD DEVIATION			
2	name	freq_flair	freq_stanza	freq_xlmr_ner	freq_flair	freq_stanza	freq_xlmr_ner	loc_stdev	org_stdev	per_stdev	avg_stdev
28	anne zernike	82	82	75	50	42	42	4.6188	1.5275	4.0415	3.4
29	hermina amersfoordt	86	76	70	37	38	40	1.5275	3.5119	8.0829	4.4
30	pieter wiedijk	69	51	48	6	8	6	1.1547	2.0817	11.3578	4.9
31	jan kops	28	25	28	22	20	20	1.1547	0.5774	1.7321	1.2
32	clara engelen	35	32	33	35	38	33	2.5166	1.5275	1.5275	1.9
33	roelf hagoort	22	16	19	22	21	22	0.5774	5.1316	3.0000	2.9
34	johannes wilhelmus boerbooms	48	32	52	51	47	47	2.3094	2.5166	10.5830	5.1
35	michael faraday	68	63	63	26	23	23	1.7321	1.5275	2.8868	2.0
36	charlotte sophie von aldenburg	116	129	117	78	52	68	13.1149	2.5166	7.2342	7.6
37	helena theodora rietberg	48	45	42	30	29	28	1.0000	3.6056	3.0000	2.5
38	heymen van 't einde	35	35	35	8	8	6	1.1547	1.7321	0.0000	1.0
39	aleida daendels	85	89	85	77	71	61	8.0829	1.1547	2.3094	3.8
40	weduwe van wouw	55	53	49	19	24	22	2.5166	2.3094	3.0551	2.6
41	berta vorkink	25	21	22	22	25	19	3.0000	0.5774	2.0817	1.9
42	johannes brommert	73	59	66	35	39	34	2.6458	4.0415	7.0000	4.6

Figure 1: We display the table of metrics in a spreadsheet. With conditional formatting, we can display it as a heatmap, where the rows with more discrepancy stand out because of the contrast of colors. This allows us to spot right away which documents show key differences in model behavior.

They report a macro F1 score of 89 in conll-02 and 94.8 on the WikiNER test (Ghaddar & Langlais, 2017).

- **Fine-tuned XLM-R:** this is a model[3] published in the HuggingFace repository and is described as a Named Entity Recognition model for ten high-resourced languages (Arabic, German, English, Spanish, French, Italian, Latvian, Dutch, Portuguese and Chinese) based on the Cross-lingual RoBERTa base model (Conneau et al., 2020). It has been fine-tuned with a neural linear layer on top that has been fine-tuned to recognize `LOC`, `ORG`, and `PER`. Given how easy it is to run models published on HuggingFace and that the model card claims to have been trained in the most important NER resources, it is feasible that someone will base their research on the information published there.

- **gpt3.5-turbo:** Finally, given the recent popularity of LLMs with the claims that they can provide very strong performance in almost any task (Min et al., 2023), we also explore the alternative of prompting gpt3.5 (Brown et al., 2020) for obtaining NER labels. Since our focus is not on prompt engineering (P. Liu et al., 2023), we measure the performance as a zero-shot setting with an intuitive prompt and visualize how it compares to applying NER-specific models on our dataset. Specifically, we provide the following prompt: *Identify and Label (PERSON, ORGANIZATION, TIME, LOCATION, ARTWORK, MISC) all of the Named Entities in the following text. Return also the character spans in which these entities appear. Return the results in TSV Format with Columns: [Entity, Label, Span Start, Span End]. Text: <ANSWER>*

4 Inspection of Model Behavior

We can spot interesting behavior in specific instances by comparing raw model outputs in parallel. An instance can be a sentence, paragraph, or document, depending on the required granularity. By looking closely at relevant instances, we can investigate which errors or tagging biases occur with the different models and make decisions accordingly. It would be impossible to inspect every single instance closely. Therefore, we propose to use the predictions of all models to compute straightforward metrics as a proxy for spotting edge cases, that is, to highlight the documents that will be the most useful to examine closely for understanding model behavior and anticipating whether that behavior is optimal for the intended use case. This is possible even when there is no human-annotated data available at all.

[3] https://huggingface.co/Davlan/xlm-roberta-base-ner-hrl

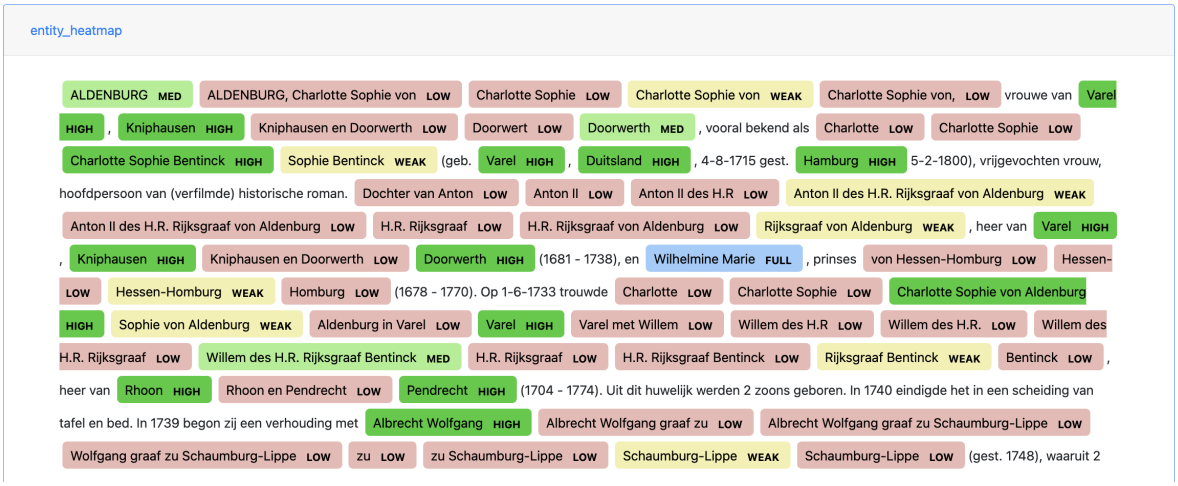

Figure 2: We can visualize which entities are preferred by most models and which ones cause more disagreement among them.

We also use a set of visualizations built on top of everyday tools to inspect the instances and make the process more user-friendly. We specifically use spreadsheets to show what we call *divergence matrices* (see Section 4.1 and Figure 1). Additionally, we created a small Flask web app[4] that integrates visualizations from `displaCy`[5] to see the spans in the text; and Google Charts[6] to show aggregated statistics.

4.1 Divergence Matrix

A divergence matrix Z is defined as a collection of P instances $\{p_0, p_1, \ldots, p_P\}$ (for our use case hee we chose a document as the instance unit), a set of N systems (models) denoted as $\{s_0, s_1, \ldots, s_N\}$, and M evaluation metrics $\{m_1, m_2, \ldots, m_M\}$, therefore Z has dimensions $P \times (S * M)$. In this context, a row represents an instance p_i, and each column $Z_{i,j,k}$ signifies the performance score of system s_j when evaluated using metric m_k on instance p_i with $\quad i = 0, 1, \ldots, P;\ j = 0, 1, \ldots, N;\ k = 0, 1, \ldots, M$. Consequently, the matrix encapsulates the evaluated models' performance across all instances and metrics (See Figure 1). Metrics can be defined as required per use case, here we show 3 example metrics:

Entity Frequency: get the raw counts of Named Entities found in each document according to each model. These are counted per category (`PER`, `LOC`, `ORG`), and also globally (the sum of all categories).

Entity Density: In this case, we divide the entity frequency by the number of tokens in the instance to get a weighted metric and be less biased towards extensive documents.

Entity Divergence: We define this metric by considering an array of frequencies from $model_0$ to $model_N$, where each element is the entity frequency predicted by a $model_n$. Then, we compute the divergence as the standard deviation of this array. The reasoning is that models will obtain the same amount of labels for easy instances (all models will agree), whereas instances with complex cases or cases with noise will have a high divergence.

To avoid falling again into the trap of looking at these metrics only globally, we can display the divergence matrix as a heatmap, where higher numbers ger darker colors and lower numbers approach to no-color. This way, it is visually easy to spot problematic instances (where color intensity changes dramatically across columns). For example, in Figure 1, the matrix is displayed in a spreadsheet, which has a low threshold for users. In this example, the instance that stands out the most is the biography of

[4] Available here: https://github.com/angel-daza/bios-dutch
[5] https://demos.explosion.ai/displacy-ent
[6] https://developers.google.com/chart/

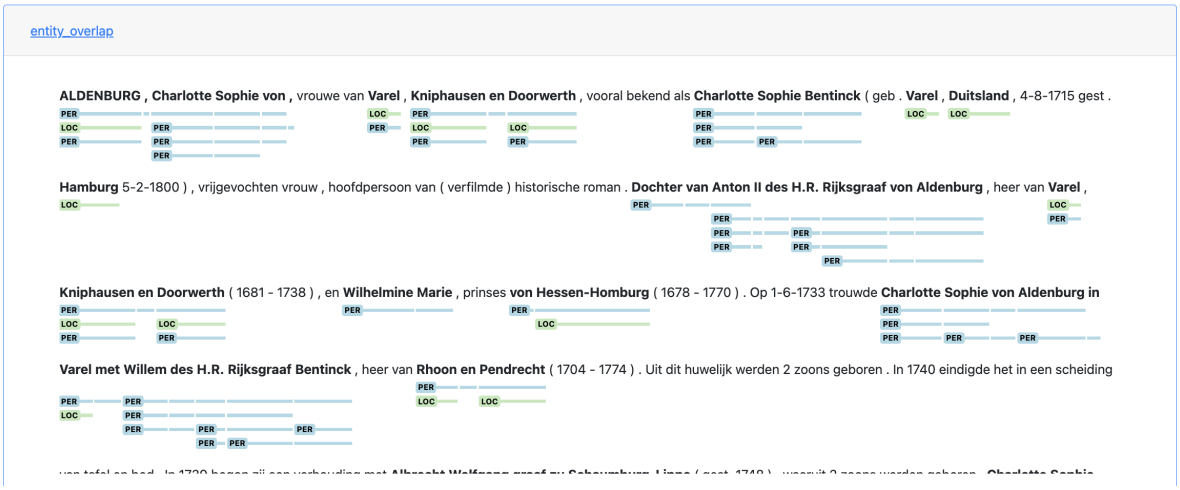

Figure 3: We can also visualize the predicted labeled-span divergences of all models in parallel. Some errors are systematic and can be later counteracted by post-processing.

Charlotte Sophie von Aldenburg, where the matrix shows that there is substantial disagreement between models in `LOC` and `PER` assignments.

4.2 Entity Agreement Heatmap

If we dig into an instance (document) with high divergence and visualize the label overlap of all models in it, we can spot where in the text they converge and where they disagree. Figure 2 illustrates the example of Charlotte Sophie von Aldenberg. We classify text spans (entity candidates) into five buckets according to the certainty of correctness using models in a voting mechanism: the entities that have the votes of N models are highlighted in blue (`FULL`), in dark green are the entities that $N-1$ models identified (`HIGH`), in light green $N-2$ (`MED`), in yellow $N-3$ (`WEAK`) and the rest of entities are labeled in red (`LOW`) meaning that the certainty of them being the right span is low since most models voted differently. For example, the last name `von Aldenburg` is obviously derived from the fact that Charlotte came from the nobility related to the `Aldenburg` location, hence the disagreement.

4.3 Parallel Span Comparison

Another example where we can see this `LOC-PER` confusion is in the entity overlap (Figure 3). Take the span `Varel, Kniphausen en Doorenwerth`, which is basically an enumeration of places that Charlotte is a *lady of*. However, some models interpreted that this segment could mean she is the *wife of* and thus classifies the subsequent Entities as people (in this case, stanza was the model that committed this confusion consistently, whereas flair got them right). Because none of the NER systems are trained to deal with these cases, we see a higher disagreement in how they are labeled and, hence, the high divergence across `PER` and `LOC` in this document. If we should pick a tool based on this aspect, we could deduce that the flair model behaves more closely to what we would expect with this kind of case.

These simple visualizations already give us a deeper insight into how models behave with our specific examples of interest without the need to spend time generating labeled data. Nevertheless, if the model is to be used at scale, relying on a handful of examples is not sufficient. The need for evaluating the model outputs vs. human annotations is still the desirable scenario. In the next section, we explore different ways of doing this.

	PER			LOC			ORG			MICRO			MACRO		
SYSTEM	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
stanza_iob	77.0	85.0	81.0	82	85.0	83.0	58.0	52.0	55.0	76.0	79.0	77.0	72.0	74.0	73.0
flair_iob	81.0	89.0	85.0	86.0	92.0	89.0	69.0	65.0	67.0	81.0	85.0	83.0	79.0	82.0	**80.0**
stanza_span	61.1	89.7	72.7	64.0	93.7	76.0	42.9	75.3	54.7	59.2	89.1	71.1	56.0	86.2	**67.8**
flair_span	53.4	83.9	65.3	66.9	93.4	78.0	47.5	78.9	59.3	56.7	86.4	68.4	55.9	85.4	67.5

Table 2: Comparison of the same models when feeding pre-tokenized data vs. feeding raw text

5 Evaluation Modes

When you intend to use automatic outputs to support your research, the ideal case is to have more than one human annotator and create a labeled subset with information on inter-annotator agreement. This labeled set can be used to compute evaluation scores that provide information on the quality of the automatic tools when applied to your data. NLP researchers tend to report results of classifiers with mainly three metrics: Precision, Recall, and F1 Score. These metrics were inherited from the field of information retrieval and adopted for other tasks (Powers, 2011). In the next section, we show that even in this standard scenario, it is essential to be careful with how we evaluate the performance scores. It remains important to keep the ultimate research goal in mind when determining which model is the *best model* for the scenario at hand. Different models will rank as *the best* according to different criteria. Often, scores are taken for granted, and differences that appear to be subtle when calculating such scores could be hiding behavior that is particularly harmful to a specific use case (Fokkens et al., 2014).

5.1 Tokenization Matters

Starting with the CoNLL-02 (Tjong Kim Sang, 2002) and CoNLL-03 shared tasks (Tjong Kim Sang & De Meulder, 2003), NER has been approached in NLP as a sequence labeling task, where each token is assigned one label in the IOB format (or related) (Ramshaw & Marcus, 1995). Some of the best-known benchmarks for NER have also taken this approach (Balasuriya et al., 2009; Tedeschi & Navigli, 2022). The conll format already presents a pre-tokenized text, which is also split into sentences. The reasoning behind this is to focus only on evaluating NER performance. The problem is that this approach can give the false impression that tokenization and sentence splitting are trivial tasks when, in fact, differences in scores are expected when applying models to untokenized text (Daza et al., 2022). This is particularly problematic, because the most common scenario for non-NLP users is to apply an out-of-the-box tool to raw text.

We run two parallel experiments: we evaluate the NER task in its default format (feeding the models with the clean and tokenized sentences) vs. the performance of the same models when processing raw text into tokenized text with Named Entities. We call the first evaluation mode *IOB match*. We performed this evaluation using the widely used Python module `seqeval`.[7] In the second approach, which we call *Span match*, we evaluate identified character spans identified by the tools in the raw text. We used character span matches to focus on the Named Entities to avoid noise in the results while comparing sentences of different sizes or tokenization divergences (each model produces different tokens and sentences). In Table 2, we see that the evaluation of raw text (*Span match*) gives significantly lower scores. Secondly, we highlight the fact that in the *IOB match* mode, the best model is flair, whereas, in the *Span match* mode, stanza is the best-performing model.

We consider the *Span match* mode to be closer to what external users of tools need; therefore, the following evaluations will be using the *Span match* mode. Within this approach, we can distinguish two submodalities: to consider a true positive only those entities whose labeling has an exact match with respect to the gold labels, or to be more lenient and consider as a true positive an entity whose span partially overlaps with the gold label (and has the same label, of course). A third evaluation modality emerges with

[7] https://github.com/chakki-works/seqeval

	PER			LOC			ORG			MICRO			MACRO		
SYSTEM	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
stanza	61.1	89.7	72.7	64.0	93.7	76.0	42.9	75.3	54.7	59.2	89.1	71.1	56.0	86.2	67.8
flair	53.4	83.9	65.3	66.9	93.4	78.0	47.5	78.9	59.3	56.7	86.4	68.4	55.9	85.4	67.5
xlmr_ner	59.8	86.5	70.7	65.0	90.2	75.6	48.8	75.7	59.3	59.8	86.2	70.7	57.9	84.1	**68.5**
gpt-3.5	0.3	8.2	0.6	0.6	32.2	1.1	0.1	11.1	0.1	0.3	11.9	0.6	0.3	17.2	0.6

Table 3: Strict evaluation NER. We measure how many predicted spans completely match the gold spans, in terms of span start, end and assigned label.

	PER			LOC			ORG			MICRO			MACRO		
SYSTEM	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
stanza	64.3	92.6	75.9	65.6	88.7	75.4	50.6	70.5	58.9	62.4	87.6	72.9	60.1	83.9	70.1
flair	59.7	97.4	74.0	68.6	95.9	80.0	54.0	80.8	64.8	61.4	94.1	74.3	60.8	91.4	**72.9**
xlmr_ner	63.6	90.9	74.8	67.4	88.2	76.4	56.0	71.0	62.6	63.6	86.6	73.3	62.3	83.4	71.3
gpt-3.5	3.8	6.0	4.6	1.7	1.5	1.6	0.5	0.4	0.5	2.8	3.5	3.1	2.0	2.7	2.2

Table 4: Partial Evaluation NER

the advent of LLMs. Some papers started to ignore spans when reporting scores by evaluating results as what we call here *bag of entities* (Ashok & Lipton, 2023), where the set of predicted entities is compared versus the set of gold entities in the text.[8] All evaluation setups could be considered valid in their own contexts; however, it is important to avoid comparing numbers obtained with different setups as they are not measuring the same signals. Next, we describe each evaluation scenario and show a table of scores when applying each setup. Note that the systems were trained to detect labeled spans (except GPT-3.5); we are only changing the way we evaluate the models, we are not re-training nor changing anything to the models. The observed performance differences thus will be solely because of the assumptions behind each way of evaluating.

5.2 Full Match

In this setting, an entity is considered correct only if it completely matches the gold span and the label. In Table 3, we can see several dimensions when comparing different models. If one only analyzes system-level scores, one would conclude that xlmr_ner is the best-performing system on the test set. However, when we inspect the performance more closely, we gain some insight into what each model is doing: When evaluating with the exact match setup, GPT 3.5's performance is definitely poor, and the reason is that with the prompt that was given, even though it gets several entities correctly, it generates pseudo-random spans. For the traditional systems, we see that Stanza performs much better for detecting `PER` than xlmr_ner and flair, which is very relevant if, for example, we are primarily interested in recognizing people. `LOC` is the easiest category for all models, suggesting fewer unknown entities were found (compared to the entities seen during training) in the biographical dataset; this is possibly because most `LOC` entities are nowadays still popular city and country names that have not changed in the last three centuries. Flair appears to be much better if we are interested in detecting `ORG`s. This category is also the weakest, perhaps because the names of organizations are more sparse, and the definition of an organization has changed through the years compared to the other two categories.

5.3 Partial Match

This mode behaves quite similarly to the full match. The main difference is that the criteria for considering a True Positive is looser. The strict match policy is too harsh for cases where the classifier almost got the whole entity, certainly with the correct label, but missed a couple of tokens compared to the gold. For example, if a model labeled `Charlotte Sophie` instead of `Charlotte Sophie von`

[8]Note that these F1 scores are compared directly to the IOB scores from previous papers, without highlighting the fact that LLMs are not predicting spans in text.

	PER			LOC			ORG			MICRO			MACRO		
SYSTEM	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
stanza	55.1	79.8	65.2	65.9	80.9	72.6	43.2	48.7	45.8	56.2	73.3	63.6	54.7	69.8	61.2
flair	45.4	72.4	55.8	70.5	86.9	77.8	47.0	57.5	51.7	53.1	74.0	61.8	54.3	72.3	61.8
xlmr_ner	54.9	77.6	64.3	66.2	80.6	72.7	48.4	51.9	50.1	57.3	73.0	64.2	56.5	70.0	**62.4**
gpt-3.5	68.9	49.4	57.5	83.5	58.3	68.6	41.5	30.5	35.1	67.6	48.3	56.3	64.6	46.1	53.8

Table 5: Evaluation of NER as bag-of-entities. This setup only considers unique entities per document and does not consider spans where they appear, nor repetitions of the same entity.

`Aldenburg` as `PER`, it is still usable information. In this example, the exact match will penalize twice the same mistake: the predicted span is a false positive since it does not exist in gold, and also, the gold span was not predicted, which counts as a false negative. This is not a recommended way of evaluating, but it is used for some cases (e.g. Luthra et al., 2023), so we keep it here for comparison purposes. Table 4 shows the results when evaluating the models on the same dataset but with the partial-match policy. As expected, numbers go up compared to the strict match. In this case, there is also a swap in the model ranking: the best model according to strict match macro F1 was xlmr, and according to partial match macro F1, the best model overall would be flair.

5.4 Bag of Entities

We could argue that in some oversimplified cases, for example, if the last goal is to build a network, we only need to encounter the right entities in the text once and do not care about their place in the text. In this case, it would not matter if *Amsterdam* is mentioned 20 times in a biography, it only matters that we recognize it once as a LOC to draw an edge between this person and this location. Nevertheless, note that this looses track of the instances where *Amsterdam* appears in the text as an ORG (for example, a metonymy) or `PER` (for example if *Amsterdam* was a last name). Assuming you are willing to take such risks, by evaluating NER as a bag of entities (the set of all unique `SurfaceText_Label` mentions in an instance), we can drop the requirement of detecting the spans, drawing more attention to the importance of recovering key entities at the expense of losing their original context. We show in Table 5 the precision, recall, and F1 when evaluating this way. At first glance, the numbers go down to the low 60s. Based on the Macro F1, we would pick again xlmr_ner as the best-performing system according to the test, but by a much narrower margin. Notably, without the Span requirement, GPT 3.5 outputs are now useful (which might make this evaluation setup tempting), giving a closer performance related to the NER-specific systems. The same trends stand at the Entity type level: `LOC` is the easiest category, followed by `PER` and ORG. If we look at `PER` metrics: Precision, Recall, and F1 on StrictMatch (Table 3 vs BoE in Table 5), we see a constant decrease in performance (Stanza F1 drops from 72 to 65; Flair F1 drops 63 to 56). This suggests that, since we are only considering Bag of Entities, the globally frequent entity names, which are very often predicted correctly, have less weight *inflating* the performance of the models. This is why the scores from BoE are lower than the Strict Match in general.

5.5 Precision vs Recall

Another critical aspect to determine is whether we would rather have a system that is biased toward False Positives (FP) or towards False Negatives (FN). Systems with a higher proportion of FPs translate into low precision, meaning that more noisy cases will appear in our categories, and systems with a high ratio of FNs, translate into low recall, indicating that many useful cases will be ignored. We can visualize the FPs and FNs of each instance separately so a close inspection can show what kind of mistakes are happening. Importantly, because the NER task is span identification and span labeling, there are different causes of error: i) The span was correctly identified, but the label is wrong, ii) The span is wrongly identified, but the label is correct, iii) The span was wrongly identified and the label is wrong, iv) The labeler did not tag the entity at all. By packing this in a single P, R, or F1 score, we lose the ability

SYSTEM	PER	LOC	ORG	TOTAL
stanza	55.10	65.86	43.22	56.22
flair	45.42	70.47	47.01	53.14
xlmr_ner	54.93	66.21	**48.44**	57.33
gpt-3.5	**68.85**	**83.51**	41.46	**67.55**

Table 6: Table of **Precisions** for each NER label, when evaluated as Bag of Entities

SYSTEM	PER	LOC	ORG	TOTAL
stanza	**79.79**	80.87	48.65	73.28
flair	72.39	**86.93**	**57.46**	**73.95**
xlmr_ner	77.62	80.64	51.87	72.95
gpt-3.5	49.43	58.27	30.45	48.28

Table 7: Table of **Recalls** for each NER label, when evaluated as Bag of Entities

to analyze the nature of the errors. However, visualizing them at the instance level can provide more explanations of the models' behavior.

If we evaluate the systems based on their precision (Table 6), the highest precision is for GPT-3.5. Notably, if we are only interested in recognizing important organizations (and do not care about recall), then it seems like xlmr is the best-performing system in the test data. On the other hand, if what we need is to maximize recall, results in Table 7 show that flair is the highest-performing model. We can also see that GPT3.5's recall is dramatically worse compared to the other systems, suggesting that (at least with the current prompt) GPT will systematically ignore several entities, even though it is precise when generating them (but, once again, this only happens when ignoring spans in the text).

5.6 Micro vs Macro(s)

Evaluating NLP classifiers with Precision, Recall, and F1 is customary. However, it is sometimes surprisingly hard to identify whether the reported metrics are done at the macro level or micro level. Normally, macro metrics are preferred because they average category-level scores. This alleviates the bias that exists across classes (although Opitz and Burst (2019) warn that sometimes different formulas are also used when computing macro scores, resulting in different numbers). Looking at Table 3, it is not surprising that stanza is the best model according to the micro F1, but xlmr is the best model according to the macro F1. The reason for this change in ranking occurs because there is a bigger proportion of `PER` entities in the corpus, a category for which stanza is much better at labeling, obtaining a higher micro; however, the macro counteracts this bias and gives a fairer score, showing the system that performs the best across categories. Hence, we reiterate that an underperforming system could be picked if one does not carefully make the difference between these two.

6 Findings and Discussion

This section provides an overview and discussion of our most important findings. As mentioned in Section 4, we built the visualizations on top of ready-to-use libraries such as Microsoft Excel, DisplaCy and Google charts. These visualizations need at most a couple of lines of basic code to run. We aim for a low technical threshold and the technical skills required to use these visualizations are comparable to those needed for running the tools. The visualizations we propose can help identify where the source of divergences is and remind us that we should check for the following aspects when evaluating. Depending on which aspects are more relevant to a specific use case, different systems can come up as the best solution for our specific use case. The following subsections highlight our main findings.

Tokenization Matters NER has traditionally been approached as a sequence labeling task, where each token is assigned one label in the IOB format (or related) (Ramshaw & Marcus, 1995). Notably, the most

common scenario for non-NLP users is to apply an out-of-the-box tool to raw text. Our experiments show that evaluation on raw text gives significantly lower scores than the evaluation assuming tokenization as a given. We consider the *Span match* mode that starts from raw text to be closer to what external users of tools need. Similar gaps between results obtained in "clean" experimental settings for standard benchmarks and real world scenarios may occur for other NLP tasks as well. It is therefore worthwhile to find out what the experimental setup that led to reported results looked like exactly to get a better feeling of what may be expected in your own use case. These verifications provide additional information next to other known factors that may determine results such as differences in genre, domain or time period in which the data was created.

At the same time, not all errors may be equally problematic. It is therefore worthwhile to consider how the output of the tools will be used to see how to weigh different kinds of errors. The following paragraphs highlight relevant aspects in approaching this question.

Full Match vs Partial Match Only entities that match entirely the gold span label are considered correct in a full match setting. This strict match policy can be too harsh for cases where the classifier almost got the whole entity with the correct label but missed a couple of tokens compared to the gold (which can also be fixed with a simple post-processing rule). For example, if a label `Charlotte Sophie` instead of `Charlotte Sophie von Aldenburg` as `PER` in her own biography can easily be mapped to the correct person leading to a fully correct outcome for making networks, thus partial match evaluation could be enough for this case.

Bag of Entities In some cases, e.g. identifying relevant documents or creating networks based on loose connections, we only need to encounter the correct entity in the text once. It then does not matter if *Amsterdam* occurs many times in the document, we only need to recognize it once as a LOC to draw an edge between this person and *Amsterdam*. Here, evaluating without the need of validating spans can be enough.

Precision vs Recall A related aspect is whether high precision (not many false positives) or high recall (not many items missed, or few false negatives) is more important. Because the NER task is span identification and span labeling, there are different causes of error: i) The span was correctly identified, but the label is wrong, ii) The span is wrongly identified, but the label is correct, iii) The span was wrongly identified and the label is wrong, iv) The labeler did not tag at all the entity. If this information remains packed in a single P, R, or F1 score, we lose the ability to analyze the errors. Visualizing the FPs and FNs of individual documents separately can show what kind of mistakes are made so we can act accordingly to fix them.

7 Limitations

This paper compares four specific NER tools for Dutch that can be applied out of the box. We are aware that comparison among more tools could provide more decisive conclusions. The tools that we used as an aid for the instance-level evaluation in principle apply to other languages and text domains, given that the tool handles "labeled spans" regardless of the nature of the labels and the expected spans; however, this paper did not provide experiments on other tasks. We focused on NER only to make a more straightforward point that even an *easy task* can produce relevant disagreement across pre-trained models.

As is widely known, the landscape of Large Language Models is evolving incredibly fast, and several far more powerful models have appeared since this research started. We want to bring attention to the fact that evaluation techniques should be fair across model architectures and task definitions. More importantly, one should proceed with caution when reading about loosely defined experiments, where LLMs are given more leniency on their *inner methods* used to produce results, and an assessment for the specific use case at hand should always be done before assuming someone else's prompt is what will work best.

Finally, we also omitted the scenario where a custom model is fine-tuned for the specific corpus. The reason for this is to emulate the scenario for a Digital Humanities researcher who wishes to use already

available tools instead of creating their own models. These two options are not mutually exclusive, as an analysis such as the one shown in this paper can motivate the researcher to train their own models if no available model satisfies the goals of their task at hand.

8 Conclusion

In this paper, we call for a more detailed evaluation of NLP span classification tasks. We apply several out-of-the-box NER models to a corpus of Dutch Biographies and compare different options for evaluation. We aim to illustrate the importance of inspecting the output of models at various levels when investigating whether their output provides the information that is needed with sufficient reliability. We also aim to show that a higher F1 score in a benchmark does not necessarily mean that the model will also be the best choice for our specific use case. We highlight the importance of carefully looking at the outputs to know whether the selected models are working in the way we intend to, going beyond just trusting global metrics in the abstract. Finally, we shared the code where we performed all these analyses as to encourage researchers from various fields to use these methods for their own use cases.

References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 54–59.

Akbik, A., Bergmann, T., & Vollgraf, R. (2019). Pooled contextualized embeddings for named entity recognition. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 724–728.

Arnoult, S. I., Petram, L., & Vossen, P. (2021). Batavia asked for advice. pretrained language models for named entity recognition in historical texts. *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, 21–30. https://doi.org/10.18653/v1/2021.latechclfl-1.3

Ashok, D., & Lipton, Z. C. (2023). Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.

Balasuriya, D., Ringland, N., Nothman, J., Murphy, T., & Curran, J. R. (2009). Named entity recognition in Wikipedia. *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, 10–18. https://aclanthology.org/W09-3302

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Colavizza, G., Blanke, T., Jeurgens, C., & Noordegraaf, J. (2021). Archives and ai: An overview of current debates and future perspectives. *J. Comput. Cult. Herit.*, *15*(1).

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

Daza, A., Fokkens, A., & Erjavec, T. (2022). Dealing with abbreviations in the Slovenian biographical lexicon. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 8715–8720. https://doi.org/10.18653/v1/2022.emnlp-main.596

De Los Reyes, D., Barcelos, A., Vieira, R., & Manssour, I. (2021). Related named entities classification in the economic-financial context. *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, 8–15. https://aclanthology.org/2021.hackashop-1.2

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Ehrmann, M., Hamdi, A., Pontes, E. L., Romanello, M., & Doucet, A. (2023). Named entity recognition and classification in historical documents: A survey. *ACM Comput. Surv.*, *56*(2).

Fokkens, A., Ter Braake, S., Ockeloen, N., Vossen, P., Legêne, S., Schreiber, G., et al. (2014). Biographynet: Methodological issues when nlp supports historical research. *LREC*, 3728–3735.

Fu, J., Liu, P., & Neubig, G. (2020). Interpretable multi-dataset evaluation for named entity recognition. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6058–6069. https://doi.org/10.18653/v1/2020.emnlp-main.489

Ghaddar, A., & Langlais, P. (2017). WiNER: A Wikipedia annotated corpus for named entity recognition. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 413–422. https://aclanthology.org/I17-1042

Jiang, G., Luo, Z., Shi, Y., Wang, D., Liang, J., & Yang, D. (2024, May). ToNER: Type-oriented named entity recognition with generative language model. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)* (pp. 16251–16262). ELRA; ICCL. https://aclanthology.org/2024.lrec-main.1412/

Kim, H., Kim, J.-E., & Kim, H. (2024, November). Exploring nested named entity recognition with large language models: Methods, challenges, and insights. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 8653–8670). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.492

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260–270. https://doi.org/10.18653/v1/N16-1030

Li, Y., Nair, P., Pelrine, K., & Rabbany, R. (2022). Extracting person names from user generated text: Named-entity recognition for combating human trafficking. *Findings of the Association for Computational Linguistics: ACL 2022*, 2854–2868. https://doi.org/10.18653/v1/2022.findings-acl.225

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, *55*(9). https://doi.org/10.1145/3560815

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luthra, M., Todorov, K., Jeurgens, C., & Colavizza, G. (2023). Unsilencing colonial archives via automated entity recognition. *Journal of Documentation*, *80*(5), 1080–1105.

Manjavacas Arevalo, E., & Fonteyn, L. (2022). Non-parametric word sense disambiguation for historical languages. *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, 123–134. https://aclanthology.org/2022.nlp4dh-1.16

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, *56*(2). https://doi.org/10.1145/3605943

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*, 3–26. https://api.semanticscholar.org/CorpusID:8310135

Nothman, J., Ringland, N., Radford, W., Murphy, T., & Curran, J. R. (2013). Learning multilingual named entity recognition from wikipedia. *Artif. Intell.*, *194*, 151–175.

Opitz, J., & Burst, S. (2019). Macro F1 and macro F1. *CoRR*, *abs/1911.03347*. http://arxiv.org/abs/1911.03347

Panchendrarajan, R., & Amaresan, A. (2018). Bidirectional LSTM-CRF for named entity recognition. *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*. https://aclanthology.org/Y18-1061

Picco, G., Martinez Galindo, M., Purpura, A., Fuchs, L., Lopez, V., & Hoang, T. L. (2023, July). Zshot: An open-source framework for zero-shot named entity recognition and relation extraction. In D. Bollegala, R. Huang, & A. Ritter (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 3: System demonstrations)* (pp. 357–368). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-demo.34

Powers, D. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Ramshaw, L., & Marcus, M. (1995). Text chunking using transformation-based learning. *Third Workshop on Very Large Corpora*. https://aclanthology.org/W95-0107

Romary, L. (2014). Natural Language Processing for Historical Texts Michael Piotrowski (Leibniz Institute of European History) Morgan & Claypool (Synthesis Lectures on Human Language Technologies, edited by Graeme Hirst, volume 17), 2012, ix+157 pp; paperbound, ISBN 978-1608459469. *Computational Linguistics*, *40*(1), 231–233. https://doi.org/10.1162/COLI_r_00180

Schweter, S., & Akbik, A. (2020). FLERT: document-level features for named entity recognition. *CoRR*, *abs/2011.06993*. https://arxiv.org/abs/2011.06993

Schweter, S., März, L., Schmid, K., & Çano, E. (2022). Hmbert: Historical multilingual language models for named entity recognition. *Experimental IR Meets Multilinguality, Multimodality, and Interaction - Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2022)*.

Tedeschi, S., & Navigli, R. (2022). MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). *Findings of the Association for Computational Linguistics: NAACL 2022*, 801–812. https://doi.org/10.18653/v1/2022.findings-naacl.60

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. https://aclanthology.org/W02-2024

Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147. https://aclanthology.org/W03-0419

Tong, Z., Ding, Z., & Wei, W. (2025, January). EvoPrompt: Evolving prompts for enhanced zero-shot named entity recognition with large language models. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, & S. Schockaert (Eds.), *Proceedings of the 31st international conference on computational linguistics* (pp. 5136–5153). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.345/

Ushio, A., & Camacho-Collados, J. (2021). T-NER: An all-round python library for transformer-based named entity recognition. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 53–62. https://doi.org/10.18653/v1/2021.eacl-demos.7

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., & Wang, G. (2023). Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Yadav, V., & Bethard, S. (2018, August). A survey on recent advances in named entity recognition from deep learning models. In E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), *Proceedings of the 27th international conference on computational linguistics* (pp. 2145–2158). Association for Computational Linguistics. https://aclanthology.org/C18-1182/

Yamada, I., Asai, A., Shindo, H., Takeda, H., & Matsumoto, Y. (2020). LUKE: Deep contextualized entity representations with entity-aware self-attention. *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), 6442–6454. https://doi.org/10.18653/v1/2020.emnlp-main.523