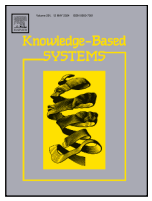
Knowledge-Based Systems 333 (2026) 115024

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



SCoRE: Streamlined corpus-based relation extraction using multi-label contrastive learning and Bayesian kNN

Luca Mariotti *, Veronica Guidetti, Federica Mandreoli

Department of Physical, Computer and Mathematical Sciences - University of Modena and Reggio Emilia, via Giuseppe Campi, 213/a, Modena, 41125, Emilia Romagna, Italy

ARTICLE INFO

Keywords:
Multi-label relation extraction
Distant supervision
Knowledge graph enrichment
Pretrained large language model
Bayesian kNN
Contrastive learning

ABSTRACT

The growing demand for efficient knowledge graph (KG) enrichment leveraging external corpora has intensified interest in relation extraction (RE), particularly under low-supervision settings. To address the need for adaptable and noise-resilient RE solutions that integrate seamlessly with pre-trained large language models (PLMs), we introduce SCoRE, a modular and cost-effective sentence-level RE system. SCoRE enables easy PLM switching, requires no finetuning, and adapts smoothly to diverse corpora and KGs. By combining supervised contrastive learning with a Bayesian k-Nearest Neighbors (kNN) classifier for multi-label classification, it delivers robust performance despite the noisy annotations of distantly supervised corpora. To improve RE evaluation, we propose two novel metrics: Correlatin Structure Distance (CSD), measuring the alignment between learned relational patterns and KG structures, and Precision at R (P@R), assessing utility as a recommender system. We also release Wiki20d, a benchmark dataset replicating real-world RE conditions where only KG-derived annotations are available. Experiments on five benchmarks demonstrate that SCoRE matches or slightly surpasses state-of-the-art methods (average gains of +3.2 in micro-F1 and +5.9 in macro-F1 against fully reproducible baselines), while reducing the training burden by more than an order of magnitude (≈ 99 % lower energy consumption in kWh) Further analyses reveal that increasing model complexity, as seen in prior work, degrades performance, highlighting the advantages of SCoRE's minimal design. Combining efficiency, modularity, and scalability, SCoRE stands as an optimal choice for real-world RE applications.

1. Introduction

Knowledge Graphs (KGs) play a crucial role in organizing and representing structured information across various domains, enhancing applications from information retrieval to complex question-answering systems [1,2]. Enriching KGs to ensure they remain up-to-date requires leveraging external data sources, particularly textual corpora. This need has driven extensive research into relation extraction (RE) [3,4], a KG enrichment task aiming at categorizing the entailed relation between two given KG entities mentioned in the text. For instance, given the sentence *"Aspirin is commonly prescribed to reduce the risk of heart attacks"* and the entity mentions *"Aspirin"* and *"heart attacks"*, a relation extraction system would predict the relation *"prevent"* to enrich a medical KG.

A key challenge for RE approaches, based on statistical and machine learning methods, is the limited availability of high-quality annotated data. Distant supervision (DS) addresses this issue by aligning textual corpora with existing KGs, thereby automatically generating relational labels at scale [5–7]. By focusing on relations' existence rather than relation mentions, this approach simplifies training by increasing data availability at the expense of introducing label noise, as automatic labeling may not always align with context-specific meanings in text [8]. As a result, recent studies have sought to mitigate DS noise with multiple-instance learning (MIL) trading sentence-level RE with entity-pair-level RE by constructing bags of sentences [8,9].

Given their outstanding ability to process natural language, in recent years, RE research has increasingly relied on deep learning and pre-trained language models (PLMs) [3,4,10,11]. In this context, the overarching trend is to combine sophisticated modeling strategies and fine-tuning protocols to mitigate DS noise and achieve more accurate RE models. Specifically, many of these methods use PLM fine-tuning on DS data, often incorporating entity markers or masking strategies to guide attention mechanisms and improve precision [12]. Bag-level formulations coupled with self-attention mechanisms are frequently adopted to mitigate label noise in DS [13–16]. Some approaches further enrich sentence representations with global contextual signals and structured knowledge from KGs to improve robustness in noisy settings [17–19].

* Corresponding author.
E-mail addresses: luca.mariotti@unimore.it (L. Mariotti), veronica.guidetti@unimore.it (V. Guidetti), federica.mandreoli@unimore.it (F. Mandreoli).

https://doi.org/10.1016/j.knosys.2025.115024
Received 10 July 2025; Received in revised form 5 November 2025; Accepted 29 November 2025
Available online 2 December 2025

However, as advocated in [20], attention-based approaches are prone to degradation with increasing noise levels, as they tend to excessively concentrate on a few high-attention sentences. Other techniques aim to refine PLM representations by performing contrastive learning (CL) pretraining or using specialized training paradigms [12,21–23]. However, while self-/semi-supervised CL frameworks are generally robust to noise, they often demand substantial quantities of relatively clean data [24], which are seldom available in DS scenarios. Finally, while most works operate at the bag level, some focus on sentence-based RE leveraging attention [25,26] or CL-based finetuning [27,28].

In this work, we consider RE from a novel perspective, emphasizing solutions that are not only accurate but also readily deployable and maintainable in realistic application scenarios. To this end, we identify several key requirements for an effective RE system: (i) it should operate at the sentence level; (ii) it should address RE as a multi-label classification task; (iii) it must be cost-effective in terms of computational resources; (iv) it must be modular enough to cope with the rapidly evolving ecosystem of PLMs and allow straightforward adjustment to different corpora and KGs.

Requirement (i) arises from the observation that even if RE performance improves, full process automation remains unattainable, and some degree of external intervention will likely remain necessary for KG enrichment. Under such conditions, bag-level RE approaches are problematic because they do not enable fine-grained predictions [29]. Instead, this granularity is necessary to explain the model's decisions and, ultimately, enable domain-expert guidance. Similarly, given the scarcity of manually curated annotated corpus in realistic domains, reliance on fully automated DS annotation becomes necessary, thus embracing a multi-label training and prediction paradigm (requirement (ii)). The remaining requirements stem from a thorough analysis of the key limitations in existing solutions, which often neglect scalability and the balance between performance and computational efficiency. For instance, while additional learning modules or advanced training strategies are introduced, they may not always yield significant performance gains and further obscure the model's decision-making process. Furthermore, methods that heavily depend on fine-tuning PLMs are manageable for smaller models (e.g., BERT) but become increasingly impractical for larger architectures [30], raising concerns about their long-term feasibility as RE solutions. Instead, a cost-effective and modular approach is essential: an approach capable of seamlessly integrating with diverse corpora, KGs, and PLMs to ensure both ready deployment and maintenance in realistic application scenarios.

Finally, we advocate that RE evaluation needs to extend beyond standard classification metrics, such as precision, recall, F1-score, and AUC. While these metrics, adapted from binary classification to the multi-label setting, typically emphasize aggregate performance across predicted relation classes, they often fail to capture multi-label performance at the per-sample level. This evaluation gap risks misrepresenting the practical utility of RE solutions, where these systems are more valuable as recommendation engines. Additionally, a thorough evaluation of the quality of learned relational patterns in relation to the underlying KG is essential to ensure their effectiveness.

This paper proposes a viable solution that meets all the above requirements and shows its effectiveness under realistic conditions. Specifically, we provide the following contributions to the RE research domain:

1. We introduce Streamlined Corpus-based Relation Extraction (SCoRE), a modular and lightweight RE framework. SCoRE is streamlined in that it avoids PLM fine-tuning for computational efficiency, adopts a modular architecture that is PLM-agnostic, and relies on a minimal training/inference pipeline. During training, SCoRE leverages multi-label supervised CL to build robust relation representations resilient to DS noise, while treating the PLM as an informed prior for head-tail mention encoding rather than a fine-tuned component. In inference, SCoRE employs a multi-label Bayesian kNN that fully exploits the hidden space structure learned by CL. This design enables SCoRE to be easily adaptable to new PLM releases, ensures minimal energy consumption, and simplifies model training and maintenance.
2. We propose two novel RE evaluation metrics to complement traditional focus on instance-wise label counts [11,31,32]. *Precision at R (P@R)* evaluates the ranking quality of predicted relations, highlighting cases where overall F1 may be low due to thresholding choices, yet the model still captures the correct ordering of relation likelihoods. *Correlation Structure Distance (CSD)* assesses the global consistency of predictions by quantifying alignment between the learned relation co-occurrence patterns and those of the underlying KG, penalizing models that achieve good F1 but systematically conflate opposite or semantically incompatible relations.
3. We release Wiki20d, the first Fully Distantly Supervised (FDS) corpus. Wiki20d is an extension of the popular benchmark Wiki20m [31] that emulates real-world RE scenarios by annotating the training set using only KG structures while maintaining the manually annotated sentences in performance assessment.

We evaluate SCoRE against state-of-the-art models using standard and novel metrics, including environmental impact. Experiments on five benchmarks demonstrate that SCoRE matches or surpasses leading models while reducing energy consumption and improving scalability, underscoring its potential for real-world applications. To further validate our choice of a minimal architectural design, we assess the impact of incorporating advanced sentence and triplet processing techniques, such as dynamic full-sentence embedding, widely advocated in the literature. The results reveal that these methods heavily depend on model fine-tuning. Without fine-tuning, they not only fail to improve performance but also lead to performance degradation while increasing computational overhead. For reproducibility, benchmarks and source code are available at https://github.com/rioma96/SCoRE.

The paper is structured as follows. Section 2 formalizes sentence-based RE as a multi-label classification task. Section 3 details the dataset creation process, training methodology, and evaluation framework for SCoRE. In Section 4, we describe both established and novel metrics used to benchmark our approach. The datasets employed in our experiments, including a comprehensive overview of Wiki20d, are presented in Section 5. Section 6 outlines the state-of-the-art solutions used for comparison and the experimental setup of SCoRE. Section 7 provides a detailed comparison of SCoRE with state-of-the-art models, along with an analysis of the impact of modifying input, architecture, and inference configurations. Lastly, Section 8 discusses related work and highlights key insights, while Section 9 draws conclusions and outlines future work.

2. Problem definition

To formally define the problem of sentence-based multi-label RE, let us start by introducing the notion of annotated corpus and then define the task in terms of multi-label classification.

Definition 1 (Annotated corpus)**.** Let us consider a reference KG denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ and $\mathcal{R}$ are the set of entities and relation types in the KG, respectively, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the set of triples (e_h, r, e_t) indicating that there exists a relation $r \in \mathcal{R}$ between the head $e_h \in \mathcal{E}$ and tail entity $e_t \in \mathcal{E}$.

Let $\mathcal{S} = \{s_i\}_{i=1}^{\mathcal{N}}$ be a corpus of $\mathcal{N}$ sentences where $s_i \in \mathcal{S}$ denotes the ith sentence and T be a tokenizer which maps s_i into a sequence of T_i tokens $T(s_i) = [t_1, \ldots, t_{T_i}]$.
A $\mathcal{G}$-based annotation of corpus $\mathcal{S}$ is denoted as

$$C^{\mathcal{G},\mathcal{S}} = \{(s_i, \{(\tau_{i,h_j}, \tau_{i,t_j}, \mathcal{R}_{i,j})\}_{j=1}^{m_i})\}_{i=1}^{\mathcal{N}}$$

where:

- $s_i \in \mathcal{S}$ is the ith sentence in $\mathcal{S}$;

Table 1
Summary of notation for sentence-based multi-label relation extraction.

Symbol	Description
$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$	Knowledge graph (KG)
$\mathcal{E}$	Set of entities in the KG
$\mathcal{R}$	Set of relation types in the KG
$\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$	Set of triples (e_h, r, e_t)
$\mathcal{S} = \{s_i\}_{i=1}^{\mathcal{N}}$	Corpus of $\mathcal{N}$ sentences s_i
$T(s_i) = [t_1, \ldots, t_{T_i}]$	Sequence of T_i tokens for s_i
$\mathcal{C}^{\mathcal{G},\mathcal{S}}$	$\mathcal{G}$-based Annotated corpus
$\mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$, $\mathcal{C}^{\mathcal{G},\mathcal{S}''}_{test}$	Training and test annotated corpus

- $\tau_{i,h_j}(\tau_{i,t_j})$ denotes the set of indexes of the tokens in $T(s_i)$ corresponding to the mention of the head (tail) entity of the jth entity pair $(e_{h_j}, e_{t_j}) \in \mathcal{E} \times \mathcal{E}$ within the sentence s_i;
- $\mathcal{R}_{i,j} \subseteq \mathcal{R}$ is the set of relation types connecting e_{h_j} to e_{t_j} in s_i.

In the following $\mathcal{C}^{\mathcal{G},\mathcal{S}}$ will be defined as an annotated corpus.

The definition above is general enough to encompass different setups. For instance, any manually annotated corpus usually assigns one relation $r \in \mathcal{R}$ to each entity pair mentions (e_{h_j}, e_{t_j}) and thus $\mathcal{R}_{i,j}$ is a singleton, while distantly-supervised annotated corpus usually refers to a subset of the existing relations connecting e_{h_j} to e_{t_j} in $\mathcal{G}$ and therefore $\mathcal{R}_{i,j} \subseteq \{r \mid (e_{h_j}, r, e_{t_j}) \in \mathcal{T}\}$.

Definition 2 (Sentence-based Relation Extraction (RE))**.** Given an annotated corpus $\mathcal{C}^{\mathcal{G},\mathcal{S}}$ split into a training and test set, $\mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$ and $\mathcal{C}^{\mathcal{G},\mathcal{S}''}_{test}$, so that $\mathcal{S}' \cup \mathcal{S}'' = \mathcal{S}$ and $\mathcal{S}' \cap \mathcal{S}'' = \emptyset$, the goal of sentence-based relation extraction is to learn from $\mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$ a multi-label classification function $f : \mathcal{S} \times \mathbb{N} \to 2^{\mathcal{R}}$ that, for each sentence $s_i \in \mathcal{S}''$ and $j \in [1, m_i]$, predicts the subset of relation types $\hat{\mathcal{R}}_{i,j} \subset \mathcal{R}$ connecting the mentions of the entity pair (e_{h_j}, e_{t_j}) in s_i.

In realistic scenarios, where obtaining an annotated corpus for every domain is often unfeasible, any sentence-based RE solution must also be evaluated under real-world conditions where ground truth is unavailable during the training phase. In this case, the training algorithm can only rely on corpora annotated with information derived from the KG. This leads to the introduction of a specific kind of annotated corpus.

Definition 3 (Fully Distantly Supervised (FDS) annotated corpus)**.** Let $\mathcal{C}^{\mathcal{G},\mathcal{S}}$ be an annotated corpus split into a training $\mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$ and a test set $\mathcal{C}^{\mathcal{G},\mathcal{S}''}_{test}$. $\mathcal{C}^{\mathcal{G},\mathcal{S}}$ is *fully distantly supervised* when, for each $s_i \in \mathcal{S}'$, each $R_{i,j} \in \mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$ is the full set of relation types connecting e_{h_j} to e_{t_j} in $\mathcal{G}$, i.e, $R_{i,j} = \{r \mid (e_{h_j}, r, e_{t_j}) \in \mathcal{T}\}$.

For ease of reference, Table 1 summarizes the introduced symbols.

3. SCoRE

Fig. 1 illustrates the overall workflow of SCoRE. In the training phase:

1. each sentence in $\mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$ goes through a single forward pass of a PLM encoder to get an hidden vector representation of its head and tail entity mention pairs. Each hidden vector is associated with the one-hot vector of the related relation types.
2. A multi-layer perceptron (MLP) maps head-tail encodings onto a hypersphere under a multi-label supervised CL framework, clustering samples with similar relational patterns.

In the testing phase, relation type prediction on unseen head-tail entity mention pairs contained in the sentences of $\mathcal{C}^{\mathcal{G},\mathcal{S}''}_{test}$ is performed by first getting the corresponding encodings and then using a non-parametric multi-label Bayesian kNN approach.

3.1. Dataset creation

Given an annotated corpus $\mathcal{C}^{\mathcal{G},\mathcal{S}}$, each sentence $s_i \in \mathcal{C}^{\mathcal{G},\mathcal{S}}$, for $i \in [1, \mathcal{N}]$, is mapped into a sequence of token embeddings via a single forward pass through the Encoder function of the PLM:

$$\mathbf{H}_i = \text{Encoder}(T(s_i)) = [\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \ldots, \mathbf{h}_{i,T_i}] \tag{1}$$

where $\mathbf{h}_{i,l} \in \mathbb{R}^h$ is the embedding of the lth token in s_i.

Then, we construct the input vector $\mathbf{x}_{i,j}$ of each entity mention pair $(\tau_{i,h_j}\tau_{i,t_j})$, for $j \in [1, m_i]$, by concatenating the average embedding of the corresponding tokens:

$$\mathbf{x}_{i,j} = [\mathbf{e}_{i,h_j}; \mathbf{e}_{i,t_j}] \in \mathbb{R}^{2h}. \tag{2}$$

where

$$\mathbf{e}_{i,h_j} = \frac{1}{|\tau_{i,h_j}|} \sum_{t \in \tau_{i,h_j}} \mathbf{h}_{i,t}\,, \quad \mathbf{e}_{i,t_j} = \frac{1}{|\tau_{i,t_j}|} \sum_{t \in \tau_{i,t_j}} \mathbf{h}_{i,t}. \tag{3}$$

The output vector $\mathbf{y}_{i,j}$ corresponding to $\mathbf{x}_{i,j}$ is represented as a one-hot encoding $\mathbf{y}_{i,j} \in \{0,1\}^R$, where $R = |\mathcal{R}|$ is the number of relation types. This vector represents set of relation types $\mathcal{R}_{i,j}$ connecting e_{h_j} to e_{t_j} in s_i, i.e. $y^k_{i,j} = (\mathbf{y}_{i,j})^k = 1$ if $r_k \in \mathcal{R}_{i,j}$, and $y^k_{i,j} = 0$ otherwise.

The aforementioned method can be seamlessly applied to both the training $\mathcal{C}^{\mathcal{G},\mathcal{S}'}_{train}$ and test $\mathcal{C}^{\mathcal{G},\mathcal{S}''}_{test}$ corpora, giving rise to a training and test dataset. For simplicity, we introduce a unified indexing scheme in the following, with a slight abuse of notation, and refer to both the training and test datasets as:

$$\mathcal{D}_{train} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\ldots,N} \qquad \mathcal{D}_{test} = \{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1,\ldots,\tilde{N}}$$

where $N = \sum_{i:s_i \in \mathcal{S}'} m_i$ and $\tilde{N} = \sum_{i:s_i \in \mathcal{S}''} m_i$ are the total number of training and test data samples respectively, with the double indexing dropped for clarity.

Note that the LLM encoder is employed solely during dataset creation, serving as an informed prior. This design prevents overfitting and reduces hallucinations that could arise from fine-tuning a large, expressive model on a limited and noisy training set, thereby preserving the encoder's ability to interpret raw text broadly. The approach can be seamlessly adapted to any PLM. Even with models supporting larger context windows or hidden dimensions, the memory overhead increases only with the hidden representation. By storing only the averaged head and tail embeddings, the storage footprint remains limited to two token-level encodings. Moreover, as the annotated corpus requires just one PLM forward pass, the overall computational cost remains minimal.

3.2. CL architecture and loss function

The next step consists of a supervised CL solution [33] relying on the following architecture. The input gets processed by a simple MLP architecture of l layers with m neurons, each with $l \ll m$ to ensure perturbative and stable behavior [34]. This is followed by a single smaller layer with $m_h < m$ neurons whose outputs get normalized to unit vectors based on L_2 norm and thus mapped onto the hypersphere $\mathbb{S}^{m_h}$.

To enable the prediction of multiple relations between entity pairs, we employ the adaptation of [33] to multi-label settings proposed by [35] that works as follows. Given the training set $\mathcal{D}_{train} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\ldots,N}$, a distance measure $d(\cdot,\cdot)$, and calling $\mathbf{z}_i = \text{MLP}(\mathbf{x}_i) \in \mathbb{S}^{m_h}$ the encoding of the ith sample via the MLP, the loss function is given by:

$$\mathcal{L} = -\frac{1}{N} \sum_{i \in [1,N]} \sum_{j \in [1,N], j \neq i} \beta_{ij} \log \frac{e^{d(\mathbf{z}_i, \mathbf{z}_j)/\tau}}{\sum_{k \in [1,N], k \neq i} e^{d(\mathbf{z}_i, \mathbf{z}_k)/\tau}} \tag{4}$$

where $\tau \in \mathbb{R}^+$ is a temperature parameter, and

$$\beta_{ij} = \frac{\mathbf{y}_i^T \cdot \mathbf{y}_j}{\sum_{k \in [1,N], k \neq i} \mathbf{y}_i^T \cdot \mathbf{y}_k}\,.$$

It can be easily seen that $\mathbf{y}_i^T \cdot \mathbf{y}_j$ is the number of shared relations between the two encoded sets and β_{ij} is its normalized version.

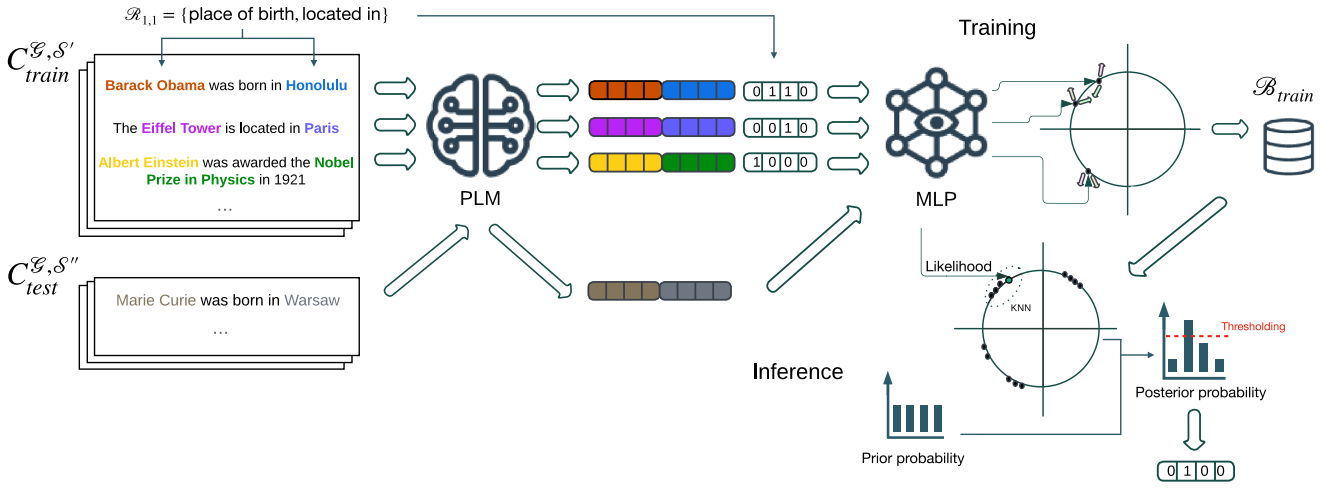


Fig. 1. Visual representation of SCoRE, showcasing the dataset creation, training, and testing stages.

The MLP weights are trained for a certain number of epochs to minimize $\mathcal{L}$. This formulation ensures that, by the end of the training, any two entity pairs that are connected through similar sets of relation types in $\mathcal{D}_{train}$ are positioned close together in the hidden feature space. In contrast, pairs linked by distinct relationships are placed farther apart.

3.3. Bayesian kNN for class prediction

As opposed to standard CL-based methods for classification, which follow a two-step approach [33], i.e., using CL for pretraining followed by classification, we estimate test set relation type probabilities right after CL training by leveraging a probabilistic kNN approach in the hidden feature space.

In particular, similarly to [35], we create a multi-dimensional datastore of the training set hidden representations and labels $\mathcal{B}_{train} = \{\mathbf{z}_i, \mathbf{y}_i\}_{i=1\dots N}$. Afterward, for each test example, $\tilde{\mathbf{x}}_j$, we obtain its hidden representation $\tilde{\mathbf{z}}_j = MLP(\tilde{\mathbf{x}}_j)$ and search the datastore $\mathcal{B}_{train}$ for its k-nearest neighbors kNN($\tilde{\mathbf{z}}_j$), according to the distance measure $d(\cdot,\cdot)$ used in the CL loss.

Then, we compute $P(r_h \mid \tilde{\mathbf{z}}_j)$, for each $r_h \in \mathcal{R}$, as follows. As shown in [36], a probabilistic multi-class kNN method requires estimating the posterior class probabilities as:

$$P(r_h|\tilde{\mathbf{z}}_j) = \frac{P(\tilde{\mathbf{z}}_j|r_h)\cdot P(r_h)}{P(\tilde{\mathbf{z}}_j)}. \tag{5}$$

where $P(\tilde{\mathbf{z}}_j|r_h)$ is the probability density of $\tilde{\mathbf{z}}_j$ given by the kNNs conditional on class r_h, $P(r_h)$ is the prior class probability, and $P(\tilde{\mathbf{z}}_j) = \sum_{i=1}^{R} P(\tilde{\mathbf{z}}_j|r_i)\cdot P(r_i)$ is the sample evidence. In standard situations, a common assumption is that the points are equally distributed and $P(\tilde{\mathbf{z}}_j|r_h) \propto \frac{k_h}{n_h V(\tilde{\mathbf{z}}_j)}$; where k_h is the number of neighbors belonging to class r_h and $V(\tilde{\mathbf{z}}_j)$ is the volume of the hypersphere centered at $\tilde{\mathbf{z}}_j$ and containing all kNNs. However, CL enforces similarity between samples based on label set overlap, so a uniform density of points is not realistic. A reasonable solution is to model the point probability conditioned on each class following the CL loss function formulation as follows:

$$P(\tilde{\mathbf{z}}_j|r_h) \propto \sum_{i:\mathbf{z}_i\in \text{kNN}(\tilde{\mathbf{z}}_j)} y_i^h \cdot e^{-\frac{d(\tilde{\mathbf{z}}_j,\mathbf{z}_i)}{\tau}}. \tag{6}$$

To make the aforementioned approach suitable for multi-label classification, we leverage Eq. (5) to perform R probabilistic kNN binary classification problems where the kth task aims to predict whether the relation r_h appears or not in the entity pair mention encoded in $\tilde{\mathbf{x}}_j$. In practice, the absence of a label in kNN members is treated as explicit information disfavoring the presence of such label in the final prediction. The posterior class probabilities are then given by:

$$P(r_h|\tilde{\mathbf{z}}_j) = \frac{\sum_{i:\mathbf{z}_i\in \text{kNN}(\tilde{\mathbf{z}}_j)} P(r_h)\cdot y_i^h \cdot e^{-\frac{d(\tilde{\mathbf{z}}_j,\mathbf{z}_i)}{\tau}}}{\sum_{i:\mathbf{z}_i\in \text{kNN}(\tilde{\mathbf{z}}_j)} \left[P(r_h)\cdot y_i^h + P(\overline{r_h})\bar{y}_i^h\right] e^{-\frac{d(\tilde{\mathbf{z}}_j,\mathbf{z}_i)}{\tau}}} \tag{7}$$

where $P(\overline{r_h}) = (1 - P(r_h))$ is the probability that the relation r_h does not appear in the kNN labels (so that $P(r_h) + P(\overline{r_h}) = 1$) and $\bar{y}_i^h = (1 - y_i^h)$. This formulation treats the presence or absence of each relation as an independent Bernoulli event. While this assumption simplifies inference, inter-label dependencies are not ignored: they are captured in the geometry of the supervised multi-label contrastive space (Section 3.2), where sentences with overlapping label sets are embedded closer together. In this sense, the independence assumption can be seen as a weak diagonal prior applied locally on top of neighborhoods that already encode label co-occurrence. As will be shown by our correlation analysis in Section 7, this approximation preserves inter-label structure in practice. Selecting an appropriate prior probability $P(r_h)$ is crucial in Bayesian inference as it reflects initial beliefs about class distributions before observing the data. The choice of prior can significantly influence predictions, especially in imbalanced or small datasets where data-driven likelihoods may be insufficient to reliably infer class probabilities.

In our setting, we follow the prescription of [36] and use an uninformative or flat prior distribution for each binary classification problem, i.e., the prior probabilities that a class appears or does not appear in the kNNs are equal, i.e., $P(r_h) = 1/2\ \forall h \in [1, R]$. Choosing flat prior probabilities is equivalent to adopting a dynamic, local neighborhood-dependent class weight. This strategy focuses on optimizing recall over precision, thereby minimizing the risk of missing positive instances (false negatives) [36]. In imbalanced classification problems, such as classifying relation type distributions in text, prioritizing recall is essential as it improves the detection of long-tail relation types (true positives), even if it leads to an increase in false positives. Moreover, if CL training is successful, relation types will be far from equally distributed in the hidden feature space, making the flat prior assumption even more reasonable.

After computing posterior class probabilities, we use the following universal thresholding to get sharp class predictions $\hat{\mathbf{y}}(\tilde{\mathbf{x}}_j)$:

$$(\hat{\mathbf{y}}(\tilde{\mathbf{x}}_j))^h = \hat{y}_j^h = \begin{cases} 1, & \text{if } P(r_h|\tilde{\mathbf{z}}_j) > c, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where $c \in [0, 1]$ is a threshold commonly shared among classes.

4. Evaluation metrics

In this section, we outline the evaluation criteria for assessing both the performance and environmental impact of sentence-level RE solutions. We start with commonly used RE metrics, explicitly defining their extensions to multi-label classification. Moreover, we illustrate the process we use to rank predictions by model confidence and define the process used to measure environmental impact. Finally, we introduce additional metrics not yet explored in RE literature that are essential for evaluating result quality and reliability of results.

4.1. MicroF1 and macroF1

Two of the most common metrics for performance evaluation in RE are micro- and macro-averaged F_1 score, named microF1 and macroF1, respectively. While originally used in multi-class classification models, these metrics can be straightforwardly adapted to multi-label settings. In particular,

$$\text{microF1} = \frac{TP}{TP + 0.5 \cdot (FN + FP)} \tag{9}$$

where TP, FP, and FN denote the total number of true positives, false positives, and false negatives across all classes, respectively. This metric emphasizes the overall accuracy of the model in predicting the correct relationships for all instances. The macroF1 score is obtained by computing the microF1 score for each class individually and then averaging the results:

$$\text{macroF1} = \frac{1}{R} \sum_{h=1}^{R} \frac{TP_h}{TP_h + 0.5 \cdot (FN_h + FP_h)}\,. \tag{10}$$

This formula helps evaluate the model's performance across different relation types, regardless of their frequency in the dataset. Note that in those datasets where only part of the labels appear in the test set, we adopt the convention of considering, in macroF1, only those classes for which $FP_h + FN_h + TP_h > 0$ as the class-specific F_1-score would be unspecified otherwise.

4.2. Ranking solutions by model confidence: micro- and macroF1@M

A standard metric applied in RE problems consists of reporting the microF1 and macroF1 values (or the precision values) on the M test set elements for which the model is most confident, denoted as microF1@M and macroF1@M, respectively. While ranking solutions according to model confidence is straightforward in multi-class classification,[1] in multi-label setting the exact formula to be employed was not clearly defined in most works.

In this paper, we apply the following scoring function to rank test set elements by model confidence. Calling $P_{\tilde{\mathbf{z}}_j} = \{P(r_h|\tilde{\mathbf{z}}_j)_{i=1,\dots,R} \mid \hat{y}_j^h = 1\}$ and $|P_{\tilde{\mathbf{z}}_j}|$ the cardinality of this set $s(\tilde{\mathbf{z}}_j)$, we define the confidence score of the model on the test sample $\tilde{\mathbf{z}}_j$ as the harmonic mean of the posterior class probabilities associated with positive predictions:

$$s(\tilde{\mathbf{z}}_j) = \left[\prod_{P(r_h|\tilde{\mathbf{z}}_j) \in P_{\tilde{\mathbf{z}}_j}} P(r_h|\tilde{\mathbf{z}}_j)\right]^{1/|P_{\tilde{\mathbf{z}}_j}|}. \tag{11}$$

The reason behind this choice is the following. If we consider the presence of different classes as independent events, then the probability of observing multiple classes is simply given by the product of the individual class probabilities. However, choosing this product as the final score would disproportionately penalize predictions containing multiple labels. By using the harmonic mean, we create a standard score that removes the distortion associated with multiple labels, summarizing them in a single "probability value". Once the data are ranked by confidence $s(\tilde{\mathbf{z}}_j)$, micro-F1 and macro-F1 scores can be recomputed on the top M most confident samples.

We note that, while @M metrics can be useful for evaluating model calibration under idealized conditions, they are influenced by the number of classes present in the most confident samples, a quantity typically unknown a priori. Consequently, these metrics often emphasize performance on a subset of relation classes, potentially distorting overall performance assessments.

4.3. Carbon footprint

Given the critical importance of sustainability in real-world settings, we extended our evaluation to include an energy consumption metric, aiming to support the development of resource-efficient relation classification methods. Energy consumption was quantified in kilowatt-hours (kWh) using CodeCarbon [37], a well-established tool that automatically detects the hardware specifications and measures consumption in real-time based on system usage.

4.4. P@R: precision at R

P@R is commonly used in information retrieval to evaluate the relevance of a system's top R ranked predictions. In sentence-level RE, it provides an instance-level indicator of how effectively a model performs as a recommender, where ranking quality is paramount and domain experts primarily judge the correctness of the top few results. Formally, P@R is the proportion of true positives among the top R predictions, reflecting the model's ability to pinpoint relevant classes for each instance. When R matches the number of ground-truth relevant classes per instance, P@R effectively measures the model's precision at exactly the point where all true relevant labels should appear.

To compute this metric, given the number true labels $R_j = \sum_{i\in[1,R]} \tilde{y}_j^i$ for each test sample $\tilde{\mathbf{x}}_j$, we rank the posterior probabilities $P(r_h|\tilde{\mathbf{z}}_j)$ in increasing order and create a vector $\hat{\mathbf{y}}_{R_j}(\tilde{\mathbf{x}}_j)$ assigning a positive prediction to the top R_j posterior probabilities.

The P@R score is then defined as the average value of the ratio of true positives in the top R posterior probabilities for all test samples:

$$\text{P@R} = \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} \frac{|\hat{\mathbf{y}}_{R_j}(\tilde{\mathbf{x}}_j) \cap \tilde{\mathbf{y}}_j|}{R_j}. \tag{12}$$

This approach offers a direct evaluation of the model's ability to retrieve all relevant labels without considering irrelevant predictions, as long as the number of relevant classes aligns with R. Furthermore, it adapts to instances with varying numbers of true relation classes.

4.5. Correlation structure distance (CSD)

In multi-label RE, the relational types that co-occur for the same entity mention pair, frequently exhibit non-trivial relationships. For instance, the relation "lives in" is often accompanied by "born in" for a person-location pair, while relations such as "father of" and "mother of" (or "son of") for a person-person pair should never appear together.

Traditional performance metrics do not quantify how well a model preserves the underlying correlation structure among labels. To achieve this goal, we introduce the Correlation Structure Distance (CSD), which quantifies the discrepancy between the correlation matrix of the true test relation types and that of the predicted ones, providing a deeper evaluation of model robustness and alignment with the underlying data structure. Specifically, using a set of one-hot encoding vectors $\mathbf{Y} = \{\mathbf{y}\}_{j=1\dots,\tilde{N}}$, we compute the correlation between each relation pair (r_h, r_p), for each $r_h, r_p \in \mathcal{R}$ using the Pearson ϕ coefficient:

$$\phi(\mathbf{Y}, r_h, r_p) = \frac{n_{11}^{(h,p)} n_{00}^{(h,p)} - n_{01}^{(h,p)} n_{10}^{(h,p)}}{\sqrt{n_{1\cdot}^{(h,p)} n_{0\cdot}^{(h,p)} n_{\cdot 1}^{(h,p)} n_{\cdot 0}^{(h,p)}}} \tag{13}$$

[1] it corresponds to sorting the test set elements according to the values of the maximum class posterior probability

Table 2
Summary of datasets and their characteristics after pre-processing operations.

Dataset	Ref. KG	Ann. Train	Ann. Test	N. Rel	Len Train	Len Val	Len Test	ML Train	ML Test
DisRex (Eng)	Wikidata	DS	DS	36	128,241	18,304	33,399	18 %	27 %
Wiki20m	Wikidata	DS	Man	80	276,260	17,485	101,861	3 %	1 %
NYT10m	Freebase	DS	Man	24	72,961	9172	6642	12 %	19 %
NYT10d	Freebase	DS	DS	55	80,542	–	4892	44 %	21 %
Wiki20d	Wikidata	FDS	Man	755	614,207	56,187	92,083	20 %	1 %

where

- $n_{11}^{(h,p)} = \sum_{j=1}^{\tilde{N}} y_j^h y_j^p$ and $n_{00}^{(h,p)} = \sum_{j=1}^{\tilde{N}} (1-y_j^h)(1-y_j^p)$ are the number of co-occurrences of positive and negative labels in the two classes, respectively;
- $n_{01}^{(h,p)} = \sum_{j=1}^{\tilde{N}} (1-y_j^h) y_j^p$ and $n_{10}^{(h,p)} = \sum_{j=1}^{\tilde{N}} \hat{y}_j^h (1-y_j^p)$ count the number of cases where the labels disagree;
- $n_{1\cdot}^{(h,p)} = \sum_{j=1}^{\tilde{N}} y_j^h$ and $n_{\cdot 1}^{(h,p)} = \sum_{j=1}^{\tilde{N}} \hat{y}_j^p$ indicate the total number of cases where r_h or r_p get value 1 in $\hat{\mathbf{Y}}$ respectively. Similar considerations hold for $n_{0\cdot}$ and $n_{\cdot 0}$.

Given the true test set labels $\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_j\}_{j=1\ldots,\tilde{N}}$ and the predicted ones $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}(\tilde{x}_j)\}_{j=1\ldots,\tilde{N}}$, we compute the distance between $\phi(\tilde{\mathbf{Y}}, r_h, r_p)$ and $\phi(\hat{\mathbf{Y}}, r_h, r_p)$ using the Frobenius norm:

$$\mathrm{CSD} = \sqrt{\sum_{h=1}^{R}\sum_{p=1}^{R} \left|\phi(\hat{\mathbf{Y}}, r_h, r_p) - \phi(\tilde{\mathbf{Y}}, r_h, r_p)\right|^2}. \tag{14}$$

The CSD values are inversely proportional to the ability of the model to reflect the ground truth relationships between labels.

5. Datasets

Five annotated corpus were considered to conduct the experiments: four are well-established benchmarks for RE, while the fifth, Wiki20d, is our released FDS annotated corpus. Their features are summarized in Table 2, detailing for each annotated corpus the reference KG, the generation method (DS - Distant Supervised, Man - Manual, FSD - Full Distant Supervised), the total number of relation types, and the dataset split size. The table also highlights the varying complexities of these datasets in terms of the number of relations and the percentage of multi-label instances in their training and test sets (ML Train/Test).

Specifically, the first four benchmarks are: **NYT10d** [8], a benchmark constructed via DS by linking mentions in the New York Times corpus to Freebase; **NYT10m** [31], a manually curated version of NYT10d with enhanced annotation quality and additional validation splits; **Wiki20m** [31] generated via DS by aligning Wikipedia articles with Wikidata and providing a manually annotated test set; **DisRex** [38], a multilingual dataset created using DS and Wikidata, designed to balance relation types and include inverse relations to verify that a model truly learns the proper ordering of entity pairs. Since we use the English version of BERT, we consider only the English portion of DisRex.

Finally, we introduce **Wiki20d**, an extension of Wiki20m obtained by processing each sentence and querying Wikidata to extract relations between identified head-tail entity pairs. By relying on FDS, Wiki20d features a substantially larger number of relation classes that are heavily imbalanced (see Fig. 2), enabling a more realistic assessment of methods in scenarios that demand the extraction of complex, multi-label relational information. Wiki20d was constructed from the 2023-06-19 Wikidata dump, using canonical Wikidata IDs for entity linking.

To ensure consistent preprocessing across datasets and align them with the multi-label RE task, we applied several data-cleaning steps. First, we removed sentences labeled with the 'NA' relation, as they introduced no meaningful relationships and could distort performance assessments. We also removed sentences where entity token positions exceeded the PLM context window, as they would impede proper representation learning.

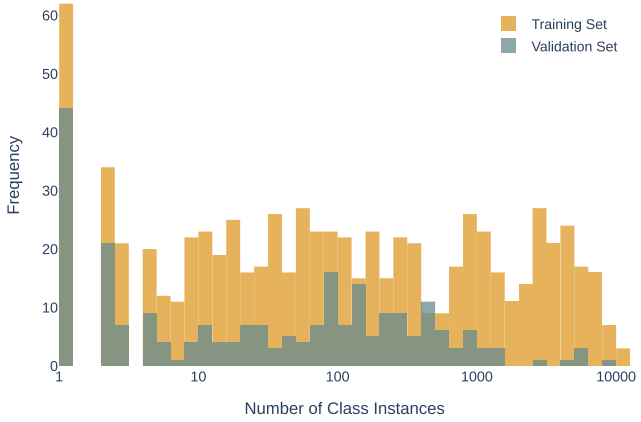


Fig. 2. Wiki20d relation class distribution.

6. Experimental setup

6.1. Settings used in SOTA models

We compare SCoRE with four sentence-level RE methods and one bag-level RE method:

- **KGPool** [39]: a sentence-level RE approach that employs a dynamic context augmentation mechanism to incorporate only the KG facts directly relevant to the sentence. The method utilizes a BiLSTM to independently encode representations for sentences, entities, and the contextual information of KG entities, constructing a Heterogeneous Information Graph (HIG). To refine this structure, KGPool applies a Self-Attention-Based Graph Convolutional Network to reduce the HIG into a more concise Context Graph. Finally, a context aggregator is employed to jointly learn from both the Context Graph and sentence-level representations.
- **PARE** [13]: a bag level RE approach that concatenates all sentences within a bag, defined by the head and tail entities, into a single passage by sequentially sampling sentences without replacement. The passage is then encoded using BERT to generate contextualized embeddings for every token. A relation query vector is subsequently employed to create a relation-aware summary of the entire passage through an attention mechanism. This summary is fed into an MLP, which outputs the probability of the corresponding relation triple.
- **HiCLRE** [27]: a sentence and bag level approach employing multi-granularity contextualization to capture cross-level structural information using multi-head self-attention across entity, sentence, and bag levels. This recontextualization aligns context-aware features from each level, refining semantic representations. Additionally, dynamic gradient adversarial perturbation improves robustness by using gradient-based CL to create pseudo-positive samples, enhancing the model's ability to distinguish relations.
- **SSLRE** [28]: a sentence-level approach that addresses the noise in DS by discarding only the labels of noisy samples and treating these instances as unlabeled. It employs a weighted k-NN graph to select confident samples as labeled data, while the rest are treated as unlabeled. The framework then uses a semi-supervised learning

Table 3
Grid search parameter configuration for MLP and loss parameters. Bold values indicate the selected or optimal ones.

Parameter	Values
`MLP - layers` (l)	[3, 4, **5**]
`MLP - depth/width` (l/m)	[**0.01**, 0.05, 0.1]
`MLP - output dims` (m_h)	[5, 10, **15**]
`MLP - activation`	[**swish**, ReLU]
`loss - distance`	[**euclidean**, cosine]
`loss - temperature` (τ)	[**0.01**, 0.05, 0.1, 0.2]
`learning_rate`	[10^{-4}, $\mathbf{5 \cdot 10^{-4}}$, 10^{-3}]
`batch_size`	[64, 128, **256**]
`kNN` (k)	[5, 10, 15, 50, 100, 150]
`probability threshold` (c)	[0.3, 0.4, 0.5, 0.6, 0.7]

approach to handle remaining label noise and effectively utilize unlabeled samples.

- **TIW** [26]: a sentence-level RE tool that uses a transitive instance weighting mechanism combined with self-distilled BERT for denoise DS sentence-level training in RE. The method fine-tunes the BERT encoder, then fixes its parameters and trains student classifiers using knowledge distillation. TIW generates dynamic instance weights to reduce noise and overfitting by considering uncertainty and consistency. Students choose between the teacher's and previous peer's outputs based on consistency, while false negative filtering and positive weighting adjust weights for negative and positive instances, respectively.

Since SCoRE operates at the sentence level, its results can be directly compared with all competitors except PARE without modifications. However, we were only partially able to reproduce KGPool's results due to the limited flexibility of the code, which made adapting it to different datasets beyond those used in the original paper challenging. Additionally, because the codes for [28] and [26] were not publicly released, we could only report the results presented in their original publications.

In contrast, since PARE is designed for bag-level predictions, we adapted it for sentence-level evaluation. To achieve this, we trained the model at the bag level and, during the prediction phase, provided one sentence at a time as input. This approach guarantees that PARE predictions are directly comparable to sentence-level methods. PARE's flexible design enabled us to reproduce results for all the datasets considered.

6.2. SCoRE configuration

For dataset creation, we employed the BERT-base model, utilizing a context window of 512 tokens, an embedding dimension of 768, and 12 transformer layers. We performed a grid search across a range of hyperparameters to optimize the MLP model and the Bayesian kNN approach. The grid search was conducted on all datasets, with the parameter configurations reported in Table 3.

Each experiment was conducted over 30 training epochs, using the AdamW optimizer, with early stopping employed to monitor the validation set loss, when available. Early stopping was triggered when the loss ceased to decrease, with a patience of 5 epochs, and the model weights were reverted to their best configuration at the point of optimal performance. To ensure reproducibility, we ran all experiments with five random seeds and adopted a simple but consistent grid search strategy. Model selection was guided by the bi-objective hypervolume over micro-F1 and macro-F1, evaluated on the validation set (or on the training set if no validation split was available). We identified a common configuration for the contrastive learning training that performed well across all datasets (highlighted in bold). To promote generality and limit per-dataset adaptation, we fixed these hyperparameters globally. The only hyperparameters that could not be fixed across datasets are related to the prediction stage: c and `kNN`. This difference is reasonable, as `kNN` regulates the local probability density estimate used to compute posterior class probabilities, while the probability threshold, c, determines how to convert these probabilities into sharp predictions. The optimal values for these hyperparameters may depend on dataset-specific characteristics, such as differences in dataset balance and the nature of the underlying relations. The optimal threshold and kNN values for each dataset were selected based on results from the validation set, or from the training set in the absence of a validation set. The corresponding values are reported in Table 4.

Table 4
The optimal threshold (c) and kNN values (k) for each dataset.

Dataset	c	k
`Nyt10m`	0.6	50
`Nyt10d`	0.7	100
`DisRex`	0.5	50
`Wiki20m`	0.5	100
`Wiki20d`	0.7	150

7. Results

All experiments were conducted on the Leonardo Booster partition, using a BullSequana X2135 "Da Vinci" single-node GPU Blade equipped with a 32-core Intel Xeon Platinum 8358 CPU (Ice Lake), 512 GB of DDR4 RAM, and four NVIDIA Ampere A100 GPUs (64GB HBM2e) interconnected via NVLink 3.0.

7.1. Performance against SOTA

In this section, we provide a detailed evaluation of our model's performance and carbon footprint, comparing them against state-of-the-art models.

Micro and MacroF1. We begin by comparing the models' performance using the commonly adopted micro- and macro-F1 metrics across various benchmark datasets. As previously mentioned, we were only partially able to reproduce some results due to limited code flexibility or the absence of publicly available repositories. The results in Table 5 present the best performance over five runs replicating the competitors. We assess the statistical significance of differences in micro- and macro-F1 scores across methods using pairwise two-tailed t-tests among the three reproducible approaches. When the results indicate a complete ranking of the methods, the best-performing method is highlighted in bold, and the second-best is underlined. In cases where two methods are statistically indistinguishable and both outperform the third, the equivalent methods are underlined.

Our findings show that SCoRE achieves competitive performance across all benchmarks, maintaining simplicity without sacrificing effectiveness. According to the statistical analysis reported in Table 5, SCoRE significantly outperforms competing models on the Nyt10d dataset, where it achieves the highest micro- and macro-F1 scores. On Nyt10m, SCoRE and PARE obtain statistically comparable results, both outperforming the remaining approaches. For DisRex, PARE exhibits a statistically significant advantage over SCoRE in both micro- and macro-F1. However, this improvement comes at the cost of predicting opposite or semantically inconsistent relations, an aspect further analyzed in the following section. Conversely, HiCLRE reaches the best performance on Wiki20m, with significantly higher scores than other models, although this advantage appears tied to datasets characterized by lower multi-label complexity and more balanced class distributions. On Wiki20d, SCoRE achieves the best macro-F1 and ranks among the top-performing methods in micro-F1, highlighting its robustness under high label imbalance.

Overall, these results confirm that increased model complexity, such as incorporating structured knowledge through KGPool or heavily fine-

Table 5
Performance comparison of SCoRE with state-of-the-art models using micro-F1 and macro-F1 metrics. results from non-reproducible methods marked by an asterisk (*).

Model	Metric	Nyt10m	Nyt10d	DisRex	Wiki20m	Wiki20d
KGPOOL	microF1	–	72.0	–	–	–
	macroF1	–	41.8	–	–	–
HiCLRE	microF1	68.7	75.2	61.2	**87.6**	68.3
	macroF1	34.7	18.4	51.3	**86.3**	06.9
SSLRE	microF1	63.8*	–	–	81.5*	–
	macroF1	–	–	–	–	–
TIW	microF1	63.8*	55.3*	–	–	–
	macroF1	35.2*	–	–	84.1*	–
PARE	microF1	77.5	86.4	**77.1**	83.8	67.2
	macroF1	38.9	46.3	**70.5**	83.7	19.6
SCoRE	microF1	77.6	**89.2**	75.4	83.5	66.9
	macroF1	40.8	49.6	62.6	80.7	**23.9**

Table 6
Label correlation matrix distance via the CSD.

Dataset	PARE	SCoRE
Nyt10m	1.38	**1.08**
Nyt10d	1.23	**0.43**
DisRex	1.99	**1.15**
Wiki20m	0.90	**0.18**
Wiki20d	5.30	**2.31**

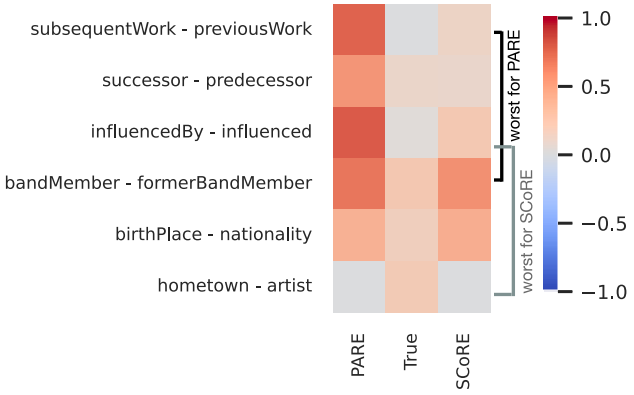


Fig. 3. PARE and SCoRE worst correlation distance entries.

Table 7
Comparison of energy consumption (kWh).

Dataset	PARE SL	PARE BAG	SCoRE	
			Dataset	Train & Test
Nyt10m	14.78	2.11	0.63	0.002
Nyt10d	12.39	1.47	0.81	0.003
DisRex	271.67	3.93	1.43	0.003
Wiki20m	743.25	2.55	2.07	0.005
Wiki20d	1561.84	6.20	5.75	0.06

tuning contextual embeddings as in HiCLRE, does not consistently yield statistically superior outcomes. While PARE remains a strong competitor, particularly on DisRex, SCoRE demonstrates comparable or second-best performance across most datasets despite its minimal architecture. For this reason, in subsequent analyses, we focus on PARE as the main reference for comparison with SCoRE. To further investigate this hypothesis, we now examine the label correlation matrix distance using the CSD.

Label Correlation Structure Distance (CSD). The comparison between SCoRE and PARE CSD values in Table 6 highlights that SCoRE consistently outperforms PARE across all five benchmark datasets, with lower CSD values signifying better alignment with the true relational structure. This is especially notable for the Nyt10m, Nyt10d, and Wiki20d datasets, which feature highly imbalanced relation labels and substantial DS noise, underscoring SCoRE's effectiveness in maintaining accurate relational alignments despite these challenges. Also in the Wiki20m dataset, which contains only a small percentage of multi-label samples in both the training (3 %) and test sets (1 %), SCoRE still demonstrates superior alignment.

Surprisingly, despite PARE's strong F1 performance, its CSD values are much higher than SCoRE's on the DisRex dataset. This counterintuitive result is further clarified through the analysis presented in Fig. 3, which compares the highest 4 misalignments of the correlation matrix entries between PARE and SCoRE against the ground truth labels. The analysis reveals that PARE struggles to differentiate between relations with opposite meanings, such as "followed" versus "followed-by" and "previous work" versus "subsequent work." By emphasizing token-level "hints" for individual relations, PARE's attention mechanisms tend to over-focus on local cues while neglecting global compatibility, which often leads to simultaneous prediction of opposite relations, resulting in superficially high F1 scores but a degraded correlation structure. This limitation is critical in RE tasks, as it can lead to the conversion of asymmetric relations into symmetric ones, thereby distorting the logical structure of the KG. Furthermore, this misalignment may violate transitivity properties, resulting in erroneous inferences and compromising the integrity of the KG.

Environmental Impact. We now compare the energy consumption of our solution, SCoRE, with that of PARE using CodeCarbon, which measures energy usage in kilowatt-hours (kWh). All measurements were obtained using CodeCarbon's default configuration, which automatically detects hardware specifications and assigns the regional carbon intensity according to the machine's physical location. GPU utilization was monitored with CodeCarbon's default sampling interval. Since all models were executed on the same machine under these identical settings, the reported energy estimates are directly comparable.

Specifically, for SCoRE, energy consumption was recorded separately during dataset preprocessing and the training/testing phases. In contrast, for PARE, a single measurement encompassed the entire process, as its model dynamically generates embeddings during each training and testing cycle. The results, presented in Table 7, demonstrate significant differences in energy efficiency, further stressed by the distinct data processing methods employed by each model.

PARE was adapted to operate on a sentence-by-sentence basis, resulting in considerable energy overhead. In this configuration, PARE creates a virtual test dataset containing only one sentence, necessitating repeated dataset loading, embedding calculation, and inference for each individual sentence during testing. This inefficiency leads to high

energy consumption, particularly for larger datasets such as Wiki20m (743 kWh). Conversely, the SCoRE model separates dataset creation from training and testing, thereby significantly reducing energy usage. For instance, SCoRE's dataset creation phase for Wiki20d requires 5.75 kWh, while subsequent training and testing demand only minimal energy (0.06 kWh for Wiki20d). For smaller datasets, SCoRE's energy consumption during training and testing is as low as 0.002-0.003 kWh, offering a more sustainable and energy-efficient approach, especially for large-scale datasets where PARE's inefficiency is most pronounced.

Additionally, it is important to note that even when PARE is utilized for bag-level RE, its energy consumption for training and testing remains substantially higher than that of SCoRE. This is due to the necessity of repeated embedding calculations and LLM weight updates for each training epoch. For example, training and testing PARE on Wiki20m at the bag level consumes approximately 2.5 kWh. In realistic settings, where training or continual learning must be performed periodically to keep the system updated, adopting a RE solution that matches the efficiency of existing models while reducing energy consumption by up to three orders of magnitude significantly enhances sustainability and lowers operational costs.

7.2. Embedding strategies and their impact

An essential component of the SCoRE pipeline is how entity mention representations are constructed from token-level embeddings. To assess this, we evaluated several embedding strategies that differ in the way they aggregate contextualized token vectors. Specifically, we considered:

- **Average pooling**: all tokens in a mention are averaged.
- **First token** and **last token**: use the representation of the boundary tokens.
- **Max pooling**: selects the most salient token dimension-wise.
- **Boundary concatenation**: concatenates the first and last token embeddings.
- **Mean_max**: concatenates the averaged and max-pooled embeddings.

It is important to note that *boundary* and *mean_max* embeddings produce representations of doubled dimensionality compared to the other strategies, potentially introducing additional computational overhead. The results in Table 8 highlight that while different strategies yield relatively consistent performance, average pooling emerges as the most effective and balanced option across datasets. Boundary and mean_max embeddings sometimes provide competitive results, but their higher input dimensionality increases model complexity without offering systematic gains. Conversely, first, last, and max pooling exhibit less stable behavior, particularly in datasets characterized by noisy annotations or strong multi-label distributions. Overall, the results suggest that *average pooling* is a reasonable choice for the default embedding strategy in SCoRE, since it provides a good trade-off between simplicity, robustness, and effectiveness.

Table 8
Micro-F1 e macro-F1 for each dataset and embedding strategy.

Dataset	Embedding	Micro-F1	Macro-F1
DisRex	average	75.23 ± 0.47	62.72 ± 0.73
	boundary	**76.12 ± 0.35**	**63.22 ± 0.37**
	first	73.34 ± 0.27	58.74 ± 1.38
	last	75.34 ± 0.12	62.48 ± 0.76
	max	74.65 ± 0.24	61.72 ± 0.51
	mean_max	75.29 ± 0.33	62.52 ± 0.45
Nyt10d	average	88.66 ± 0.48	**48.85 ± 1.13**
	boundary	**88.91 ± 0.37**	47.10 ± 1.21
	first	87.89 ± 0.36	43.19 ± 0.78
	last	88.20 ± 0.38	46.55 ± 2.12
	max	88.12 ± 0.69	46.79 ± 2.76
	mean_max	88.45 ± 0.30	46.17 ± 1.64
Nyt10m	average	77.12 ± 0.33	39.59 ± 1.33
	boundary	**77.26 ± 0.31**	**39.86 ± 0.25**
	first	77.10 ± 0.53	38.39 ± 0.43
	last	76.87 ± 0.56	38.91 ± 0.96
	max	77.23 ± 0.35	39.55 ± 0.72
	mean_max	77.17 ± 0.33	39.45 ± 1.12
Wiki20d	average	63.20 ± 2.16	23.33 ± 0.42
	boundary	65.09 ± 1.51	23.56 ± 0.48
	first	57.44 ± 2.52	21.67 ± 0.67
	last	63.44 ± 1.87	23.13 ± 0.35
	max	60.55 ± 0.52	22.86 ± 0.59
	mean_max	**65.68 ± 2.55**	**23.78 ± 0.67**
Wiki20m	average	83.56 ± 0.13	80.80 ± 0.18
	boundary	83.60 ± 0.10	80.73 ± 0.16
	first	80.33 ± 0.23	76.98 ± 0.36
	last	82.20 ± 0.17	78.45 ± 0.26
	max	82.92 ± 0.16	79.95 ± 0.05
	mean_max	**83.63 ± 0.15**	**80.89 ± 0.19**

Table 9
Effects of architectural enhancements.

Dataset	Metric	SCoRE + CLS	SE + A	SE + A + CL
Nyt10m	microF1	77.4	65.1	72.5
	macroF1	40.4	23.2	23.7
	kWh	0.64	3.88	7.8
Nyt10d	microF1	89.8	67.4	75.1
	macroF1	48.9	23.3	23.9
	kWh	0.92	14.42	15.12
DisRex	microF1	75.2	36.2	52.7
	macroF1	63.1	23.6	31.3
	kWh	1.44	4.05	18.15
Wiki20m	microF1	83.8	63.7	63.6
	macroF1	81.2	58.1	59.9
	kWh	2.08	13.54	19.5
Wiki20d	microF1	66.8	48.6	50.6
	macroF1	22.9	15.6	16.4
	kWh	5.83	40.1	57.3

7.3. Anti-ablation analysis: the adverse effects of architectural enhancements.

In contrast to ablation studies, which identify critical model components by systematically removing modules and measuring the resulting performance decline, we investigate the impact of increasing architectural complexity. Specifically, the experiments in this section are designed with the following objectives. The overall results are presented in Table 9.

7.3.0.1. ***SCoRE + CLS.*** In the experiment, we assess whether the input structure associated with encoding head-tail pairs during dataset creation (Section 3.1) is sufficient or if it can benefit from additional information, without substantially modifying the SCoRE pipeline. To this aim we explored the impact of incorporating the CLS token into triplet mention embeddings, following the standard approach of concatenating the CLS embedding with those of the head and tail entities appearing in the sentence [23]. A grid-search was performed to find the best hyperparameters, following Section 6.2. The results in Table 9, reveal that, contrary to expectations, the inclusion of the CLS token did not lead to any performance variation. Both micro and macro F1 SCoREs remained largely unchanged, suggesting that the CLS token does not provide substantial contributions to the entity pair embeddings.

Subsequently, considering the robust results of PARE across datasets of various types, we investigate whether replacing the dataset creation phase with a more commonly used approach in the literature can lead to improved outcomes. To this end, we perform two experiments. The

Table 10
Performance on overall datasets (left) and most confident M samples (right) across Bayesian kNN configurations.

Dataset	kNN	Overall performance				@M performance			
		microF1	macroF1	CSD	P@R	m@100	M@100	m@1000	M@1000
Nyt10d	UU	89.1 ± 0.5	48.8 ± 0.6	**0.4 ± 0.0**	**92.0 ± 0.4**	93.3 ± 0.4	68.2 ± 6.1	93.2 ± 0.4	53.1 ± 0.9
	UC	78.7 ± 0.5	28.2 ± 0.7	7.0 ± 0.0	**92.0 ± 0.4**	**94.7 ± 1.1**	**95.2 ± 0.6**	**95.2 ± 0.6**	**59.8 ± 3.9**
	IU	86.9 ± 0.3	40.8 ± 0.6	0.7 ± 0.1	91.3 ± 0.6	92.9 ± 1.7	72.5 ± 2.0	90.8 ± 0.8	52.0 ± 0.8
	IC	**89.2 ± 0.6**	**48.8 ± 1.0**	0.8 ± 0.1	91.3 ± 0.6	92.9 ± 2.8	71.4 ± 15.4	93.1 ± 0.5	52.0 ± 6.0
Nyt10m	UU	**77.5 ± 0.1**	40.0 ± 0.9	**1.1 ± 0.0**	78.4 ± 0.3	86.6 ± 4.0	60.7 ± 13.7	85.9 ± 0.2	43.5 ± 1.6
	UC	69.0 ± 0.4	**42.7 ± 0.6**	3.0 ± 0.1	78.4 ± 0.3	85.9 ± 0.8	46.9 ± 2.0	**86.6 ± 1.0**	42.0 ± 1.4
	IU	76.8 ± 0.7	32.4 ± 1.1	**1.1 ± 0.0**	**80.6 ± 0.3**	**87.8 ± 4.2**	54.0 ± 4.8	85.4 ± 0.3	42.9 ± 1.9
	IC	77.1 ± 0.2	40.5 ± 0.8	1.2 ± 0.0	**80.6 ± 0.3**	86.6 ± 0.7	**63.5 ± 5.8**	85.6 ± 0.2	**43.7 ± 4.0**
DisRex	UU	**75.2 ± 0.2**	**62.9 ± 0.5**	**1.2 ± 0.1**	**76.7 ± 0.1**	**86.6 ± 1.9**	64.2 ± 7.6	**86.7 ± 0.3**	58.2 ± 1.8
	UC	60.8 ± 0.3	51.9 ± 0.5	4.4 ± 0.0	**76.7 ± 0.1**	83.8 ± 1.9	**66.2 ± 4.7**	85.9 ± 0.2	**66.6 ± 4.0**
	IU	65.3 ± 0.1	38.0 ± 1.0	1.7 ± 0.1	76.5 ± 0.2	84.4 ± 2.3	55.8 ± 5.5	84.0 ± 1.0	56.1 ± 3.6
	IC	**75.2 ± 0.2**	**62.9 ± 0.5**	**1.2 ± 0.1**	76.5 ± 0.2	85.3 ± 1.5	62.2 ± 3.8	86.3 ± 0.1	57.5 ± 0.8
WIKI20M	UU	**83.5 ± 0.1**	**80.8 ± 0.1**	**0.2 ± 0.0**	**83.0 ± 0.1**	97.7 ± 1.3	94.5 ± 2.9	**98.3 ± 0.1**	**93.7 ± 1.8**
	UC	62.4 ± 1.0	65.8 ± 0.8	6.4 ± 0.1	**83.0 ± 0.1**	**98.0 ± 1.4**	**95.3 ± 3.4**	97.8 ± 0.2	92.1 ± 2.5
	IU	70.3 ± 1.2	60.3 ± 1.5	0.6 ± 0.0	82.4 ± 0.1	96.7 ± 1.3	90.5 ± 3.8	98.1 ± 0.3	93.2 ± 0.8
	IC	**83.5 ± 0.1**	**80.8 ± 0.1**	**0.2 ± 0.0**	82.4 ± 0.1	98.0 ± 0.0	94.6 ± 0.3	98.1 ± 0.4	92.8 ± 1.6
WIKI20D	UU	65.1 ± 2.5	**23.2 ± 0.8**	2.3 ± 0.1	**67.2 ± 1.3**	**96.8 ± 2.4**	89.1 ± 6.9	**95.7 ± 0.3**	77.3 ± 3.7
	UC	20.8 ± 1.4	5.1 ± 0.2	32.3 ± 0.4	**67.2 ± 1.3**	92.0 ± 2.8	89.0 ± 2.8	93.3 ± 1.2	77.7 ± 2.7
	IU	24.5 ± 3.0	16.3 ± 1.1	**1.0 ± 0.0**	64.7 ± 1.7	95.7 ± 1.3	**90.5 ± 2.4**	95.1 ± 0.2	75.7 ± 4.7
	IC	**67.2 ± 1.5**	21.2 ± 0.8	3.6 ± 0.1	64.7 ± 1.7	93.7 ± 3.3	88.3 ± 6.6	95.0 ± 1.1	**82.3 ± 0.7**

first is not directly related to the SCoRE architecture; it represents a sentence-level adaptation of PARE without fine-tuning and serves solely as a comparison for the final experiment. The second replaces SCoRE's dataset creation phase with PARE's dynamic sentence encoding mechanism and attention mechanism.

7.3.0.2. ***Sentence Embedding + Attention (SE+A).*** In this experiment, we explore whether PARE's approach, which involves dynamic sentence encoding supported by an attention mechanism for RE, can remain effective when the fine-tuning of the PLM is removed and training happens at the sentence level. Freezing PLM weights requires additional modifications to the input processing compared to PARE. In fact, the use of attention mechanisms in RE tasks typically yields suboptimal performance unless entity markers or masks are employed [12,22]. While PARE uses special entity markers, this approach is not feasible in our case, as the model cannot learn new special tokens without fine-tuning. Consequently, we adopt the strategy of using MASK tokens to replace the head and tail mentions and train PARE's architecture using a multi-label cross-entropy loss and the hyperparameter setting described in [13]. We train the model for 60 epochs and use early stopping as described in Section 6.2. The results for the SE + A configuration demonstrate significant underperformance relative to both SCoRE and PARE, underscoring the critical role of PLM fine-tuning in attention-based models that leverage full sentence embeddings.

Sentence Embedding + Attention + CL (SE+A+CL). This final experiment involves utilizing PARE's dynamic sentence encoding mechanism and attention mechanism by replacing SCoRE's dataset creation phase. Therefore, in this version, sentences are processed dynamically and follow PARE's architecture with masked head and tail entity as in SE + A. However, the remainder of the learning and inference process adheres to the steps defined in the SCoRE pipeline: supervised multi-label CL based on the attention head's output and Bayesian kNN during inference. To make this possible, we removed the softmax activation function from the last layer of SE + A and added the minimal component to perform CL sensibly, i.e. a single fully connected layer with a small number of neurons m_h whose output gets normalized to unit vectors based on L_2 norm. This model is then trained using the multilabel CL loss function described in Eq. (4). A grid-search was performed to find the best hyperparameters for the loss, learning rate, batch size, m_h, kNN, and probability threshold, as described in Section 6.2. The results show that, although the SE + A + CL approach underperforms SCoRE, it yields comparable or even superior results relative to the SE + A configuration. This indicates the effectiveness of supervised CL in managing noisy datasets, as it emphasizes learning robust representations that differentiate between similar and dissimilar instances.

Overall, these experiments demonstrate that increasing input expressiveness or modifying the model's architecture can often degrade performance, particularly when PLM fine-tuning is avoided. This underscores that the interplay between complexity and performance is not always a straightforward trade-off.

7.4. Impact of Bayesian kNN configurations on SCoRE performance

While the previous section highlighted the drawbacks of some architectural modifications, here we focus on examining whether different configurations in the model's prediction stage can enhance performance.

Using a flat prior in Bayesian kNN neglects class frequencies, potentially overemphasizing rare classes and biasing predictions, thereby reducing accuracy for majority classes. Therefore, it may be interesting to investigate how different prior choices affect prediction outcomes. We performed a sensitivity analysis to evaluate the impact of using an informative prior, calculated from class frequencies as $P(r_i) = n_i / \sum_j n_j$, where n_i is the number of training instances in class r_i. While this method can address class imbalance and improve calibration, in a kNN framework it may diminish the influence of rare classes in the posterior. Specifically, when a rare class is among the nearest neighbors, the frequency-based prior reduces its effect on the posterior probabilities.

Selecting an appropriate decision thresholding is essential for ensuring result quality. Universal thresholding (Eq. (8)) applies a single threshold across all classes, while class-specific thresholding assigns distinct thresholds to each class. The latter is particularly effective in imbalanced datasets, as it helps prevent the under-prediction of rare classes by aligning sensitivity and specificity with class prevalence. However, determining optimal thresholds poses a challenge. Various methods exist, including Bayesian decision theory, cross-validation-based probability calibration, and utilizing prior class probabilities. To maintain simplicity and due to the absence of a preferred error type, we adopt the use of prior class probabilities thesholding, defined as

$$\hat{y}(\tilde{\mathbf{z}})_i = \begin{cases} 1, & \text{if } P(r_i|\tilde{\mathbf{z}}) > P(r_i), \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

In our experiments, we tested how performance were affected by both prior and decision threshold choices. In particular, we will refer to the configurations as:

- UU: Uninformative prior/Universal threshold (default SCoRE setup).
- UC: Uninformative prior/Class-specific threshold.
- IU: Informative prior/Universal threshold.
- IC: Informative prior/Class-specific threshold.

For each of the previous configurations, we computed micro-/macroF1 SCoREs and CSD values. Each experiment was repeated 10 times to facilitate the comparison among different configurations. For what concerns universal thresholding, we retained the threshold value providing the best performance on the validation set, where available.

The results on the left-hand side of Table 10 demonstrate that the UU (SCoRE) and IC configurations consistently achieve the best micro- and macro-F1 SCoREs compared to other methods. However, UU exhibits greater agreement with the ground truth label correlation matrix, as indicated by the smaller CSD values. In contrast, methods like UC and IU experience significant performance degradation, with IU underperforming due to its sensitivity to class variability from the informative prior, and UC struggling because of its reliance on prior probability thresholds, which may not accurately reflect local label distributions.

7.5. From @M performance to P@R

Let us now focus on the reliability degree of model confidence by looking at the m@M and M@M performance. The results, shown on the right hand side of Table 10, clearly confirm that, regardless of the configuration, all setups drastically improve performance when requested to judge only the most confident samples based on our ranking method in Eq. (11). The configuration UU confirms to be the best with 16 podium positions (6 best, 10 second best), followed by UC (9 best, 1 second best), IC (3 best, 5 second best), and finally IU (2 best, 4 second best). On this task, the UC configuration works surprisingly well, especially when compared to the relatively low performance on the whole dataset, suggesting that class-specific thresholding can provide marginal benefits. However, upon careful consideration, no single setup consistently outperforms the others across all scenarios. This is particularly surprising, especially considering the performance gap observed on the full dataset, raising questions about the actual utility of these metrics.

To more effectively evaluate model performance under real-world conditions, we shift the focus from class-based to instance-based performance by analyzing the behavior of P@R metrics across different Bayesian kNN configurations. The results, presented in Table 10, demonstrate that P@R values are consistent within each prior type and exhibit minimal variation between universal and class-specific thresholding methods. This indicates that fluctuations in micro- and macro-F1 scores are largely driven by threshold selection rather than changes in the predictive model itself. Indeed, although the UU (SCoRE) and IC methods use different priors, a different choice of decision thresholds can significantly realign their predictive performance. Notably, P@R values for SCoRE are comparable to or exceed those of micro- and macro-F1 scores, highlighting its strong performance as a recommender system. Furthermore, uninformative priors generally yield higher P@R values across most datasets, establishing SCoRE (Section 3) as the optimal configuration among those evaluated.

8. Related works

Recent RE research has increasingly focused on deep learning solutions. Among those, non-PLM approaches often leverage alternative architectures and external knowledge resources, such as KgPool [39], which enriches input context with KG facts for single-label extraction, and graph NN-based methods like RECON [40], which align extracted relations with KG entries for improved accuracy. However, such approaches can hardly be adapted to different case studies and KGs.

In PLM-based DS RE, fine-tuning has become a dominant approach, enabling models to specialize their embeddings by adjusting parameters to capture relational patterns [41].

Several studies frame RE as a bag-level classification task via MIL to address noise in DS annotations. Apart from PARE [13], the approaches leveraging attention mechanisms aim to enhance robustness by using sophisticated learning techniques such as employing syntactic trimming [14], adding sequential layers like Bi-LSTM [15], or introducing hierarchical attention to align sentence- and bag-level representations [16]. To further mitigate noise and improve generalization, some MIL works enrich instance representations with external knowledge. For example, Liu et al. [19] integrates fine-grained alignment and inductive signals from KG neighbors to address long-tail relations. Another example [17] combines global context with knowledge-aware embeddings to guide denoising, while the authors in [18] reconstruct latent structural graphs and refine them through iterative optimization, leveraging pretrained KBs to improve sentence-level representation learning. Another popular approach finetunes the PLM using CL to align sentences with similar entity pair mentions or triplets. For example, Wan et al. [23] uses sentence-level predictions and scores to compute prior weights to guide bag-level CL training, whereas [22] employs contrastive pre-training with sentence- and attention-derived bag encodings.

In this paper, we advocate that sentence-level RE is the only approach enabling domain expert guidance and oversight. In principle, models trained at bag-level can be used to infer single sentences, as we did with PARE. However, MIL training often focuses on high-attention sentences, under-utilizing data, and increasing noise sensitivity [20,42]. Additionally, its effectiveness relies on multiple sentences per entity pair, which is challenging for long-tail relations. These limitations become evident in the comparison with PARE provided in Section 7, where we show that it consistently predicts opposite relations (a problem that could only be highlighted thanks to sentence-level predictions).

Some recent efforts continue to rely on PLM fine-tuning but operate at the sentence level. Among these, we directly compared SCoRE with TIW [26], HiCLRE [27], and SSLRE [28] in 7. Another relevant method is [25], which aims to improve implicit and long-tail relation performance by combining a fine-tuned LLM's output with a memory-based kNN classifier leveraging training triplets encoding. Although Eq. (6) is similar to the one used in Wan et al. [25], their kNN formulation does not derive distance weighting from a metric induced in the hidden feature space. Moreover, for inference, they combine a fine-tuned classifier with the kNN prediction by linearly interpolating the two outputs. This approach is heuristic and limited to single-label prediction. In contrast, SCoRE structures the representation space through supervised multi-label CL, and then derives a principled Bayesian posterior per label that accounts for both presence and absence information. This design naturally enables multi-label inference, and provides a probabilistic framework for incorporating priors.

9. Conclusions and future works

In this paper, we introduced SCoRE, a sentence-based multi-label RE tool designed to effectively handle the noise inherent in DS annotations. To the best of our knowledge, SCoRE is the first approach that completely avoids finetuning, using the PLM solely as an informed prior during dataset creation. This approach minimizes computational costs and ensures adaptability to advancements in LLMs, as the PLM is employed only in a single forward pass. Although relying solely on supervised CL may appear simplistic, this method has been demonstrated to enhance robustness against input noise and hyperparameter variability [33], efficiently utilizing the information in positive pairs while retaining the capacity of CL to mine negative samples. Departing from prevailing trends in the literature, SCoRE does not use CL as a pretraining step followed by a classifier layer. Instead, we streamline the training and testing process by employing a local non-parametric method, i.e., Bayesian kNN, for direct inference on test set labels based on the metric

induced by CL in the hidden feature space. Furthermore, we choose prior class probabilities to enhance recall, improving the detection of long-tail relation types and reinforcing SCoRE's capability to handle complex, imbalanced datasets.

We demonstrated that SCoRE's minimal architecture matches or surpasses state-of-the-art models, offering a lightweight, adaptable, and interpretable solution for RE. This effectiveness is confirmed on our realistic dataset, Wiki20d, which simulates real-world conditions requiring reliance on FDS annotations. By introducing the CSD metric, we showed that SCoRE better aligns with relational patterns found in KG relations. Additionally, we highlighted the detrimental impact of incorporating more complex input and sentence-processing techniques, which, while suitable for fine-tuning approaches, proved counterproductive for SCoRE's design. Additionally, we evaluated SCoRE's sensitivity to prior probabilities and thresholding, identifying a flat prior and universal thresholding as the optimal configuration. This analysis further reveals that micro-F1@M and macro-F1@M metrics are poor indicators of real-world performance because they primarily capture a subset of high-confidence samples, limiting their practical utility. For this reason, we recommend focusing on metrics like P@R for instance-based evaluation and a more accurate assessment of RE systems in decision support contexts. SCoRE's strong P@R results highlight its effectiveness as a recommender system, maintaining robust performance across different prior probability settings and demonstrating substantial potential for real-world applications.

Despite these strengths, some limitations remain. First, while the use of flat priors enhances recall, it may underperform under extreme class imbalance, suggesting that adaptive priors could further improve rare relation detection. Second, although the framework is compatible with multilingual PLMs, our evaluation focused exclusively on English, and extending SCoRE to multilingual or low-resource languages would require additional investigation. Finally, as with most lightweight methods, performance under strong domain shifts may depend on careful preprocessing of input corpora.

However, given the promising results achieved by the SCoRE pipeline across different benchmarks, we plan to extend our approach in future work to encompass entity linking and relational triple extraction [10,11,43–45]. Additionally, we aim to enhance the applicability of SCoRE as a recommender system by integrating it into a human-in-the-loop framework [46,47]. This integration would facilitate expert interaction with the data-driven learning pipeline, improving model adaptability and maintenance while streamlining continuous updates.

CRediT authorship contribution statement

Luca Mariotti: Writing – original draft, Software, Methodology, Data curation, Conceptualization; **Veronica Guidetti:** Writing – original draft, Software, Methodology, Formal analysis, Conceptualization; **Federica Mandreoli:** Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization.

Data availability

I have included my GitHub link of code and data in my manuscript. https://github.com/rioma96/SCoRE.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, A.-C.N. Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (4) (2021) 1–37.
[2] C. Peng, F. Xia, M. Naseriparsa, F. Osborne, Knowledge graphs: opportunities and challenges, Artif. Intell. Rev. 56 (11) (2023) 13071–13102.
[3] W. Hogan, An overview of distant supervision for relation extraction with a focus on denoising and pre-training methods, arXiv preprint arXiv:2207.08286 (2022).
[4] K. Detroja, C.K. Bhensdadia, B.S. Bhatt, A survey on relation extraction, Intell. Syst. Appl. 19 (2023) 200244.
[5] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009, pp. 1003–1011.
[6] R. Bunescu, R. Mooney, Learning to extract relations from the web using minimal supervision, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 576–583.
[7] M. Craven, J. Kumlien, et al., Constructing biological knowledge bases by extracting information from text sources, in: ISMB, 1999, 1999, pp. 77–86.
[8] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III, ECML PKDD'10, Springer-Verlag, Berlin, Heidelberg, 2010, p. 148-163.
[9] X. Jiang, Q. Wang, P. Li, B. Wang, Relation extraction with multi-instance multi-label convolutional neural networks, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1471–1480.
[10] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: a roadmap, IEEE Trans. Knowl. Data Eng. 36 (07) (2024) 3580–3599.
[11] X. Zhao, Y. Deng, M. Yang, L. Wang, R. Zhang, H. Cheng, W. Lam, Y. Shen, R. Xu, A comprehensive survey on relation extraction: recent advances and new frontiers, ACM Comput. Surv. 56 (11) (2024) 1–39.
[12] H. Peng, T. Gao, X. Han, Y. Lin, P. Li, Z. Liu, M. Sun, J. Zhou, Learning from context or names? an empirical study on neural relation extraction, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 3661–3672.
[13] V. Rathore, K. Badola, P. Singla, Mausam, PARE: a simple and strong baseline for monolingual and multilingual distantly supervised relation extraction, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 340–354.
[14] D. Christou, G. Tsoumakas, Improving distantly-supervised relation extraction through BERT-based label and instance embeddings, IEEE Access 9 (2021) 62574–62582.
[15] H. Yin, S. Liu, Z. Jian, Distantly supervised relation extraction via contextual information interaction and relation embeddings, Symmetry 15 (2023) 1788.
[16] J. Zhang, M. Cao, Distant supervision for relation extraction with hierarchical attention-based networks, Expert Syst. Appl. 220 (C) (2023) 119727.
[17] J. Gao, H. Wan, Y. Lin, Exploiting global context and external knowledge for distantly supervised relation extraction, Knowl. Based Syst. 261 (2023) 110195. https://www.sciencedirect.com/science/article/pii/S0950705122012916. https://doi.org/https://doi.org/10.1016/j.knosys.2022.110195
[18] Q. Zhou, Y. Zhang, D. Ji, Distantly supervised relation extraction with KB-enhanced reconstructed latent iterative graph networks, Knowl. Based Syst. 260 (2023) 110108. https://www.sciencedirect.com/science/article/pii/S0950705122012047. https://doi.org/https://doi.org/10.1016/j.knosys.2022.110108
[19] M. Liu, F. Zhou, J. He, X. Yan, Knowledge graph attention mechanism for distant supervision neural relation extraction, Knowl. Based Syst. 256 (2022) 109800. https://www.sciencedirect.com/science/article/pii/S0950705122009145. https://doi.org/https://doi.org/10.1016/j.knosys.2022.109800
[20] Z. Hu, Y. Cao, L. Huang, T.-S. Chua, How knowledge graph and attention help? a qualitative analysis into bag-level relation extraction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4662–4671.
[21] L.B. Soares, N. Fitzgerald, J. Ling, T. Kwiatkowski, Matching the blanks: distributional similarity for relation learning, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2895–2905.
[22] T. Chen, H. Shi, S. Tang, Z. Chen, F. Wu, Y. Zhuang, CIL: Contrastive instance learning framework for distantly supervised relation extraction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 6191–6200.
[23] Z. Wan, F. Cheng, Q. Liu, Z. Mao, H. Song, S. Kurohashi, Relation extraction with weighted contrastive pre-training on distant supervision, in: Findings of the Association for Computational Linguistics: EACL 2023, 2023, pp. 2580–2585.
[24] Y. Xue, K. Whitecross, B. Mirzasoleiman, Investigating why contrastive learning benefits robustness against label noise, in: International Conference on Machine Learning, PMLR, 2022, pp. 24851–24871.
[25] Z. Wan, Q. Liu, Z. Mao, F. Cheng, S. Kurohashi, J. Li, Rescue implicit and long-tail cases: nearest neighbor relation extraction, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 1731–1738.

[26] X. Lin, W. Jia, Z. Gong, Self-distilled transitive instance weighting for denoised distantly supervised relation extraction, in: Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 168–180.
[27] D. Li, T. Zhang, N. Hu, C. Wang, X. He, HiCLRE: a hierarchical contrastive learning framework for distantly supervised relation extraction, in: Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 2567–2578.
[28] X. Sun, Q. Liu, S. Wu, Z. Wang, L. Wang, Noise-robust semi-supervised learning for distantly supervised relation extraction, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 13145–13157.
[29] W. Jia, D. Dai, X. Xiao, H. Wu, ARNOR: attention regularization based noise reduction for distant supervision relation classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1399–1408.
[30] V.B. Parthasarathy, A. Zafar, A.I. khan, A. Shahid, The ultimate guide to fine-tuning LLMs from basics to breakthroughs: an exhaustive review of technologies, research, best practices, applied research challenges and opportunities, arXiv:2408.13296 abs/2408.13296 (2024).
[31] T. Gao, X. Han, Y. Bai, K. Qiu, Z. Xie, Y. Lin, Z. Liu, P. Li, M. Sun, J. Zhou, Manual evaluation matters: reviewing test protocols of distantly supervised relation extraction, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1306–1318.
[32] G. Nolano, M. Blum, B. Ell, P. Cimiano, Pointing out the shortcomings of relation extraction models with semantically motivated adversarials, arXiv preprint arXiv:2402.19076 (2024).
[33] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, Adv. Neural Inf. Process. Syst. 33 (2020) 18661–18673.
[34] D.A. Roberts, S. Yaida, B. Hanin, The Principles of Deep Learning Theory, 46, Cambridge University Press Cambridge, MA, USA, 2022.
[35] R. Wang, X. Dai, et al., Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2022, pp. 672–679.
[36] J.M.N. Gøttcke, A. Zimek, Handling class imbalance in K-nearest neighbor classification by balancing prior probabilities, in: Similarity Search and Applications: 14th International Conference, SISAP 2021, Dortmund, Germany, September 29–October 1, 2021, Proceedings 14, Springer, 2021, pp. 247–261.
[37] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, LiamConnell, A. Saboni, Inimaz, supatomic, M. Léval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, et al., mlco2/codecarbon: v2.4.1, 2024. Zenodo https://doi.org/10.5281/zenodo.11171501
[38] A. Bhartiya, K. Badola, Mausam, DiS-ReX: a multilingual dataset for distantly supervised relation extraction, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 849–863.
[39] A. Nadgeri, A. Bastos, K. Singh, I.O. Mulang', J. Hoffart, S. Shekarpour, V. Saraswat, KGPool: dynamic knowledge graph context selection for relation extraction, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 535–548.
[40] A. Bastos, A. Nadgeri, K. Singh, I.O. Mulang', S. Shekarpour, J. Hoffart, M. Kaul, RECON: relation extraction using knowledge graph context in a graph neural network, Proc. Web Conf. 2021 Association for Computing Machinery, New York, NY, USA, (2021), pp. 1673–1685. https://doi.org/10.1145/3442381.3449917
[41] M. Szép, D. Rueckert, R. von Eisenhart-Rothe, F. Hinterwimmer, A practical guide to fine-tuning language models with limited data, arXiv preprint arXiv:2411.09539 (2024).
[42] T. Chen, H. Shi, L. Liu, S. Tang, J. Shao, Z. Chen, Y. Zhuang, Empower distantly supervised relation extraction with collaborative adversarial training, in: Proceedings of the AAAI Conference on Artificial Intelligence, 35, 2021, pp. 12675–12682.
[43] W. Zhang, J. Wang, C. Chen, W. Lu, W. Du, H. Wang, J. Liu, T. Ruan, A bidirectional extraction-then-evaluation framework for complex relation extraction, IEEE Trans. Knowl. Data Eng. 36 (12) (2024) 7442–7454.
[44] F. Ren, L. Zhang, X. Zhao, S. Yin, S. Liu, B. Li, A simple but effective bidirectional framework for relational triple extraction, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 824-832.
[45] X. Zhao, M. Yang, Q. Qu, R. Xu, J. Li, Exploring privileged features for relation extraction with contrastive student-teacher learning, IEEE Trans. Knowl. Data Eng. 35 (08) (2023) 7953–7965.
[46] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, L. He, A survey of human-in-the-loop for machine learning, Future Gener. Comput. Syst. 135 (C) (2022) 364-381.
[47] T. Bikaun, M. Stewart, W. Liu, CleanGraph: human-in-the-loop knowledge graph refinement and completion, arXiv preprint arXiv:2405.03932 (2024).