Uncertainty Quantification for Named Entity Recognition via Conformal Prediction

Matthew Singer
North Carolina State University

Karl Pazdernik
Pacific Northwest National Laboratory

Srijan Sengupta
North Carolina State University

Abstract

Named Entity Recognition (NER) is a foundational component in many language tasks, such as knowledge graph construction, information extraction, and question answering. However, existing NER models typically output a single predicted label sequence without any quantification of uncertainty, leaving downstream applications vulnerable to cascading errors. We introduce a conformal prediction framework for NER that produces prediction sets over full label sequences with finite-sample coverage guarantees, serving an analogous role to confidence intervals in classical statistics. To tailor the general conformal prediction methodology to the NER application, we propose the use of Mondrian conformal prediction according to input length and language, hybrid probability-index nonconformity scores, and a modified RAPS procedure for sequence labeling. These techniques mitigate the problem of overly large prediction sets while maintaining valid coverage. Experiments on CoNLL++, CoNLL-Reduced, and WikiNEuRal benchmarks demonstrate that our methods consistently achieve the target confidence while producing efficient prediction sets across diverse base models. This work establishes a statistically principled approach to uncertainty-aware NER with direct benefits for downstream knowledge-driven NLP systems.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Named entity recognition (NER) involves identifying spans of text corresponding to specific categories of real-world *named entities* such as persons, locations, and organizations (Weischedel et al., 2011). NER serves as a critical first step for many complex natural language processing (NLP) applications, including named entity disambiguation (NED), information extraction, question answering, and text summarization (Yamada et al., 2016; Mollá et al., 2006; Nan et al., 2021). Formally, the NER task can be defined as follows: Given a dataset of N labeled samples $\mathcal{D} = \{(\mathbf{w}_i, \mathbf{y}_i)\}_{i=1}^{N}$, each observation $\mathbf{w}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,t_i})$ denotes the observed input word sequence of length t_i and $\mathbf{y}_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,t_i})$ the associated sequence of class labels, where $y_{i,j} \in \mathcal{L}$. The label space $\mathcal{L}$ consists of entity types together with the non-entity label and special start/stop indicators. Because entities may span multiple words, the inside-outside-beginning (IOB2) tagging scheme (Sang and Veenstra, 1999) is typically used.

As an illustration, consider the example sentence: *Sarah is from New York City.* An accurate NER system using the IOB2 tagging scheme would output Table 1. These entity tags enable downstream modules to further process the text, such as linking "Sarah" to a specific individual in a database or connecting "New York City" to a specific location.

Word	Sarah	is	from	New	York	City
Label	B-PER	O	O	B-LOC	I-LOC	I-LOC

Table 1: IOB2 NER labels for an English sentence.

Despite their critical role, NER systems often provide only a single predicted label sequence for an input sentence, without any measure of confidence or uncertainty (see Table 1). This lack of uncertainty quan-

tification can lead to error propagation in downstream tasks. For instance, if the NER model incorrectly identifies "New York" (rather than "New York City") as a location, a subsequent NED system might link it to the state of New York instead of the city, introducing semantic inaccuracies. These kinds of errors are particularly detrimental when NER outputs are used in high-stakes domains such as biomedical literature mining, legal document analysis, or automated knowledge base population. Yet current NER frameworks, including those based on sequence labeling (Ma and Hovy, 2016; Lample et al., 2016), substring classification (Zhong and Chen, 2020), and machine reading comprehension (Li et al., 2019), typically offer no way to quantify the uncertainty of their predictions.

In this paper, we propose a rigorous and flexible framework to incorporate uncertainty quantification into NER models. Rather than producing a single output that is either correct or incorrect, our method generates a *set* of plausible labels that is guaranteed to contain the true label with a certain probability. This probability can be specified by the user. This prediction set plays a role analogous to a confidence interval in classical statistics, offering a statistically valid quantification of uncertainty tailored to the NER task.

Specifically, we focus on the *sequence labeling* formulation of NER, where each word in the sentence is mapped to a tag indicating whether it is part of an entity and what type it belongs to (see Table 1). Our proposed approach produces sets of plausible full-sentence labelings that are guaranteed to contain the correct labeling with a user-specified confidence level. For the example sentence in Table 1, rather than outputting only one labeling (e.g., tagging "New York" as the location), our method might produce a set that includes both "New York" and "New York City" as possible location spans, thereby providing a clear and objective quantification of uncertainty and mitigating the risk of downstream error propagation.

We choose the sequence labeling formulation of NER over other *span-based* formulations due to the output's ability to contain multiple NER entities with contextual information relating one entity to another. The full NER label space is defined as

$$\mathcal{L} = \{l_0, l_1, \ldots, l_c, l_{\text{start}}, l_{\text{end}}\},$$

where c is the number of entity types, l_0 is the non-entity label, and $l_{\text{start}}, l_{\text{end}}$ are special start/stop indicators. The goal of NER is to learn a mapping $\mathbf{y}$ for each sentence, where the space of possible labelings is $|\mathcal{L}|^t$ and t is the length of the sentence.

In related work, incorporating uncertainty quantification (UQ) into NLP tasks is a growing field of research with many techniques and applications (Shorinwa et al., 2025; Hu et al., 2023; Campos et al., 2024). UQ seeks to quantify both aleatoric (inherent variability in data) and epistemic uncertainty (stemming from model limitations). The increasing deployment of black-box machine learning models has driven a parallel rise in the importance of UQ techniques, as black-box models are unable to be understood/reasoned with and provide little to no statistical guarantees when compared to more principled statistical techniques. While this paper focuses on UQ in model outputs, there has been substantial research on utilizing uncertainty for auxiliary tasks such as model training (Nie et al., 2025).

UQ in NER remains a nascent topic. Zhang et al. (2024) performs NER via two models, a local span-identification model and an LLM-based classification model, with uncertainty-aware outputs from the span-identification model being an input to the second classification step. Prior UQ methods for NER also include model calibration (Liang et al., 2021), ensemble-based approaches (Yang et al., 2024; Akkasi and Varoğlu, 2016; He et al., 2023), Bayesian inference (Akkasi and Varoğlu, 2016; Maragoudakis et al., 2006; He et al., 2023), among active learning and other statistical techniques (Nguyen et al., 2021; Liu et al., 2022a; Vazhentsev et al., 2022). Each of these methods has merits but also notable limitations in the NER context. Calibration methods aim to align predicted probabilities with empirical frequencies. Well-calibrated outputs can be used to generate prediction sets, but these methods are highly vulnerable to *miscalibration* since calibration failures can compromise coverage guarantees. Ensemble methods rely on model diversity, such as bootstrapping, dropout, or varying seeds, to estimate uncertainty. However, they are computationally expensive and difficult to deploy at scale. Bayesian methods offer a theoretically sound alternative, but full Bayesian neural networks are challenging to train and scale.

Conformal prediction (Angelopoulos et al., 2023; Shafer and Vovk, 2008), in contrast, provides a statistically principled and scalable UQ framework, making it especially well-suited to the challenges of NER. Conformal prediction is a class of techniques for constructing prediction sets with finite-sample coverage guarantees under the assumption of exchangeability. This assumption, which posits that the joint distribution of a sequence of observations remains invariant under permutations, is notably much weaker than the commonly assumed independent and identically distributed (i.i.d.) condition in classical uncertainty quantification techniques. Using this assumption, conformal prediction methods estimate the quantile dis-

tribution of a non-conformity score function in a leave-one-out fashion (*transductive* conformal prediction) or via a held-out calibration set (*inductive* conformal prediction). Given a new observation, a prediction set with coverage $1 - \alpha$ can be constructed using the estimated quantile distribution. In this paper, we adopt the inductive conformal prediction paradigm because of its greater scalability and computational efficiency. Despite the slow adoption of conformal prediction to the domain of NER, conformal prediction techniques have been developed across a wide variety of applications (Zhou et al., 2025; Campos et al., 2024; Gong et al., 2025).

To our knowledge, the only prior work on conformal prediction in NER is Fisch et al. (2022), where the authors utilized a substring multi-class classification framework in which every possible subsequence is evaluated individually. We previously described this as a span-based formulation of NER. This approach is computationally expensive and fails to capture joint uncertainty across multiple entities.

In contrast, we construct prediction sets over entire label sequences, allowing us to account for contextual dependencies and utilize the structured prediction framework of a CRF. This formulation captures contextual relationships among co-occurring entities (e.g., 'if *Sarah* is labeled as a person, then *New York City* is likely a location'). This full-sentence (i.e. full-sequence) formulation of the problem complicates class-conditional calibration, as multiple entity types must be jointly considered. Unlike the span-based conformal prediction methodology, the coverage guarantee of our technique is evaluated over the entire sequence, capturing all NER entities, together, in-context. This eliminates the permutation problem present in span-based NER, where there is no guarantee or guidance as to how to combine multiple span-level prediction sets, causing there to be a large number of possible prediction combinations. To the best of our knowledge, no prior work has studied conformal prediction at the sentence-level for NER.

The next sections describe NER models and the conformal prediction methodology. After presenting the basic unconditional conformal prediction methodology, we also demonstrate how the conformal prediction technique can be calibrated for class conditional coverage based on sentence length, language, and entity type. We further demonstrate how standard non-conformity scores may be adapted to include rank-based information, therefore increasing the efficiency of our prediction sets. These methods make UQ in NER both practical and theoretically sound.

2 NER MODELS

NER is a foundational task in NLP, originating from the Message Understanding Conferences (MUC) in the 1990s. The field has undergone numerous advancements over the years. Early NER systems relied on hand-crafted entity lists and rule-based grammar patterns. Around the early 2000s, researchers began employing feed-forward neural networks (Hammerton, 2003), followed by recurrent neural networks.

A major breakthrough occurred in 2011, when researchers incorporated conditional random fields (CRFs) into convolutional neural network (CNN) architectures (Collobert et al., 2011) and later by 2016 to long short-term memory networks (LSTM) (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Clark et al., 2018). The CRF layer enabled models to learn structured prediction rules and significantly improved benchmark performance. Although CRF-based decision heads continue to be used in state-of-the-art systems, recent research has shifted toward more expressive embedding methods, reformulating NER as a non-sequential task (Fisch et al., 2022; Liu et al., 2022b), incorporating external knowledge sources (Wang et al., 2022), and improving training methodologies (Zhou and Chen, 2021; Conneau, 2019).

The architecture adopted in this work is a BERT-BiLSTM-CRF model (Tedeschi et al., 2021). At a high level, the CRF defines the probability of a label sequence as a normalized score over possible sequences, where emission scores depend on the word embeddings and transition scores depend on successive label pairs. With neural features, these score functions simplify to linear transformations of embeddings and a label–label transition matrix, as shown in equation (1):

$$\mathbb{P}(\mathbf{y}_{a:a+b} \mid \mathbf{x}; y_{a-1}, y_{a+b+1}) = \frac{s(\mathbf{y}_{a:a+b}, \mathbf{x})}{\sum_{\mathbf{z} \in \mathcal{L}^m} s(\mathbf{z}_{a:a+b}, \mathbf{x})}, \tag{1}$$

where $s(\cdot)$ is a compact way of writing the unnormalized value function for a label sequence:

$$s(\mathbf{z}_{a:a+b}, \mathbf{x}) = \exp\{\mathbf{W}^*(z_{a-1}, :) \cdot x_{a-1} + \sum_{j=i}^{a+b+1} \mathbf{W}(z_{j-1}, z_j) + \mathbf{W}^*(z_j, :) \cdot x_j\}.$$

Please see the supplement for full mathematical details of the model including a visual schematic, as well as Beam search decoding details.

In this work, we adopt CRFs as the NER models because they provide a mathematically rigorous founda-

tion for sequence labeling: their factorized structure yields well-defined sequence probabilities, transparent scoring functions, and efficient exact/approximate decoding. This mathematical tractability makes CRFs suitable for deriving non-conformity scores and establishing finite-sample coverage with our conformal procedure. Although large language models have become popular for NER, CRF-based decision layers continue to deliver competitive results on several benchmarks, especially when paired with strong contextual encoders (Jiang et al., 2022; Han et al., 2024; Bhadauria et al., 2024). Most importantly, our contribution is foundational and modular: the CRF serves as the scoring backbone that produces probabilities for computing non-conformity scores, and the conformal prediction wrapper then converts these into valid prediction sets. This separation cleanly decouples modeling from uncertainty quantification, so newer or stronger CRF variants (e.g., neural or span-structured CRFs) can be dropped in without changing the conformal machinery.

3 CONFORMAL PREDICTION

Consider a desired confidence level of $1-\alpha \in (0,1)$, and a dataset of N labeled observations $\mathcal{D} = \{(\mathbf{w}_i, \mathbf{y}_i)\}_{i=1}^{N}$. For notational convenience, we will often suppress the observation index i; therefore, each word j in sentence i may be written as $w_{i,j} = w_j$. This word is mapped to a vector representation $x_{i,j} = x_j \in \mathbb{R}^d$. This vector representation is the output of the Bi-LSTM into the CRF layer. The dataset is partitioned into a training set $\mathcal{I}_{\text{train}}$ and a calibration set $\mathcal{I}_{\text{cal}}$, and a CRF model is trained using $\mathcal{I}_{\text{train}}$. Let $nc(\mathbf{y} \mid \mathbf{x})$ denote a generic non-conformity score based on Equation 1 such that higher values of this function correspond to greater non-conformity between the value y and the expected value of y given x and the previously observed calibration data. Therefore, large non-conformity scores are less desirable. The conformal prediction objective is to determine a threshold τ such that for a new observation $(\mathbf{x}_{new}, \mathbf{y}_{new})$ the conformal prediction set $\mathcal{C}(\mathbf{x}_{\text{new}}, \tau)$ satisfies

$$\mathbb{P}\left(\mathbf{y}_{\text{new}} \in \mathcal{C}(\mathbf{x}_{\text{new}}, \tau)\right) \geq 1-\alpha. \tag{2}$$

The prediction set $\mathcal{C}(\mathbf{x}, \tau)$ is defined as:

$$L(\mathbf{x}, \tau) = \min\left\{r : nc(\mathbf{y}^{(r)} \mid \mathbf{x}) \geq \tau\right\}. \tag{3}$$

$$\mathcal{C}(\mathbf{x}, \tau) = \left\{\mathbf{y}^{(r)} : r \leq L(\mathbf{x}, \tau)\right\}. \tag{4}$$

Here, $\mathbf{y}^{(r)}$ represents the r-th most probable output label sequence, ranked by the value function in Equation 1. An acceptable threshold τ that achieves the desired $1-\alpha$ level of coverage is given by the $\lceil(1-\alpha)(1+|\mathcal{I}_{\text{cal}}|)\rceil$-th largest non-conformity score on the calibration set. Provided that the calibration and test sets are exchangeable, this threshold guarantees the desired coverage. Proposition 1 in the supplemental materials proves the coverage of this baseline conformal prediction procedure. If an alternative conformity score is used in which 'higher' values indicate 'greater conformity', then Equation 3 becomes:

$$L(\mathbf{x}, \tau) = \max\left\{r : nc(\mathbf{y}^{(r)} \mid \mathbf{x}) \leq \tau\right\} \tag{5}$$

However, all conformity scores may be transformed into non-conformity scores by multiplying the conformity score by negative one; therefore, for convenience, we will only utilize the non-conformity score formulation present in Equation (4). Notably, Equation (4) is written for a generic non-conformity score $nc(\mathbf{y}^{(k)} \mid \mathbf{x})$, however, any non-conformity score may be utilized, and all non-conformity scores will achieve at least the desired coverage. Despite this, all non-conformity scores do not achieve the same prediction set efficiency and some non-conformity scores produce a larger-sized prediction set on average than others.

While this standard procedure produces valid prediction sets with coverage at least $1-\alpha$, in this paper, we use the adaptive conformal prediction (ACP) (Romano et al., 2020), which produces more efficient prediction sets. Assume that a new test observation $(\mathbf{x}_{\text{test}}, \mathbf{y}_{\text{test}})$ is exchangeable with the calibration dataset $\mathcal{I}_{\text{cal}}$. The threshold τ is set to the $(1-\alpha)$-quantile of the non-conformity scores computed on the calibration set:

$$\begin{aligned}\tau &= Q_{1-\alpha}\left(\{nc(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}_{\text{cal}}}\right) \\ &= \lceil(1-\alpha)(1+|\mathcal{I}_{\text{cal}}|)\rceil^{th} \text{ largest score in } \mathcal{I}_{\text{cal}}.\end{aligned}$$

In practice, when prediction sets are formed using Equation (3), coverage tends to slightly exceed the target level. This is because non-conformity scores increase in discrete steps as each additional sequence is added to the prediction set. Consequently, the final element often causes the total score to overshoot τ, especially when the target coverage is low and each discrete step is large. To mitigate this, ACP estimates the overshoot and applies a probabilistic rule: the final element is included with probability proportional to how close the nonconformity score is to the threshold. The adjusted prediction set is defined as

$$\mathcal{C}(\mathbf{x}, \tau, \alpha) = \begin{cases}\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(r)}\} & \text{if } u \leq V(\cdot) \\ \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(r-1)}\} & \text{otherwise}\end{cases}, \tag{6}$$

where $nc(\mathbf{y}^{(r-1)} \mid \mathbf{x}) < \tau < nc(\mathbf{y}^{(r)} \mid \mathbf{x})$, and $u \sim \text{Uniform}(0,1)$. The probability of including $\mathbf{y}^{(v)}$ is defined by the overshoot function:

$$V(\cdot) = \frac{f(1-\alpha) - \hat{\pi}(\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(v-1)}\})}{\hat{\pi}(\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(v)}\}) - \hat{\pi}(\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(v-1)}\})}. \tag{7}$$

where $\hat{\pi}(\cdot)$ is an empirical estimate of the coverage function and $f(1-\alpha)$ corresponds to the desired coverage level. The 'best' choice of $\hat{\pi}$ and f is not known and depends on the non-conformity score utilized.

3.1 Three Baseline Non-conformity Scores

So far, we have described conformal prediction utilizing a generic, undefined non-conformity score. However, as mentioned, the efficiency of the prediction sets generated by conformal prediction is dependent on the choice of non-conformity score. We will now define three non-conformity scores that achieve efficient prediction sets and can be easily calculated with our chosen CRF NER model. A commonly used non-conformity score in classification tasks is based on the model's confidence in a predicted sequence. In analogy to regression, where non-conformity is measured as the residual, the classification setting defines

$$\mathbf{nc1}(\mathbf{y} \mid \mathbf{x}) = 1 - \hat{\mathbb{P}}(\mathbf{y} \mid \mathbf{x}),$$

where $\hat{\mathbb{P}}(\mathbf{y} \mid \mathbf{x})$ is the predicted probability of label sequence $\mathbf{y}$ given the input embedded word sequence $\mathbf{x}$ arising from the trained CRF model. Previously, all NER and CRF notation was defined using the true probability $\mathbb{P}$, however, we will now assume that $\mathbb{P}(\mathbf{y} \mid \mathbf{x}) \approx \hat{\mathbb{P}}(\mathbf{y} \mid \mathbf{x})$ and will utilize the estimated probability measure for the remainder of our conformal prediction work. We refer to the non-conformity score described in the above equation as **nc1** and use it as a baseline throughout this work. Here **nc1** is an appropriate choice of non-conformity score for the classification setting due to how the CRF model is fit. On average, $\mathbf{x}, \mathbf{y}$ pairs that are common within the training dataset should produce higher predicted probabilities than uncommon or absent observation pairs. Therefore, a high predictive probability indicates that an observed pair conforms to the training data.

Alternatively, instead of considering the probability of one observation, the non-conformity score could be made such that it accounts for the predicted probability of all responses within the top r outputs. Under this paradigm, a small predicted probability summation indicates that the overall set of responses does not conform sufficiently to the training data; therefore, elements are added to the prediction set until a sufficient predicted probability is achieved. We define this alternate *cumulative probability* non-conformity score:

$$\mathbf{nc2}(\mathbf{y}^{(r)} \mid \mathbf{x}) = \sum_{k=1}^{r} \hat{\mathbb{P}}(\mathbf{y}^{(k)} \mid \mathbf{x}).$$

Later, we will show how for the task of NER, the **nc2** performs worse on average than **nc1** due to what we call the 'run-away prediction problem'.

When constructing a conformal prediction set with **nc1**, elements are added to the prediction set until the last added element exceeds a threshold, meanwhile when utilizing **nc2**, elements are added until the cumulative probability exceeds a threshold. Because **nc2** does not evaluate individual probabilities, sometimes there becomes a point where the appropriate threshold has not been reached yet the remaining sequences which have been included in the set have near-zero probabilities. Therefore, in order to achieve the desired cutoff threshold τ, many small probability sequences must be included in the prediction set, drastically decreasing the efficiency of this non-conformity score.

To contrast with these probabilistic scores, we introduce a simple 'rank-based' score:

$$\mathbf{nc3}(\mathbf{y}^{(r)} \mid \mathbf{x}) = r.$$

This score, denoted **nc3**, produces prediction sets of fixed size and is robust to vanishing probability values. Like **nc1**, on average, observations that exist frequently in the training data should achieve a higher predicted rank when compared to observations that are rare or non-existent in the training data. Although **nc3** does not adapt to instance difficulty, it avoids the scenarios in which **nc1** and **nc2** produce large prediction sets with many low probability responses. All three of the aforementioned non-conformity scores are effective due to their ability to represent model accuracy, sensitivity to uncertainty, and ease of interpretation. It is logical to think that predictions with a small individual probability or a large index are unlikely to occur from an accurate model.

Additionally, when evaluating equation (7) with probabilistic scores such as **nc1** and **nc2**, we found that the following approximation works well:

$$V(\cdot) = \frac{\tau - \hat{\mathbb{P}}(\mathbf{y}^{(v-1)})}{\hat{\mathbb{P}}(\mathbf{y}^{(v)}) - \hat{\mathbb{P}}(\mathbf{y}^{(v-1)})},$$

where τ is the prediction set cutoff, and $\hat{\mathbb{P}}$ is the estimated model probability for the given value of y. For the rank-based score **nc3**, an approximation is obtained using the calibration quantile function:

$$V(\cdot) = \frac{(1-\alpha) - Q(nc(\mathbf{y}^{(v-1)}))}{Q(nc(\mathbf{y}^{(v)}) - Q(nc(\mathbf{y}^{(v-1)}))},$$

where $Q(nc(\mathbf{y}^{(v)}))$ is the empirical CDF value (percentile) of the score within the calibration set $\mathcal{I}_{\text{cal}}$. The adaptive approximation of **nc3** produces prediction sets closer to the true coverage when compared to **nc1** or **nc2**. This is because **nc3** produces prediction sets of constant size without adaptive coverage such that the quantile function of **nc3** on the calibration

data is a good estimator of the empirical coverage of the prediction set.

4 CONFORMAL PREDICTION FOR NER

In this section, we propose several modifications of the base methodology customized for NER. Notably, we use a standard top-K approximation of the sequence probability; see the supplement for details.

4.1 Unconditional Conformal Prediction

Depending on the type of model used, conformal prediction sets may be made for various levels of granularity. For CRF based models, the most direct application of conformal prediction is to form prediction sets over the set of fully labeled word sequences. These conformal prediction sets return a set of sentence-level labels such that each element in the prediction set is a full sequence with one label for every input word. The probability of the correct full label sequence being included is calibrated to the desired confidence level.

Input:	*Sarah*	*is*	*from*	*New*	*York*	*City*	nc2($\mathbf{y}$|$\mathbf{x}$)
$\mathbf{y}^{(1)}$:	O	O	O	B-LOC	I-LOC	I-LOC	.28
$\mathbf{y}^{(2)}$:	B-PER	O	O	B-LOC	I-LOC	I-LOC	.66
$\mathbf{y}^{(3)}$:	O	O	O	O	O	O	.98
$\mathbf{y}^{(4)}$:	B-PER	O	O	B-ORG	B-ORG	I-ORG	.995

Table 2: Illustrative example of an prediction set.

Table 2 provides an example of the top four predicted sentence-level outputs for the sentence 'Sarah is from New York City'. Given a non-conformity score $nc(\mathbf{y}|\mathbf{x})$ and input sequence $\mathbf{x}$, let $\hat{Q}_{1-\alpha}$ denote the calibrated threshold estimated from the calibration set. Unconditional conformal prediction is then performed by applying adaptive conformal prediction as described in Equation 6. Please see the supplement for algorithmic details. Using Table 2, if the cutoff threshold were to be .60 then the resulting conformal prediction set would be either $\{\mathbf{y}^{(1)}\}$ or $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}\}$ depending on the uniform random variable used in adaptive conformal prediction. As mentioned, there are no separate prediction sets for 'Sarah' and 'New York', as this method only produces full-label-sequence predictions.

Next, in Table 3 we demonstrate the effectiveness of the adaptive coverage method when utilizing our three proposed baseline non-conformity scores on the multilingual WikiNEuRal benchmark dataset and the Babelscape model (Tedeschi et al., 2021). The adaptive coverage method achieves empirical coverage levels that more closely align with the desired threshold of 90%. Additionally, the table highlights a key characteristic of the **nc3** method: its use of a fixed set size. As a result, when early stopping is applied, an empty prediction set is consistently returned. Table 3 also demonstrates how **nc2** constructs significantly larger prediction sets than **nc1** and **nc3** while also overshooting the desired coverage. This will be shown to be a common trend with the **nc2** non-conformity score as applied to NER.

Dataset	**Method**	**Confidence**	**Set Size**
NC1	Early Stop	0.8983	0.9269
	Full Prediction	0.9525	1.9269
	ACP	0.9289	1.2585
NC2	Early Stop	0.8981	36.64
	Full Prediction	0.9951	37.64
	ACP	0.9950	37.19
NC3	Early Stop	0.0000	0.0000
	Full Prediction	0.9351	1.0000
	ACP	0.8946	0.9567

Table 3: Average coverage and set size of unconditional predictions sets for **NC1**, **NC2**, and **NC3** when the Babelscape model is evaluated on the multilingual-WikiNEuRal benchmark.

4.2 Class-conditional Conformal Prediction

A key property of the exchangeability requirement for conformal prediction is that mixtures of exchangeable sequences remain exchangeable. This enables calibration across confounding or lurking variables, which may impact the nonconformity score, such as sentence length and language. Overall calibration is near the target when aggregated across inputs (see Figure 2 in the supplement for details). However, despite good *unconditional* calibration, conditional miscalibration can occur when the data is stratified by confounding variables such as sequence length and language. For the WikiNEuRal dataset, language stratification shows systematic under/over-coverage without stratification (Fig. 3 in the supplement), and length stratification similarly shows miscalibration for longer sequences (Fig. 4 in the supplement).

The language and length examples are specific cases of miscalibration in conformal prediction problems when there are crucial confounders that play a role in the partitions of the training and test datasets. When constructing prediction sets for a natural language model, it is desirable to be able to make as small a prediction set as possible while maintaining validity. If there are sub-populations of the input data that are easily identifiable with minimal error, then it is valuable to stratify the overall dataset, provided that we have a sufficient amount of data. By stratifying the population dataset into varying subsections, we are able to isolate poorly performing sections, such that their bad

performance does not increase the average prediction set size. This procedure may also be viewed as isolating well-performing sections, such that we are able to decrease their average prediction set size relative to the population's average. This method of partitioning the event space is known as *Mondrian* conformal prediction (Vovk et al., 2005). The limit to the number of strata in which we are able to stratify your data set is based on the number of observations, the ability to accurately group each stratum, and, importantly, the belief that observations belonging to different strata should behave differently from one another. In the case of NER, we identify two such grouping variables: sentence length and language.

When splitting our dataset in accordance with sentence length, it is impractical to separate each 'length' in its entirety as the benchmark calibration set becomes too small for rarely occurring sentence lengths. Therefore, in order to ensure an appropriate sample size in our calibration dataset, we group sentence lengths together such that similar length sentences are grouped together. This method of grouping similar class structures is motivated by Ding et al. (2023) in the clustering of granular classes for regular classification problems. For all benchmark datasets, sentence-length strata are grouped into ranges of 1–10, 11–20, 21–30, 31–40, and 40+ words. To formally evaluate the effect of binning, we carried out pairwise Kolmogorov–Smirnov (KS) tests whose results are reported in the supplement. For each length and language bin, we calculate independent cutoff thresholds and modify equation 4 as follows:

$$L(\mathbf{x}, \tau_{lang,length}) = \min\{k : nc(\mathbf{y}^{(k)} \mid \mathbf{x}) \geq \tau_{lang,length}\},$$
$$\mathcal{C}(\mathbf{x}, \tau, lang, length) = \left\{\mathbf{y}^{(k)} : k \leq L(\mathbf{x}, \tau_{lang,length})\right\} \tag{8}$$

Where $\tau_{lang,length}$ is a language and length-specific cutoff for the stratified (i.e. Mondrian) prediction set. This cutoff is equal to the $(1-\alpha)$ quantile of the calibration subsection that matches the desired language/length bin. The proof that Equation 8 is well calibrated is provided as part of Theorem 1 in the supplementary materials. Figure 1 shows the recalibrated coverage across different sentence lengths for the English language. In Figure 1, we calculate 'empirical coverage' only utilizing the top-100 responses, not allowing the prediction set to expand beyond 100 elements. Therefore, empirical coverage is bound by the maximum accuracy of those first 100 predictions, causing each coverage line to dip below the diagonal as the desired coverage approaches one.

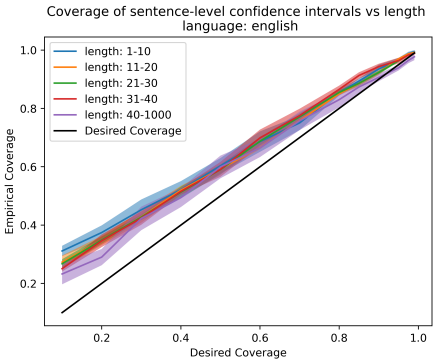

Figure 1: Ccoverage across sentence lengths for the English stratum of the WikiNEuRal benchmark with **nc1**.

4.3 Index-Based Non-Conformity Scores

As discussed earlier, conformal prediction for NER can be applied in multiple configurations with various non-conformity scores. However, when constructing prediction sets for NER, users often run into a common issue: the creation of large prediction sets populated by low-probability outputs. This is particularly problematic when using the **nc2** score. The top few predictions may account for most of the confidence mass, but still fall short of the threshold τ, requiring the inclusion of many low-confidence predictions. This inflates prediction set size and undermines efficiency.

Notably, this run-on behavior affects both **nc1** (probability) and **nc2** (cumulative probability), but not **nc3** (rank-based). Because **nc3** enforces a fixed-size threshold, it naturally limits growth. We propose two hybrid strategies and evaluate one existing strategy that combines probability-based and index-based conformal prediction to harness their respective strengths. The effectiveness of each hybrid non-conformity score is evaluated in Section 5.

4.3.1 Naive Intersection of Sets

This naive approach relies on the assumption that, in the worst case, two prediction sets are minimally overlapping. If there are two sets that each contain at least 95% of the probability, then their intersection must contain no less than 90% of the probability. Let C_{idx} and C_{prob} be prediction sets calibrated at levels $1-\alpha$ and $1-\beta$, respectively. Additionally, C_{idx} is constructed with a index based nonconformity score which limits set growth (**nc3**), and C_{prob} is constructed with a model probability based nonconformity score which is able to stop earlier if its threshold is satisfied (**nc1** or **nc2**). Then, without any further assumptions, we have $\mathbb{P}\left(\mathbf{y}_{\text{new}} \in C_{idx}(\mathbf{x}) \cap C_{prob}(\mathbf{x})\right) \geq 1-\alpha-\beta$. The previous statement is proved as part of Theorem 2 in

the supplementary materials. When the confidence levels α and β are selected appropriately, the resulting hybrid method yields a prediction set that is at least as efficient as those produced by either non-conformity score alone. Importantly, α and β can be optimized via a grid search over the calibration dataset. In scenarios where one method consistently outperforms the other, it is always possible to set either α or β to zero and assign the entire confidence budget to the superior method in order to meet the desired coverage.

4.3.2 Conditional Prediction Sets

An alternative is to constrain the probability-based score to be conditioned on the index being below a threshold τ_{idx}. Let: $C_{\text{idx}}(\mathbf{x}) = \{\mathbf{y} : \text{rank}(\mathbf{y}) \leq \tau_{\text{idx}}\}$, such that the desired coverage of $C_{\text{idx}}(\mathbf{x})$ is $1-\alpha$. We may then define a second set using only those calibration samples that fall within this index range:

$$\mathbb{P}(\mathbf{y}_{\text{new}} \in C_{\text{prob}}(\mathbf{x}) \mid \mathbf{y}_{\text{new}} \in C_{\text{idx}}(\mathbf{x})) \geq 1-\beta.$$

The total coverage is thus bounded by:

$$\mathbb{P}(\mathbf{y}_{\text{new}} \in C_{\text{idx}}(\mathbf{x}) \cap C_{\text{prob}}(\mathbf{x})) \geq (1-\alpha)(1-\beta).$$

This approach enforces tighter control than the naive method while reducing unnecessary set growth. The proof of the coverage bound is contained within the supplementary materials, Theorem 3. Like the naive prediction set, the best value of α and β is dependent on the specific application and behavior of its nonconformity scores. This technique is additionally not limited to the use of one index and one probability-based non-conformity score; it can instead be used for any two non-conformity score combinations. Unlike the naive approach, there is no 'slack' in the prediction sets caused by this method, and therefore, it does produce tighter prediction sets at the cost of being slightly more burdensome to implement.

4.3.3 Regularized Prediction Sets

A third approach is the RAPS (Regularized Adaptive Prediction Sets) procedure (Angelopoulos et al., 2020), which modifies the conformity score to include a linear penalty on the index. Unlike the previous methods, the RAPS procedure is a modified non-conformity score and does not utilize two separate non-conformity scores or two prediction sets. The RAPS procedure was initially proposed using a variation of **nc2**, but below we define it using a generic non-conformity score, as later we will show how it may be improved by utilizing **nc1** instead.

$$\text{Score} = nc(\mathbf{y}^{(k)}|\mathbf{x}) + \lambda \cdot \max(k - \tau_{\text{idx}}, 0)$$

Like the previous two methods, the addition of a regularization term reduces the amount of low-probability results that may be included in the prediction set. While effective, this method assumes a fixed linear relationship (λ) and does not allow for flexible control of how much the index contributes to total error. In both the naive and conditional approaches, the user can set a maximum set size based on the chosen index and understand approximately how much error is associated with each non-conformity score. In contrast, the RAPS procedure does allow for the user to set a maximum set size by selecting a large value of λ, but does not easily allow for the user to know how much error is due to the choice of λ, τ_{idx}, and k versus the used non-conformity score.

5 EXPERIMENTS

We evaluate our methods using three benchmark datasets: CoNLL++ (Wang et al., 2019), CoNLL-reduced, and WikiNEuRal (Tedeschi et al., 2021), and four base models: Babelscape, Dslim, Jean-Baptiste, and TNER (Tedeschi et al., 2021; Devlin et al., 2018a; Polle, 2022; Ushio and Camacho-Collados, 2021). Additional details on datasets and models are provided in the supplement. Our experiments assess (i) the choice of nonconformity score for sentence-level prediction sets, (ii) the effect of index-based refinements such as naive, conditional, and RAPS, and (iii) model- and dataset-level performance. Across all conditions, we report empirical coverage and prediction set size, focusing on a target confidence of 95%.

5.1 Comparing index-based Methods

Table 4 shows that, among the three base nonconformity scores, **nc1** yields the most efficient sets at high coverage levels, whereas **nc3** is preferable when smaller sets at lower coverage are acceptable.

Sentence-Level						
Desired Coverage	nc1		nc2		nc3	
	Cov.	Size	Cov.	Size	Cov.	Size
0.8	0.8483	0.8999	0.9949	27.6058	0.7937	0.8496
0.9	0.9275	1.2564	0.995	37.6617	0.8939	0.9569
0.95	0.9681	1.8099	0.9951	46.1498	0.9434	1.2503
0.975	0.9834	2.3949	0.9951	53.9261	0.9795	3.0
0.99	0.9907	4.5512	0.9952	59.9726	0.9895	11.0

Table 4: Average coverage and set size for **nc1**, **nc2**, and **nc3** with Babelscape model on WikiNEuRal dataset.

Table 5 displays the efficiency and coverage of the six index-based hybrid non-conformity scores when evaluated on sentence-level prediction sets for the WikiNEuRal Benchmark with the Babelscape model. As a general trend, we can observe that all hybrid methods, when utilizing **nc1** perform better than their **nc2**

counterparts when both methods achieve the desired coverage. The RAPS procedure also performs slightly better than the conditional and naive methods for all methods that utilize **nc1**; however, naive and conditional **nc1** implementations are an improvement over the RAPS-**nc2** implementation. This is notable because the original RAPS methodology was proposed utilizing the **nc2** non-conformity score. We believe that part of the disparity between the conditional and RAPS-based prediction sets is the granularity of the grid search performed to estimate the optimal values of α and β for the conditional prediction sets. RAPS prediction sets also perform a grid search over the hyperparameter term of λ, but the impact of a sparse grid on the efficiency of the RAPS non-conformity score is unclear when compared to a grid search for the conditional non-conformity scores.

	Sentence-Level (NC1)					
Desired Coverage	**Naive + nc1**		**RAPS + nc1**		**Cond. + nc1**	
	Cov.	Size	Cov.	Size	Cov.	Size
0.8	0.8504	0.9101	0.8503	0.8702	0.8504	0.9100
0.9	0.9276	1.2972	0.9228	0.9807	0.9274	1.2962
0.95	0.9676	1.8308	0.9675	1.8286	0.9675	1.8287
0.975	0.9811	2.2526	0.9770	2.1126	0.9811	2.2524
0.99	0.9913	5.3971	0.9907	5.3778	0.9913	5.3968
	Sentence-Level (NC2)					
Desired Coverage	**Naive + nc2**		**RAPS + nc2**		**Cond. + nc2**	
	Cov.	Size	Cov.	Size	Cov.	Size
0.8	0.7844	0.5105	0.9396	1.033	0.9307	1.0000
0.9	0.8885	0.9546	0.9397	1.0332	0.9307	1.000
0.95	0.9438	1.3345	0.9704	1.9823	0.9700	1.9741
0.975	0.9742	2.5008	0.9788	2.9940	0.9786	3.0000
0.99	0.9903	15.0000	0.9909	15.8872	0.9907	15.9298

Table 5: Average coverage and set size for the naive, raps, and conditional methods using **nc1** or **nc2** on WikiNEuRal benchmark with the Babelscape model.

5.2 Benchmark Comparisons

Table 6 summarizes performance across the three benchmarks at 95% coverage using the conditional **nc1** method. As anticipated, all models achieved empirical coverage close to the target level. However, performance varied notably depending on each model's training regime. The Jean-Baptiste model outperformed others on both the CoNLL and CoNLL_Red datasets, likely due to its original training on CoNLL. In contrast, the Dslim model underperformed on the CoNLL dataset despite also being trained on it, potentially due to its significantly smaller parameter count relative to the other models. Babelscape, being the only model trained directly on the WikiNEuRal dataset, exhibited superior performance on that benchmark compared to the others. Interestingly, Dslim outperformed Jean-Baptiste on WikiNEuRal, despite its weaker performance on the simpler CoNLL-based tasks. Despite the large differences in set size among the CoNLL and WikiNEuRal benchmarks, there is a relatively small difference in prediction sizes for the reduced CoNLL benchmark. This reduced CoNLL benchmark reduces CoNLL to an entity identification problem without class prediction. Therefore, we believe that most models, when fine-tuned, struggle the most in identifying the new class boundaries but not the span boundaries.

CoNLL	Coverage	Set Size
Babelscape	0.942 ± 0.007	25.19 ± 0.969
Dslim	0.947 ± 0.007	23.021 ± 0.882
Jean	0.948 ± 0.006	4.556 ± 0.38
Tner	0.943 ± 0.004	18.492 ± 0.678
CoNLL_red	Coverage	Set Size
Babelscape	0.941 ± 0.009	4.025 ± 0.096
Dslim	0.951 ± 0.004	2.089 ± 0.015
Jean	0.96 ± 0.008	1.786 ± 0.023
Tner	0.954 ± 0.001	2.557 ± 0.006
WikiNEuRal_en	Coverage	Set Size
Babelscape	0.955 ± 0.003	2.465 ± 0.022
Dslim	0.951 ± 0.004	4.927 ± 0.076
Jean	0.953 ± 0.003	5.676 ± 0.052
Tner	0.95 ± 0.002	6.621 ± 0.134

Table 6: Average coverage and set size for three benchmark datasets across four models at 95% target coverage. Conformal prediction sets are computed utilizing the **nc1** non-conformity score.

6 CONCLUSION

This paper presents a framework for incorporating conformal prediction into existing NER models to enable uncertainty quantification. We evaluated multiple methods across multiple benchmarks, including the CoNLL, CoNLL-Reduced, WikiNEuRal datasets. Additionally, we evaluate the effectiveness of multiple non-conformity scores as applied to sentence-level prediction sets. We further identify the importance of accounting for both the language and length of the input sentence to ensure well-calibrated prediction sets across diverse inputs, demonstrating the use of Mondrian conformal prediction for NLP problems. The methodology can be expanded in multiple ways, most notably by determining how the conformal prediction set output of our methods can be utilized by downstream NLP tasks. Additionally, the stratification of NER may be improved for models with more data and the ability to stratify input sentences by other categories, such as author or source domain.

References

Oshin Agarwal, Yinfei Yang, Byron C Wallace, and Ani Nenkova. Interpretability analysis for named entity recognition to understand system predictions and how they can improve. *Computational Linguistics*, 47(1):117–140, 2021.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018.

Abbas Akkasi and Ekrem Varoğlu. Improving biochemical named entity recognition using pso classifier selection and bayesian combination methods. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(6):1327–1338, 2016.

Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

Divya Bhadauria, Alejandro Sierra-Múnera, and Ralf Krestel. The effects of data quality on named entity recognition. In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 79–88, 2024.

Margarida Campos, António Farinhas, Chrysoula Zerva, Mário AT Figueiredo, and André FT Martins. Conformal prediction for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 12:1497–1516, 2024.

Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370, 2016.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*, 2018.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12 (ARTICLE):2493–2537, 2011.

A Conneau. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

Nguyen Viet Cuong, Nan Ye, Wee Sun Lee, and Hai Leong Chieu. Conditional random field with high-order dependencies for sequence labeling and segmentation. *The Journal of Machine Learning Research*, 15(1):981–1009, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018a. URL `http://arxiv.org/abs/1810.04805`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018b.

Tiffany Ding, Anastasios Angelopoulos, Stephen Bates, Michael Jordan, and Ryan J Tibshirani. Class-conditional conformal prediction with many classes. *Advances in neural information processing systems*, 36:64555–64576, 2023.

Greg Durrett and Dan Klein. Neural crf parsing. *arXiv preprint arXiv:1507.03641*, 2015.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Conformal prediction sets with limited false positives. In *International Conference on Machine Learning*, pages 6514–6532. PMLR, 2022.

Xiuwen Gong, Nitin Bisht, and Guandong Xu. Conformal prediction for partial label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16862–16870, 2025.

Chirag Gupta, Arun K Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022.

James Hammerton. Named entity recognition with long short-term memory. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 172–175, 2003.

Daojun Han, Zemin Wang, Yunsong Li, Xiangbo Ma, and Juntao Zhang. Segmentation-aware relational graph convolutional network with multi-layer crf for nested named entity recognition. *Complex & Intelligent Systems*, 10(6):7893–7905, 2024.

Jianfeng He, Linlin Yu, Shuo Lei, Chang-Tien Lu, and Feng Chen. Uncertainty estimation on sequential labeling via uncertainty transmission. *arXiv preprint arXiv:2311.08726*, 2023.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. Uncertainty in natural language processing: Sources, quantification, and applications. *arXiv preprint arXiv:2306.04459*, 2023.

Zhentao Hu, Wei Hou, and Xianxing Liu. Deep learning for named entity recognition: a survey. *Neural Computing and Applications*, 36(16):8995–9022, 2024.

Junzhe Jiang, Mingyue Cheng, Qi Liu, Zhi Li, and Enhong Chen. Nested named entity recognition from medical texts: an adaptive shared network architecture with attentive crf. In *CAAI International Conference on Artificial Intelligence*, pages 248–259. Springer, 2022.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, 2019.

Shining Liang, Linjun Shou, Jian Pei, Ming Gong, Wanli Zuo, and Daxin Jiang. Calibrenet: Calibration networks for multilingual sequence labeling. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 842–850, 2021.

Mingyi Liu, Zhiying Tu, Tong Zhang, Tonghua Su, Xiaofei Xu, and Zhongjie Wang. Ltp: a new active learning strategy for crf-based named entity recognition. *Neural Processing Letters*, 54(3):2433–2454, 2022a.

Tianyu Liu, Yuchen Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. Autoregressive structured prediction with language models. *arXiv preprint arXiv:2210.14698*, 2022b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

Manolis Maragoudakis, Katia Kermanidis, Aristogiannis Garbis, and Nikos Fakotakis. Dealing with imbalanced data using bayesian techniques. In *LREC*, pages 1045–1050, 2006.

Clara Meister, Tim Vieira, and Ryan Cotterell. If beam search is the answer, what was the question? *arXiv preprint arXiv:2010.02650*, 2020.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Australasian Language Technology Association Workshop*, pages 51–58. Australasian Language Technology Association, 2006.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. Entity-level factual consistency of abstractive text summarization. *arXiv preprint arXiv:2102.09130*, 2021.

Minh-Tien Nguyen, Guido Zuccon, Gianluca Demartini, et al. Loss-based active learning for named entity recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

Binling Nie, Yiming Shao, and Yigang Wang. Improving distantly supervised named entity recognition by emphasizing uncertain examples. *Pattern Analysis and Applications*, 28(1):13, 2025.

Jean Baptiste Polle. Jean-baptiste/roberta-large-ner-english, 2022. URL `https://huggingface.co/Jean-Baptiste/roberta-large-ner-english`.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

Erik Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, 1999.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*, 2025.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561, 2022.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. Wikineural: Combined neural and knowledge-based silver data creation for multilingual ner. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, 2021.

Asahi Ushio and Jose Camacho-Collados. T-NER: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-demos.7. URL `https://aclanthology.org/2021.eacl-demos.7`.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, et al. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252, 2022.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yue Ting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. *ArXiv*, abs/2203.00545, 2022.

Yaqiang Wang, Zhonghua Yu, Li Chen, Yunhui Chen, Yiguang Liu, Xiaoguang Hu, and Yongguang Jiang. Supervised methods for symptom name recognition in free-text clinical records of traditional chinese medicine: an empirical study. *Journal of biomedical informatics*, 47:91–104, 2014.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. Crossweigh: Training named entity tagger from imperfect annotations. *arXiv preprint arXiv:1909.01441*, 2019.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 17, 2011.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23 (170):20, 2013.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*, 2016.

Kang Yang, Zhiwei Yang, Songwei Zhao, Zhejian Yang, Sinuo Zhang, and Hechang Chen. Uncertainty-aware contrastive learning for semi-supervised named entity recognition. *Knowledge-Based Systems*, 296:111762, 2024.

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. Linkner: Linking local named entity recognition models to large language models using uncertainty. In *Proceedings of the ACM Web Conference 2024*, pages 4047–4058, 2024.

Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. *arXiv preprint arXiv:2010.12812*, 2020.

Wenxuan Zhou and Muhao Chen. Learning from noisy labels for entity-centric information extraction. *arXiv preprint arXiv:2104.08656*, 2021.

Xiaofan Zhou, Baiting Chen, Yu Gui, and Lu Cheng. Conformal prediction: A data perspective. *ACM Computing Surveys*, 2025.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/**No**/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes/No/**Not Applicable**]
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/**No**/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/**Not Applicable**]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/**No**/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/**No**/Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/**No**/Not Applicable]
   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

7 Supplement for Section 2: NER models and Conditional Random Fields

7.1 Conditional Random Fields

The model architecture adopted in this work is a variant of the BERT-BiLSTM-CRF framework Tedeschi et al. (2021). It employs a pre-trained multilingual transformer encoder (e.g., BERT Devlin et al. (2018b) or RoBERTa Liu et al. (2019)) to generate contextualized embeddings. These embeddings are then passed through a bidirectional long short-term memory (BiLSTM) network and subsequently into a CRF layer for structured prediction Durrett and Klein (2015). A CRF-based model was selected over more complex alternatives such as encoder-decoder architectures, large language models (LLMs), or other multi-entity frameworks due to its strong performance on state-of-the-art NER tasks Akbik et al. (2018); Wang et al. (2014); Hu et al. (2024) and the interpretability of its decision spaceAgarwal et al. (2021)[1]. The structured output of the CRF makes it particularly well-suited for uncertainty quantification. An overview of the adopted CRF architecture is provided in Figure 2.

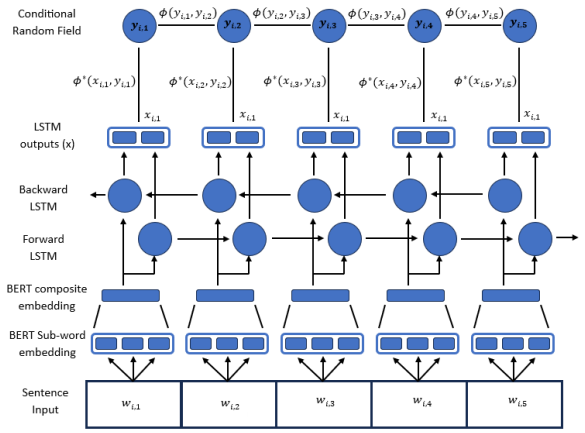

Figure 2: NER BiLSTM-CRF model architecture. $w_{i,1}$, $x_{i,1}$, and $y_{i,1}$ denote the j^{th} word, word-embedding, and predicted label of the i^{th} observation, respectively. CRF transmission and emission feature functions $\phi(y_{j-1}, y_j)$ and $\phi^*(x_j, y_j)$ are defined in Section 7.1.

The CRF decision head depicted in figure 2 is a type of Markov random field (MRF), where each label, conditioned on the input sequence, satisfies the Markov property with respect to its neighboring labels. For the following equations, the index of the observation is irrelevant and therefore the subscript i is dropped from the notation as described in section 2. Following the adapted notation, the conditional independence assumption is expressed as

$$\mathbb{P}(y_j \mid x_j, \{y_k : k \neq j\}) = \mathbb{P}(y_j \mid x_j, y_{j-1}, y_{j+1}), \quad (9)$$

indicating that the probability of label y_j depends only on the numerical representation of the j^{th} word, x_j, and its immediate neighbors y_{j-1} and y_{j+1}. In general, inference on full CRFs is NP-hard and may significantly increase the computational cost of a model Cuong et al. (2014). In order to decrease the computational overhead, in this work, we use a linear-chain CRF with two classes of feature functions that model first-order emission and transition probabilities.

The remainder of this subsection describes the standard linear chain CRF method as it is commonly applied to neural networks. Under the CRF model it is assumed that the probability of a label subsequence $\mathbf{y}_{a:a+b} = (y_a, y_{a+1}, \ldots, y_{a+b})$ given the input sequence of embedded words $\mathbf{x}_{a:a+b}$ may be modeled as

$$\begin{aligned}&\mathbb{P}(\mathbf{y}_{a:a+b} \mid \mathbf{x}_{a-1:a+b+1}; y_{a-1}, y_{a+b+1}) \\ &= \frac{\phi^*(x_{a-1}, y_{a-1}) \prod_{j=a}^{a+b+1} \phi(y_{j-1}, y_j)\phi^*(x_j, y_j)}{\sum_{\mathbf{z}\in\mathcal{L}^m} \phi^*(x_{a-1}, z_{a-1}) \prod_{j=a}^{a+b+1} \phi(y_{j-1}, z_j)\phi^*(x_j, z_j)},\end{aligned} \quad (10)$$

where $\mathcal{L}^b$ is the set of all possible labeling sequences of size b, the model probability described in equation (10) is similar to that of a softmax function, where the numerator is the function that denotes the 'value' of the given subsequence. The denominator of the probability function denotes the sum of the values of all possible subsequences. In this way, the probability of a labeling sequence is equal to the proportion of the sequence's value to all other values. The score functions described in this probability statement are related to the set of transmission feature functions $(\phi(y_{j-1}, y_j))$, which denote how probable a label is given its neighboring labels, and the emission feature functions $(\phi^*(x_{a-1}, y_{a-1}))$, which denote how probable a label is given the related input vector. Each feature function is an exponential family parameterized by a weight vector λ or λ^*, yielding:

$$\phi^*(x_j, y_j) = \exp\left(\sum_{k=1}^{e^*} \lambda_k^* f_k^*(x_j, y_j)\right) \quad (11)$$

[1]The increased 'interpretability' of CRFs is derived from the ability to observe the relationships among the input word embeddings and prediction class labels via the manually specified transmission and emission feature functions.

$$\phi(y_{j-1}, y_j) = \exp\left(\sum_{k=1}^{e} \lambda_k f_k(y_{j-1}, y_j)\right) \tag{12}$$

Functions $f^*(x_j, y_j)$ and $f(y_{j-1}, y_j)$ are any real-valued differentiable functions that take as input either the input vector/label pair or two label/label pairs, respectively. There is no limit to the number of transmission and emission feature functions that may be used to model the CRF, and the number of emission feature functions may differ from the number of transmission functions as denoted by e and e^* in the above equations. When performing NER with a neural network, because the input word embeddings x_j are an output of a BiLSTM, the feature functions used in the linear chain CRF are simplified to single-feature matrices Ma and Hovy (2016). It is expected that feature extraction is performed automatically as a function of the leading neural network. The simplified feature functions may thus be given as.

$$\phi^*(x_j, y_j) = \exp\left(\mathbf{W}^*(y_j, :) \cdot x_j\right), \tag{13}$$

$$\phi(y_{j-1}, y_j) = \exp\left(\mathbf{W}(y_{j-1}, y_j)\right), \tag{14}$$

where $\mathbf{W}^*(y_j, :) \cdot x_j$ depicts the dot product of the input vector x_j and the emission feature matrix $\mathbf{W}^* \in \mathbb{R}^{|\mathcal{L}| \times d}$. Similarly $\mathbf{W}(y_{j-1}, y_j)$ depicts the transmission feature matrix values corresponding by the labels of y_{j-1} and y_j. Utilizing this simplification, equation 10 now becomes

$$\mathbb{P}(\mathbf{y}_{a:a+b} \mid \mathbf{x}; y_{a-1}, y_{a+b+1}) = \frac{s(\mathbf{y}_{a:a+b}, \mathbf{x})}{\sum_{\mathbf{z} \in \mathcal{L}^m} s(\mathbf{z}_{a:a+b}, \mathbf{x})}, \tag{15}$$

where $s(\cdot)$ is a compact way of writing the unnormalized value function for a label sequence:

$$s(\mathbf{z}_{a:a+b}, \mathbf{x}) = \exp\{\mathbf{W}^*(z_{a-1}, :) \cdot x_{a-1} + \sum_{j=i}^{a+b+1} \mathbf{W}(z_{j-1}, z_j) + \mathbf{W}^*(z_j, :) \cdot x_j\}.$$

7.2 CRF Decoding via beam search

As seen in equation (15), the computation of $\mathbb{P}(\mathbf{y}_{a:a+b} \mid \mathbf{x})$ depends on identifying the most probable label sequence, which is itself a non-trivial task due to the exponential size of the label space (e.g., $|\mathcal{L}|^b$ for a sequence of length b). To address this, CRF models employ the beam search decoding algorithm Meister et al. (2020).

Beam search begins at the designated `START` token (y_0) and determines the top-K best transmissions to the next label at time step one (y_1) by maximizing the sum of emission and transmission scores from the feature functions depicted in equations (13) and (14), i.e., $\mathbf{W}^*(y_1) \cdot x_{y_1} + \mathbf{W}(y_0, y_1)$. Once the top-K transmissions have been determined for time step one, the algorithm evaluates the next K most probable continuations at time step two, resulting in up to K^2 sequences. This expanded set is pruned back to the top K sequences, and the process repeats until an `END` token is reached. If K is larger than the size of the CRF's state space, this process will end in the set of the top-K most probable sequences, of which the sequence with the highest score is the output of the NER model. If the value of K is less than the size of the state space, then beam search performs a greedy search that is not guaranteed to find the true most likely output. Because beam search is a greedy algorithm, in larger sequence generation tasks, beam search is known to produce low-diversity outputs. To mitigate this, alternative decoding algorithms such as contrastive beam search and diverse beam search Su et al. (2022); Vijayakumar et al. (2016) have been proposed and shown to improve diversity in downstream tasks such as machine translation and image captioning. In this paper, the K values chosen for the beam search algorithm also determine the maximum size of a prediction set before the set of all possible sequences is returned (guaranteeing 100% accuracy). Because we do not wish to limit the size of our prediction sets to a small number of outputs before returning the full set, we utilize a K value of 100.

8 Supplementary Material for Section 4

8.1 A. Approximation in "Conformal Prediction for NER"

The probability function listed in equation (15) is computationally expensive due to requiring the calculation of the denominator. We find that in NER applications, the score of the k'th most likely sequence quickly approaches zero as k increases. Therefore, it is appropriate to approximate the denominator using only the top-K highest-value sequences. Any sequence that does not belong to the top-K highest-value sequences has its value function essentially set to zero. This approximation is common for supervised machine learning methods, which contain a large output space. Let $\mathbf{y}^{(r)}$ denote the r-th highest value sequence. The approximate probability becomes:

$$\hat{\mathbb{P}}(\mathbf{y}_{a:a+b} \mid \mathbf{x}) \approx \frac{s(\mathbf{y}_{a:a+b})}{\sum_{r=1}^{K} s(\mathbf{y}^{(r)}_{a:a+b})} \tag{16}$$

This approximation allows tractable computation for conformal prediction while maintaining high accuracy in practice. The choice of K value limits the maximum size of the generated prediction set before the complete set of all possible sequences is returned. This, in turn, limits the effective maximum coverage of the returned prediction sets. We find that in our simulation studies, we utilize K=100 as we find that it is able to achieve well above the maximum desired coverage of 99% while still being relatively cheap computationally.

8.2 B. Algorithmic Details (Unconditional Sequence-level CP)

The following algorithm depicts the process of constructing general unconditional prediction sets.

Algorithm 1: Unconditional Conformal Prediction (Sequence-level)

Input: Training/calibration data $\mathcal{I}_{\text{train}}, \mathcal{I}_{\text{cal}}$; test input $(\mathbf{x}_{\text{test}}, \mathbf{y}_{\text{test}})$; confidence level α; non-conformity score nc; beam width k; overshoot function $V(\cdot)$

Output: Prediction set $C(\mathbf{x}_{\text{test}}, \tau)$

Train CRF model M on $\mathcal{I}_{\text{train}}$
Compute $\tau = \hat{Q}_{1-\alpha}(\{nc(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}_{\text{cal}}})$
Predict top-k sequences $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(k)} = M(\mathbf{x}_{\text{test}})$
Initialize $\mathcal{C} \leftarrow \{\}$; $i \leftarrow 1$
while $nc(\mathbf{y}^{(i)} \mid \mathbf{x}_{test}) < \tau$ ***and*** $i \leq k$ **do**
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{y}^{(i)}\}$; $i \leftarrow i + 1$
end
$u \sim \text{Unif}(0, 1)$; compute the overshoot function $V(\cdot)$; **if** $u \leq V(\cdot)$ **then**
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{y}^{(i)}\}$
end
return $\mathcal{C}$

8.3 C. Class-conditional Conformal Prediction for NER

The methodology described in Algorithm 8.2; will produce prediction sets with an average coverage that is the supplied $1-\alpha$ confidence level. However, when applying the conformal prediction technique, it is readily apparent that there are ways in which you may split the dataset such that a given split may be either under-calibrated or over-calibrated. The most apparent method of splitting the data to induce a miss-calibration in a multilingual model is dividing the data up into different language bins.

A key property of the exchangeability requirement for conformal prediction is that mixtures of exchangeable sequences remain exchangeable. This enables calibration across confounding or lurking variables, which may impact the nonconformity score, such as sentence length and language. Figure 3 displays the empirical coverage and desired coverage of the three non-conformity scores **nc1**, **nc2**, and **nc3**. The empirical coverage of a prediction set is the proportion of prediction sets, constructed on a withheld benchmark test partition ($\mathcal{I}_{test}$) that contain the correct response.

$$\text{Empirical Coverage} = \frac{1}{|\mathcal{I}_{test}|} \sum_{i \in \mathcal{I}_{test}} \mathbb{I}(\mathbf{y}_i \in \mathcal{C}(\mathbf{x}_i)) \tag{17}$$

Where $C(\mathbf{x}_i)$ is the conformal prediction set which is being evaluated. Figure 3 shows that the sentence-level conformal prediction methods achieve the desired coverage when aggregated over all inputs regardless of their language or sentence length. This coverage figure was generated for the multilingual WikiNEuRal benchmark dataset with the Babelscape model. For ease of comparison, all future figures displayed in this section rely on the **nc1** non-conformity score.

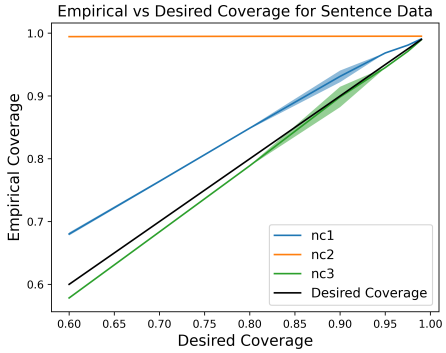

Figure 3: Overall calibration of sentence-level prediction sets for **nc1**, **nc2**, and **nc3** when computed with the Babelscape model on the multilingual WikiNEuRal benchmark dataset.

However, despite good unconditional calibration, conditional miscalibration can occur when the data is stratified by confounding variables such as sequence length and language. Figure 4 illustrates how when unaccounted for, different language groups exhibit systematic over- or under-coverage in the constructed prediction sets. As seen in Figure 4, the Babelscape model, when trained on the Multilingual dataset, and calibrated without language separation, tends to construct prediction sets with poor coverage for the Russian language while producing excessively large prediction sets for the Spanish language.

The language example provided above is one example of what may occur in conformal prediction problems

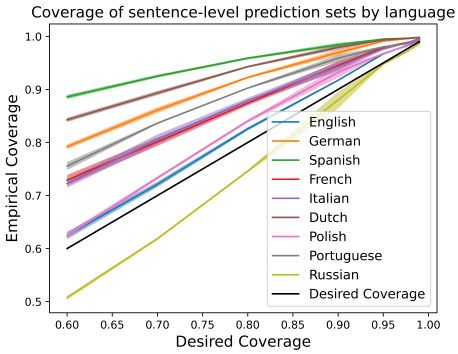

Figure 4: Left: initial calibration per language in multilingual NER. Right: calibration per sentence length bin.

when there are important partitions of the training and test datasets. When constructing prediction sets for a natural language model, it is desirable to be able to make as small a prediction set as possible while maintaining validity. If there are sub-populations of the input data that are easily identifiable with minimal error, then it is valuable to stratify your overall dataset, provided you have a sufficient amount of data. By stratifying the population dataset into varying subsections, you are able to isolate poorly performing sections, such that their bad performance does not increase the average prediction set size. This procedure may also be viewed as isolating well-performing sections, such that you are able to decrease their average prediction set size relative to the population's average.

The limit to the number of strata in which you are able to stratify your data set is based on the number of observations, the ability to accurately group each stratum, and, importantly, the belief that observations belonging to different strata should behave differently from one another. In the case of NER, we identify two important strata in which the model may behave differently, that is, input sentence length and input language.

When splitting our dataset in accordance with sentence length, due to the amount of data available, it is impractical to separate each 'length' in its entirety as the benchmark calibration set becomes too small for rarely occurring sentence lengths. Therefore, in order to ensure an appropriate sample size in our calibration dataset, we group sentence lengths together such that similar length sentences are grouped together. This method of grouping similar class structures is motivated by the work of Ding et alDing et al. (2023) in the clustering of granular classes for regular classification problems. For all benchmark datasets, sentence-length strata are grouped into ranges of 1–10, 11–20, 21–30, 31–40, and 40+ words. In applications, as the amount of data available for calibration increases, the size of the sentence-length groupings may be decreased such that each group is more granular. The original calibration of the Babelscape model on the multilingual WikiNEuRal benchmark for each sentence length grouping is depicted in Figure 5. As this figure shows, without stratification, the baseline NER model constructs prediction sets for long sentences with a higher than desired coverage while constructing invalid prediction sets for sentences above length 21.

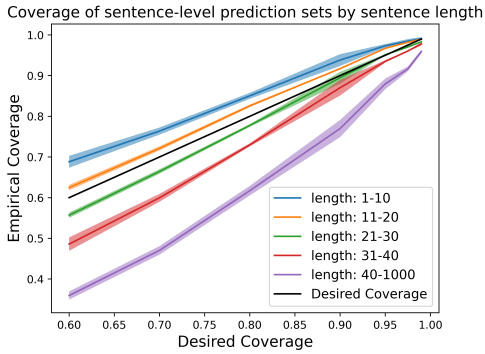

Figure 5: Left: initial calibration per language in multilingual NER. Right: calibration per sentence length bin.

Although Figure 4 and Figure 5 display the missclassification of language and length independently, when accounting for such confounding variables, it is possible to construct prediction sets that account for both language and length at the same time. When doing so, locally adjusted thresholds must be computed for all combinations of sentence length and language. For each length and language bin, we calculate independent cutoff thresholds and modify the conformal prediction set construction equations as follows:

$$L(\mathbf{x}, \tau_{lang,length}) = \min\left\{k : nc(\mathbf{y}^{(k)} \mid \mathbf{x}, lang(\mathbf{x}), length(\mathbf{x})) \geq \tau_{lang,length}\right\},$$

$$\mathcal{C}(\mathbf{x}, \tau, lang, length) = \left\{\mathbf{y}^{(k)} : k \leq L(\mathbf{x}, \tau_{lang,length})\right\}$$

Where $lang(\mathbf{x})$ and $length(\mathbf{x})$ are functions which determine the language and sentence length of input x. Additionally, $\tau_{lang,length}$ is a language and length-specific cutoff for the stratified prediction set. This cutoff is equal to the $(1-\alpha)$ quantile of the calibration subsection that matches the desired language/length bin. Figure 6 shows the recalibrated coverage across

different sentence lengths for the English language. As you can see, unlike Figures 4 and 5, each sentence length grouping is well calibrated for the English subsection of the dataset. Similar figures may also be constructed for each language. We do note that the grouping procedure performed on the sentence length bins does ensure conditional coverage for each language-length bin combination, but does not ensure proper calibration within the bin. For example, the input length 19 may be under-calibrated because its cutoff threshold is based on all sentences from length 11-20, each of which has a slightly different non-conformity score distribution.

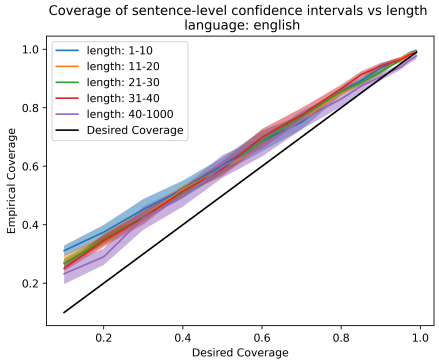

Figure 6: Stratified coverage across sentence lengths for the English language stratum of the multilingual WikiNEuRal benchmark dataset. The coverage depicted is calculated utilizing the conditional non-conformity score with **nc1**.

In order to visualize the difference in distributions between different sentence lengths, we plot a heatmap of p-values obtained from Kolmogorov–Smirnov (KS) tests comparing distributions of **nc1** non-conformity scores between sentence length groups in Figure 7. Lighter cells indicate similar distributions (high p-values), while darker cells indicate statistically significant differences. To perform the Kolmogorov–Smirnov (KS) test, a sample of 100 observations was drawn from each input length group of size 10 to size 30. For each observation, the nonconformity score corresponding to the correct answer was computed. These scores were then used to conduct the KS test, and the resulting p-value was recorded. This entire process was repeated 100 times, and the average p-value across repetitions is presented in the heatmap above. The heatmap illustrates that, for most sentence lengths, an increase of as few as four words can significantly alter the distribution of nonconformity scores. Note: despite some values obtaining small p-values on the heatmap, we do know that each sentence length has a different distribution. The heatmap is meant as an illustrative tool to show how easily the differences in distributions are detected even with a limited sample size and, subsequently, limited power. We are not advising the use of this heatmap as a principled statistical test of equal distributions.

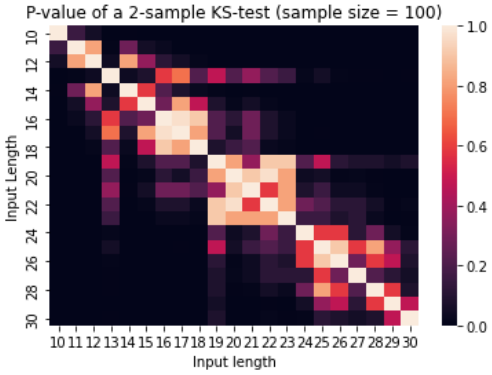

Figure 7: Kolmogorov-Smirnov between multiple 100-sample non-conformity scores from different sentence-length groups within the English language partition of the multilingual WikiNEuRal benchmark.

9 Supplementary materials for Experiments: Details and Results

9.1 Benchmark Datasets

We evaluate our methods on three primary benchmark datasets: CoNLL++ Wang et al. (2019), CoNLL-reduced, and WikiNEuRal Tedeschi et al. (2021).

CoNLL++ is a refined version of the original CoNLL-2003 benchmark, featuring a reduced number of labeling errors. It includes four entity classes: `PER`, `LOC`, `ORG`, and `MISC`. We use the English partition of CoNLL++, which consists of annotations from Reuters newswire articles collected between August 1996 and August 1997.

In application, some NER downstream tasks do not use the class labeling aspect of NER and only utilize the identified entity span as an input; for that reason, we propose CoNLL-Red as a reduced and simplified version of CoNLL++, in which all entity classes are merged into a single 'Entity' category. This benchmark provides a way to evaluate the uncertainty around entity identification without classification. This evaluation metric is vital because, in the slight fine-tuning stage, the underlying models may be more able to adapt to the slightly changed 'entity-identification' task while being less able to adapt to the 'entity classification'. Therefore, the NER output may have more wrong-classification errors than wrong-span errors.

WikiNEuRal is a recent multilingual benchmark built

from Wikipedia articles. Named entities in WikiNEuRal are generated through a silver-standard data creation process and align with Wikipedia's own entity annotations. It adopts the CoNLL class schema and spans nine languages: Dutch, English, French, German, Italian, Polish, Portuguese, Russian, and Spanish. We include WikiNEuRal primarily for its relevance to downstream tasks such as Named Entity Disambiguation and Relationship Extraction. Furthermore, we utilize the multilingual form of WikiNEuRal throughout sections 4-6 to demonstrate various techniques for generating language-specific coverage. When comparing base models and benchmark datasets, we only report the results evaluated on the English partition of the WikiNEuRal dataset. This is because all models outside of Babelscape were not trained on any non-English languages and are unable to tokenize non-English characters.

9.2 Base Models

We evaluate four different base models to see how the selection of an underlying model affects the conformal prediction procedures for each benchmark dataset. All base models were chosen due to their effectiveness at performing NER and their public availability at huggingface.co.

Babelscape Tedeschi et al. (2021) is a multilingual NER model trained on the WikiNEuRal dataset. It was selected to evaluate NER performance in the multilingual context provided by the WikiNEuRal benchmark. Other baseline models were not trained with the tokenizer necessary for multilingual NER and, as such, are not evaluated on this task. As shown in our results, Babelscape is the best-performing NER base model that we evaluated for the WikiNEuRal dataset while being comparable to Jean-Baptiste for CoNLL. For this reason, the Babelscape model is used throughout this paper to demonstrate coverage of various techniques and compare the non-conformity scores of the three conformal prediction methodologies.

Dslim Devlin et al. (2018a) is the smallest model among those considered, containing approximately 110 million parameters, compared to 170M in Babelscape, 350M in Jean-Baptiste, and 350M in TNER. Trained on the CoNLL-2003 benchmark, Dslim achieved a base F1 score of 0.926. It was included to assess how a smaller model performs after fine-tuning, relative to the larger Jean-Baptiste model, which was also trained on CoNLL-2003.

Jean-Baptiste Polle (2022)is the second model in our evaluation trained on the CoNLL-2003 dataset. Unlike the other base models, it leverages a RoBERTa-based transformer for its word embeddings.

TNER Ushio and Camacho-Collados (2021) is a large NER model trained on the OntoNotes dataset Weischedel et al. (2013) and is the only OntoNotes-based baseline included in our evaluation. Ontonotes is a NER benchmark with a significantly different NER ontology compared to Conll++ and WikiNEuRal. Therefore, we selected TNER to illustrate how effectively a model that was trained on out-of-distribution data can be fine-tuned for conformal prediction on the CoNLL dataset.

9.3 Extended Tables

The following two tables present the results from Tables 4 and 5 with standard errors computed from twenty different calibration/test splits.

Sentence-Level		
Desired Coverage	**nc1**	
	Coverage	Size
0.8	0.8488 ± 0.0013	0.9000 ± 0.0015
0.9	0.9312 ± 0.009	1.3526 ± 0.2542
0.95	0.9685 ± 0.0002	1.8092± 0.0156
0.975	0.9809 ± 0.0003	2.1735 ± 0.0011
0.99	0.9908 ± 0.0002	4.5451 ± 0.0165
Desired Coverage	**nc2**	
	Coverage	Size
0.8	0.9950±0.0001	27.5074± 0.2131
0.9	0.9910 ±0.0001	37.45611± 0.2422
0.95	0.9951±0.0001	45.8718±0.2508
0.975	0.9952± 0.0001	52.3022± 0.2342
0.99	0.9952±.0001	59.8476± 0.2262
Desired Coverage	**nc3**	
	Coverage	Size
0.8	**0.7889 ± 0.0007**	0.8443±0.0005
0.9	0.8984 ± 0.0162	0.9611±0.0171
0.95	**0.9452 ± 0.0006**	1.2858±0.0016
0.975	**0.9711 ± 0.0004**	2 ± 0.000
0.99	**0.9895 ± 0.0001**	11 ± 0.000

Table 7: Results from Table 4 with standard errors computed from twenty calibration/test splits

Coverages reported in Table 7 are bolded if the 95% confidence interval formed by the average and standard error does not contain the desired coverage. As we can see, this only occurs with **nc3** and as explained in the main paper, is primarily due to poor estimation of the cutoff threshold (Equation 7) in the ACP procedure.

Similarly to Table 7, Table 8 reports the results of Table 5 with standard errors. Coverages are bolded if the 95% confidence interval formed by the average and reported standard error does not contain the desired coverage. This occurs only for the Naive + **nc2** method, most likely due to a mis-calibration of Equation 7 during Adaptive Conformal Prediction. Despite the miss-

calibration, the empirical coverages are within 2% of the desired level. Additionally, by undershooting the desired coverage, the prediction sets formed by Naive + **nc2** at the 80% and 90% confidence level are smaller than all other methods.

10 Theorems, Propositions, and Proofs

The following subsections describe multiple Theorems, Propositions, and their Proofs related to the coverage claims of our conditional, naive, and class-conditional conformal prediction methods.

10.1 Theory of General Conformal Prediction

The construction of the following proposition follows from Gupta et al. (2022); we simply repeat their proof with our differing notation.

Suppose we observe data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$. Let this dataset be partitioned into two subsections, $\mathcal{I}_{\text{train}}$ and $\mathcal{I}_{\text{cal}}$. A CRF model is trained using $\mathcal{I}_{\text{train}}$. Let $nc(\mathbf{y}|\mathbf{x})$ denote a generic non-conformity score based on the outputs of the trained CRF model. To construct a prediction set based on the above non-conformity score, calculate the $(1-\alpha)$-quantile on the calibration dataset $\mathcal{I}_{\text{cal}}$.

$$\begin{aligned}\tau =& Q_{1-\alpha}\left(\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i\in\mathcal{I}_{\text{cal}}}\right)\\ =& \lceil (1-\alpha)(1+|\mathcal{I}_{\text{cal}}|)\rceil^{th} \text{ largest score in } \mathcal{I}_{\text{cal}} \end{aligned} \tag{18}$$

Next, the prediction set may be constructed using $nc(\cdot, \mathbf{x}_i)$ and τ:

$$C(\mathbf{x}, \tau) = \{\mathbf{y} : \mathbf{nc}(\mathbf{y}|\mathbf{x_i}) \leq \tau\} \tag{19}$$

Proposition 1. *Assuming $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i\in\mathcal{I}_{cal}\cup[n+1]}$ are exchangeable, then the prediction set $C(\cdot)$ in Equation 19 satisfies:*

$$\mathbb{P}(\mathbf{y}_{n+1} \in C(\mathbf{x}_{n+1}, \tau)) \geq 1-\alpha \tag{20}$$

Proof. By the construction of the confidence interval $C(\mathbf{x}, \tau)$:

$$\mathbf{y}_{n+1} \in C(\mathbf{x}_{n+1}, \tau) \text{ iff } \mathbf{nc}(\mathbf{y}|\mathbf{x_i}) \leq \tau$$

Therefore

$$\mathbb{P}(\mathbf{y}_{n+1} \in C(\mathbf{x}_{n+1}, \tau)) = \mathbb{P}(nc(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \leq \tau)$$

The exchangeability of $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i\in\mathcal{I}_{\text{cal}}\cup[n+1]}$, implies that $\{\mathbf{nc}(\mathbf{y}_i|\mathbf{x}_i)\}_{i\in\mathcal{I}_{\text{cal}}\cup[n+1]}$ is also exchangeable. Thus, Lemma 2 of Romano et al. (2019) yields

$$\mathbb{P}(\mathbf{nc}(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \leq \tau) \geq 1-\alpha$$

□

10.2 Theory of stratified NER

Theorem 1. *Let E be the sample space for all possible NER inputs and outputs $(\mathbf{x}, \mathbf{y})$. Consider a partition of E into m mutually exclusive and exhaustive subsets such that $\bigcup_{j=1}^m E_j = E, \mathbb{P}(\bigcup_{j=1}^m E_j) = 1$ and $\forall j \neq k, E_j \cap E_k = \varnothing, \mathbb{P}(E_j \cap E_k) = 0$.*

Then the prediction set formed by the following equations

$$\begin{aligned}\tau_j &= Q_{1-\alpha}\left(\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{(\mathbf{x_i},\mathbf{y_i})\in E_j}\right)\\ C_j(\mathbf{x}, \tau) &= \{\mathbf{y} : \mathbf{nc}(\mathbf{y}|\mathbf{x_i}) \leq \tau_j\}\end{aligned}$$

is well calibrated for observations belonging to each subset such that:

$$\mathbb{P}(\mathbf{y}_{n+1} \in C_j(\mathbf{x}_{n+1}, \tau_j)|(\mathbf{x_{n+1}}, \mathbf{y_{n+1}}) \in E_j) \geq 1-\alpha \tag{21}$$

Proof. Let E be the sample space for all possible NER inputs and outputs $(\mathbf{x}, \mathbf{y})$. Consider a partition of E into m mutually exclusive and exhaustive subsets such that $\bigcup_{j=1}^m E_j = E, \mathbb{P}(\bigcup_{j=1}^m E_j) = 1$ and $\forall j \neq k, E_j \cap E_k = \varnothing, \mathbb{P}(E_j \cap E_k) = 0$.

Now consider the calibration dataset $\mathcal{D}_{cal} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i\in\mathcal{I}_{\text{cal}}}$. Assuming that the calibration data is exchangeable, we may partition the calibration dataset based on the sample space partition $\{E_j\}_{j=1}^m$. Let

$$\mathcal{D}_{cal}^j = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{cal} : (\mathbf{x}_i, \mathbf{y}_i) \in E_j\} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i\in\mathcal{I}_{\text{cal}}^j},$$

where $\mathcal{I}_{\text{cal}}^j$ is the set of indices corresponding to observations belonging to the E_j. Clearly, each partition D_{cal}^j of the calibration data inherits the exchangeable property.

We will now define a conformal prediction set, conditional on the true observation belonging to the partition E_j.

$$\begin{aligned}\tau_j =& Q_{1-\alpha}\left(\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{(\mathbf{x_i},\mathbf{y_i})\in E_j}\right)\\ =& Q_{1-\alpha}\left(\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i\in\mathcal{I}_{cal}^j}\right)\\ =& \lceil (1-\alpha)(1+|\mathcal{I}_{cal}^j|)\rceil^{th} \text{ largest}\\ & \text{score in } \mathcal{I}_{cal}^j\end{aligned}$$

A prediction set may then utilize this cutoff to generate a conditionally-calibrated prediction set:

$$C_j(\mathbf{x}, \tau) = \{\mathbf{y} : \mathbf{nc}(\mathbf{y}|\mathbf{x_i}) \leq \tau_j\}$$

Given the above, let a new observation $(\mathbf{x}_{n+1}, \mathbf{y_{n+1}})$ be exchangeable with the calibration data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in \mathcal{I}_{cal}}$. We must have $(\mathbf{x_{n+1}}, \mathbf{y_{n+1}}) \in E_j$ for some $1 \leq j \leq m$. Then, $\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}^j_{cal} \cup [n+1]}$ is exchangeable. Therefore, via Proposition 1:

$$\mathbb{P}(\mathbf{y}_{n+1} \in C_j(\mathbf{x}_{n+1}, \tau_j) | (\mathbf{x_{n+1}}, \mathbf{y_{n+1}}) \in E_j) \geq 1 - \alpha \tag{22}$$

□

10.3 Theory of naive Prediction Sets Proof

Theorem 2. *let C_{idx} and C_{prob} be conformal prediction sets calibrated at levels $1-\alpha$ and $1-\beta$ then the intersection of these two prediction sets obtains a coverage above $1-\alpha-\beta$:*

$$\mathbb{P}\left(\mathbf{y}_{new} \in C_{idx}(\mathbf{x}) \cap C_{prob}(\mathbf{x})\right) \geq 1 - \alpha - \beta$$

Proof. Let us construct two conformal prediction sets utilizing two distinct non-conformity scores **nc** and $\mathbf{nc}^*$:

$$\begin{aligned}\tau =& Q_{1-\alpha}\left(\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}_{\text{cal}}}\right) \\ =& \lceil (1-\alpha)(1 + |\mathcal{I}_{\text{cal}}|)\rceil^{th} \text{ largest value of } \mathbf{nc} \text{ in } \mathcal{I}_{\text{cal}}\end{aligned}$$

$$\begin{aligned}\tau^* =& Q_{1-\beta}\left(\{\mathbf{nc}^*(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}_{\text{cal}}}\right) \\ =& \lceil (1-\beta)(1 + |\mathcal{I}_{\text{cal}}|)\rceil^{th} \text{ largest value of } \mathbf{nc}^* \text{ in } \mathcal{I}_{\text{cal}}\end{aligned}$$

Prediction sets for each quantile and non-conformity score are constructed via:

$$C(\mathbf{x}, \tau) = \{\mathbf{y} : \mathbf{nc}(\mathbf{y}|\mathbf{x_i}) \leq \tau\}$$

$$C^*(\mathbf{x}, \tau^*) = \{\mathbf{y} : \mathbf{nc}^*(\mathbf{y}|\mathbf{x_i}) \leq \tau^*\}$$

Via Proposition 1:

$$\mathbb{P}(\mathbf{y}_{n+1} \in C(\mathbf{x}_{n+1}, \tau)) \geq 1 - \alpha$$

$$\mathbb{P}(\mathbf{y}_{n+1} \in C^*(\mathbf{x}_{n+1}, \tau^*)) \geq 1 - \beta$$

Following the basic law of probability that $P(A \cap B) \geq P(A) + P(B) - 1$, and then define the events: $A = \{\mathbf{y}_{\text{new}} \in C_{idx}(\mathbf{x})\}$ and $B = \{\mathbf{y}_{\text{new}} \in C_{prob}(\mathbf{x})\}$:

$$\begin{aligned}&\mathbb{P}\big[(A \cap B\big] \\ =&\mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}\big[A \cup B\big] \\ \geq&\mathbb{P}(A) + \mathbb{P}(B) - 1 \geq 1 - \alpha - \beta\end{aligned}$$

□

10.4 Theory of conditional Prediction Sets

Theorem 3. *Let: $C_{idx}(\mathbf{x})$, be a conformal prediction set such that the desired coverage is $1-\alpha$. We may then define a second prediction set ($C_{prob}(\mathbf{x})$)such that:*

$$\mathbb{P}(\mathbf{y}_{new} \in C_{prob}(\mathbf{x}) \mid \mathbf{y}_{new} \in C_{idx}(\mathbf{x})) \geq 1 - \beta$$

We propose that the coverage of the intersection of these two sets is bounded by:

$$\mathbb{P}(\mathbf{y}_{new} \in C_{idx}(\mathbf{x}) \cap C_{prob}(\mathbf{x})) \geq (1-\alpha)(1-\beta)$$

Proof. Let us construct two conformal prediction sets utilizing a baseline non-conformity score and a secondary non-conformity score. The quantile used to calculate the baseline prediction set is as follows:

$$\begin{aligned}\tau =& Q_{1-\alpha}\left(\{\mathbf{nc}(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}_{\text{cal}}}\right) \\ =& \lceil (1-\alpha)(1 + |\mathcal{I}_{\text{cal}}|)\rceil^{th} \text{ largest value of } \mathbf{nc} \text{ in } \mathcal{I}_{\text{cal}}\end{aligned}$$

Now, define the subset of the calibration data below τ as $\mathcal{I}_{cal,\tau} = \{i : \mathbf{nc}(\mathbf{y}_i|\mathbf{x}_i) \leq \tau, i \in \mathcal{I}_{cal}\}$. Now define the secondary quantile as follows:

$$\begin{aligned}\tau^* =& Q_{1-\beta}\left(\{\mathbf{nc}^*(\mathbf{y}_i \mid \mathbf{x}_i)\}_{i \in \mathcal{I}_{cal,\tau}}\right) \\ =& \lceil (1-\beta)(1 + |\mathcal{I}_{cal,\tau}|)\rceil^{th} \text{ largest value of } \mathbf{nc} \text{ in } \mathcal{I}_{cal,\tau}\end{aligned}$$

Prediction sets for each quantile and non-conformity score are constructed via:

$$C(\mathbf{x}, \tau) = \{\mathbf{y} : nc(\mathbf{y}|\mathbf{x_i}) \leq \tau\} \tag{23}$$

$$C^*(\mathbf{x}, \tau^*) = \{\mathbf{y} : nc^*(\mathbf{y}|\mathbf{x_i}) \leq \tau^*\} \tag{24}$$

Following Proposition 1:

$$\mathbb{P}(\mathbf{y}_{n+1} \in C(\mathbf{x}_{n+1}, \tau)) \geq 1 - \alpha$$

Additionally, by construction:

$$\mathbb{P}\left[\mathbf{y}_{n+1} \in C^*(\mathbf{x}_{n+1}, \tau^*) \middle| \mathbf{y}_{n+1} \in C(\mathbf{x}_{n+1}, \tau)\right]$$
$$= \mathbb{P}\left[nc^*(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \leq \tau^* \middle| nc(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \leq \tau)\right]$$

Now, let E be the event space of all possible NER inputs and outputs $(\mathbf{x}, \mathbf{y})$. Let us partition this event space into two regions $\{E_1, E_2\}$, where $E_1 = \{(\mathbf{x}, \mathbf{y}) : \mathbf{nc}(\mathbf{y}|\mathbf{x}) \leq \tau\}$ and $E_2 = \{(\mathbf{x}, \mathbf{y}) : \mathbf{nc}(\mathbf{y}|\mathbf{x}) > \tau\}$. If the new observation $(\mathbf{x_{n+1}}, \mathbf{y_{n+1}}) \in E_1$, then $\{\mathbf{nc}^*(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in \mathcal{I}^1_{\text{cal}} \cup [n+1]} = \{\mathbf{nc}^*(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in \mathcal{I}_{\text{cal},\tau} \cup [n+1]}$ is exchangeable. Therefore, by Theorem 1:

$$\mathbb{P}\left[nc^*(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \leq \tau^* \middle| nc(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \leq \tau\right] \geq 1 - \beta$$

Theorem 3 now follows from the following basic probability result: for any two events A and B,

$$P(A \cap B) = P(A)P(B|A).$$

Where $A = \{\mathbf{y}_{\text{n+1}} \in C(\mathbf{x}, \tau)\}$ and $B = \{\mathbf{y}_{\text{n+1}} \in C^*(\mathbf{x}, \tau^*)\}$. By definition $P(A) \geq (1 - \alpha)$ and $P(B|A) \geq (1 - \beta)$.

Now, Let $C_{\text{idx}}(\mathbf{x}) = C(\mathbf{x}, \tau)$ and $C_{\text{prob}}(\mathbf{x}) = C^*(\mathbf{x}, \tau^*)$. Then it is clear that

$$\mathbb{P}(\mathbf{y}_{\text{new}} \in C_{\text{idx}}(\mathbf{x}) \cap C_{\text{prob}}(\mathbf{x})) \geq (1 - \alpha)(1 - \beta).$$

□

Sentence-Level (NC1)		
Desired Coverage	**Naive + nc1**	
	Coverage	Size
0.8	0.8504±0.0012	0.9101±0.0011
0.9	0.9276±0.0010	1.2972±0.0015
0.95	0.9676±0.0006	1.8308±0.0012
0.975	0.9811±0.0004	2.2526±0.0020
0.99	0.9913±0.0002	5.3971±0.0242
Desired Coverage	**RAPS + nc1**	
	Coverage	Size
0.8	0.8503±0.0012	0.8702±0.0008
0.9	0.9228±0.0010	0.9807±0.0003
0.95	0.9675±0.0006	1.8286±0.0012
0.975	0.9770±0.0005	2.1126±0.0012
0.99	0.9907±0.0002	5.3778±0.0143
Desired Coverage	**Conditional + nc1**	
	Coverage	Size
0.8	0.8504±0.0012	0.9100±0.0011
0.9	0.9274±0.0010	1.2962±0.0015
0.95	0.9675±0.0006	1.8287±0.0012
0.975	0.9811±0.0004	2.2524±0.0020
0.99	0.9913±0.0002	5.3968±0.0217
Sentence-Level (NC2)		
Desired Coverage	**Naive + nc2**	
	Coverage	Size
0.8	**0.7844 ± 0.0010**	0.5105±0.0052
0.9	**0.8885 ± 0.0012**	0.9546±0.0007
0.95	**0.9438 ± 0.0010**	1.3345±0.0016
0.975	0.9742 ± 0.0007	2.5008±0.0017
0.99	0.9903 ± 0.0001	15.0000±0.0000
Desired Coverage	**RAPS + nc2**	
	Coverage	Size
0.8	0.9396±0.0011	1.033±0.0004
0.9	0.9397±0.0011	1.0332±0.0004
0.95	0.9704±0.0006	1.9823±0.0005
0.975	0.9788±0.0005	2.9940±0.0004
0.99	0.9909±0.0001	15.8872±0.0024
Desired Coverage	**Conditional + nc2**	
	Coverage	Size
0.8	0.9307±0.0011	1.0000±0.0001
0.9	0.9307±0.0011	1.000±0.0001
0.95	0.9700±0.0006	1.9741±0.0006
0.975	0.9786±0.0005	3.0000±0.0001
0.99	0.9907±0.001	15.9298±0.0020

Table 8: Results from Table 5 with standard errors computed from twenty calibration/test splits.