Decomposing Unitization and Typing for Efficient and Consistent Span-Bound Concept Annotation

Anonymous ACL submission

Abstract

In specialized domains that require expert annotators and high inter-annotator agreement, high-quality datasets with span-bound semantic concept annotations remain expensive to develop. Substantial resources are typically spent on *unitizing*, the task of identifying precise span boundaries for entity mentions. Unitizing is a significant source of inter-annotator disagreement, a poor use of expensive domain expertise, and very time-consuming. We propose a lighter annotation procedure that concentrates manual efforts on typed position annotations, marking positions in the text that overlap with mentions of each entity type, abstracting away span boundary decisions. With as few as 100-200 example sentences, we train span boundary detection models to unitize typed position annotations. Through evaluation over three datasets: CRAFT (biomedical), GENIA (molecular biology), and POLIANNA (climate/energy policy text), we demonstrate that (1) annotating typed positions in the text instead of full concept annotation is a more efficient use of time in low-resource settings, and (2) model-inferred span boundaries result in higher agreement at both the annotator training and corpus annotation phases, without sacrificing utility.

1 Introduction

Semantic concept annotation refers to the extraction of span-bound mentions of concepts, which can broadly refer to concrete classes in ontologies, named entities, or abstract themes and characteristics. High quality concept annotations for specialized domains are expensive (Kim et al., 2003; Li et al., 2016; Krallinger et al., 2015). It took thousands of hours to annotate concepts for the CRAFT biomedical article dataset (Bada et al., 2012). For the POLIANNA policy design dataset, Sewerin et al. (2023) report over 600 hours of annotation.

To reduce the cost of concept annotation, LLMs have been proposed as a sufficient substitute for manual annotation, where models are prompted to extract typed entities (Katz et al., 2023). Large LLM-generated NER datasets such as PileNER (Zhou et al., 2023) and NuNER (Bogdanov et al., 2024) are widely used. This approach, however, often does not transfer to specialized domains where LLMs under-perform in few-shot settings (Golde et al., 2025; Volkanovska, 2025; Jimenez Gutierrez et al., 2022). High-performing model-based annotation for specialized domains with large label spaces still need hundreds of annotated examples (Zhou et al., 2023; Sainz et al., 2023).

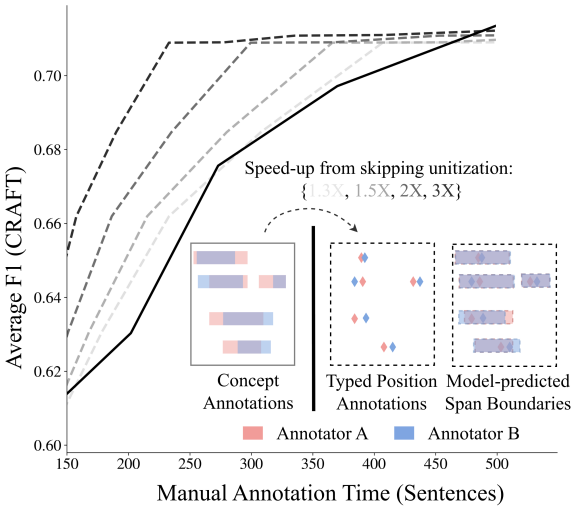

Figure 1: We propose an annotation procedure with manual human effort focused on identifying typed positions, augmented by model-detected span boundaries (right), as an alternative to full concept annotation (left). Decomposing the task in this way is a more efficient use of manual annotation time (dashed lines), achieving higher accuracy in less time compared to full annotations (solid line). We further find that model-inferred span boundaries result in higher consistency between pairs of annotators.

Traditionally, the high cost of developing a dataset is distributed between two phases: (1) Obtaining high consistency among annotators for a subset of documents, and (2) annotators independently annotating the remainder of the corpus

(Stubbs, 2012; Hovy et al., 2006). For concept annotation, manual efforts are spent on identifying mentions of entity types and identifying span boundaries for each mention or *unitizing* (Artstein and Poesio, 2008; Krippendorff, 2004). Unitization is a documented source of inter-annotator disagreement, bottlenecking the first phase (Reiss et al., 2020; Krallinger et al., 2015). Subjective span boundary decisions (i2b2, 2010; Kim et al., 2003) can be reduced through detailed specification of how to select boundaries (Bada et al., 2012; Krallinger et al., 2015), but this may represent additional overhead for annotators. In the second phase, unitizing is an expensive, time-consuming sub-task (Finin et al., 2010; Andrade et al., 2024; Fort et al., 2012). Further, adherence to complex rules about which case modifiers or supporting phrases should be included in a span is poor use of expensive domain expertise. While linguistic principles are critical for developing high-quality datasets, they are not necessarily well-aligned with domain expertise of annotators. At the same time, LLMs can follow linguistic rules (Zhang et al., 2024), suggesting that the unitization sub-task may be better suited to perform with a model.

In this work, we demonstrate that decomposing the concept annotation task into typed position identification and unitization results in more efficient use of manual annotation time. We develop an alternate annotation procedure, concentrating manual effort where it provides the most utility, annotating typed positions, rather than the complete annotation of type and span boundaries (Figure 1). We show that a relatively cheap model for span boundary detection, using at most 200 annotated sentences as training data, is sufficient to automate span boundary detection with adequate performance across most settings. The small amount of noise in inferred silver span boundaries is outweighed by the additional examples that we can afford to annotate with our proposed procedure.

Our evaluation focuses on expensive datasets that required domain expertise to annotate: Biomedical research articles in CRAFT (Bada et al., 2012), biology abstracts in GENIA (Ohta et al., 2002), energy and climate policies in POLIANNA (Sewerin et al., 2023). In simulated annotation over these datasets, we demonstrate that in settings with adequate span boundary detection performance, the additional data that we can afford to annotate with typed positions instead of full concepts results in stronger performing concept annotation models. For example, if we spend the time required for full concept annotation of 100 additional CRAFT sentences on annotating typed positions instead, we observe a 2.7 point performance improvement. We also measure consistency of our procedure at the annotator training phase and corpus annotation phase, which we are able to simulate using unreleased single-blind raw annotator files from annotator training sessions for CRAFT and public double-blind corpus annotation data for POLIANNA. At each phase of annotation, we find that model-inferred spans are significantly more consistent than manually annotated spans. Additionally, we describe a utility-aligned modification to conventional relaxed concept annotation evaluation for settings with a higher tolerance for noisy span boundaries.

While our results on time-efficiency and inter-annotator agreement are restricted to concept annotation, we believe that task decomposition is a natural strategy for optimally distributing annotation efforts between humans and models. We argue that inspecting sub-tasks along the axes of time-efficiency, accuracy, and consistency for models and humans can clarify how to best reduce manual effort for challenging annotation tasks generally.

2 Concept Annotation Procedure

We design an annotation procedure that is both (1) an efficient use of expensive domain expert time and (2) results in consistent and accurate concept annotations. Our proposed annotation procedure achieves this by concentrating manual annotation effort on annotating typed positions in the text instead of complete concept annotations, as we find that unitizing spans with a model both saves manual annotation effort and leads to more consistent annotations.

2.1 Formalization

Given a corpus C and a fixed manual annotation budget B, the annotation procedure produces a set of concept annotations over a subset of the corpus, exhausting B.

The annotation procedure design should be selected to maximize the number of annotated documents at sufficiently high annotation quality. We assume a low-budget setting where B is insufficient to annotate the entire corpus. The budget is distributed over the following annotation tasks, and we refer to the set of documents that can be

annotated for task t with B resources as C_B^t. B represents a time or money resource, with $|C_B^t|$ indicating the number of examples that can be annotated for that task under that resource constraint.

We assume that all approaches additionally involve annotating a held out set of examples for evaluation, which is not impacted and is therefore omitted from this discussion for brevity.

2.2 Annotation Tasks

Full Concept Annotation. For a closed set of concepts, we extract for each document a list of tuples, where each tuple contains a concept and span boundaries (start and end indices). In this work we restrict our analysis to flat concept annotation where spans do not overlap. The task is denoted with superscript^f.

Typed Position Annotation. Similar to concept annotation, typed position annotation extracts for each document a list of tuples, where each tuple contains a concept and an index specifying the where an instance of the specified concept appears in the document. There is no restriction on which position in the span is selected as the index. The task is denoted with superscript^p.

Span Boundary Detection. Given a typed position tuple and a document, span boundary detection produces start and end indices representing a span of the specified concept. The task is denoted with superscript^{p2s}.

2.2.1 Performing Annotation Tasks

Annotation tasks are performed by either a human annotator, A, or a model, $\mathcal{A}$. We assume that model annotators make use of task-specific training data, either as in-context exemplars (Brown et al., 2020) or as finetuning training data (Gururangan et al., 2020) for a large language model.

2.3 Baseline Manual Annotation

For a set of annotators $A_1, \ldots, A_k$, part of the budget B is spent obtaining annotators that can produce concept annotations that meet a quality threshold representing either inter-annotator agreement for each pair of annotators (double-blind) or agreement between annotators and a curator annotator (single-blind). Once this threshold is met, the remainder of the budget is exhausted on independent annotation effort, where each annotator A_i annotates disjoint subsets of the corpus for the full concept annotation task. This procedure results in $|C_B^f|$ gold annotated documents.

2.4 Baseline Model-based Annotation

In model-based annotation, B is exhausted by obtaining $|C_B^f|$ gold annotated documents using manual annotation (§2.3). A model $\mathcal{A}^f$ is trained to perform concept annotation using the gold documents. This procedure results in $|C_B^f|$ gold annotated documents and silver concept annotations over the remainder of the corpus. We select this baseline as our focus is on tasks where zero-shot LLM performance is insufficient, requiring additional expert input to clearly define the task. In this setting, full concept annotations are commonly used as gold exemplars in model-based annotation (Naraki et al., 2024; Goel et al., 2023a).

2.5 Decomposed Annotation

We introduce a modification to the baseline model-based annotation procedure to minimize manual unitizing effort. A subset of the budget, B_1, is spent obtaining gold-standard concept annotations over a small sample of the corpus $C_{B_1}^f$ (which we empirically find to be between 100-200 sentences; see §4.1). Over $C_{B_1}^f$, we insert a special character ♦ at a random position on each gold annotated span to indicate a typed position annotation. Then, we train a model $\mathcal{A}^{p2s}$ for the span boundary detection task over the typed position augmented $C_{B_1}^f$ sample.

The remainder of the budget, B_2, is spent annotating gold typed positions over unannotated documents in the corpus, resulting in $|C_{B_2}^p|$ annotated documents. Using $\mathcal{A}^{p2s}$, we infer span boundaries over $C_{B_2}^p$, resulting in gold-like concept annotations over the sample $C_{B_2}^p$. Since typed positions are necessarily cheaper than full concept annotation, $|C_B^f| < |C_{B_1}^f| + |C_{B_2}^p|$. Finally, $C_{B_1}^f \cup C_{B_2}^p$ is used as training data for $\mathcal{A}^f$. This procedure results in $|C_{B_1}^f|$ gold documents, $|C_{B_2}^p|$ gold-like documents with concept annotations, and the remainder of the corpus annotated with silver concepts.

3 Experimental Results

We compare the quality of data resulting from typed position annotation and full concept annotation for varying annotation budgets. Specifically, we measure: (1) the utility of annotated data as the accuracy of models that can be trained with the number of annotations (§3.3), and (2) inter-

annotator agreement calculated over the annotations (§3.4) at both the annotator training phase and corpus annotation phase.

3.1 Datasets

POLIANNA The policy design annotations (POLIANNA) dataset is a collection of EU directives and regulations on climate mitigation and renewable energy. The corpus contains 412 EU legal acts spanning 18 policies. 20,577 spans are annotated using a hierarchical scheme of 42 entity types (Sewerin et al., 2023). In addition to a curated set of annotations, POLIANNA includes raw annotations performed by 6 individual annotators, where each article is doubly annotated.

CRAFT The Colorado Richly Annotated Full-Text Corpus (CRAFT) contains 97 biomedical research articles (Bada et al., 2012). The documents are annotated for concepts from 11 separate biomedical ontologies. We specifically run experiments with the annotations from Sequence Ontology (SO), NCBI Taxonomy (NCBITaxon), Gene Ontology Biological Process (GO BP), Gene Ontology Molecular Function (GO MF), Chemical Entities of Biological Interest (CHEBI), where our primary experiments are focused on NCBITaxon, SO, and GO BP. We select these ontologies to represent a wide range of inter-annotator agreement as reported in Bada et al. (2012).

We additionally obtained raw annotation data for individual annotators during the course of annotator training. We consider a subset of ontologies that was annotated single-blind, where a curator corrected annotations performed by other annotators: GO BP, GO MF, SO, and CHEBI. These ontologies were most challenging to achieve satisfactory agreement, with the highest number of training sessions and the lowest initial consistency.

GENIA GENIA 1.0 is a set of research article abstracts from the molecular biology domain containing 18,546 sentences (Ohta et al., 2002). Five entity types are annotated: cell line, cell type, DNA, RNA and protein.

3.2 Models

We experiment with instruction-tuned autoregressive language models including Llama-3.1-8B-Instruct, Qwen3-8B-Instruct, Mistral-7B-Instruct-v0.3, Llama-3.2-1B-Instruct, and Falcon3-10B-Instruct. We primarily report finetuning results, as we found that in-context learning performs on average 27 points below finetuning in preliminary experiments with Llama-3.1-8B-Instruct, likely a result of the large amount of entity types (Milios et al., 2023). In this preliminary experiment, we used k-NN (Liu et al., 2022) to order exemplars as is standard for in-context NER (Wang et al., 2025a; Berger et al., 2025; Monajatipoor et al., 2024). For finetuning, We adopt the task formulation UniNER-7B-all-in-one as presented in UniversalNER with multi-turn chat format (Zhou et al., 2023). All finetuning results are reported over 10 random seeds.

3.3 Measuring Data Efficiency

We compare the data quality resulting from concentrating manual effort on typed positions (§2.5) and full concept annotation (§2.4).

Data Splits. We divide the corpus into gold, silver, and test splits. All manual annotation efforts are focused on the gold split, and we report results varying the size of the gold split with all other data splits fixed. The human-annotated gold split is used as training data for a model. Then we perform inference with the model over the silver split. We train a second model with the inferred annotations in the silver split. We focus our analysis on results over the test split as a utilitarian measure of silver data quality. We report standard NER metrics for strict (Tjong Kim Sang and De Meulder, 2003) and relaxed string matching (Bossy et al., 2013).

Annotation Procedure. For the baseline procedure (§2.4), we take the human annotations over the gold split as concept annotations (C_B^f). To simulate our decomposed annotation procedure (§2.5), the complete gold split is annotated with typed positions (C_B^p), and a small sample is augmented with span boundaries ($C_{B_1}^f$). In our experiments, we select a position uniformly at random from within the range of the curated span boundaries as we did not find a significant benefit from more elaborate sampling strategies (see Figure 2). The augmented gold sample is used as training data for a span boundary detection model, and this model infers the remainder of the gold split resulting in gold-like concept annotations over the complete gold split.

We select 100 as gold split sample size for the CRAFT datasets, and 200 for GENIA and POLIANNA. For CRAFT datasets and GENIA, we sample 1500 examples for the test split and for POLIANNA, we sample 1000 due to limitations in corpus size. We vary the size of gold split to

represent at most 500 time units.[1] We define one unit of time as the amount of time required to fully annotate one sentence for typed spans.

Note that in this work, we do not measure how much faster it is to annotate typed points than annotating typed spans, as this speed-up is dataset dependent. Based on previous work performing an empirical study over Japanese clinical notes and the high inferential load of span boundary annotation (Andrade et al., 2024; Finin et al., 2010; Fort et al., 2012), we report results for a range of reasonable speed-ups $m \in \{1.3, 1.5, 2.0, 3.0\}$. With n units of time, an annotator could annotate n sentences for concepts or $n' + m(n - n')$ sentences for typed positions, where a small subset of the gold split (n' sentences) is annotated for concepts to train the span boundary detection model and the remaining $n - n'$ sentences are annotated with typed positions. Across all experiments, we fix n' as 100 for CRAFT ontologies and 200 for GENIA and POLIANNA, based on experimental results for the span boundary detection task (Figure 2).

3.4 Measuring consistency

We measure how annotating positions instead of spans affects inter-annotator agreement at both the annotator training and corpus annotation stage.

We compare consistency at the annotator training stage with a subset of CRAFT ontologies for which we obtained raw annotator files from training sessions conducted between 2008-2010. For the first d documents annotated by both the curator and trainee annotator in the training sessions, we simulate typed position annotation, where the curation span boundaries are taken as the output for a given typed position. We train Llama-3.1-8B-Instruct models for span boundary detection over these documents. Then, we perform inference over the remainder of the documents. For $d \in \{5, 10, 20\}$, we report consistency for span match F1, where true positives are the instances of identical span boundaries for each pair of annotators. This is a relaxed version of the span and entity type match metric used in Bada et al. (2012).[2]

We additionally compare consistency at the corpus annotation stage using double-blind raw annotations for POLIANNA. For each annotator, we simulate uniformly random typed position annotation based on the individual annotator's span boundary annotation. We train a Llama-3.1-8B-instruct model on a sample of 100 sentences for the span boundary identification task where we take the curation span boundaries as the output for a given typed position. We use the span boundary detection model to infer span boundaries for each individual annotator's synthetic position annotations. We then compute agreement metrics for each pair of annotators, comparing the setting where span boundaries are marked by the model and the setting where span boundaries are marked by individual annotators. We report a suite of consistency metrics including F1, boundary accuracy, Levenshtein distance (Levenshtein et al., 1966), and Krippendorff's unitizing alpha (Artstein and Poesio, 2008).

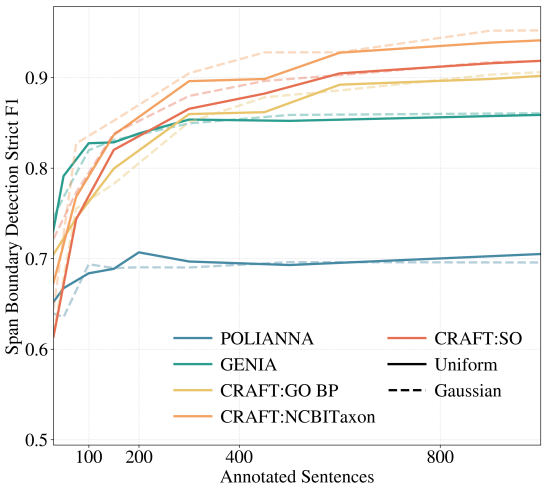

Figure 2: Llama-3.1-8B-instruct span boundary detection F1 over an increasing amount of annotated sentences. For POLIANNA and GENIA, performance tends to plateau at 200 sentences, and CRAFT requires 100 annotated sentences for at least .75 strict F1.

4 Results and Analysis

4.1 Span Boundary Inference is Cheap

In Figure 2, we measure the cost of developing a span boundary detection model. Using a Llama-3.1-8B-Instruct model, we observe that for all domains, we can achieve 0.7 strict F1 with at most 200 annotated sentences. Further, there is a negligible difference between marking positions at random on the span (Uniform) and closer the center (Guassian). This suggests that models are robust to the precise position selected, further reducing the annotator overhead.

In Figure 2, we also observe that the CRAFT domains tend to have the strongest span boundary detection performance followed by GENIA and

[1] Across the domains, 500 sentences make up between 1.7-18% of the dataset, representing a low-resource setting.

[2] The metrics originally computed with Knowtator annotation tool (Ogren, 2006) were re-implemented in Python.

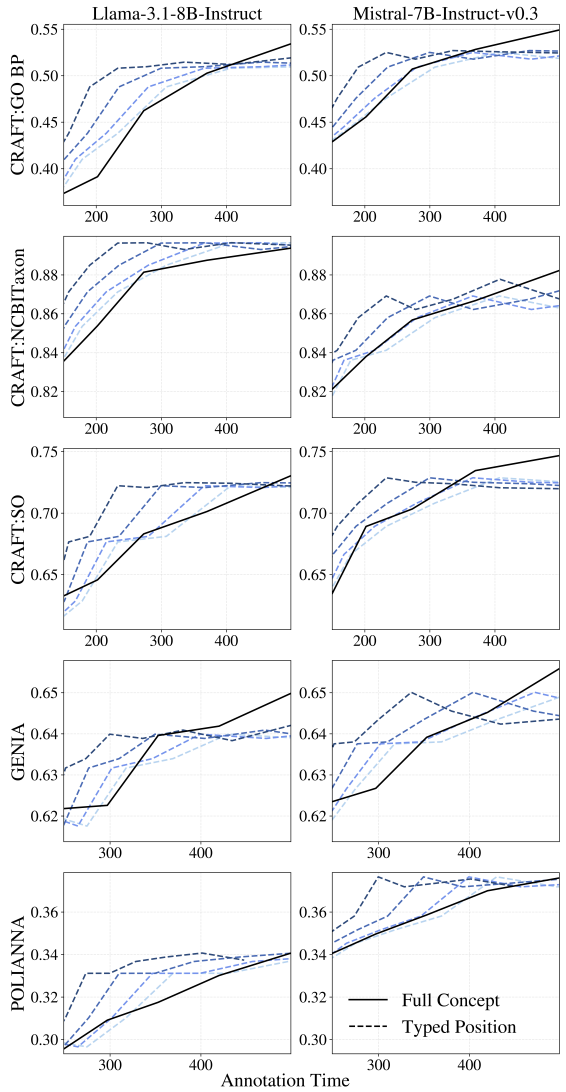

Figure 3: For varying amounts of annotator time, we report strict F1 where annotation time is spent on typed positions or full concept annotation. We project the performance improvement for varying speedups (1.3X, 1.5X, 2X, 3X) resulting from annotating positions instead of spans.

finally POLIANNA. There may be a relationship between average span length and the span boundary identification strict F1. The longest spans are in POLIANNA (~21 characters), followed by GENIA (~15 characters), and finally the CRAFT domains (~8 characters). This is consistent with previous annotation studies noting lower segmentation agreement for longer units of discourse (Hearst, 1997; Reynar, 1999; Ries, 2001).

4.2 Annotating Positions is More Efficient

In Table 1, we report strict F1 over the test split, and we compare between spending 100 additional units of time on typed position annotation and concept annotation, assuming that typed positions can be annotated twice as fast as full concepts. For our strongest performing models Llama-3.1-8B and Mistral-7B, typed position annotation performs better than full concept annotation on average. The largest performance improvements from typed position annotation are focused on CRAFT domains, and this is likely a result of low span boundary noise relative to POLIANNA and GENIA (Figure 2). For GO BP, for example, there is a 3-7 point difference in F1 when annotating typed positions instead of concepts. The performance for our smallest model (Llama-3.2-1B-Instruct) is especially sensitive to lower performance over the silver split, resulting in more instability than the larger models over the test split. Similar to Bai et al. (2025), we find that the Qwen3-7B-Instruct model tends to under-perform for concept annotation.

For our best performing models, we vary the annotation budget beyond 100 additional units of time in Figure 3. We plot test split performance for manual time spent on full concept annotation and typed positions for multiple potential speed-ups. Across potential speed-ups, the performance improvement from annotating typed positions is highest when there are fewer than 350 units of time available, but the improvement is negligible or negative for larger budgets. This results from inferred span boundary errors outweighing the benefit of additional data that we can afford to annotate. For low annotation resource settings however, annotating typed points results in stronger performance than full concept annotation.

4.3 Inferred Span Boundaries are More Consistent

For CRAFT, we compare in Figure 4 the span boundary agreement resulting from inferred span boundaries and manual concept annotation. With as few as five annotated documents, the trained span boundary detection model results in a 3-20 point improvement in agreement in the first training session that it would be deployed. We found that for GO BP + GO MF[3] (p-value < .001) and SO (p-value < .05), inferred span boundaries are more consistent than manual concept annotation over five documents. This is especially the case for both ontologies in the first third of training sessions (p-value < .01). In general, the inferred span boundaries tend towards less extreme valleys than manual concept annotation. The results are mixed for CHEBI, where inferred span boundaries

[3]GO BP and GO MF are combined, as both ontologies were annotated together.

Model	Manual Annotation	GO BP	NCBITaxon	SO	GENIA	POLIANNA	Average
Llama-3.2-1B	Full Concept	$0.225_{\pm 0.103}$	$0.565_{\pm 0.218}$	$0.501_{\pm 0.046}$	$0.483_{\pm 0.039}$	$0.144_{\pm 0.039}$	0.384
	Typed Position	$\mathbf{0.299}_{\pm 0.083}$	$\mathbf{0.690}_{\pm 0.041}$	$\mathbf{0.518}_{\pm 0.035}$	$\mathbf{0.496}_{\pm 0.049}$	$\mathbf{0.151}_{\pm 0.047}$	0.431
Qwen3-8B	Full Concept	$0.305_{\pm 0.118}$	$0.753_{\pm 0.047}$	$0.324_{\pm 0.197}$	$0.584_{\pm 0.019}$	$0.269_{\pm 0.017}$	0.447
	Typed Position	$\mathbf{0.366}_{\pm 0.076}$	$\mathbf{0.789}_{\pm 0.040}$	$\mathbf{0.383}_{\pm 0.138}$	$\mathbf{0.585}_{\pm 0.022}$	$\mathbf{0.289}^{**}_{\pm 0.017}$	0.482
Falcon-3-10B	Full Concept	$0.371_{\pm 0.048}$	$0.785_{\pm 0.033}$	$0.569_{\pm 0.030}$	$\mathbf{0.612}_{\pm 0.013}$	$0.278_{\pm 0.012}$	0.523
	Typed Position	$\mathbf{0.422}_{\pm 0.048}$	$\mathbf{0.797}_{\pm 0.025}$	$\mathbf{0.577}_{\pm 0.034}$	$0.589_{\pm 0.026}$	$\mathbf{0.279}_{\pm 0.022}$	0.533
Llama-3.1-8B	Full Concept	$0.391_{\pm 0.054}$	$0.853_{\pm 0.021}$	$0.645_{\pm 0.026}$	$0.623_{\pm 0.010}$	$0.310_{\pm 0.016}$	0.564
	Typed Position	$\mathbf{0.452}^{**}_{\pm 0.029}$	$\mathbf{0.875}^{**}_{\pm 0.013}$	$\mathbf{0.678}^{**}_{\pm 0.024}$	$\mathbf{0.633}^{*}_{\pm 0.013}$	$\mathbf{0.325}^{*}_{\pm 0.015}$	0.593
Mistral-7B	Full Concept	$0.455_{\pm 0.016}$	$0.837_{\pm 0.024}$	$0.687_{\pm 0.028}$	$0.627_{\pm 0.011}$	$0.350_{\pm 0.008}$	0.591
	Typed Position	$\mathbf{0.485}^{**}_{\pm 0.028}$	$\mathbf{0.846}_{\pm 0.016}$	$\mathbf{0.695}_{\pm 0.037}$	$\mathbf{0.638}^{*}_{\pm 0.009}$	$\mathbf{0.356}_{\pm 0.009}$	0.604

Table 1: Strict F1 scores comparing utility resulting from 100 additional units of time on differing annotation tasks (§2.2). Here, we assume annotating typed positions is twice as fast as full concept annotation; see Figure 3 for alternate projected speed-ups. We report mean and standard deviation across 10 seeds. Concentrating manual effort on typed position annotation is a significantly better use of time on average than full concept annotation for our best-performing models over domains with high span boundary detection performance. *and **denote statistical significance with $p < .05$ and $p < .01$.

Metric	Manual	Model	Model-Manual
↑ NER relaxed F1	0.6697	0.8420	0.1723^{***}
↑ NER strict F1	0.6023	0.7559	0.1536^{***}
↑ Start accuracy	0.7298	0.8677	0.1379^{***}
↑ End accuracy	0.7719	0.7988	0.0269
↓ Levenshtein dist	12.3619	5.7474	-6.6145^{***}
↑ Krippendorff's α	0.4308	0.5997	0.1689^{**}

Table 2: For POLIANNA, we compare agreement for manual concept annotation with manual typed positions and model-inferred spans using a Llama-3.1-8B-Instruct model, averaged over annotator pairs. Inferring span boundaries improves consistency across all metrics. **and ***denote statistical significance with $p < .01$ and $p < .001$. respectively.

Sample Size	Strict	PoG Strict	Relaxed	PoG Relaxed
250	0.296	0.313	0.324	0.329
300	0.309	0.327	0.340	0.344
500	0.341	0.360	0.372	0.378

Table 3: Alternative pool-of-gold metrics (PoG) with strict and relaxed F1 for a Llama-3.1-8B-instruct model over POLIANNA for varying amounts of full concept annotations.

under-perform manual annotation for fewer than 20 annotated documents in the first half of the training sessions.

Overall, the results suggest that there is on average no loss in span boundary consistency when annotation trainees concentrate their efforts on learning to annotate entity types correctly, rather than the span boundaries. When we consider span boundary agreement in isolation, we reach acceptable levels of agreement in fewer training sessions with inferred span boundaries. In addition to the consistency gains, there are time savings from annotating typed positions over the remaining documents instead of full concept annotations.

We report pairwise consistency averaged over 6 annotators for POLIANNA in Table 2, comparing typed position annotations with inferred span boundaries and manual concept annotations. We find that typed point annotation results in higher consistency than full concept annotation across all metrics. Inspecting scores for each annotator pair, we found that the consistency improvement resulting from the use of the model is especially large in cases of low human agreement. For example, for the least consistent annotator pair, inferring span boundaries improves the Krippendorff's alpha score by 0.399.

4.4 Implications for Evaluation

To better align with utility-based evaluation (Tsai et al., 2006; Herman Bernardim Andrade et al., 2023), we report alternative *pool-of-gold* metrics in Figure 7, where predicted spans are matched with any human annotated span overlapping with the set of curated spans. We observe a 1.8% performance improvement with pool-of-gold metrics for strict F1 and a smaller 0.05% improvement for relaxed F1. Relaxed F1, which is based on thresholded edit-distance, is designed to represent the pool-of-gold strict F1, and the performance gap between them is likely a result of insufficient candidate annotated spans for each curated span, as documents in POLIANNA are annotated by only two annotators. As we move towards a generative paradigm in information extraction, relaxed string matching is an exceedingly natural choice for extrinsic evaluation (Herman Bernardim Andrade et al., 2023), and rather than relying on edit distance, the metric could be made more accurate by constructing semantically equivalent, text-bound span sets for each curation span.

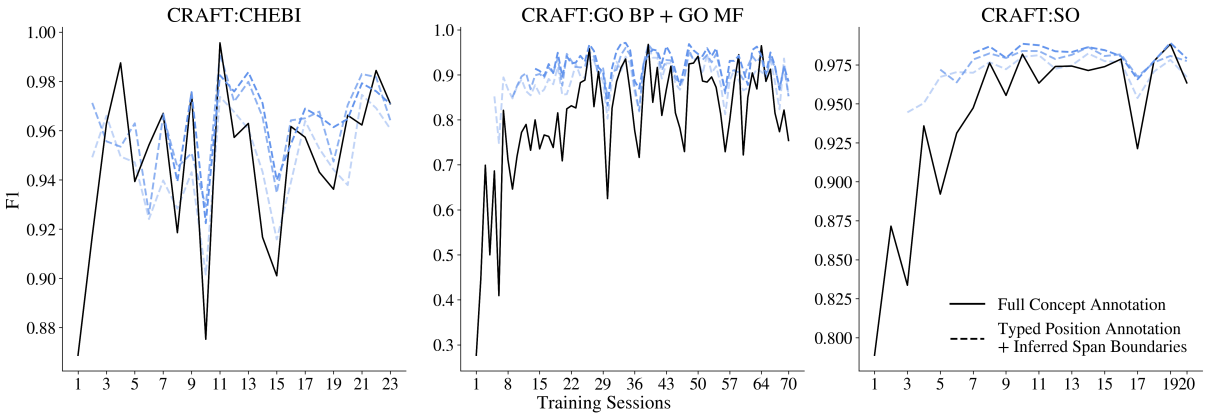

Figure 4: Span boundary F1 representing concept annotation agreement and typed position annotation with inferred boundaries agreement. The first 5 (light blue), 10 (medium blue), or 20 (dark blue) annotated documents are used to train a Llama-3.1-8B-Instruct model to infer span boundaries. For the first third of training sessions, we observe stronger consistency boundaries from inferred spans for GO BP + GO MF (p-value < .001) and SO (p-value < .01) for as few as 5 training documents for span boundary detection.

5 Related Work

Concept annotation disagreement has been linked to span length (Amigo et al., 2025; Artstein and Poesio, 2008), large label spaces (Papay et al., 2020; Fort et al., 2012), and ambiguity in annotation schema (Peng et al., 2024). To mitigate disagreement, annotation efforts often reduce task scope (Hovy et al., 2006) or introduce increasingly detailed guidelines (Bada et al., 2012), which can limit task utility or increase annotator burden. In contrast, we reduce disagreement by modifying the annotation task itself, removing span boundary selection as a source of disagreement.

Recent work has explored using off-the-shelf large language models and few-shot learning to reduce manual effort in concept annotation. For NER, this includes generating synthetic data via paraphrasing (Sharma et al., 2023), prompt-engineering (Tang et al., 2023; Santoso et al., 2024), and in-context learning (Berger et al., 2025). However, few-shot performance is highly sensitive to retrieved examples (Sainz et al., 2023), and improvements based on retrieval modules (Wang et al., 2025b) or label statistics (Bai et al., 2025) still require large pools of high-quality annotations.

Decomposition has been suggested as a general strategy to reduce annotation burden (Stubbs, 2012; Gandhi et al., 2023). For example, Wang et al. (2021); Ma et al. (2022); Wang et al. (2022); Morand et al. (2025) decompose NER to span extraction and entity typing, and Xie et al. (2023) elicit mentions of each entity type one-by-one. The value of these decompositions have been demonstrated to improve model performance, but it remains unclear if the decomposition translates to time savings for expert annotators. Our work differs by selecting a decomposition that has been empirically shown to be cheap (Andrade et al., 2024).

Manual annotation efforts can also be reduced with interface design (Giachelle et al., 2021; Kummerfeld, 2019), pre-annotation in well-resourced domains (Roberts et al., 2008), active learning (Liu and Wong, 2024), or incorporating LLMs (Goel et al., 2023b; Naraki et al., 2024). These approaches in combination with our proposed annotation procedure would reduce annotation costs further.

6 Conclusion

We demonstrate that it is a better use of annotator time to mark positions in the text overlapping with mentions instead of span boundaries. The additional examples that we are able to cover with position annotations outweigh the errors resulting from inferred span boundaries in low-resource settings. We additionally find that inferred span boundaries are more consistent at both the corpus annotation and annotator training phases. Considering the low cost of span boundary detection, a direction for future work could be in distributing span-bound concept annotations as typed positions, where custom span lengths are configured with a small amount of lightweight annotation. As low-resource concept annotation performance improves asymmetrically for different domains (Jimenez Gutierrez et al., 2022; Nagar et al., 2025; Volkanovska, 2025), it may better serve end-users to routinely evaluate what sub-tasks we still need to manually annotate, as opposed to whether we should annotate at all.

7 Limitations

We find that annotating typed points is most effective in settings where the span boundary detection model is high-performing (at least 0.85 strict F1). Our strongest results were focused on high-consistency datasets (CRAFT sub-domains), whereas results were mixed over POLIANNA. Span boundary detection performance is likely partially a result of inter-annotator consistency. Users may be able to a priori estimate this level of agreement in advance using crude heuristics such as span length.

Another limitation of this work is that we do not measure the speed-up that results from replacing concept annotation with typed position annotation. For a new domain, it would be necessary to estimate (1) whether span boundary detection performance is strong and (2) what the speed-up resulting from typed position annotation is.

Most annotation software, particularly for concept annotation is not designed for position annotation. This is additional infrastructure that would be necessary to develop in future work.

In this work, we primarily develop models for annotation using supervised finetuning instead of training-free approaches to NER such as in-context learning. This decision is based on empirical results about the performance of finetuning, but it may be a limitation for use in settings with low compute access. We also do not inspect the potential of models with fewer than 1B parameters for span boundary detection in this work.

Relatedly, we restrict our notion of cost to manual annotation resources, without considering the cost of finetuning. In future work, the scope could be broadened to corpus development cost more generally.

We do not include discontinuous or nested spans in our analysis. While our proposed procedure is not incompatible with such spans, it is unclear to what extent our results would generalize.

References

Enrique Amigo, Elena Álvarez-Mellado, Julio Gonzalo, and Jorge Carrillo-de Albornoz. 2025. Evaluating sequence labeling on the basis of information theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27849–27860, Vienna, Austria. Association for Computational Linguistics.

Gabriel Herman Bernardim Andrade, Shuntaro Yada, and Eiji Aramaki. 2024. Is boundary annotation necessary? evaluating boundary-free approaches to improve clinical named entity annotation efficiency: Case study. *JMIR Medical Informatics*, 12(1):e59680.

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner Jr, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, and 1 others. 2012. Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):161.

Fan Bai, Hamid Hassanzadeh, Ardavan Saeedi, and Mark Dredze. 2025. LLMs are better than you think: Label-guided in-context learning for named entity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28360–28380, Suzhou, China. Association for Computational Linguistics.

Uri Berger, Tal Baumel, and Gabriel Stanovsky. 2025. In-context learning on a budget: A case study in token classification. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 7–14, Albuquerque, New Mexico. Association for Computational Linguistics.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne P Bernard. 2024. NuNER: Entity recognition encoder pre-training via LLM-annotated data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11829–11841, Miami, Florida, USA. Association for Computational Linguistics.

Robert Bossy, Wiktoria Golik, Zorana Ratkovic, Philippe Bessières, and Claire Nédellec. 2013. BioNLP shared task 2013 – an overview of the bacteria biotope task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 161–169, Sofia, Bulgaria. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with amazon's mechanical turk*, pages 80–88.

Karën Fort, Adeline Nazarenko, and Sophie Rosset. 2012. Modeling the complexity of manual annotation tasks: a grid of analysis. In *Proceedings of COLING 2012*, pages 895–910, Mumbai, India. The COLING 2012 Organizing Committee.

Nupoor Gandhi, Anjalie Field, and Emma Strubell. 2023. Annotating mentions alone enables efficient domain adaptation for coreference resolution. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10543–10558, Toronto, Canada. Association for Computational Linguistics.

Fabio Giachelle, Ornella Irrera, and Gianmaria Silvello. 2021. Medtag: a portable and customizable annotation tool for biomedical documents. *BMC Medical Informatics and Decision Making*, 21(1):352.

Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, Jean Steiner, Itay Laish, and Amir Feder. 2023a. Llms accelerate annotation for medical information extraction. In *Proceedings of the 3rd Machine Learning for Health Symposium*, volume 225 of *Proceedings of Machine Learning Research*, pages 82–100. PMLR.

Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, Jean Steiner, Itay Laish, and Amir Feder. 2023b. Llms accelerate annotation for medical information extraction. *ArXiv*, abs/2312.02296.

Jonas Golde, Patrick Haller, Max Ploner, Fabio Barth, Nicolaas Jedema, and Alan Akbik. 2025. Familiarity: Better evaluation of zero-shot named entity recognition by quantifying label shifts in synthetic training data. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 820–834, Albuquerque, New Mexico. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Gabriel Herman Bernardim Andrade, Shuntaro Yada, and Eiji Aramaki. 2023. Comparative evaluation of boundary-relaxed annotation for entity linking performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8238–8253, Toronto, Canada. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

i2b2. 2010. Fourth i2b2/va shared-task and workshop: Challenges in natural language processing for clinical data (relations task). `https://www.i2b2.org/NLP/Relations/`.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Uri Katz, Matan Vetzler, Amir Cohen, and Yoav Goldberg. 2023. NERetrieve: Dataset for next generation named entity recognition and retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3340–3354, Singapore. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, and 1 others. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(Suppl 1):S2.

Klaus Krippendorff. 2004. Measuring the reliability of qualitative text analysis data. *Quality and quantity*, 38(6):787–800.

Jonathan K. Kummerfeld. 2019. SLATE: A super-lightweight annotation tool for experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Florence, Italy. Association for Computational Linguistics.

Vladimir I Levenshtein and 1 others. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Jiaxing Liu and Zoie SY Wong. 2024. Utilizing active learning strategies in machine-assisted annotation for clinical named entity recognition: a comprehensive analysis considering annotation costs and target effectiveness. *Journal of the American Medical Informatics Association*, 31(11):2632–2640.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022. Decomposed meta-learning for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1584–1596, Dublin, Ireland. Association for Computational Linguistics.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184, Singapore. Association for Computational Linguistics.

Masoud Monajatipoor, Jiaxin Yang, Joel Stremmel, Melika Emami, Fazlolah Mohaghegh, Mozhdeh Rouhsedaghat, and Kai-Wei Chang. 2024. Llms in biomedicine: A study on clinical named entity recognition. *arXiv preprint arXiv:2404.07376*.

Victor Morand, Nadi Tomeh, Josiane Mothe, and Benjamin Piwowarski. 2025. Tommer–efficient entity mention detection from large language models. *arXiv preprint arXiv:2510.19410*.

Aishik Nagar, Viktor Schlegel, Thanh-Tung Nguyen, Hao Li, Yuping Wu, Kuluhan Binici, and Stefan Winkler. 2025. LLMs are not zero-shot reasoners for biomedical information extraction. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 106–120, Albuquerque, New Mexico. Association for Computational Linguistics.

Yuji Naraki, Ryosuke Yamaki, Yoshikazu Ikeda, Takafumi Horie, Kotaro Yoshida, Ryotaro Shimizu, and Hiroki Naganuma. 2024. Augmenting ner datasets with llms: towards automated and refined annotation. *arXiv preprint arXiv:2404.01334*.

Philip V. Ogren. 2006. Knowtator: A protégé plug-in for annotated corpus construction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Demonstrations*, pages 273–275, New York City, USA. Association for Computational Linguistics.

Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, Hideki Mima, and Junichi Tsujii. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the human language technology conference*, pages 73–77. Morgan Kaufmann Publishers Inc. San Francisco.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895, Online. Association for Computational Linguistics.

Siyao Peng, Zihang Sun, Sebastian Loftus, and Barbara Plank. 2024. Different tastes of entities: Investigating human label variation in named entity annotations. In *Proceedings of the Third Workshop on Understanding Implicit and Underspecified Language*, pages 73–81, Malta. Association for Computational Linguistics.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226, Online. Association for Computational Linguistics.

Jeffrey C. Reynar. 1999. Statistical models for topic segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 357–364, College Park, Maryland, USA. Association for Computational Linguistics.

Klaus Ries. 2001. Segmenting conversations by topic, initiative, and style. In *Workshop on Information Retrieval Techniques for Speech Applications*, pages 51–66. Springer.

Angus Roberts, Robert Gaizasukas, Mark Hepple, and Yikun Guo. 2008. Combining terminology resources and statistical methods for entity recognition: an evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv preprint arXiv:2310.03668*.

Joan Santoso, Patrick Sutanto, Billy Cahyadi, and Esther Setiawan. 2024. Pushing the limits of low-resource NER using LLM artificial data generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9652–9667, Bangkok, Thailand. Association for Computational Linguistics.

Sebastian Sewerin, Lynn H Kaack, Joel Küttel, Fride Sigurdsson, Onerva Martikainen, Alisha Esshaki, and Fabian Hafner. 2023. Towards understanding policy design through text-as-data approaches: The policy design annotations (polianna) dataset. *Scientific Data*, 10(1):896.

Saket Sharma, Aviral Joshi, Yiyun Zhao, Namrata Mukhija, Hanoz Bhathena, Prateek Singh, and Sashank Santhanam. 2023. When and how to paraphrase for named entity recognition? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7052–7087, Toronto, Canada. Association for Computational Linguistics.

Amber Stubbs. 2012. Developing specifications for light annotation tasks in the biomedical domain. In *Proceedings of the Third Workshop on Building and*

Evaluating Resources for Biomedical Text Mining, page 71.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. 2006. Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics*, 7(1):92.

Elena Volkanovska. 2025. Large language models as annotators of named entities in climate change and biodiversity: A preliminary study. In *Proceedings of the 1st Workshop on Ecology, Environment, and Natural Language Processing (NLP4Ecology2025)*, pages 24–33, Tallinn, Estonia. University of Tartu Library.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025a. GPT-NER: Named entity recognition via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275, Albuquerque, New Mexico. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025b. Gpt-ner: Named entity recognition via large language models. In *Findings of the association for computational linguistics: NAACL 2025*, pages 4257–4275.

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. Learning from language description: Low-shot named entity recognition via decomposed framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1618–1630, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot NER with ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956, Singapore. Association for Computational Linguistics.

Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15654–15669, Bangkok, Thailand. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. Universalner: Targeted distillation from large language models for open named entity recognition.

A Appendix

A.1 Implementation Details

Finetuning Models are trained over 15 epochs with a learning rate of 2e-4. We perform quantized low-rank adaptation finetuning for target modules: "q_proj", "v_proj".

Pre-processing We split documents for CRAFT and POLIANNA into sentences. For CRAFT, this results in high mention sparsity, so we downsample examples without entities in the training data, preserving only 20% of negative examples. We additionally restrict our analysis to contiguous spans and we filter out discontinuous mentions in the CRAFT dataset.

A.1.1 Extended Results

In Table 5, we report pairwise consistency metrics for POLIANNA corpus annotation. We find that the consistency improvement resulting from the use of the model is especially large in cases of low human agreement. For least consistent annotator pairs A-C and C-F, we can observe high Levenshtein distance and low Krippendorff's alpha for traditional NER annotation, but when the model is used to infer span boundaries over the synthetic points, we can observe much stronger metrics. This improvement is smaller for pairs of annotators that exhibit high agreement for traditional NER annotation. This suggests that using a model to identify span boundaries could be especially useful for settings where boundary agreement is harder to obtain or human annotators are less consistent with one another.

Trivial Span Boundary Annotation Conflict Examples	
... bioliquids or biomass fuels produced from food and **feed crops** for which a significant expansion of the production ...	... change-risk biofuels, bioliquids or biomass fuels produced from **food and feed crops** for which a significant expansion of the production ...
... with copies of the national communications and biennial **reports** submitted to the unfccc secretariat.	... commission with copies of the national communications and **biennial reports** submitted to the unfccc secretariat.
... an assessment of their implementation respectively in their **integrated national energy and climate plans** and progress reports pursuant to regulation (eu) 2018/1999.	... assessment of their implementation respectively in their integrated **national energy and climate plans** and progress reports pursuant to regulation (eu) 2018/1999.
... voluntary financial payment to the union renewable energy **financing mechanism** set up at union level, contributing to renewable ...	... making a voluntary financial payment to the union **renewable energy financing mechanism** set up at union level, contributing to renewable ...
Meaningful Span Boundary Annotation Conflict Examples	
... — including electricity from renewable sources, electricity from **coal** and lignite, and cogeneration of electricity and useful ...	... gas, electricity — including electricity from renewable sources, **electricity from coal and lignite** , and cogeneration of electricity and useful heat ...
... before 30 april each year statistics on national **electricity** and heat production from high and low efficiency ...	... before 30 april each year statistics on national **electricity and heat** production from high and low efficiency cogeneration, in ...

Table 4: Examples of span boundary conflicts for overlapping spans labeled with the same entity type. In some examples, span boundary conflicts represent conflicting meanings wrt the entity type, while in other examples the span boundary conflicts are produced by inclusion of an adjective or article. Sometimes, the boundary conflicts are meaningful.

User: Member states shall bring into force the laws, regulations and administrative provisions necessary to comply with this directive by 5 December 2010

Assistant: I've read this text.

User: Your task is to extract all entities and identify their entity types.
The output should be in a list of tuples of the following format:[("entity 1", "type of entity 1") ...

Assistant: [("5 December 2010","Compliance time"), ("regulations","Regulatory instrument"), ("administrative provisions","Unspecified instrument"), ("member states","Default addressee")]

Figure 5: Task format for NER to extract the entity types and spans given a sentence.

User: Figure 9 ♦Expression in Precursors Is Not Sufficient for the Emergence of OT/VS Neurons (A-C) X-gal staining of coronal sections ...

Assistant: I've read this text.

User: Extract the span marked by ♦ that describes gene expression.

Assistant: Expression

Figure 6: Task format for span boundary identification to extract the span covering the special character ♦ of a specified entity type.

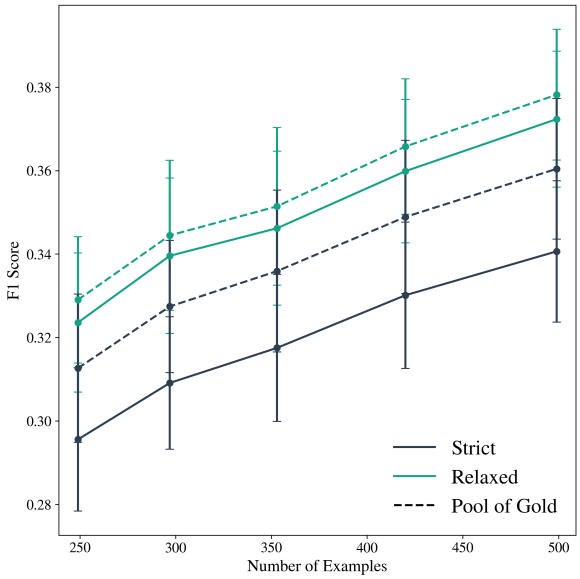

Figure 7: We report alternative pool-of-gold metrics: strict and relaxed F1 for a Llama-3.1-8B-instruct over POLIANNA.

Annotators	Metric	Human	Model	Δ (Model - Human)
F-B	↑ NER relaxed F1	$0.7059_{\pm 0.0076}$	$0.8706_{\pm 0.0050}$	0.1647^{***}
	↑ NER strict F1	$0.6227_{\pm 0.0077}$	$0.7904_{\pm 0.0083}$	0.1678^{***}
	↑ Start boundary accuracy	$0.7995_{\pm 0.0003}$	$0.8705_{\pm 0.0268}$	0.0710^{*}
	↑ End boundary accuracy	$0.8345_{\pm 0.0031}$	$0.7767_{\pm 0.0337}$	-0.0578
	↓ Levenshtein distance	$7.6011_{\pm 0.0650}$	$7.0640_{\pm 0.7488}$	-0.5372
	↑ Krippendorff's alpha	$0.5586_{\pm 0.0038}$	$0.5726_{\pm 0.0316}$	0.0140
A-F	↑ NER relaxed F1	$0.6939_{\pm 0.0031}$	$0.8450_{\pm 0.0264}$	0.1510^{**}
	↑ NER strict F1	$0.6556_{\pm 0.0031}$	$0.7740_{\pm 0.0340}$	0.1184^{**}
	↑ Start boundary accuracy	$0.7868_{\pm 0.0038}$	$0.8743_{\pm 0.0261}$	0.0875^{**}
	↑ End boundary accuracy	$0.8150_{\pm 0.0012}$	$0.8077_{\pm 0.0156}$	-0.0073
	↓ Levenshtein distance	$7.7720_{\pm 0.0980}$	$4.7418_{\pm 0.8080}$	-3.0302^{**}
	↑ Krippendorff's alpha	$0.5454_{\pm 0.0054}$	$0.6324_{\pm 0.0282}$	0.0871^{*}
A-B	↑ NER relaxed F1	$0.6702_{\pm 0.0012}$	$0.8618_{\pm 0.0228}$	0.1916^{***}
	↑ NER strict F1	$0.5890_{\pm 0.0018}$	$0.7702_{\pm 0.0208}$	0.1812^{***}
	↑ Start boundary accuracy	$0.7488_{\pm 0.0003}$	$0.8786_{\pm 0.0189}$	0.1298^{***}
	↑ End boundary accuracy	$0.8435_{\pm 0.0008}$	$0.8532_{\pm 0.0542}$	0.0097
	↓ Levenshtein distance	$9.4226_{\pm 0.1031}$	$5.4391_{\pm 1.6521}$	-3.9835^{*}
	↑ Krippendorff's alpha	$0.5211_{\pm 0.0010}$	$0.6711_{\pm 0.0861}$	0.1501
C-B	↑ NER relaxed F1	$0.6908_{\pm 0.0099}$	$0.8365_{\pm 0.0164}$	0.1457^{***}
	↑ NER strict F1	$0.6056_{\pm 0.0105}$	$0.7539_{\pm 0.0223}$	0.1483^{**}
	↑ Start boundary accuracy	$0.7765_{\pm 0.0109}$	$0.9074_{\pm 0.0730}$	0.1309
	↑ End boundary accuracy	$0.8021_{\pm 0.0079}$	$0.8549_{\pm 0.0883}$	0.0528
	↓ Levenshtein distance	$9.7415_{\pm 0.4642}$	$3.2225_{\pm 1.6599}$	-6.5189^{**}
	↑ Krippendorff's alpha	$0.4871_{\pm 0.0188}$	$0.7205_{\pm 0.1698}$	0.2334
C-F	↑ NER relaxed F1	$0.6388_{\pm 0.0026}$	$0.8086_{\pm 0.0128}$	0.1698^{***}
	↑ NER strict F1	$0.5898_{\pm 0.0022}$	$0.7304_{\pm 0.0037}$	0.1406^{***}
	↑ Start boundary accuracy	$0.7116_{\pm 0.0015}$	$0.8146_{\pm 0.0820}$	0.1030
	↑ End boundary accuracy	$0.7279_{\pm 0.0017}$	$0.7862_{\pm 0.0280}$	0.0583^{*}
	↓ Levenshtein distance	$14.1821_{\pm 0.1379}$	$6.9178_{\pm 0.6260}$	-7.2643^{***}
	↑ Krippendorff's alpha	$0.3705_{\pm 0.0032}$	$0.5002_{\pm 0.1462}$	0.1297
A-C	↑ NER relaxed F1	$0.6187_{\pm 0.0018}$	$0.8294_{\pm 0.0192}$	0.2107^{***}
	↑ NER strict F1	$0.5511_{\pm 0.0015}$	$0.7164_{\pm 0.0183}$	0.1653^{***}
	↑ Start boundary accuracy	$0.5556_{\pm 0.0011}$	$0.8606_{\pm 0.0376}$	0.3051^{***}
	↑ End boundary accuracy	$0.6085_{\pm 0.0010}$	$0.7140_{\pm 0.0139}$	0.1055^{***}
	↓ Levenshtein distance	$25.4523_{\pm 0.4212}$	$7.0994_{\pm 1.9446}$	-18.3529^{***}
	↑ Krippendorff's alpha	$0.1023_{\pm 0.0011}$	$0.5016_{\pm 0.0429}$	0.3993^{***}

Table 5: Annotator-pair performance metrics sorted by Human Krippendorff's alpha (descending). We observe larger model improvements for annotator pairs with lower human agreement, particularly for NER F1 scores and Levenshtein distance.