EMPIRE: EXPLICIT METHOD FOR DATASET PARTITIONING USING INTEGER PROGRAMMING FOR NAMED ENTITY RECOGNITION TO REDUCE DATA LEAKAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluations of Named Entity Recognition (NER) models are often confounded by data leakage between dataset splits. *Entity leakage* occurs when the same entity appears in both training and test data, and *context leakage* when test sentences share their surrounding context with training sentences; both let models score well through memorization rather than generalization. We introduce EMPIRE (Explicit Method for dataset Partitioning using Integer programming for Entity Recognition), which formulates dataset splitting as an integer linear program. It reduces entity and context leakage, weighted by a single parameter α, while fixing the split sizes and bounding each entity class's share of every split. On 24 NER datasets, EMPIRE matches the entity disjointness of a minimum-cut baseline while preserving class balance, and with α=0 it reduces context leakage more than a similarity-based baseline. Removing entity leakage lowers the test F1 of both a BiLSTM-CRF and a fine-tuned Llama-2-7B, whereas removing context leakage alone does not, indicating that much of the F1 on native splits reflects entities seen during training. The class-balance constraint is essential for this comparison: without it, splits strand entity classes and lower F1 for reasons unrelated to generalization.

1 INTRODUCTION

Named Entity Recognition (NER) underpins many natural language processing (NLP) pipelines, from knowledge graph construction (Zhong et al., 2023) to biomedical text mining (Sang et al., 2018), and recent models report near-human scores on standard benchmarks (He et al., 2023; Marone et al., 2025; OpenAI., 2026). A growing body of work, however, shows that these scores overestimate generalization: test sets often contain entities that also appear in training, and models perform far better on such entities than on unseen ones (Golde et al., 2025; Fu et al., 2020; Elangovan et al., 2021). As models are increasingly applied to new domains (Levy et al., 2017; Xian et al., 2018), benchmarks must measure generalization rather than memorization.

Datasets are usually split at random or by time (Gorman & Bedrick, 2019), which allows *data leakage* (Kaufman et al., 2012) in two forms (Elangovan et al., 2021; Fu et al., 2020). *Entity leakage* occurs when the same entity appears in both training and test data, so a model can recognize it from memory. *Context leakage* occurs when test sentences share their surrounding context with training sentences (e.g., *"admitted to [X] for observation"*), so a model can infer the entity from the pattern alone. Reported F1 correlates strongly with entity leakage (Babl et al., 2025; Golde et al., 2025), suggesting that much of it reflects memorization. Leakage is especially likely in domains with a limited set of entities, such as biomedicine, where the roughly 20,000 human protein-coding genes recur across studies.

Prior work has identified and quantified these forms of leakage (Augenstein et al., 2017; Fu et al., 2020; Golde et al., 2025), but methods for building splits that avoid them remain limited. MinCut (Babl et al., 2025) separates entities across splits but ignores context similarity and class proportions, and similarity-based splitting such as DataSAIL (Joeres et al., 2025) assumes one label per sample, which does not hold for NER sentences with multiple entities. No existing method jointly controls entity leakage, context leakage, and the class distribution of each split. Without the last, a split can reduce leakage only by leaving some entity classes with little or no training or test data.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

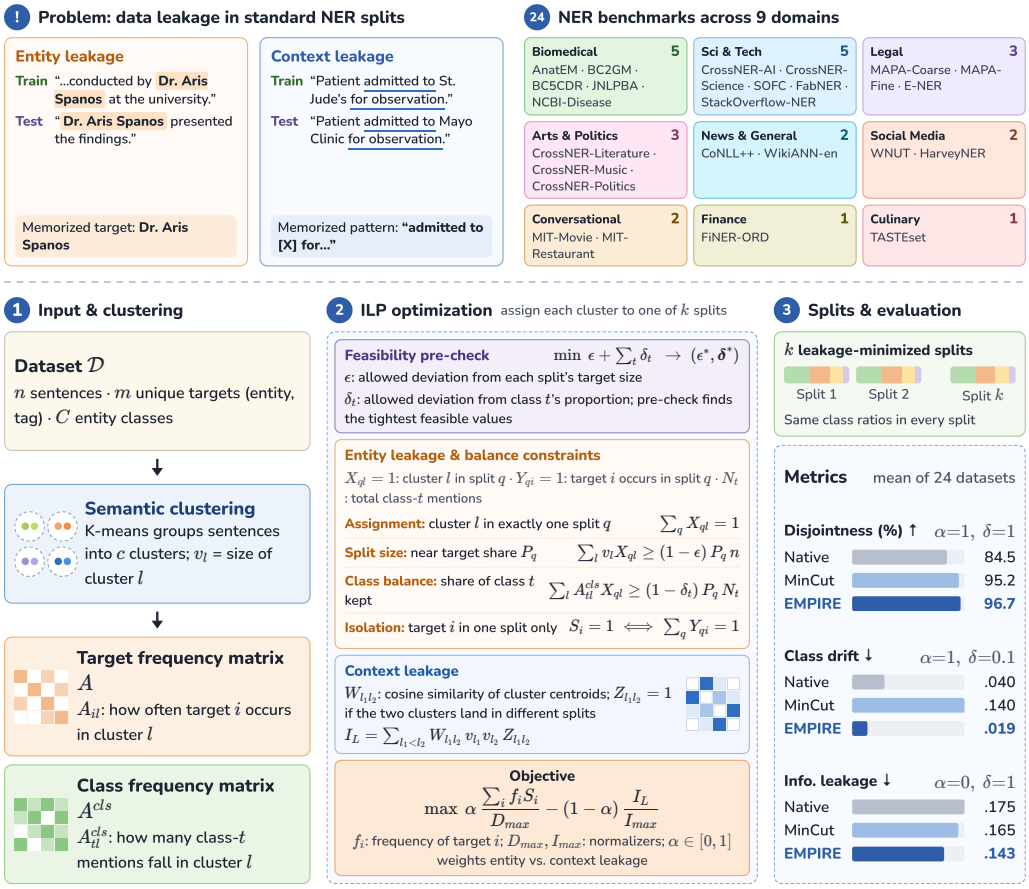

Figure 1: **Overview of the EMPIRE dataset partitioning framework. Top:** Standard NER splits suffer from *entity leakage* and *context leakage*. We evaluate leakage across 24 NER benchmarks spanning 9 domains. **Bottom:** The three-stage EMPIRE pipeline: **(1) Input & Clustering:** Sentences are grouped via K-means to construct target and class frequency matrices. **(2) ILP Optimization:** Cluster assignments are constrained by pre-checked feasibility limits (ϵ^*, δ^*), maximizing entity disjointness and minimizing context similarity, weighted by α, subject to split proportions and class balance. **(3) Splits & Evaluation:** Depending on α, the resulting splits reduce entity or context leakage relative to the native splits while preserving class proportions.

We propose **EMPIRE** (Explicit Method for dataset Partitioning using Integer programming for Entity Recognition), which formulates dataset splitting as an integer linear program (ILP). A single weight α sets the balance between entity disjointness and context dissimilarity, while constraints fix the split sizes and bound each class's share of every split. We compare EMPIRE with the native splits, MinCut, and a similarity-based baseline on 24 NER datasets from diverse domains (Figure 1), using four split-level metrics (entity disjointness, entity coverage ratio, Hellinger distance, and information leakage) and the test F1 of a BiLSTM-CRF (Huang et al., 2015; Lample et al., 2016) and a fine-tuned Llama-2-7B (Touvron et al., 2023).

Our contributions are:

1. EMPIRE, a customizable ILP formulation for splitting NER datasets that controls entity leakage, context leakage, split sizes, and class balance, with a single parameter α that selects which form of leakage to remove.

2. An evaluation on 24 NER datasets against native splits and two baselines, showing that EMPIRE matches MinCut's entity disjointness while preserving class balance, and that at fixed split sizes full

disjointness is feasible only when the largest group of sentences linked by shared entities fits within one split.

3. Evidence from both BiLSTM-CRF and fine-tuned Llama-2-7B that removing entity leakage lowers test F1 while removing context leakage alone does not, and, for the BiLSTM-CRF across all 24 datasets, that without a class-balance constraint, splits strand entity classes and lower F1 for reasons unrelated to generalization.

2 Related Work

The generalization of NER models to unseen entities has been widely examined (Augenstein et al., 2017; Bernier-Colborne & Langlais, 2020; Lin et al., 2021; Tu & Lignos, 2021; Fu et al., 2020; Lee et al., 2022; Golde et al., 2025). Golde et al. (2025), for instance, show a log-linear relationship between F1 and how often a test entity occurs in the training split. These studies quantify the effect of leakage but do not propose a way to construct splits; most approaches to evaluating generalization instead augment the training or test data with unseen entities or perturbations (Lin et al., 2021; Morris et al., 2020).

Closest to our work, Babl et al. (2025) show that random splits inflate NER performance through entity leakage, and propose splitting with a minimum graph cut (Karypis & Kumar, 1998) to reduce entity overlap between training and test. Their method does not consider context similarity or class proportions, and it can deviate from the target split sizes to reach full separation (Section 5). DataSAIL (Joeres et al., 2025) partitions data by similarity to evaluate generalization beyond the training distribution. However, it assumes one global label per sample (e.g., a property of a molecule), which does not fit NER sentences that contain several entities. To our knowledge, EMPIRE is the first splitting method that jointly controls entity leakage, context leakage, and class proportions.

3 Methodology

We cast dataset splitting as an ILP that partitions a dataset into k splits while preserving a predefined split ratio and the semantic class proportions (overview in Figure 1).

Let $\mathcal{D}$ be a dataset of n instances (e.g., sentences or abstracts) to be partitioned into k disjoint splits indexed by $q \in \{1, \ldots, k\}$, and let $\mathcal{E}$ be its set of $m = |\mathcal{E}|$ unique *targets*, i.e., (entity, tag) pairs such as ("Dr. Aris Spanos", PERSON), indexed by $i \in \{1, \ldots, m\}$. Entity surface forms are stemmed with NLTK's English `SnowballStemmer` before building $\mathcal{E}$, so morphological variants such as *transcription factor(s)* and *transcriptional factor(s)* map to the same target.

Let $\mathcal{C}$ be the set of $C = |\mathcal{C}|$ semantic classes (entity types), indexed by $t \in \{1, \ldots, C\}$. To keep pairwise similarity computations for context leakage tractable, instances are grouped into K clusters of semantically similar sentences, indexed by $l \in \{1, \ldots, K\}$, where cluster l holds v_l sentences (Appendices A.3 and A.5). Two count matrices summarize the data: $A \in \mathbb{N}^{m \times K}$, where A_{il} is the number of occurrences of target i in cluster l, and $A^{cls} \in \mathbb{N}^{C \times K}$, where A^{cls}_{tl} is the number of occurrences of class t in cluster l. Each split q has a target ratio $P_q \in (0, 1)$ with $\sum_{q=1}^{k} P_q = 1$ (e.g., 0.8/0.1/0.1 for train/validation/test). Deviations from these ratios are bounded by a tolerance $\epsilon \in [0, 1]$, and deviations from the global class proportions by per-class tolerances $\delta_t \in [0, 1]$, $t \in \{1, \ldots, C\}$.

3.1 Decision Variables

The ILP uses four sets of binary variables: 1) $X_{ql} = 1$ if cluster l is assigned to split q; 2) $Y_{qi} = 1$ if target i occurs in split q; 3) $S_i = 1$ if target i occurs in exactly one split (i.e., it is disjoint); and 4) $Z_{l_1,l_2} = 1$ if clusters l_1 and l_2 are assigned to different splits.

3.2 Entity Leakage Constraints

1. Cluster Assignment Constraint: Each cluster is assigned to exactly one split:

$$\sum_{q=1}^{k} X_{ql} = 1 \quad \forall l \in \{1, \ldots, K\} \tag{1}$$

2. Entity Presence Projection (Big-M Method): A big-M constraint links the cluster assignments to the target indicators Y_{qi}, with $M = \max_i \sum_{l=1}^{K} A_{il} + 1$:

$$\sum_{l=1}^{K} A_{il} X_{ql} \leq M \cdot Y_{qi} \ \forall q \in \{1, \ldots, k\}, \quad \forall i \in \{1, \ldots, m\} \tag{2}$$

This forces $Y_{qi} = 1$ whenever target i occurs in split q. The converse is not enforced: a spurious $Y_{qi} = 1$ can only prevent $S_i = 1$ and never improves the objective, so it leaves the optimum unchanged, and omitting the converse constraint halves the number of presence constraints.

3. Cardinality Bounds Constraint: Each split must receive at least a $(1 - \epsilon)$ fraction of its target size; because the total n is fixed, this also caps every split at $(P_q + \epsilon(1 - P_q))\, n$, so no upper-bound constraint is needed:

$$\sum_{l=1}^{K} v_l X_{ql} \geq (1 - \epsilon) \cdot P_q \cdot n \quad \forall q \in \{1, \ldots, k\} \tag{3}$$

4. Class Distribution Equilibrium Constraint: Each split must also receive at least a $(1 - \delta_t)$ fraction of its proportional share of every class t, keeping class proportions close to those of the full dataset:

$$\sum_{l=1}^{K} A^{cls}_{tl} X_{ql} \geq (1 - \delta_t) \cdot P_q \cdot N^{cls}_t, \ \forall q \in \{1, \ldots, k\}, \quad \forall t \in \{1, \ldots, C\} \tag{4}$$

where $N^{cls}_t = \sum_{l=1}^{K} A^{cls}_{tl}$ is the total count of class t.

5. Exactly-One Disjoint Isolation: Rather than comparing every pair of splits, we count the splits that contain each target; S_i can equal 1 only if target i occurs in exactly one split:

$$\sum_{q=1}^{k} Y_{qi} \leq 1 + (1 - S_i) \cdot M \quad \forall i \in \{1, \ldots, m\} \tag{5}$$

If target i occurs in two or more splits, the constraint forces $S_i = 0$; if it occurs in exactly one, S_i is unconstrained and, for $\alpha > 0$, the disjointness term of the objective sets it to 1. No explicit pairwise (XOR) constraints between splits are therefore needed.

3.3 Similarity Based Constraints

To limit context leakage, similar clusters should be kept in the same split. Let $W \in [0, 1]^{K \times K}$ be the pairwise cluster similarity, $W_{l_1, l_2} = \frac{1}{2}\left(1 + \cos(\bar{u}_{l_1}, \bar{u}_{l_2})\right)$ with $W_{ll} = 0$, where $\bar{u}_l$ is the L_2-normalized centroid embedding of cluster l. The indicators Z_{l_1, l_2} are tied to the assignments by:

$$X_{q, l_1} - X_{q, l_2} \leq Z_{l_1, l_2}, \quad \forall q \in \{1, \ldots, k\} \tag{6}$$

$$Z_{l_1, l_2} \leq 2 - X_{q, l_1} - X_{q, l_2}, \quad \forall q \in \{1, \ldots, k\} \tag{7}$$

Eq. (6) forces $Z_{l_1, l_2} = 1$ when the two clusters lie in different splits, and Eq. (7) forces $Z_{l_1, l_2} = 0$ when they share one; both hold for every cluster pair $l_1 < l_2$.

The cluster-level information leakage I_L sums the similarity of every cluster pair placed in different splits, weighted by both cluster sizes (Joeres et al., 2025):

$$I_L = \sum_{l_1=1}^{K-1} \sum_{l_2=l_1+1}^{K} (W_{l_1, l_2} \cdot v_{l_1} \cdot v_{l_2}) Z_{l_1, l_2} \tag{8}$$

Minimizing I_L keeps contextually similar sentences in the same split, reducing context leakage.

3.4 Objective Function and Normalization

A parameter $\alpha \in [0, 1]$ trades off entity disjointness against information leakage. Each target's contribution to disjointness is weighted by its mention frequency $f_i = \sum_{l=1}^{K} A_{il}$, so frequent entities, which leak most when shared, count more. The two terms live on different scales, so each is normalized to $[0, 1]$ by its maximum: $D_{max} = \sum_{i=1}^{m} f_i$ (every target disjoint) and $I_{max} = \sum_{l_1<l_2} W_{l_1,l_2} v_{l_1} v_{l_2}$ (every cluster pair separated). The objective is

$$\max \left(\alpha \cdot \frac{\sum_{i=1}^{m} f_i \cdot S_i}{D_{max}} - (1 - \alpha) \cdot \frac{I_L}{I_{max}} \right) \tag{9}$$

Normalization makes α a direct interpolation weight; Appendix A.4 works through an example.

4 Experimental Setup

4.1 Datasets

We use 24 NER datasets spanning biomedical, legal, financial, social-media, and other specialized domains: JNLPBA (Collier et al., 2004), BC5CDR (Li et al., 2016), bc2gm (Smith et al., 2008), ncbi_disease (Doğan et al., 2014), AnatEM (Pyysalo & Ananiadou, 2014), conllpp (Wang et al., 2019), wnut (Derczynski et al., 2017), the five CrossNER domains (Liu et al., 2021), MAPA (coarse and fine) (Arranz et al., 2022), E-NER (Au et al., 2022), FiNER-ORD (Shah et al., 2023), HarveyNER (Chen et al., 2022), wikiann_en Pan et al. (2017), FabNER (Kumar & Starly, 2022), mit-movie, mit-restaurant Liu et al. (2013), SOFC Friedrich et al. (2020), StackOverflow-NER (Tabassum et al., 2020), and TASTEset (Wróblewska et al., 2022). They range from 1 to 25 entity classes and 700 to 40,000 instances; Table 1 gives details and license information in Appendix A.1.

4.2 Baselines

We compare against three baselines: **Native**, the split as provided in the respective dataset source; **MinCut** (Babl et al., 2025); and **TopSim**, a simple similarity-aware, greedy baseline we implemented (refer to Appendix A.6 for more details).

4.3 EMPIRE Configurations

We evaluate EMPIRE with three settings of α: α=1 (entity disjointness), α=0 (context leakage), and α=0.5 (both). For α=1 the similarity term is inactive, so EMPIRE assigns individual sentences, i.e., no clustering (K=0); for $\alpha < 1$ we use K=50 clusters over entity-masked embeddings. Each setting is run with two class-balance tolerances: δ^*, the tightest feasible tolerance with lower bound 0.1 found by the feasibility pre-check (Appendix A.2), and δ=1, which removes the class-balance constraint, as in MinCut. We set ϵ=0 throughout, so every split matches the target ratio exactly.

4.4 Embedding Model

For $\alpha < 1$, W is built from L_2-normalized all-MiniLM-L6-v2 (Wang et al., 2020) sentence embeddings computed after entity masking: every entity mention is replaced by the same placeholder token, regardless of its class or length (e.g., the CoNLL sentence *"EU rejects German call to boycott British lamb"* becomes "`entmask` *rejects* `entmask` *call to boycott* `entmask` *lamb"*). The embeddings thus encode context rather than entity identity or type, so sentences with different entities but similar context are recognized as similar. The same masked embeddings are used for the information-leakage metric (Section 4.5).

4.5 Evaluation Metrics

We assess each split with four metrics.

Entity Disjointness Score The percentage of unique targets that occur in exactly one split, computed from the final assignment of every method, where $S_i = 1$ if target i occurs in exactly one split;

targets are matched by exact surface form and class:

$$\text{Disjoint Score (D.S.)} = \left(\frac{1}{m}\sum_{i=1}^{m} S_i\right) \times 100\% \tag{10}$$

Entity Coverage Ratio (ECR) Following Fu et al. (2020), ECR measures how often entity mentions in one split also occur, with the same label, in the others. Because we produce k splits rather than a fixed train/test pair, we compute it leave-one-out: each split q is held out in turn and compared with the union of the rest. Let $L(e)_q^y$ be the number of mentions of entity e with label y in split q, and $L(e)_{\setminus q}^y$ its count in all other splits:

$$\text{ECR} = \frac{\sum_{q=1}^{k} \sum_{e \in E_q} \sum_{y} \frac{L(e)_{\setminus q}^y}{L(e)_{\setminus q}} \times L(e)_q^y}{\sum_{q=1}^{k} \sum_{e \in E_q} L(e)_q} \tag{11}$$

where E_q is the set of unique entities in split q, and $L(e)_q$ and $L(e)_{\setminus q}$ sum over labels. ECR is reported as a percentage.

Hellinger's Distance Measures class-distribution drift between each split and the full dataset. Let p_t be the share of entity mentions of class t in the full dataset and $p_t^{(q)}$ its share in split q; the mean Hellinger distance over the k splits is

$$\bar{H} = \frac{1}{k}\sum_{q=1}^{k} \frac{1}{\sqrt{2}} \sqrt{\sum_{t=1}^{C} \left(\sqrt{p_t} - \sqrt{p_t^{(q)}}\right)^2}. \tag{12}$$

Lower values indicate that the splits better preserve the dataset's class proportions.

Information Leakage (I.L.) The mean cosine similarity over all sentence pairs placed in different splits. With S_{cross} the set of such pairs and u_a the L_2-normalized (entity-masked) embedding of sentence s_a,

$$\text{I.L.} = \frac{1}{|S_{\text{cross}}|} \sum_{(s_a, s_b) \in S_{\text{cross}}} u_a \cdot u_b \tag{13}$$

Unlike the cluster-level I_L that EMPIRE optimizes, which uses rescaled centroid similarities, I.L. is computed over individual sentences with raw cosine similarities.

4.6 Downstream Evaluation

The metrics above characterize the splits themselves, not how models trained on them behave. We therefore train a BiLSTM-CRF (Huang et al., 2015; Lample et al., 2016) from scratch on each split, and fine-tune Llama-2-7B (Touvron et al., 2023) on five of the datasets, reporting test F1 (training details in Appendix A.8). We expect splits with less leakage to yield lower but more trustworthy F1, since the model can no longer rely on memorized entities or contexts.

5 Results

5.1 α Controls Which Leakage Is Removed

The two objective terms act on different kinds of leakage. Figure 2 reports each configuration's paired difference from Native across the 24 datasets. With α=1, EMPIRE raises entity disjointness by 10.1 percentage points at δ^* and 12.2 at δ=1, and lowers ECR by 29.9 and 39.2 points (all $p < 0.001$), but at δ^* it leaves information leakage unchanged. With α=0, EMPIRE lowers information leakage by 0.020 at δ^* and 0.062 at δ=1 ($p < 0.001$), with no significant change in entity disjointness. α=0.5 falls between the two on both metrics (Section 5.2). The choice of α thus determines which form of leakage a split removes. At δ^*, all three settings also keep the Hellinger distance at or below

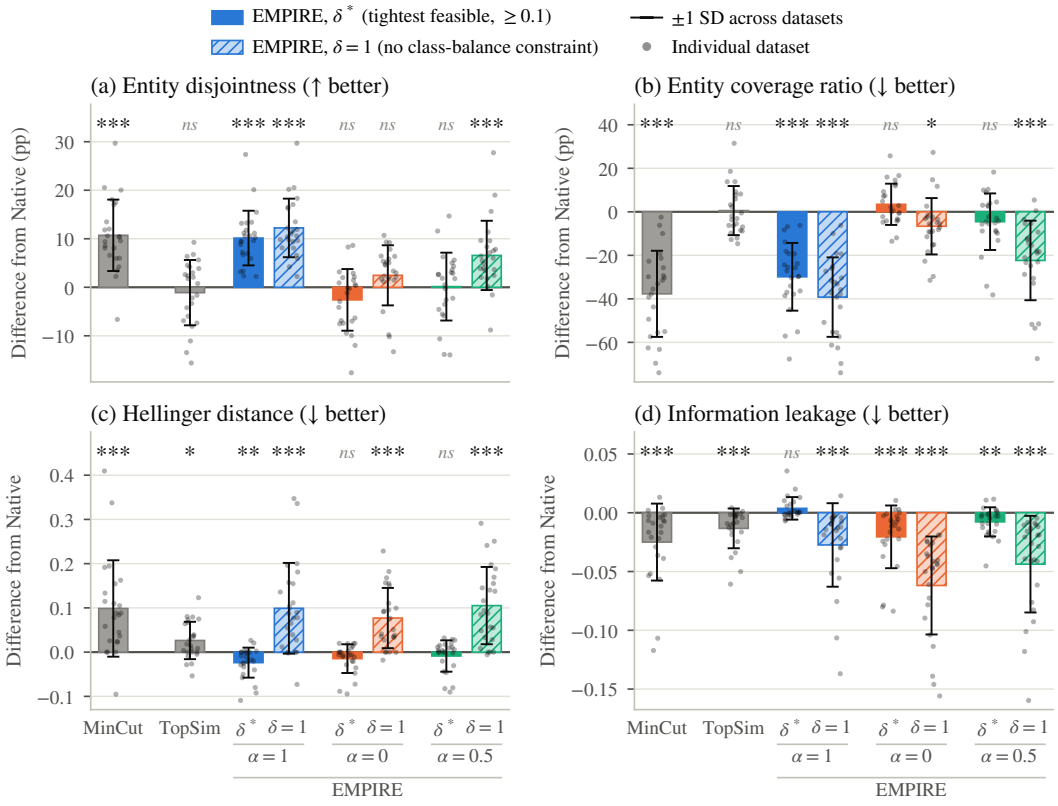

Figure 2: **Split-quality summary across 24 NER datasets.** Each bar is the mean paired difference from the Native split (configuration − Native); error bars show ±1 standard deviation across datasets and dots show individual datasets. Entity disjointness and entity coverage ratio (ECR) are in percentage points. Significance: two-sided Wilcoxon signed-rank test of the paired differences, Holm-corrected across the eight configurations within each metric (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, ns: not significant). δ^* is the tightest feasible per-class tolerance with lower bound 0.1; δ=1 removes the class-balance constraint. For, EMPIRE with α=1 operates on individual sentences (K=0); $\alpha < 1$ uses K=50 clusters. In all cases, $\epsilon = 0$. Per-dataset values are in Tables 3–6.

Native, so this control does not cost class balance. MinCut, which also targets entity leakage, matches EMPIRE (α=1, δ^*) in entity disjointness (+10.7 vs. +10.1 points) but not in class balance: it raises the Hellinger distance by 0.099 over Native ($p < 0.001$), whereas EMPIRE lowers it by 0.024 ($p < 0.01$). Part of MinCut's disjointness also comes from relaxing the split sizes. On AnatEM, for instance, it reaches 100% by enlarging the training split from 49.6% to 51.1% of the data (Table 2), exactly the size of the largest group of sentences linked by shared entities; EMPIRE with K=0 and ϵ=0 cannot do this. TopSim, the context-aware baseline, lowers information leakage by only 0.013 ($p < 0.001$), against 0.020 and 0.062 for EMPIRE (α=0) at δ^* and δ=1, and it raises the Hellinger distance by 0.026 ($p < 0.05$).

Where EMPIRE (α=1) falls short of full disjointness, the cause is mostly the data rather than the solver. A split with no shared entities must keep every group of sentences linked by shared entities within a single split, so at ϵ=0 full disjointness is possible only if the largest such group fits in the largest split (Table 10). This holds for 12 of the 24 datasets, and on all 12 EMPIRE (α=1, δ=1) reaches 100% disjointness with a proven-optimal solution (Table 9). On the other 12 it averages 93.4%. Seven of these still exceed 98% with MIP gaps below 4%. The lowest values occur on FabNER, TASTEset, and bc5cdr (80.1–84.1%), where the largest group covers 99.8%, 100%, and 70% of the sentences and the solver stops with gaps of 61–186%. For α=0, the reduction in information leakage grows with corpus size. It is largest on the larger corpora (up to 0.156 on jnlpba) and smallest on the five CrossNER domains (0.019–0.027), which have fewer than 1,400 sentences each.

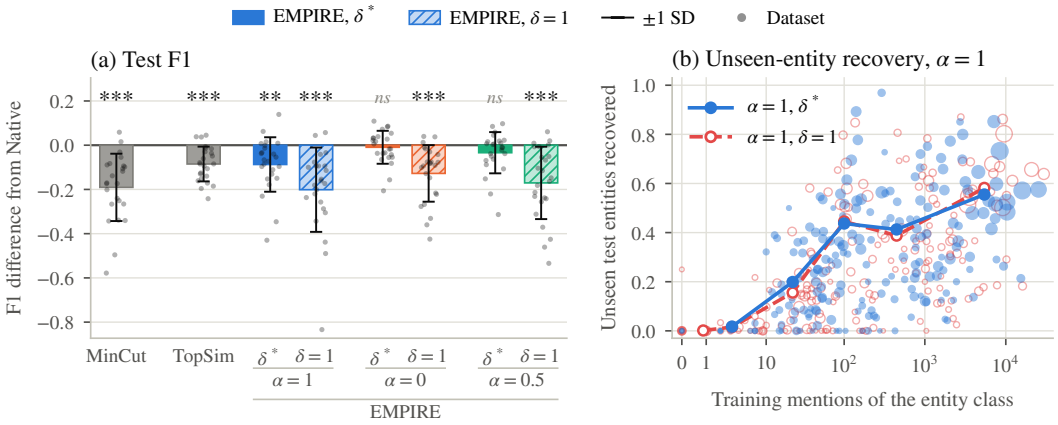

Figure 3: **Downstream evaluation with a BiLSTM-CRF. (a)** Test F1 as the mean paired difference from the Native split across 24 datasets ($\Delta F1$ = $F1$ configuration - $F1$ native); error bars show ± 1 standard deviation and dots show individual datasets. Significance: two-sided Wilcoxon signed-rank test, Holm-corrected across the eight configurations. **(b)** Test entities which does not occur in the training split, for EMPIRE with α=1, against the number of training mentions of their class. Each dot is one entity class in one dataset (area proportional to its number of unseen test entities); lines are span-weighted recovery rates per training-support band (0, 1–9, 10–49, 50–199, 200–999, 1000+; bands with fewer than 30 entities omitted). At δ=1, 5.8% of unseen test entities belong to classes with fewer than 10 training mentions, against 0.1% at δ^*. δ^* is the tightest feasible per-class tolerance with lower bound 0.1; δ=1 removes the class-balance constraint. Training details are in Appendix A.8.

5.2 Combining Both Objectives (α=0.5)

Setting α=0.5 interpolates between the two objectives rather than improving on both. At δ^*, its entity disjointness lies between those of α=0 and α=1 on 22 of the 24 datasets, and its information leakage on 17. Per dataset, it retains a median of 20% of the disjointness gain of α=1 over α=0, and 52% of the leakage reduction of α=0 over α=1 (28% and 38% at δ=1). Part of its shortfall in disjointness likely reflects clustering rather than the weight itself: α=0.5 assigns K=50 clusters, whereas α=1 assigns individual sentences. Its downstream F1 likewise falls between the two (Section 5.3).

5.3 Effect on Model Performance

Figure 3a reports BiLSTM-CRF test F1 as the paired difference from Native. Removing entity leakage lowers F1: EMPIRE (α=1, δ^*) scores 0.087 below Native on average and lower on 19 of the 24 datasets ($p < 0.01$), and MinCut scores 0.191 below ($p < 0.001$). Removing context leakage alone does not. EMPIRE (α=0, δ^*) stays within 0.009 of Native and scores lower on only 12 datasets (not significant), and α=0.5 falls in between (-0.034, not significant). Much of the F1 measured on native splits therefore reflects entities seen during training, which a model can recognize without generalizing. TopSim also lowers F1 (-0.085, $p < 0.001$), although its splits are no more entity-disjoint than Native. Because it places in test the sentences whose nearest neighbor is least similar, it removes near-duplicate contexts. This is a form of context leakage that information leakage, averaged over all cross-split pairs, registers only weakly. The larger drops of all δ=1 configurations (-0.13 to -0.20) are examined in Section 5.4. These trends are not specific to the BiLSTM-CRF. We also fine-tune Llama-2-7B on five of the datasets (SOFC, mit-restaurant, StackOverflow-NER, conllpp, and jnlpba) using the same splits (Appendix A.8). Its F1 differences from Native closely track those of the BiLSTM-CRF (Pearson $r = 0.85$ over 30 dataset–configuration pairs; Figure 4). EMPIRE (α=1, δ^*) again lowers F1 on all five datasets (-0.115 on average), while EMPIRE (α=0, δ^*) does not ($+0.014$). With five datasets we do not test these differences for significance.

5.4 Why the Class-Balance Constraint Matters

Removing the class-balance constraint lowers F1 much further. Every $\delta{=}1$ configuration scores 0.13–0.20 below Native (Figure 3a), and for each α the δ^* split scores higher than the $\delta{=}1$ split on 20–21 of the 24 datasets (by 0.11–0.14 on average, $p < 0.001$). Part of this drop is expected, since $\delta{=}1$ also removes more leakage (2.1–6.4 more points of entity disjointness). But the constraint also governs how entity classes are distributed across splits. Without it, the solver is free to concentrate a class's mentions in one split: 20–32 entity classes are missing from the training, validation, or test split under $\delta{=}1$, against 5–6 under δ^* and 14 under Native, and 2–8 datasets are left with fewer than 200 gold test entities (Figure 5). Figure 3b isolates the effect on test entities that never occur in training, for EMPIRE ($\alpha{=}1$). For classes with similar amounts of training data, such entities are recovered at the same rate under both δ values. What differs is how many of them belong to classes with almost no training data: 5.8% have fewer than 10 training mentions of their class at $\delta{=}1$, against 0.1% at δ^* (see also Figure 6). Applying the $\delta{=}1$ recovery rate within each band of training support to the δ^* distribution fully accounts for the gap in their recovery. A lower score under $\delta{=}1$ thus partly reflects classes the split starved of training data rather than harder generalization, and δ^* avoids this, at a cost in leakage that is small for $\alpha{=}1$ but larger for $\alpha{=}0$ (Section 5.5).

5.5 Ablations

Number of clusters: Increasing K from 50 to 200 (EMPIRE $\alpha{=}0$, $\delta{=}1$, all 24 datasets) leaves mean entity disjointness (86.9–87.6%) and information leakage (0.331–0.334) within the variation between repeated solves of the same configuration (up to 0.92 points and 0.017). It does, however, raise the median MIP gap from 5.0% to 11.3%, and the number of solves stopped by the time limit from 12 to 23 of 24 (Figure 7). Finer clustering adds $O(K^2)$ pairwise variables that the solver cannot exploit within the time limit, so $K{=}50$ suffices.

Embedding model: Generating the splits with the larger Qwen3-Embedding-0.6B Zhang et al. (2025b) or Jasper-Token-Compression-600M Zhang et al. (2025a) instead of all-MiniLM-L6-v2 (EMPIRE $\alpha{=}0$, $K{=}50$, five datasets) does not lower information leakage when the leakage of all three is measured on MiniLM embeddings (Table 11). On four datasets it changes by at most 0.02, and on jnlpba Jasper raises it from 0.24 to 0.43 ($\delta{=}1$). The encoders differ mainly in how widely they spread sentence similarities (Figure 8). MiniLM spreads them most and Jasper least on all five datasets, and at $\delta{=}1$ a narrower spread comes with a larger MIP gap on four of the five. Scored in each encoder's own space, the same splits would show up to 0.51 more information leakage, which reflects that encoder's similarity scale rather than the split.

Class-balance floor: Lowering the floor from $\delta{=}1$ to 0.1 (five datasets; Figure 9) costs EMPIRE ($\alpha{=}1$) at most 3.2 points of entity disjointness. For $\alpha{=}0$ the cost varies by dataset. On StackOverflow-NER and conllpp, disjointness falls by 15.3 and 12.8 points and information leakage rises by up to 0.14, while on SOFC and mit-restaurant the floor barely matters. Tighter floors also make the problem harder: for $\alpha{=}1$ on conllpp, the MIP gap grows from 0% to 27%.

6 Conclusion and Future Work

We presented EMPIRE, an ILP formulation that partitions NER datasets while controlling entity leakage, context leakage, split sizes, and class balance. Across 24 datasets, the weight α determines which kind of leakage a split removes: shared entities ($\alpha{=}1$) or similar contexts ($\alpha{=}0$), with $\alpha{=}0.5$ in between. Where full entity disjointness is not reached, the limit lies mostly in the data. At fixed split sizes, it is possible only when the largest group of sentences linked by shared entities fits within one split. Removing entity leakage lowers test F1 for both a BiLSTM-CRF and a fine-tuned Llama-2-7B, while removing context leakage alone does not, indicating that much of the F1 measured on native splits comes from entities seen during training. The class-balance constraint is essential. Without it, splits strand entity classes and leave test entities with little or no training data for their class, lowering F1 for reasons unrelated to generalization. Compared with MinCut, EMPIRE reaches similar disjointness while preserving class proportions and exact split sizes. The formulation is not specific to NER. It applies to any dataset whose samples share discrete units that a model can memorize and whose labels should stay balanced across splits, by redefining which units the entity matrix records. Relation extraction is a direct extension, with entity pairs or relation triples as the shared units and relation types as the classes.

Reproducibility Statement

We have uploaded all code as supplementary material, including the EMPIRE implementation, the feasibility pre-check, the baselines, split evaluation, and model training. The ILP formulation is given in Section 3, with the feasibility analysis, clustering, and computational complexity in Appendix A.2–A.5. The EMPIRE configurations are listed in Section 4.3, solver settings and hardware in Appendix A.7, and model architectures, hyperparameter search spaces, and training procedures in Appendix A.8. All 24 datasets are publicly available (Table 1). All runs use a fixed random seed (42).

AI Use Statement

We used generative AI tools as an assistant for editing the text of this paper. All experiments, split generation, and model training were designed and run by the authors, and the authors reviewed and verified every figure's caption, number, and sentence produced with this assistance. The authors take full responsibility for the content of the paper.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

Victoria Arranz, Khalid Choukri, Montse Cuadros, Aitor García-Pablos, Lucie Gianola, Cyril Grouin, Manuel Herranz, Patrick Paroubek, and Pierre Zweigenbaum. MAPA project: Ready-to-go open-source datasets and deep learning technology to remove identifying information from text documents. In *Proceedings of the Workshop on Ethical and Legal Issues in Human Language Technologies and Multilingual De-Identification of Sensitive Data In Language Resources within the 13th Language Resources and Evaluation Conference*, pp. 64–72, 2022.

Ting Wai Terence Au, Vasileios Lampos, and Ingemar J Cox. E-NER: An annotated named entity recognition corpus of legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, 2022.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83, 2017.

Florian Babl, Moritz Hennen, Jakob Murauer, and Michaela Geierhos. Random splitting negatively impacts ner evaluation: Quantifying and eliminating the overestimation of ner performance. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9724–9738, 2025.

Gabriel Bernier-Colborne and Philippe Langlais. HardEval: Focusing on challenging tokens to assess robustness of NER. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1704–1711, 2020.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3329–3339, 2022.

Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pp. 73–78, 2004.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 140–147, 2017.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10, 2014.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. Memorization vs. generalization: Quantifying data leakage in nlp performance evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1325–1335, 2021.

Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. The SOFC-Exp corpus and neural approaches to information extraction in the materials science domain. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Jinlan Fu, Pengfei Liu, and Qi Zhang. Rethinking generalization of neural models: A named entity recognition case study. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7732–7739, 2020.

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

Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2786–2791, 2019.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *Proceedings of The Eleventh International Conference on Learning Representations*, 2023.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Roman Joeres, David B Blumenthal, and Olga V Kalinina. Data splitting to avoid information leakage with DataSAIL. *Nature Communications*, 16(1):3337, 2025.

George Karypis and Vipin Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*, 38:7–1, 1998.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

Aman Kumar and Binil Starly. FabNER: information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33(8): 2393–2407, 2022.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270, 2016.

Juhyuk Lee, Min-Joong Lee, June Yong Yang, and Eunho Yang. Does it really generalize well on unseen data? systematic evaluation of relational triple extraction methods. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3849–3858, 2022.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, 2017.

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

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3728–3737, 2021.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. ASGARD: A portable architecture for multilingual dialogue systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8386–8390, 2013. doi: 10.1109/ICASSP.2013.6639301.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. CrossNER: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13452–13460, 2021.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. mmBERT: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*, 2025.

John Morris, Jin Yong Yoo, and Yanjun Qi. TextAttack: Lessons learned in designing python frameworks for NLP. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pp. 126–131, 2020.

Hiroki Nakayama. seqeval: A python framework for sequence labeling evaluation. *Software available from https://github. com/chakki-works/seqeval*, pp. 123, 2018.

OpenAI. Gpt-5.5, 2026. URL `https://developers.openai.com/api/docs/models/gpt-5.5`.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1946–1958, 2017.

Sampo Pyysalo and Sophia Ananiadou. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875, 2014.

Shengtian Sang, Zhihao Yang, Lei Wang, Xiaoxia Liu, Hongfei Lin, and Jian Wang. SemaTyP: A knowledge graph based literature mining method for drug discovery. *BMC Bioinformatics*, 19(1): 193, 2018.

Agam Shah, Abhinav Gullapalli, Ruchit Vithani, Michael Galarnyk, and Sudheer Chava. FiNER-ORD: Financial named entity recognition open research dataset. *arXiv preprint arXiv:2302.11157*, 2023.

Larry Smith, Lorraine K Tanabe, Rie Johnson Nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. Overview of BioCreative II gene mention recognition. *Genome Biology*, 9(Suppl 2):S2, 2008.

Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. Code and named entity recognition in stackoverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4913–4926, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jingxuan Tu and Constantine Lignos. TMR: Evaluating NER recall on tough mentions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 155–163, 2021.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5787, 2020.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5154–5163, 2019.

Ania Wróblewska, Agnieszka Kaliska, Maciej Pawłowski, Dawid Wiśniewski, Witold Sosnowski, and Agnieszka Ławrynowicz. TASTEset–recipe dataset and food entities recognition benchmark. *arXiv preprint arXiv:2204.07775*, 2022.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018.

Dun Zhang, Ziyang Zeng, Yudong Zhou, and Shuyang Lu. Jasper-token-compression-600M technical report. *arXiv preprint arXiv:2511.14405*, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62, 2023.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *International Conference on Learning Representations*, volume 2024, pp. 12276–12294, 2024.

A APPENDIX

A.1 PER-DATASET RESULTS

Table 1 summarizes the 24 datasets. Tables 2–7 report the split ratios, the four split-quality metrics summarized in Figure 2, and the BiLSTM-CRF test F1 summarized in Figure 3, for every dataset and configuration.

Table 1: Summary of Datasets used in this work. Total instances represent the combined native splits.

Dataset	Domain	Total Instances	Semantic Classes	Licensing Info.
AnatEM	Bio (Anatomy)	11,809	1	CC BY-SA 3.0
CrossNER_ai	AI	881	14	MIT License
CrossNER_literature	Literature	916	12	MIT License
CrossNER_music	Music	945	13	MIT License
CrossNER_politics	Politics	1,392	9	MIT License
CrossNER_science	Science	1,193	17	MIT License
FabNER	Manufacturing	13,682	12	Custom (research use)
MAPA_coarse	Legal	1,399	5	CC BY 4.0
MAPA_fine	Legal	1,399	17	CC BY 4.0
conllpp	General (News)	20,744	4	No known license
mit-movie	Conversational/Query	12,218	12	No known license
mit-restaurant	Conversational/Query	9,181	8	No known license
wikiann_en	General (Wikipedia)	40,000	3	ODC-BY / Wikipedia terms
E-NER	Legal (SEC filings)	11,642	7	CC BY-NC-SA 4.0 (non-commercial)
HarveyNER	Crisis/Social Media	6,571	4	No known license
SOFC	Materials Science	876	4	CC BY 4.0
StackOverflow-NER	Software/Programming	15,307	25	CC BY-SA 4.0
TASTEset	Culinary/Food	700	9	CC BY 4.0
FiNER-ORD	Finance	4,739	3	CC BY-NC 4.0
bc2gm	Bio (Genes)	20,000	1	No known license
bc5cdr	Bio	16,423	2	Academic Use
jnlpba	Bio	22,402	5	Academic Use
ncbi_disease	Bio (Disease)	7,298	1	Unknown
wnut	Social Media	5,690	6	No known license

Table 2: Split Ratios across 24 NER datasets and splitting methodologies. Ratios are reported as train:validation:test. Bold indicates a ratio that differs from the Native split at two decimal precision. In all cases we set the lower bound of $\epsilon = 0$; however, for $\alpha < 0$, clustering with $k = 50$ can make this infeasible, and so the feasible tolerance may be higher.

				EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0$)		**EMPIRE** ($\alpha = 0.5$)	
Dataset	**Native**	**MinCut**	**TopSim**	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$
AnatEM	0.50:0.18:0.32	**0.51:0.17:0.31**	0.50:0.18:0.32	0.50:0.18:0.32	0.50:0.18:0.32	0.50:0.18:0.32	0.50:0.18:0.32	0.50:0.18:0.32	0.50:0.18:0.32
CrossNER_ai	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49
CrossNER_literature	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45	0.11:0.44:0.45
CrossNER_music	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	0.11:0.40:0.49	**0.12:0.40:0.48**	0.11:0.40:0.49
CrossNER_politics	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47	0.14:0.39:0.47
CrossNER_science	0.17:0.38:0.46	0.17:0.38:0.46	0.17:0.38:0.46	0.17:0.38:0.46	0.17:0.38:0.46	**0.17:0.38:0.45**	0.17:0.38:0.46	**0.18:0.37:0.45**	0.17:0.38:0.46
FabNER	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15	0.69:0.16:0.15
MAPA_coarse	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29
MAPA_fine	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29	**0.63:0.07:0.29**	0.64:0.07:0.29	0.64:0.07:0.29	0.64:0.07:0.29
conllpp	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17	0.68:0.16:0.17
mit-movie	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	**0.69:0.12:0.20**	0.70:0.10:0.20
mit-restaurant	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20
wikiann_en	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50	0.25:0.25:0.50
E-NER	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20
HarveyNER	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20	0.60:0.20:0.20
SOFC	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20
StackOverflow-NER	0.61:0.19:0.20	0.61:0.19:0.20	0.61:0.19:0.20	0.61:0.19:0.20	0.61:0.19:0.20	0.61:0.19:0.20	0.61:0.19:0.20	**0.60:0.19:0.20**	0.61:0.19:0.20
TASTEset	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20	0.70:0.10:0.20
FiNER-ORD	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23	0.69:0.08:0.23
bc2gm	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25	0.63:0.13:0.25
bc5cdr	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36	0.32:0.32:0.36
jnlpba	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17	0.83:0.00:0.17
ncbi_disease	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13	0.74:0.13:0.13
wnut	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23	0.60:0.18:0.23

A.2 FEASIBILITY ANALYSIS & PARAMETER SELECTION

Before executing the main EMPIRE optimization, a practitioner must specify ϵ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_C)$. However, for a given dataset and cluster count K, not all $(\epsilon, \boldsymbol{\delta})$ combinations admit a valid partition. To guard against this, we employ a lightweight feasibility pre-check that identifies the minimum

Table 3: Entity Disjointness Score (%) (↑) across 24 NER datasets and splitting methodologies. The best-performing score for each dataset is shown in **bold**, while the second-best score is underlined. In all cases, $\epsilon = 0$.

				EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0, K = 50$)		**EMPIRE** ($\alpha = 0.5, K = 50$)	
Dataset	**Native**	**MinCut**	**TopSim**	δ^* (lb 0.1)	$\delta = 1$	δ^* (lb 0.1)	$\delta = 1$	δ^* (lb 0.1)	$\delta = 1$
AnatEM	82.5	**100.0**	74.5	92.1	98.8	75.6	87.3	79.0	90.0
CrossNER_ai	86.2	**98.2**	88.6	97.4	98.1	86.6	91.4	90.5	94.6
CrossNER_literature	88.9	97.5	92.8	98.5	**98.7**	91.1	91.9	91.5	93.8
CrossNER_music	86.5	98.4	92.2	97.7	**98.8**	89.4	91.6	91.6	94.4
CrossNER_politics	88.1	98.5	90.2	98.3	**99.1**	86.9	89.3	88.6	92.0
CrossNER_science	90.0	99.6	93.0	98.9	**99.7**	92.0	95.4	93.7	96.5
FabNER	63.6	57.0	57.6	70.8	**80.1**	57.3	58.6	57.7	62.0
MAPA_coarse	95.8	**100.0**	84.7	98.1	**100.0**	86.3	85.6	90.2	99.4
MAPA_fine	79.5	**100.0**	66.0	82.6	**100.0**	67.5	69.7	65.5	95.1
conllpp	82.9	**100.0**	78.7	96.3	**100.0**	78.8	90.8	82.7	92.6
mit-movie	81.0	99.1	87.4	96.5	**99.4**	80.9	85.6	82.8	83.5
mit-restaurant	82.8	90.9	84.6	**94.6**	92.7	83.4	84.6	85.9	83.0
wikiann_en	94.1	**100.0**	96.7	**100.0**	**100.0**	95.5	95.5	95.9	96.1
E-NER	70.3	**100.0**	77.1	97.7	**100.0**	78.9	76.7	81.9	98.1
HarveyNER	79.8	99.8	89.1	99.9	**100.0**	88.1	90.5	94.5	98.8
SOFC	79.8	85.8	84.0	**93.1**	90.2	83.0	83.2	85.4	85.8
StackOverflow-NER	90.6	**100.0**	83.2	97.3	**100.0**	80.6	96.4	80.0	96.1
TASTEset	72.5	76.0	73.3	81.9	**84.1**	72.3	74.9	75.4	76.4
FiNER-ORD	91.9	**100.0**	85.0	98.9	**100.0**	84.5	98.3	87.3	99.0
bc2gm	89.4	**100.0**	93.9	**100.0**	**100.0**	90.3	98.7	92.1	99.0
bc5cdr	73.2	83.7	57.6	**83.8**	81.4	55.6	59.9	59.4	64.4
jnlpba	93.4	**100.0**	90.1	96.8	**100.0**	86.5	95.5	90.8	96.9
ncbi_disease	86.5	**100.0**	84.9	99.7	**100.0**	79.0	95.7	92.8	97.9
wnut	97.8	**100.0**	95.6	**100.0**	**100.0**	94.9	99.4	95.5	99.6

Table 4: Entity Coverage Ratio (%) (↓) across 24 NER datasets and splitting methodologies. ECR is the leave-one-out share of entity mentions in each split that also occur, with the same label, in the other splits (Section 4.5); lower indicates less entity leakage. The best-performing score for each dataset is shown in **bold**, while the second-best score is underlined. In all cases, $\epsilon = 0$.

				EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0, K = 50$)		**EMPIRE** ($\alpha = 0.5, K = 50$)	
Dataset	**Native**	**MinCut**	**TopSim**	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$
AnatEM	63.2	**0.0**	72.4	24.7	2.0	70.7	54.1	65.7	44.2
CrossNER_ai	41.0	5.6	37.4	11.3	**5.3**	40.7	31.8	31.2	16.3
CrossNER_literature	30.7	8.0	23.8	6.9	**5.6**	27.0	25.3	25.1	18.5
CrossNER_music	34.6	4.8	25.8	9.2	**2.9**	30.3	25.7	26.0	16.2
CrossNER_politics	35.5	4.0	32.4	5.8	**3.1**	37.6	33.0	33.2	22.7
CrossNER_science	31.1	0.8	25.6	6.7	**0.8**	26.9	19.7	21.9	10.0
FabNER	89.8	68.5	92.2	71.7	**63.8**	92.7	92.4	92.7	90.3
MAPA_coarse	10.0	**0.0**	41.5	3.2	**0.0**	35.7	37.3	18.8	0.7
MAPA_fine	62.5	**0.0**	81.1	54.2	**0.0**	79.2	77.3	80.8	9.1
conllpp	55.5	**0.0**	58.8	29.8	**0.0**	60.4	40.5	50.0	26.8
mit-movie	71.3	1.7	61.5	34.9	**1.5**	71.4	65.8	67.0	48.2
mit-restaurant	64.1	14.8	62.2	19.2	**13.3**	63.4	61.8	59.1	48.7
wikiann_en	19.3	**0.0**	13.1	**0.0**	**0.0**	16.0	15.9	14.7	13.9
E-NER	74.0	**0.0**	61.9	6.3	**0.0**	62.0	65.8	53.1	6.5
HarveyNER	55.3	0.2	40.6	0.1	**0.0**	41.7	36.9	17.1	3.8
SOFC	64.3	38.6	57.9	30.2	**27.1**	59.5	59.4	51.2	46.3
StackOverflow-NER	39.2	**0.0**	47.4	12.0	**0.0**	51.2	24.0	49.2	15.9
TASTEset	85.5	83.0	85.0	71.6	**66.3**	85.9	83.9	82.6	81.9
FiNER-ORD	33.1	**0.0**	47.0	2.8	**0.0**	49.1	10.4	36.1	2.5
bc2gm	37.8	**0.0**	25.1	**0.0**	**0.0**	35.6	6.4	29.0	5.0
bc5cdr	72.3	**42.9**	86.1	53.0	45.0	86.8	84.0	82.5	77.7
jnlpba	43.7	**0.0**	49.6	6.7	**0.0**	56.4	36.5	47.6	29.3
ncbi_disease	57.4	**0.0**	48.9	0.3	**0.0**	64.5	27.7	23.2	5.7
wnut	6.2	**0.0**	13.9	**0.0**	**0.0**	15.5	3.0	10.7	0.8

achievable values $(\epsilon^*, \delta_1^*, \ldots, \delta_C^*)$ prior to the main solve. Rather than fixing ϵ and δ_t as constants, we promote them to continuous decision variables:

$$\min_{\{X_{ql}\},\, \epsilon,\, \boldsymbol{\delta}} \quad \epsilon + \sum_{t=1}^{C} \delta_t \tag{14}$$

subject to Equations (1), (3), and (4) from the main EMPIRE formulation, under lower bound constraints $\epsilon \geq \epsilon_{lb}$ and $\delta_t \geq \delta_{lb}\ \forall t$. Because each δ_t appears additively and independently in the

Table 5: Hellinger Distance (↓) across 24 NER datasets and splitting methodologies. Divergence of each split's class distribution from the unsplit whole; lower indicates a closer match (better stratification). The best-performing score for each dataset is shown in **bold**, while the second-best score is underlined. In all cases, $\epsilon = 0$.

				EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0, K = 50$)		**EMPIRE** ($\alpha = 0.5, K = 50$)	
Dataset	**Native**	**MinCut**	**TopSim**	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$
AnatEM	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
CrossNER_ai	0.071	0.266	0.139	**0.031**	0.271	0.036	0.239	0.041	0.312
CrossNER_literature	0.041	0.100	0.116	**0.021**	0.129	0.030	0.141	0.046	0.135
CrossNER_music	0.053	0.245	0.176	**0.028**	0.250	0.032	0.209	0.064	0.242
CrossNER_politics	0.120	0.165	0.157	0.040	0.185	**0.026**	0.211	0.038	0.236
CrossNER_science	0.056	0.218	0.135	**0.027**	0.237	0.046	0.284	0.085	0.347
FabNER	0.064	0.086	0.055	**0.006**	0.062	0.016	0.046	0.020	0.092
MAPA_coarse	0.067	0.151	0.082	**0.034**	0.171	0.039	0.218	0.047	0.318
MAPA_fine	0.152	0.562	0.124	**0.043**	0.499	0.080	0.248	0.062	0.350
conllpp	0.023	0.133	0.041	**0.006**	0.134	0.017	0.099	0.040	0.108
mit-movie	**0.012**	0.350	0.093	0.015	0.348	0.032	0.169	0.039	0.182
mit-restaurant	0.011	0.126	0.085	**0.001**	0.147	0.031	0.122	0.031	0.099
wikiann_en	**0.003**	0.091	0.067	0.029	0.081	0.022	0.036	0.007	0.051
E-NER	0.029	0.183	0.063	**0.028**	0.119	0.042	0.067	0.040	0.184
HarveyNER	**0.018**	0.106	0.057	0.038	0.175	0.020	0.110	0.027	0.155
SOFC	0.009	0.114	0.037	**0.008**	0.064	0.029	0.060	0.015	0.064
StackOverflow-NER	0.053	0.187	0.048	**0.029**	0.208	0.047	0.235	0.068	0.198
TASTEset	0.018	0.042	0.027	**0.010**	0.057	0.012	0.045	0.017	0.027
FiNER-ORD	0.028	0.106	0.032	**0.006**	0.105	0.020	0.064	0.028	0.168
bc2gm	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
bc5cdr	**0.002**	0.039	0.010	0.005	0.014	0.011	0.037	0.034	0.059
jnlpba	0.050	0.076	0.024	**0.000**	0.077	0.030	0.050	0.004	0.083
ncbi_disease	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
wnut	0.116	**0.021**	0.062	0.024	0.043	0.027	0.157	0.036	0.110

Table 6: Information Leakage (↓) across 24 NER datasets and splitting methodologies, measured as mean embedding cosine similarity (entity-masked all-MiniLM-L6-v2) between sentences placed in different splits. Lower indicates splits are more semantically distinct from one another. The best-performing score for each dataset is shown in **bold**, while the second-best score is underlined. In all cases, $\epsilon = 0$.

				EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0, K = 50$)		**EMPIRE** ($\alpha = 0.5, K = 50$)	
Dataset	**Native**	**MinCut**	**TopSim**	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$
AnatEM	0.1759	0.1194	0.1721	0.1743	0.1244	0.1489	0.1149	0.1598	**0.1144**
CrossNER_ai	0.4954	0.4866	0.4997	0.4955	0.4833	0.4957	**0.4679**	0.4955	0.4823
CrossNER_literature	0.5212	0.5185	0.5224	0.5200	0.5169	0.5157	**0.5021**	0.5196	0.5130
CrossNER_music	0.5509	0.5442	0.5467	0.5494	0.5414	0.5478	**0.5253**	0.5480	0.5321
CrossNER_politics	0.5112	0.5085	0.5123	0.5119	0.5086	0.5048	**0.4863**	0.5102	0.4955
CrossNER_science	0.4477	0.4402	0.4463	0.4465	0.4400	0.4467	**0.4236**	0.4503	0.4270
FabNER	0.5133	0.5020	0.4986	0.5075	0.5089	0.4994	**0.4785**	0.5119	0.5031
MAPA_coarse	0.2261	0.2299	0.2246	0.2404	0.2233	0.2097	**0.1849**	0.2369	0.2143
MAPA_fine	0.2239	0.2259	0.2222	0.2341	0.2108	0.2267	**0.1773**	0.2354	0.2150
conllpp	0.3249	0.2181	0.2641	0.3450	0.2184	0.2447	**0.1789**	0.3072	0.2068
mit-movie	0.7551	0.7261	0.7447	0.7554	0.7258	0.7436	**0.6825**	0.7309	0.7359
mit-restaurant	0.7171	0.7006	0.6986	0.7169	0.6985	0.6949	**0.6694**	0.7012	0.6910
wikiann_en	0.7097	0.7057	0.6971	0.7111	0.7058	**0.6640**	0.6694	0.6645	0.6705
E-NER	0.1837	0.1618	0.1757	0.1815	0.1619	0.1761	0.1648	0.1804	**0.1425**
HarveyNER	0.2569	0.2551	0.2323	0.2574	0.2503	0.2489	**0.2201**	0.2366	0.2283
SOFC	0.4728	0.4664	0.4637	0.4801	0.4633	0.4355	**0.4297**	0.4698	0.4680
StackOverflow-NER	0.2370	0.1835	0.2362	0.2427	0.1615	0.2397	**0.0980**	0.2411	0.1359
TASTEset	0.7327	0.7160	0.7163	0.7279	0.7175	**0.6489**	0.6543	0.7230	0.7222
FiNER-ORD	0.1716	0.1354	0.1624	0.1688	0.1317	0.1491	**0.0820**	0.1590	0.0889
bc2gm	0.1647	0.1439	0.1265	0.1739	0.1347	0.1405	**0.0607**	0.1556	0.0720
bc5cdr	0.2940	0.2551	0.2909	0.2937	0.2797	0.2832	**0.2498**	0.2848	0.2540
jnlpba	0.3955	0.2783	0.3614	0.4311	0.2586	0.4060	0.2396	0.3910	**0.2359**
ncbi_disease	0.2734	0.2214	0.2233	0.2664	0.2178	0.1947	**0.1597**	0.2543	0.1971
wnut	0.1536	0.1667	0.1510	0.1666	0.1680	0.1511	**0.1035**	0.1551	0.1129

objective, the minimization drives every class's tolerance down to its own achievable minimum, independent of how constrained any other class is.

The solution $(\epsilon^*, \delta_1^*, \ldots, \delta_C^*)$ represents the tightest tolerances under which any size and class-balanced partition can exist for the given dataset and K. We note that the $(\epsilon, \boldsymbol{\delta})$ values used in our experiments are not necessarily equal to $(\epsilon^*, \boldsymbol{\delta}^*)$. The pre-check serves strictly as a diagnostic: if a

Table 7: Downstream Test F1 across 24 NER datasets, using a BiLSTM-CRF trained from scratch on each generated split. The lowest F1 for each dataset is shown in **bold**, while the second-lowest is underlined. In all cases, $\epsilon = 0$.

				EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0, K = 50$)		**EMPIRE** ($\alpha = 0.5, K = 50$)	
Dataset	**Native**	**MinCut**	**TopSim**	$\delta = 1$	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$	$\delta = 1$	$\delta = 0.1$
AnatEM	0.781	**0.530**	0.768	0.561	0.684	0.764	0.797	0.730	0.717
CrossNER_ai	0.425	0.159	0.229	**0.120**	0.275	0.349	0.371	0.169	0.318
CrossNER_literature	0.369	0.280	0.284	0.275	0.315	0.291	0.352	**0.265**	0.300
CrossNER_music	0.398	0.224	**0.213**	0.218	0.397	0.318	0.381	0.251	0.376
CrossNER_politics	0.499	0.381	0.392	0.405	0.507	0.538	0.571	**0.317**	0.544
CrossNER_science	0.433	0.334	0.293	0.318	0.395	0.339	0.396	**0.201**	0.422
E-NER	0.861	0.522	0.690	0.423	0.431	0.639	0.806	**0.327**	0.709
FabNER	**0.837**	0.853	0.874	0.882	0.900	0.850	0.891	0.898	0.922
FiNER-ORD	0.633	0.459	0.671	0.479	0.454	**0.356**	0.672	0.395	0.639
HarveyNER	0.606	0.265	0.364	0.262	0.260	0.338	0.400	**0.180**	0.292
MAPA_coarse	0.647	0.541	0.553	**0.446**	0.633	0.565	0.582	0.667	0.609
MAPA_fine	0.835	0.256	0.756	**0.001**	0.842	0.687	0.753	0.593	0.853
SOFC	0.761	**0.667**	0.700	0.684	0.688	0.798	0.792	0.762	0.783
StackOverflow-NER	0.503	0.301	0.487	0.342	0.417	**0.078**	0.586	0.191	0.519
TASTEset	0.921	**0.866**	0.872	0.879	0.887	0.911	0.886	0.930	0.911
bc2gm	0.729	0.646	0.603	0.635	0.598	0.369	0.762	**0.268**	0.641
bc5cdr	0.810	**0.544**	0.780	0.710	0.825	0.719	0.837	0.742	0.813
conllpp	0.809	0.569	0.725	**0.552**	0.769	0.734	0.831	0.802	0.815
jnlpba	0.699	0.561	0.693	**0.526**	0.633	0.657	0.785	0.693	0.758
mit-movie	0.872	**0.376**	0.742	0.382	0.775	0.743	0.835	0.708	0.834
mit-restaurant	0.775	0.605	0.624	**0.538**	0.623	0.745	0.732	0.660	0.784
ncbi_disease	0.780	0.437	0.743	0.492	0.549	0.449	0.786	**0.409**	0.730
wikiann_en	0.718	0.687	0.557	0.684	0.607	0.501	0.561	0.504	**0.497**
wnut	0.257	0.316	0.302	0.314	0.396	**0.156**	0.366	0.202	0.354

practitioner's chosen tolerances fall below $(\epsilon^*, \boldsymbol{\delta}^*)$, no valid partition exists and the tolerances must be relaxed. Finally, the feasible region is sensitive to K, as combinations infeasible at small K may become feasible as K increases, where finer-grained clusters afford the solver greater flexibility. Table 8 lists the per-class tolerances δ_t^* found by the pre-check, and Table 9 the relative MIP gap of every EMPIRE solve. Table 10 shows for which datasets full entity disjointness is feasible at ϵ=0 (Section 5).

A.3 Systematic Constraint Reduction via Clustering

A primary mathematical obstacle during ILP optimization is combinatorial explosion driven by large $O(n^2)$ Boolean interactions when minimizing Information Leakage. Native pair-wise similarity assessments rapidly decay solver efficiency.

In our methodology, we execute a systematic refinement: by identifying heavily overlapping representations and clustering identical or highly similar examples into super-nodes (clusters), we drastically reduce the variable space from n individual instances to K generalized centroids.

For this refinement, we employ the K-means clustering algorithm over a hybrid feature space. Because individual instances must be grouped by both their contextual embeddings and their underlying entity targets, we represent each sentence by concatenating its normalized context embedding (v_{struct}) with its normalized entity occurrence vector (v_{ent}). Crucially, we utilize the parameter α to weight this concatenation, defining the fused representation as $[\sqrt{\alpha} \cdot v_{ent} \parallel \sqrt{1-\alpha} \cdot v_{struct}]$. Performing standard K-means clustering on this concatenated space mathematically resembles a weighted distance calculation. Specifically, the squared Euclidean distance between any two instances inherently evaluates to $\alpha||v_{ent}^{(1)} - v_{ent}^{(2)}||^2 + (1-\alpha)||v_{struct}^{(1)} - v_{struct}^{(2)}||^2$. This elegant formulation ensures that the geometric distance evaluated by K-means explicitly balances entity overlap against structural similarity precisely according to α. Consequently, the resulting cluster centroids (super-nodes) reliably group instances that are analogous in both syntactic form and target content.

Consequently, the restrictive structural constraints dictating Z_{l_1,l_2} shift from processing massive $n \times n$ grids to tightly confined $K \times K$ domains. By injecting the physical sentence volumes v_l of these cluster super-nodes directly back into the cardinality limits $\sum_{l=1}^{K} v_l X_{ql}$ and class distribution equilibrium limits $\sum_{l=1}^{K} A_{tl}^{cls} X_{ql}$, the optimization model successfully mirrors full statistical accuracy across

Table 8: Per-class feasible tolerance δ_t^* at $K = 50$ for $\alpha \in \{0, 0.5\}$, $\epsilon = 0$ (lower bound $\delta_{lb} = 0.1$).

	EMPIRE ($\alpha = 0$)			**EMPIRE** ($\alpha = 0.5$)		
Dataset	**Min** δ_t^*	**Mean** δ_t^*	**Max** δ_t^*	**Min** δ_t^*	**Mean** δ_t^*	**Max** δ_t^*
AnatEM	0.100	0.100	0.100	0.100	0.100	0.100
CrossNER_ai	0.100	0.104	0.119	0.100	0.147	0.382
CrossNER_literature	0.100	0.100	0.100	0.100	0.109	0.208
CrossNER_music	0.100	0.101	0.105	0.100	0.142	0.284
CrossNER_politics	0.100	0.100	0.100	0.100	0.108	0.151
CrossNER_science	0.100	0.132	0.243	0.100	0.209	0.488
FabNER	0.100	0.100	0.100	0.100	0.100	0.100
MAPA_coarse	0.100	0.280	1.000	0.100	0.280	1.000
MAPA_fine	0.100	0.368	1.000	0.100	0.318	1.000
conllpp	0.100	0.100	0.100	0.100	0.100	0.100
mit-movie	0.100	0.105	0.159	0.100	0.121	0.331
mit-restaurant	0.100	0.100	0.100	0.100	0.100	0.100
wikiann_en	0.100	0.100	0.100	0.100	0.100	0.100
E-NER	0.100	0.237	1.000	0.100	0.281	1.000
HarveyNER	0.100	0.100	0.101	0.100	0.100	0.100
SOFC	0.100	0.100	0.100	0.100	0.100	0.100
StackOverflow-NER	0.100	0.175	1.000	0.100	0.228	1.000
TASTEset	0.100	0.100	0.100	0.100	0.100	0.100
FiNER-ORD	0.100	0.100	0.100	0.100	0.100	0.100
bc2gm	0.100	0.100	0.100	0.100	0.100	0.100
bc5cdr	0.100	0.100	0.100	0.100	0.100	0.100
jnlpba	0.100	0.100	0.100	0.100	0.100	0.100
ncbi_disease	0.100	0.100	0.100	0.100	0.100	0.100
wnut	0.100	0.100	0.100	0.100	0.100	0.100

millions of samples while circumventing NP-hard memory exhaustion. This architectural paradigm allows the Information Leakage reduction logic to execute smoothly for $\alpha < 1$ implementations.

A.4 An illustrative Example for Alpha Interpolation

To rigorously demonstrate how the parameter α dynamically balances Entity Disjointness against Information Leakage, consider a toy dataset with $K = 3$ clusters (G_1, G_2, G_3) and $m = 3$ unique entities (E_1, E_2, E_3), targeting $k = 2$ splits.

Entity Distribution:

- G_1 contains $\{E_1, E_2\}$
- G_2 contains $\{E_3\}$
- G_3 contains $\{E_2\}$

Maximum possible disjoint entities $D_{max} = 3$.

Structural Similarities (W):

- $W_{1,2} = 0.8$ (High structural similarity between G_1 and G_2)
- $W_{2,3} = 0.2$ (Low similarity between G_2 and G_3)
- $W_{1,3} = 0.0$ (No similarity)

Maximum possible information leakage $I_{max} = 1.0$ (assuming volumes $v = 1$).

The normalized objective function evaluates: Obj $= \alpha \cdot \frac{D}{3} - (1 - \alpha) \cdot \frac{I_L}{1.0}$

Scenario 1: $\alpha = 1$ (Strict Entity Disjointness) The solver completely ignores structural similarities and strictly maximizes D.

- **Optimal Partition:** $\{G_1, G_3\}$ in Split 1, and $\{G_2\}$ in Split 2.

Table 9: Gurobi relative MIP gap (%) at termination for each EMPIRE solve, |bound − incumbent|/|incumbent|; lower is closer to proven optimal. † marks solves stopped by the time limit (7,200 s for α=1, 14,400 s for $\alpha < 1$); unmarked solves terminated before it. Gaps above 100% occur when the bound remains loose at the time limit. δ^* is the tightest feasible per-class tolerance with lower bound 0.1. EMPIRE with $\alpha < 1$ uses entity-masked embeddings. In all cases, $\epsilon = 0$.

	EMPIRE ($\alpha = 1, K = 0$)		**EMPIRE** ($\alpha = 0, K = 50$)		**EMPIRE** ($\alpha = 0.5, K = 50$)	
Dataset	δ^*	$\delta = 1$	δ^*	$\delta = 1$	δ^*	$\delta = 1$
AnatEM	23.6†	0.9†	8.3†	6.5†	10.7†	0.0
CrossNER_ai	11.6†	1.9†	0.0	6.8†	4.3	0.0
CrossNER_literature	6.1†	3.4†	10.2†	7.3†	4.2	0.0
CrossNER_music	16.3†	1.0†	4.7	6.8†	0.0	0.0
CrossNER_politics	8.3†	0.4†	10.6†	8.5†	0.0	4.4
CrossNER_science	4.8†	0.0	1.2	7.8†	0.0	0.0
FabNER	247.4†	185.7†	7.7†	7.7†	13.7†	77.5†
MAPA_coarse	0.0	0.0	4.9	4.7	3.6	4.9
MAPA_fine	21.7†	0.0	0.0	4.9	5.0	3.6
conllpp	61.0†	0.0	6.6†	5.0	4.6	4.7
mit-movie	64.3†	0.4†	4.9	5.0†	0.0	3.7
mit-restaurant	30.6†	18.8†	5.0	5.0	4.7	0.0
wikiann_en	0.0	0.0	4.4	5.0	32.8†	38.0†
E-NER	2.2†	0.0	1.7	6.0†	0.0	4.8
HarveyNER	0.0	0.0	4.9	9.6†	4.9	5.0
SOFC	22.3†	17.2†	5.0	5.2†	63.4†	43.4†
StackOverflow-NER	21.3†	0.0	1.5	5.0	4.9	5.0
TASTEset	108.2†	60.7†	4.9	5.0	13.6†	12.0†
FiNER-ORD	0.5†	0.0	5.0	5.0	0.0	4.2
bc2gm	0.0	0.0	5.3†	5.0	3.8	5.0
bc5cdr	136.4†	96.5†	5.0	4.7	5.6†	4.8
jnlpba	6.7†	0.0	3.2	1.0	2.2	4.6
ncbi_disease	0.0	0.0	4.9	4.8	5.0	4.7
wnut	0.0	0.0	5.0	8.2†	0.0	4.7
Hit time limit (of 24)	18	11	6	12	6	4

- **Result:** Split 1 contains $\{E_1, E_2\}$, Split 2 contains $\{E_3\}$. All three entities are perfectly isolated ($D = 3$).
- **Consequence:** Because G_1 and G_2 were forced into different splits, Information Leakage is severely penalized ($I_L = W_{1,2} + W_{2,3} = 1.0$).

Scenario 2: $\alpha = 0$ **(Strict Information Leakage Minimization)** The solver completely ignores entity overlap and strictly minimizes cut similarities.

- **Optimal Partition:** $\{G_1, G_2\}$ in Split 1, and $\{G_3\}$ in Split 2.
- **Result:** The highly similar G_1 and G_2 are safely kept together. The only similarity cut is between G_2 and G_3, resulting in minimal leakage ($I_L = 0.2$).
- **Consequence:** Entity disjointness drops. Because E_2 exists in both G_1 (Split 1) and G_3 (Split 2), it leaks across boundaries. Only E_1 and E_3 remain isolated ($D = 2$).

Scenario 3: $\alpha = 0.5$ **(Balanced Trade-off)** The solver evaluates the mathematical compromise between the two extremes.

- **Evaluating** $\alpha = 1$ **Partition:** $0.5 \cdot (\frac{3}{3}) - 0.5 \cdot (\frac{1.0}{1.0}) = 0.5 - 0.5 = 0.0$
- **Evaluating** $\alpha = 0$ **Partition:** $0.5 \cdot (\frac{2}{3}) - 0.5 \cdot (\frac{0.2}{1.0}) = 0.333 - 0.1 = 0.233$

At exactly $\alpha = 0.5$, the solver mathematically realizes that the $+33\%$ gain in Entity Disjointness (from $D = 2$ to $D = 3$) is drastically outweighed by the massive -80% penalty in Information

Table 10: Feasibility of full entity disjointness. Sentences are linked when they share an entity (exact surface form and class); a split with no shared entity must keep each connected group of linked sentences within one split, so at ϵ=0 it requires at least that the largest group fits in the largest split. *Largest split* is the size of the largest split at the native ratio, *Largest group* the size of the largest connected group and % its share of all sentences, and *Fits* whether the group fits in that split. The last two columns report entity disjointness (%) and the relative MIP gap (%) of EMPIRE (α=1, K=0, δ=1). Grouping sentences by stemmed entities gives the same verdict on every dataset. Rows are ordered by the ratio of the largest group to the largest split.

			Largest group			**EMPIRE** (α=1, δ=1)	
Dataset	**Sentences**	**Largest split**	**Size**	**%**	**Fits**	**Disj.**	**Gap**
bc5cdr	16,423	5,865	11,500	70.0	no	81.4	96.5
CrossNER_politics	1,392	651	1,153	82.8	no	99.1	0.4
CrossNER_literature	916	416	720	78.6	no	98.7	3.4
CrossNER_ai	881	431	730	82.9	no	98.1	1.9
CrossNER_music	945	465	725	76.7	no	98.8	1.0
FabNER	13,682	9,435	13,652	99.8	no	80.1	185.7
TASTEset	700	489	700	100.0	no	84.1	60.7
SOFC	876	613	864	98.6	no	90.2	17.2
CrossNER_science	1,193	543	729	61.1	no	99.7	0.0
mit-restaurant	9,181	6,426	7,810	85.1	no	92.7	18.8
mit-movie	12,218	8,552	9,167	75.0	no	99.4	0.4
AnatEM	11,809	5,861	6,037	51.1	no	98.8	0.9
conllpp	20,744	14,041	13,789	66.5	yes	100.0	0.0
jnlpba	22,402	18,546	17,612	78.6	yes	100.0	0.0
StackOverflow-NER	15,307	9,263	6,734	44.0	yes	100.0	0.0
FiNER-ORD	4,739	3,262	1,870	39.5	yes	100.0	0.0
ncbi_disease	7,298	5,433	2,991	41.0	yes	100.0	0.0
MAPA_fine	1,399	893	399	28.5	yes	100.0	0.0
E-NER	11,642	8,149	3,401	29.2	yes	100.0	0.0
bc2gm	20,000	12,500	4,711	23.6	yes	100.0	0.0
MAPA_coarse	1,399	893	319	22.8	yes	100.0	0.0
wikiann_en	40,000	20,000	3,881	9.7	yes	100.0	0.0
HarveyNER	6,571	3,967	642	9.8	yes	100.0	0.0
wnut	5,690	3,394	82	1.4	yes	100.0	0.0

Leakage (from $I_L = 0.2$ to $I_L = 1.0$). Consequently, the solver intelligently selects the $\alpha = 0$ partition ($\{G_1, G_2\}$ vs $\{G_3\}$) as the globally optimal compromise.

A.5 Computational Complexity

EMPIRE operates over K clusters, m targets, k splits, and C semantic classes. The total complexity dynamically scales with the interpolation parameter α:

- When $0 < \alpha < 1$: The solver must evaluate both entity disjointness and structural similarity. It initializes exactly $K \cdot k + m \cdot k + m + \frac{K(K-1)}{2}$ binary decision variables and enforces $K+m\cdot k+k+k\cdot C+m+\frac{3}{2}\cdot k\cdot K(K-1)$ constraints. Because $K \ll n$, condensing the n raw instances into a constant-sized set of K pre-computed clusters reduces the mathematically NP-hard $\mathcal{O}(n^2)$ similarity bounds into a manageable, highly scalable formulation. More details on clustering is provided in Appendix A.3.

- When $\alpha = 1$: The optimization focuses strictly on entity/triplet disjointness. The formulation reduces to exactly $K \cdot k + m \cdot k + m$ decision variables bounded by $K + m \cdot k + k + k \cdot C + m$ constraints. Because the quadratic $\frac{K(K-1)}{2}$ terms are entirely absent, this specific formulation completely avoids combinatorial scaling, rendering the disjoint optimization significantly less complex.

- When $\alpha = 0$: The model focuses strictly on minimizing structural context leakage. The formulation reduces to exactly $K \cdot k + \frac{K(K-1)}{2}$ decision variables governed by $K + k + k \cdot C + \frac{3}{2} \cdot k \cdot K(K-1)$ constraints.

A.6 TopSim Baseline

TopSim is a simple baseline for context leakage that requires no optimization. Each sentence s_a is embedded with all-MiniLM-L6-v2 (without entity masking) and L_2-normalized to u_a. Its isolation score is its maximum cosine similarity to any other sentence in the dataset,

$$m_a = \max_{b \neq a}; u_a \cdot u_b. \tag{15}$$

Sentences are sorted by m_a in ascending order and assigned in that order to the test, validation, and training splits, each filled to its native size. The most isolated sentences therefore form the test split, and near-duplicates remain in training. For jnlpba, which has no validation split, only the test and training splits are filled. TopSim is deterministic and preserves the native split sizes exactly, but has no entity or class-balance constraint. Computing all scores takes $O(n^2 d)$ time for n sentences with d-dimensional embeddings; processing the similarity matrix in blocks keeps memory linear in n. Because it considers only each sentence's single nearest neighbor, TopSim removes near-duplicate contexts but changes the average similarity between splits only slightly.

A.7 Hardware and software information

EMPIRE is implemented in Python with CVXPY 1.8.2 and solved with Gurobi 13.0.2. Each solve runs on 8 CPU cores with 32,GB of memory, and Gurobi's thread count is left at its default, so it uses all allocated cores. The feasibility pre-check and the main ILP each use the same time limit: 7,200 s for α=1 and 14,400 s for $\alpha < 1$. The feasibility pre-check uses a relative MIP gap tolerance of 0, so that δ^* is the tightest tolerance it can prove feasible within the limit. The main ILP uses a tolerance of 0 for α=1 and 5% for $\alpha < 1$. All solves use seed 42. The BiLSTM-CRF is trained on one NVIDIA L40S GPU with 8 CPU cores and 48 GB of memory. Llama-2-7B is fine-tuned either on two NVIDIA H100 GPUs with 16 CPU cores and 128 GB of memory or on four H100 GPUs with 48 CPU cores and 480 GB, and evaluated on a single H100 with 8 CPU cores and 32 GB of memory.

A.8 Model Training Details

BiLSTM-CRF. We use a single-layer bidirectional LSTM (hidden size 256) with a CRF output layer (Huang et al., 2015; Lample et al., 2016). Sentences are tokenized with the all-MiniLM-L6-v2 subword tokenizer, and the embedding table is initialized from that model's input embeddings and fine-tuned at one tenth of the learning rate of the LSTM and CRF. We train with AdamW for 50 epochs under a warmup–stable–decay schedule (10% linear warmup, 20% decay), with gradient-norm clipping at 5.0 and batch sizes of 64–256 depending on dataset size, and restore the checkpoint with the best validation F1. For each dataset, we tune the learning rate ($10^{-4}, 5{\cdot}10^{-4}, 10^{-3}, 5{\cdot}10^{-3}, 10^{-2}$), weight decay ($0, 10^{-4}, 10^{-3}, 10^{-2}, 0.1$), and dropout ($0.2, 0.3, 0.4, 0.5$) with 20 trials of Optuna's TPE sampler (Akiba et al., 2019) on the native split. We then reuse the selected values for every other split of that dataset, so no split configuration receives its own tuning. Test F1 is entity-level micro F1 computed with seqeval (Nakayama, 2018). jnlpba provides only training and test data, so we hold out a random 15% of each training split for validation. All runs use seed 42 on a single NVIDIA L40S GPU.

Llama-2-7B. We fully fine-tune Llama-2-7B (base) (Touvron et al., 2023) in the conversational instruction format of UniversalNER (Zhou et al., 2024). Each training sentence becomes one conversation that presents the text and then asks, for each entity type of the dataset, "What describes *[type]* in the text?", with the list of matching entity strings as the answer. We train for up to 20 epochs with learning rate 2×10^{-5} (cosine schedule, 4% warmup, no weight decay), a global batch size of 128, bf16 precision, and a maximum sequence length of 1024 tokens. Training stops early once the validation loss has not improved for three epochs, and we evaluate the final model. Training uses FSDP on two or four NVIDIA H100 GPUs, with the global batch size held fixed. At test time, the model generates each answer by always choosing its most probable next token (up to 128 new tokens) and compute micro F1 between predicted and gold entity strings per query, after lowercasing

and removing articles and punctuation, following UniversalNER. The hyperparameters are identical across datasets and splits.

Figure 4 compares the test F1 of both models on the five datasets used for Llama-2-7B (Section 5.3).

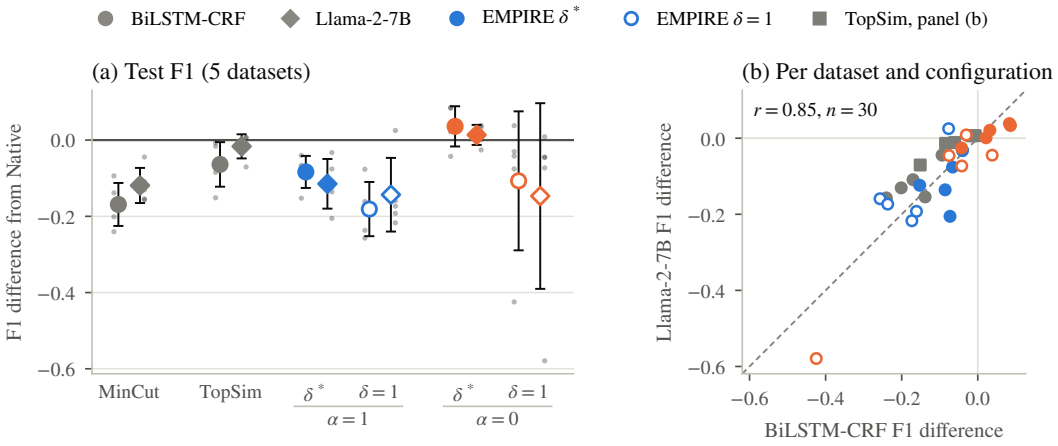

Figure 4: **Model ablation: BiLSTM-CRF vs. fine-tuned Llama-2-7B** on SOFC, mit-restaurant, StackOverflow-NER, conllpp and jnlpba (identical splits for both models). **(a)** Mean paired test-F1 difference from the Native split with ± 1 standard deviation across the five datasets; small dots are individual datasets. **(b)** Test-F1 difference from Native for every dataset and configuration, BiLSTM-CRF against Llama-2-7B (dashed: $y{=}x$; Pearson $r = 0.85$ over 30 cells). EMPIRE with $\alpha{=}0.5$ was not trained with Llama-2-7B. δ^* is the tightest feasible per-class tolerance with lower bound 0.1; $\delta{=}1$ removes the class-balance constraint. Training details are in Appendix A.8.

A.9 Test-Set Composition under Each Class-Balance Setting

These figures support Section 5.4. Figure 5 shows the number of gold entities in each test split, and Figure 6 shows how many test entities belong to classes with little training data, with and without the class-balance constraint.

A.10 Split-Parameter Ablations

This section gives the details of the ablations in Section 5.5: the number of clusters K (Figure 7), the embedding model (Figure 8 and Table 11), and the class-balance floor δ (Figure 9). All splits are scored with the same entity-masked all-MiniLM-L6-v2 embeddings.

A.11 Limitations

EMPIRE provides a mathematically rigorous framework for dataset partitioning that jointly controls entity leakage, context leakage, and class imbalance, but it has several limitations. First, the optimization is formulated as an NP-hard ILP, making runtime and memory usage sensitive to dataset size and cluster count K. Although K-means clustering reduces the variable space from $O(n^2)$ to $O(K^2)$, large K values can still lead to substantial computational overhead and solver timeouts. Second, EMPIRE relies primarily on commercial solvers such as Gurobi for efficient optimization. While open-source alternatives like SCIP and CBC are available, they are considerably slower on large-scale ILPs, limiting accessibility for non-academic users. Third, for $\alpha < 1$ EMPIRE optimizes a cluster-level surrogate: context similarity is computed between cluster centroids rather than individual sentences, introducing a mismatch between the optimized objective and the sample-level information-leakage metric. As a result, the solution obtained is optimal for the surrogate objective but not necessarily for the exact sample-level formulation. Finally, EMPIRE produces static partitions; incorporating new samples requires re-running the optimization, which is computationally expensive and may alter existing assignments, reducing benchmarking consistency in dynamic dataset settings.

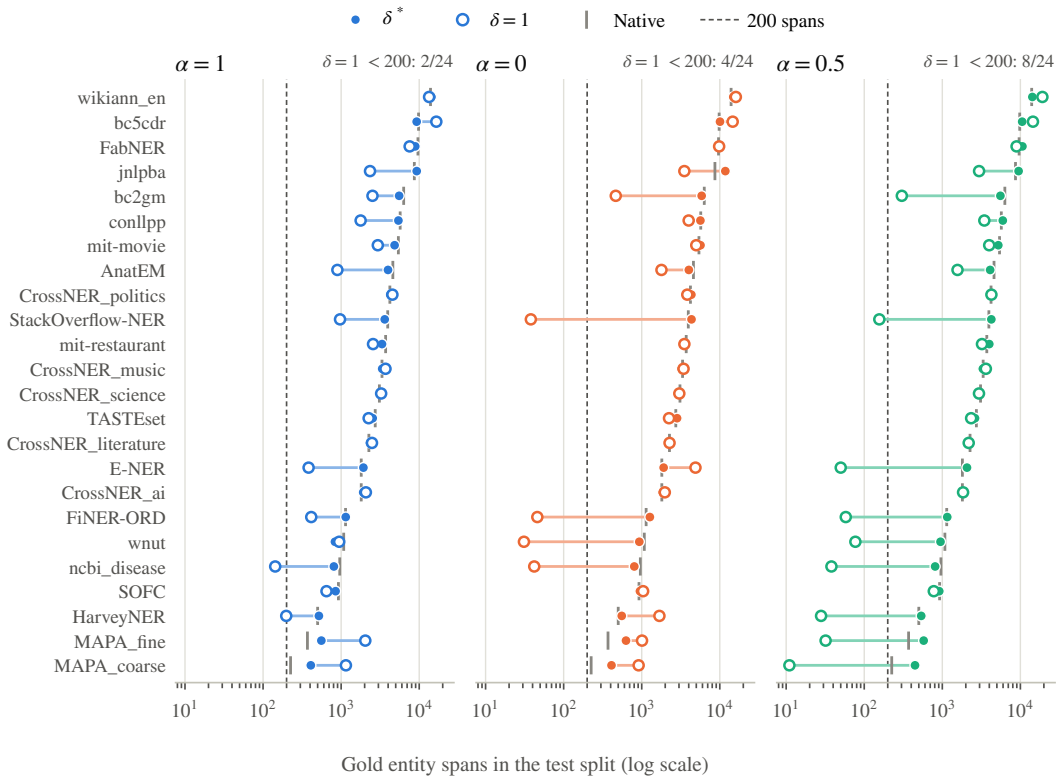

Figure 5: **Test-set size under each class-balance setting.** Gold entity spans in the test split of each dataset (log scale) for EMPIRE with $\alpha \in \{1, 0, 0.5\}$ at δ=1 (hollow) and δ^* (filled), with the Native test set as a gray tick; datasets are ordered by Native test size. The dashed line marks 200 spans. Without the class-balance constraint the solver is free to concentrate entity mentions on one side of the split: 2 (α=1), 4 (α=0), 8 (α=0.5) datasets fall below 200 test spans at δ=1, and none do at δ^*.

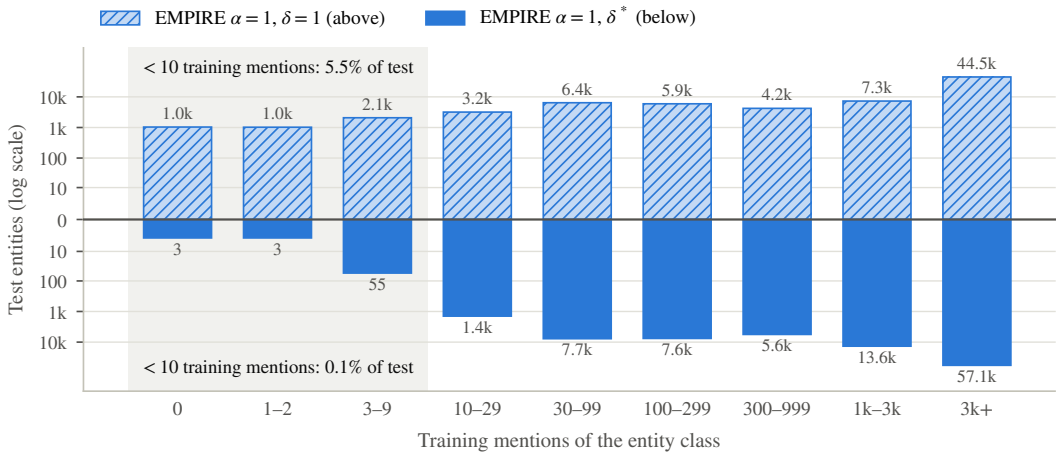

Figure 6: **Test entities by the training support of their class** (EMPIRE α=1, all 24 datasets pooled). Bars count every gold test entity by the number of training mentions of its class (log scale); δ=1 above the axis, δ^* below. At δ=1, 5.5% of test entities belong to classes with fewer than 10 training mentions, against 0.1% at δ^*.

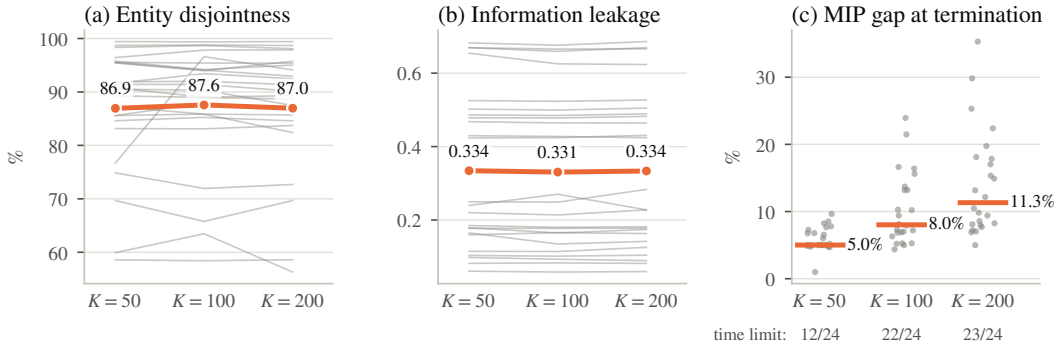

Figure 7: **Effect of the number of clusters** K (EMPIRE α=0, δ=1, entity-masked embeddings, 24 datasets). **(a, b)** Entity disjointness and information leakage; gray lines are individual datasets and the bold line is the mean. **(c)** Relative MIP gap at termination (dots: datasets; bar: median); the count below each K is the number of solves stopped by the 14,400 s time limit. Mean disjointness (86.9–87.6) and information leakage (0.331–0.334) stay within the run-to-run noise (up to 0.92 pp and 0.017), while the median gap grows from 5.0% to 11.3%.

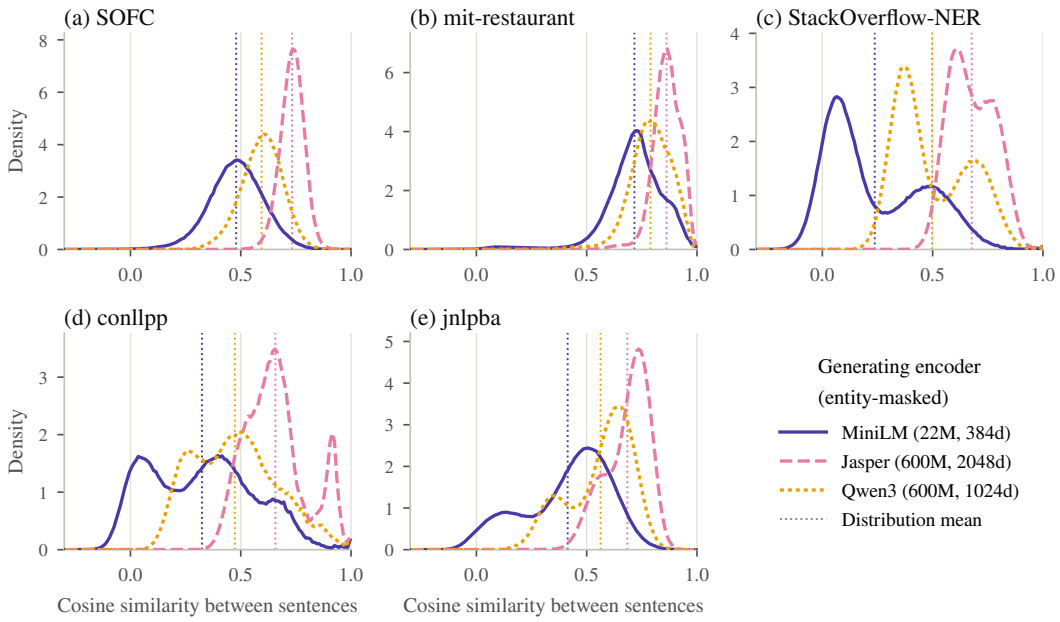

Figure 8: **Sentence-similarity distributions of the three generating encoders.** For each ablation corpus, the density of pairwise cosine similarity between 3,000 randomly sampled sentences (the same sample for all encoders), computed on entity-masked embeddings from all-MiniLM-L6-v2, Jasper-Token-Compression-600M and Qwen3-Embedding-0.6B. Dotted lines mark each distribution's mean. Because the encoders place the same sentences at very different similarity levels, information leakage must be scored in one fixed embedding space to compare splits generated with different encoders.

Table 11: Embedding-model ablation (EMPIRE α=0, K=50, entity-masked embeddings). *Sim.* gives the mean and standard deviation of the pairwise cosine similarity between sentences in each generating encoder's space (Figure 8). For each class-balance setting: entity disjointness (%), information leakage measured on all-MiniLM-L6-v2 embeddings (I.L.) and on the generating encoder's own embeddings (I.L. own; identical to I.L. for MiniLM), and the relative MIP gap (%). δ^* is the tightest feasible per-class tolerance with lower bound 0.1. MiniLM rows are the main-study splits (Figure 2).

		Sim.		δ=1				δ^*			
Dataset	**Encoder**	Mean	SD	Disj.	I.L.	I.L. own	Gap	Disj.	I.L.	I.L. own	Gap
SOFC	MiniLM	0.48	0.12	83.2	0.430	–	5.2	83.0	0.436	–	5.0
	Jasper	0.73	0.05	83.9	0.441	0.717	6.3	83.0	0.454	0.721	5.8
	Qwen3	0.59	0.09	83.2	0.440	0.565	5.0	83.3	0.454	0.570	5.0
mit-restaurant	MiniLM	0.72	0.13	84.6	0.669	–	5.0	83.4	0.695	–	5.0
	Jasper	0.86	0.06	83.4	0.686	0.841	6.3	83.4	0.694	0.852	5.8
	Qwen3	0.79	0.09	84.7	0.679	0.752	5.9	82.8	0.694	0.769	5.9
StackOverflow-NER	MiniLM	0.24	0.22	96.4	0.098	–	5.0	80.6	0.240	–	1.5
	Jasper	0.68	0.10	98.4	0.113	0.620	10.1	80.1	0.252	0.683	0.0
	Qwen3	0.50	0.17	96.7	0.101	0.389	8.5	80.8	0.244	0.500	0.0
conllpp	MiniLM	0.32	0.24	90.8	0.179	–	5.0	78.8	0.245	–	6.6
	Jasper	0.66	0.14	89.0	0.173	0.576	7.9	80.2	0.245	0.601	8.3
	Qwen3	0.47	0.19	89.4	0.175	0.353	5.7	77.2	0.258	0.412	7.1
jnlpba	MiniLM	0.41	0.20	95.5	0.240	–	1.0	86.5	0.406	–	3.2
	Jasper	0.68	0.10	88.7	0.429	0.690	4.9	87.2	0.416	0.688	5.0
	Qwen3	0.56	0.14	94.4	0.277	0.454	4.4	86.7	0.456	0.593	5.0

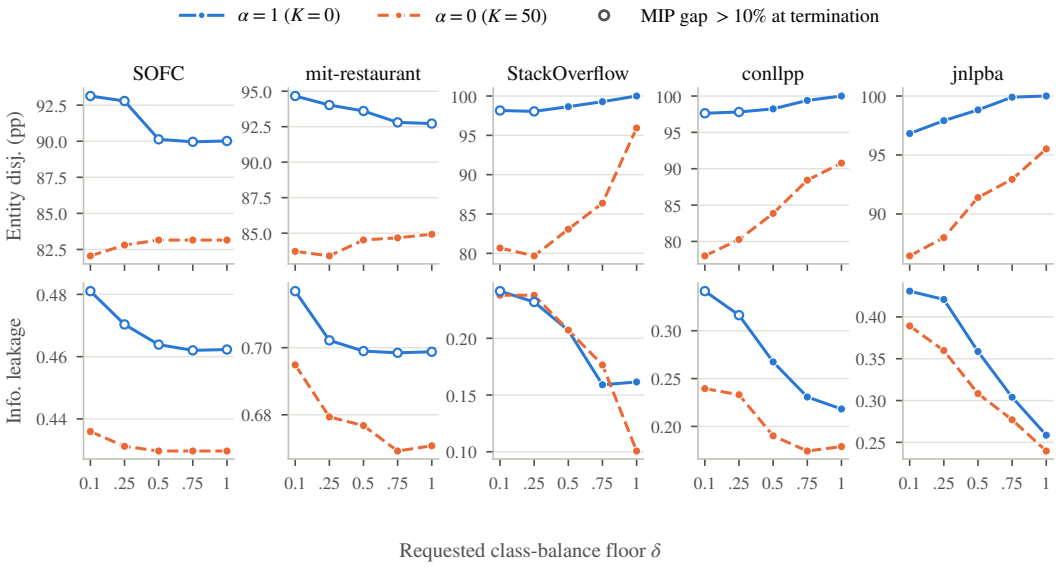

Figure 9: **Effect of the class-balance floor** δ on five datasets, for EMPIRE α=1 (K=0) and α=0 (K=50, entity-masked embeddings). The x-axis is the requested lower bound on the per-class tolerance; for StackOverflow-NER the achieved mean is slightly higher because classes spanning fewer clusters than splits are pinned at δ_c=1. Hollow markers denote solves that stopped with a MIP gap above 10%. Information leakage is measured on entity-masked all-MiniLM-L6-v2 embeddings; ϵ=0 throughout.