Let's Stop Incorrect Comparisons in End-to-end Relation Extraction!

Bruno Taillé[1,2], Vincent Guigue[1], Geoffrey Scoutheeten[2] and Patrick Gallinari[1,3]

[1]Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6

[2]BNP Paribas

[3]Criteo AI Lab

`{bruno.taille, vincent.guigue, patrick.gallinari}@lip6.fr`
`geoffrey.scoutheeten@bnpparibas.com`

Abstract

Despite efforts to distinguish three different evaluation setups (Bekoulis et al., 2018a,b), numerous end-to-end Relation Extraction (RE) articles present unreliable performance comparison to previous work. In this paper, we first identify several patterns of invalid comparisons in published papers and describe them to avoid their propagation. We then propose a small empirical study to quantify the most common mistake's impact and evaluate it leads to overestimating the final RE performance by around 5% on ACE05. We also seize this opportunity to study the unexplored ablations of two recent developments: the use of language model pretraining (specifically BERT) and span-level NER. This meta-analysis emphasizes the need for rigor in the report of both the evaluation setting and the dataset statistics. We finally call for unifying the evaluation setting in end-to-end RE [1].

1 Introduction

Named Entity Recognition (NER)[2] and Relation Extraction (RE) are key Information Extraction tasks, for example at the heart of Knowledge Graph Construction along with Coreference Resolution and Entity Linking. In the traditional pipeline approach, these tasks are treated with two models trained separately and applied sequentially (Bach and Badaskar, 2007). Nevertheless, combining information from both submodules is beneficial (Roth and Yih, 2002) and end-to-end RE models tackling both tasks jointly have been proposed to better model their interdependency and overcome cascading errors (Li and Ji, 2014).

This end-to-end setting has recently received more attention in the wake of improved language models (LM). However, in this prolific and competitive domain, authors have used several evaluation settings to compare their performance. And despite the attempt to clearly identify three main setups (Bekoulis et al., 2018a,b), this multiplication of settings makes the apprehension of the literature difficult and confusing, but more importantly, it has led to erroneous comparisons and conclusions.

In this paper, we first present a quick literature review of the recent advances in end-to-end RE. Our main contribution is then the identification of invalid comparison patterns in recent publications. We list them with the hope of stopping the propagation of erroneous results and presenting a curated list of published results. To further this contribution, we propose a small empirical study to quantify the impact of switching the two main metrics and estimate it can lead to a relative overestimation of around 5% in the end-to-end RE results on ACE05.

As a second contribution, we take advantage of this quantitative study to perform the omitted ablations of two recent developments in the literature: LM pretraining and Span-level NER. It confirms that recent empirical gains are mainly due to LM pretraining, while there is no evidence for quantitative gains from Span-level NER.

Finally, we argue that the main cause for previously identified mistakes is the lack of reproducibility and, consequently, of previous work reproductions. We call for a more rigorous report of both evaluation settings and dataset statistics in general and particularly in end-to-end RE. And we also suggest unifying our evaluation setting to reduce the chance of future mistakes and enable more meaningful cross-dataset analyses.

2 A Quick Literature Review

In order to have a global view of recent evolutions, we present a quick literature review of end-to-end RE models. We focus on supervised extraction of

---

[1]Code available at github.com/btaille/sincere

[2]We will also use NER to refer to Entity Mention Detection (EMD) when entities of interest are not Named Entities.

Reference	Criterion	Code	Representations						Enc.	NER		RE
			LM	Word	Char	Hand	POS	DEP	Tag	Tag	Dec.	Dec.
(Giorgi et al., 2019)	S	*	B						-	B	MLP	Biaff.
(Eberts and Ulges, 2020)	SB ✓	✓	B						-	S	MLP	PMaxPool
(Wadden et al., 2019)	B ✓	✓	B						-	S	MLP	Biaff.
(Li et al., 2019)	S		B						-	-	MT QA	MT QA
(Dixit and Al-Onaizan, 2019)	S		E	S	C				L	S	MLP	Biaff.
(Luan et al., 2019)	B ✓	✓	E	G	ns				L	S	MLP	Biaff.
(Nguyen and Verspoor, 2019)	SR ✓	✓		G	L				L	B	MLP	MHS-Biaff.
(Sanh et al., 2019)	- ✗	✓	E	G	C				L	B	CRF	MHS-Lin.
(Luan et al., 2018)	B	✓	E	G	ns					S	MLP	Biaff.
(Sun et al., 2018)	S ✓	≈		ns	C				L	B	MLP	PCNN
(Bekoulis et al., 2018a,b)	SBR ✓	✓		S/W	L				L	B	CRF	MHS-Lin.
(Zhang et al., 2017)	S ✓	≈		G	C		✓	✓	L	B	I-LSTM	I-LSTM
(Li et al., 2017)	S ✓	≈		ns	C		✓	✓	L	B	MLP	SP LTSM
(Katiyar and Cardie, 2017)	S ✓			W					L	B	I-MLP	I-Pointer
(Zheng et al., 2017)	S ✓			ns					L	B	MLP	PCNN
(Adel and Schütze, 2017)	R ✓	✓		W					-	B	CNN	PCNN+CRF
(Gupta et al., 2016)	R ✓			T		✓	✓		-	B	I-RNN	I-RNN
(Miwa and Bansal, 2016)	S ✓	✓		ns			✓	✓	L	B	MLP	SP LSTM
(Miwa and Sasaki, 2014)	S ✓	✓				✓			-	B	I-SVM	I-SVM
(Li and Ji, 2014)	SB ✓					✓			-	B	I-Perc.	I-Perc.

Table 1: Proposed classification of end-to-end RE models in antichronological order. Criterion: **S**trict / **B**oundaries / **R**elaxed and presence of statement (✗: incorrectly stated). Code: source code availability (≈ : no documentation / *:WIP). Language Model pretraining: **E**LMo (Peters et al., 2018) / **B**ERT (Devlin et al., 2019). Word embeddings: **S**ENNA (Collobert and Weston, 2011) / **W**ord2Vec (Mikolov et al., 2013) / **G**loVe (Pennington et al., 2014) / **T**urian (Turian et al., 2010). Character embeddings pooling: **C**NN / (Bi)**L**STM. Hand: handcrafted features. POS/DEP: use of Ground Truth or external Part-of-Speech tagger or Dependency Parser. Encoder: (Bi)**L**STM. NER Tag: **B**ILOU / **S**pan. Decoders: I- = Incremental, MHS=Multi-Head Selection, SP=Shortest Dependency Path. ns=Not Specified, for words it might be randomly initialized embeddings.

intra-sentence binary relations in English corpora. A summary is proposed in Table 1.

Local classifiers The first attempts to model the interdependency between NER and RE combined the predictions of independent local classifiers according to global constraints (e.g. the arguments of the "Live In" relation must be a Person and a Location); either with Probabilistic Graphical Models (Roth and Yih, 2002), Integer Linear Programming (Roth and Yih, 2004) or Card Pyramid Parsing (Kate and Mooney, 2010).

Incremental Joint Training Li and Ji (2014) propose the first joint model using a structured perceptron to parse a sentence with a set of two actions: append a mention to detected entities and possibly link it with a relation to a previous mention. Katiyar and Cardie (2017) adopt the same framing but replace handcrafted features with word embeddings and use a BiLSTM for NER and a Pointer Network for RE. Miwa and Sasaki (2014) simplify this setting by sequentially filling a table containing all entity and relation information. Gupta et al. (2016)

take up this Table Filling (TF) approach but use an RNN with a multitask approach. Similarly, (Zhang et al., 2017) use LSTMs but add syntactic features from (Dozat and Manning, 2017)'s Dependency Parser.

Entity Filtering Other models use entity filtering as in the pipeline setting where RE is viewed as classification given a sentence and a pair of arguments. This requires passing each pair of candidate entities through the RE classifier. The only difference is that the NER and RE models share some parameters in end-to-end RE, often in a BiLSTM encoder. Indeed, as in the previous incremental setting, NER is modeled as sequence labeling using BILOU tags (Ratinov and Roth, 2009) and the NER module is often a BiLSTM as in (Huang et al., 2015). The two modules are jointly trained by optimizing for the (weighted) sum of their losses.

Miwa and Bansal (2016) use a sequential BiLSTM for NER and a Tree-LSTM over the shortest dependency path between candidate arguments given by an external parser and Li et al. (2017) apply this model to biomedical data.

Adel and Schütze (2017), Zheng et al. (2017) and Sun et al. (2018) all rely on the Piecewise CNN (PCNN) architecture for RE (Zeng et al., 2015). The sentence is split into three pieces: before the first argument, between the arguments, and after the last argument. The RE classifier is fed with CNN pooled representations of these three pieces and of both arguments. Adel and Schütze (2017) add a CRF to model the argument type / relation type dependencies while Sun et al. (2018) use minimum risk training to incorporate global F1 scores in the loss and make loss functions more interdependent.

Multi-Head Selection   To avoid relying explicitly on NER prediction, Bekoulis et al. (2018b,a), propose Multi-Head Selection where RE classification is made for every pair of words. As in Table Filling, relations should only be predicted between the last words of entity mentions to avoid redundancy and inconsistencies. This enables end-to-end RE in a single pass, but contextual information must be implicitly encoded in all word representations since the Linear RE classifier is only fed with representations of both arguments and a label embedding of BILOU NER predictions. Nguyen and Verspoor (2019) replace this linear RE classifier by the bilinear scorer from Dozat and Manning (2017)'s Dependency Parser. A similar architecture is extended with BERT representations in (Giorgi et al., 2019). Finally, Sanh et al. (2019) build on (Bekoulis et al., 2018b) to explore a broader multi-task setting incorporating Coreference Resolution (CR) and another corpus for NER. They use ELMo contextualized embeddings (Peters et al., 2018).

Span-level NER   With the same idea of jointly training CR along with joint NER and RE, Luan et al. (2018) replace the traditional sequence labeling framing of NER by span-level classification inspired by end-to-end CR (Lee et al., 2017) and Semantic Role Labeling (SRL) (He et al., 2018). In this setting, all spans (up to a fixed length) are independently classified as entities, which enables detecting overlapping entities, and they use an element-wise biaffine RE classifier to classify all pairs of detected spans. In (Luan et al., 2019), they then propose to iteratively refine predictions with dynamic graph propagation of RE and CR confidence scores. This work is adapted with BERT as an encoder in (Wadden et al., 2019).

Dixit and Al-Onaizan (2019) use a model very similar to Luan et al. (2018)'s but restrict to end-to-end RE. Eberts and Ulges (2020) recently use span-level NER with BERT as an encoder. They add a pooled representation of the middle context for RE, similarly to piecewise models.

Question Answering   RE can also be framed as Question Answering (QA) in the zero-shot (Levy et al., 2017) or end-to-end (Li et al., 2019) settings. The latter Multi-Turn QA uses templates of questions to identify entity mentions and their relations.

3   Datasets and Metrics

Datasets   Although a variety of datasets have been used, we limit our report to the five we identified as the most frequently studied for brevity.

Following (Roth and Yih, 2002), end-to-end RE has traditionally been explored on English news articles, which is reflected in the domain of its historical benchmarks, CoNLL04 and the ACE datasets. **CoNLL04** (Roth and Yih, 2004) is annotated for four entity types and five relation types and specifically only contains sentences with at least one relation. The **ACE04** dataset (Doddington et al., 2004) defines seven coarse entity types and seven relation types. **ACE05** resumes this setting but merges two relation types leading to six of them.

More recently, Gurulingappa et al. (2012) propose the **ADE** dataset in the biomedical domain, which focuses on one relation, the Adverse Drug Event between a Drug and one of its Adverse Effects. In the scientific domain, Luan et al. (2018) introduce **SciERC** composed of 500 scientific article abstracts annotated with six types of scientific entities, coreference clusters, and seven relations between them.

Metrics   The traditional metrics for assessing both NER and RE performance are Precision, Recall and F1 scores. However, there are two points of attention: the use of micro or Macro averaged metrics across types and the criterion used to consider a prediction as true positive.

On this second point, there is no difficulty for NER where the consensus is to both consider detection and typing. However, compared to the pipeline Relation Classification, this end-to-end RE setting adds a source of mistake in the identification of arguments. And while there is an agreement that the relation type must be correctly detected, several evaluation settings have been introduced with different argument detection requirements.

Hence, Bekoulis et al. (2018a) distinguishes three evaluation settings:

Strict: both the boundaries and the entity type of each argument must be correct.

Boundaries: argument type is not considered and boundaries must be correct.

Relaxed: NER is reduced to Entity Classification i.e. predicting a type for each token. A multi-token entity is considered correct if at least one token is correctly typed.

4 Identified Issues in Published Results

This variety of evaluation settings, visible in Table 1, leads to confusion which in turn favors recurring mistakes. By a careful examination of previous work and often only thanks to released source codes and/or sufficiently detailed descriptions, we identified several of them. Because these precious sources of information are sometimes missing, we cannot assert we are exhaustive. However, we will now list them to avoid their propagation and present a curated summary of supposedly comparable results in Table 2.

4.1 Comparing Boundaries to Strict results on ACE datasets

The most common mistake is the comparison of Strict and Boundaries results. Indeed, several works (Zheng et al., 2017; Luan et al., 2019; Wadden et al., 2019) use the Boundaries setting to compare to previous Strict results. However, because the Strict setting is more restrictive, this leads to overestimating the benefit of the proposed model over previous SOTA. We propose a quantification of the resulting improper gain in section 5.4.

4.2 Confusing Settings on CoNLL04

On the CoNLL04 dataset, the two settings that have been used are even more different. Indeed, while Miwa and Sasaki (2014) use the Strict evaluation, Gupta et al. (2016), who build upon the same Table Filling idea, introduce a different setting. They 1) use the Relaxed criterion; 2) discard the "Other" entity type; 3) release another train / test split; 4) use Macro-F1 scores.

This inevitably leads to confusions, first on the train / test splits, e.g. Giorgi et al. (2019) claim to use the splits from (Miwa and Sasaki, 2014) while they link to (Gupta et al., 2016)'s. Second, Nguyen and Verspoor (2019) unconsciously introduce a different *Strict setup* because it ignores the "Other" entity type and considers Macro-F1 instead of micro-F1 scores. This leads to unfair comparisons.

4.3 Altering both Metrics and Data

Sanh et al. (2019) propose a multitask Framework for NER, RE and CR and use ACE05 to evaluate end-to-end RE. However, they combine two mistakes: incorrect metric comparison and dataset alteration. First, they use the typical formulation to describe a Strict setting but, in fact, use a setting looser than Boundaries. Indeed, they do not consider the type of arguments and only their last word must be correctly detected. Second, they truncate the ACE05 dataset to sentences containing at least one relation both in train and test sets, which leads to an even more favorable setting.

What is worrisome is that both these mistakes are almost invisible in their paper and can only be detected in their code. The only hint for incorrect evaluation is that they report a score for a setting where they only supervise RE, which is impossible in any standard setting. For the dataset, the fact that they do not use the standard preprocessing from (Miwa and Bansal, 2016)[1] might be a first clue.

4.4 Are We Even Using the Same Data?

Without going this far into data alteration, a first source of ambiguity resides in the use or not of the validation set as additional training data. While on CoNLL04, because there is no agreement on a dev set, the final model is trained on train+dev by default; the situation is less clear on ACE. And our following experiments show that this point is already critical w.r.t SOTA claims.

Considering data integrity and keeping the ACE datasets example, even when the majority of works refer to the same preprocessing scripts[1] there is no way to check the integrity of the data without a report of complete dataset statistics. This is especially true for these datasets whose license prevents sharing of preprocessed versions.

Yet, we have to go back to (Roth and Yih, 2004) to find the original CoNLL04 statistics and (Li and Ji, 2014) for ACE datasets. To our knowledge, only a few recent works report in-depth datasets statistics (Adel and Schütze, 2017; Sanh et al., 2019; Giorgi et al., 2019). We report them for CoNLL04 and ACE05 in Table 3 along with our own.

---

[1] github.com/tticoin/LSTM-ER

Reference	ACE 05		ACE 04		CoNLL04		ADE		SciERC		
	Ent	Rel	Ent	Rel	Ent	Rel	Ent	Rel	Ent	Rel	
Strict Evaluation	μF1		μF1		μF1		MF1		μF1		
(Giorgi et al., 2019)	**87.2**†	58.6†	**87.6**†	**54.0**†	89.5†	66.8†	**89.6**	**85.8**			+
(Eberts and Ulges, 2020)	-	-	-	-	**88.9**†	**71.5**†	89.3†	79.2†			-
(Dixit and Al-Onaizan, 2019)	86.0	**62.8**	-	-	-	-	-	-			-
(Li et al., 2019)	84.8	60.2	83.6	49.4	87.8*	68.9*	-	-			+
(Sun et al., 2018)	83.6	59.6	-	-	-	-	-	-			+
(Bekoulis et al., 2018a)	-	-	81.6†	47.5†	83.6†	62.0†	86.7	75.5			+
(Bekoulis et al., 2018b)	-	-	81.2†	47.1†	83.9†	62.0†	86.4	74.6			+
(Zhang et al., 2017)	83.6	57.5	-	-	85.6*	67.8*	-	-			-
(Li et al., 2017)	-	-	-	-	-	-	84.6	71.4			+
(Katiyar and Cardie, 2017)	82.6	53.6	79.6	45.7	-	-	-	-			-
(Li et al., 2016)	-	-	-	-	-	-	79.5	63.4			-
(Miwa and Bansal, 2016)	83.4	55.6	81.8	48.4	-	-	-	-			-
(Miwa and Sasaki, 2014)	-	-	-	-	80.7*	61.0*	-	-			-
(Li and Ji, 2014)	80.8	49.5	79.7	45.3					-	-	-
Boundaries Evaluation											
(Eberts and Ulges, 2020)	-	-	-	-					**70.3**†	**50.8**†	-
(Wadden et al., 2019) ✗	**88.6**	**63.4**	-	-					67.5	48.4	+
(Luan et al., 2019) ✗	88.4	63.2	**87.4**	**59.7**					65.2	41.6	+
(Luan et al., 2018)	-	-	-	-					64.2	39.3	+
(Zheng et al., 2017) ✗	-	52.1	-	-					-	-	-
(Li and Ji, 2014)	80.8	52.1	79.7	48.3					-	-	-
Relaxed Evaluation					MF1						
(Nguyen and Verspoor, 2019) ✗					**93.8**	**69.6**					-
(Bekoulis et al., 2018a)					93.0†	68.0†					+
(Bekoulis et al., 2018b)					93.3†	67.0†					+
(Adel and Schütze, 2017)					82.1	62.5					-
(Gupta et al., 2016)					92.4†	**69.9**†					-
Not Comparable											
(Sanh et al., 2019) ✗	85.5	60.5									-

Table 2: Summary of recently published results in end-to-end RE on five datasets.
* = partition from (Miwa and Sasaki, 2014). † = explicit use of train+dev. **+** = experiments on additional datasets. ✗ = some results were incorrectly reported as Strict. Models over the dashed lines use LM pretraining.

We observe differences in the number of sentences, entity mentions and relations. Minor differences in the number of annotated mentions likely come from evolutions in datasets versions. Their impact on performance comparison should be limited, although problematic. But we also observe more impactful differences, e.g. with (Giorgi et al., 2019) for both datasets and despite using the same setup and preprocessing.

Such a difference in statistics reminds us that the dataset is an integral part of the evaluation setting. And in the absence of sufficiently detailed reports, we cannot track when and where they have been changed since their original introduction.

CoNLL04	(R&Y, 04)	(A&S, 17)	(G, 19)	Ours
# sents	1,437	-	-	1,441
# ents	5,336	5,302	14,193	5,349
# rels	2,040	2,043	2,048	2,048

ACE05	(L&J, 14)	(S, 19)	(G, 19)	Ours
# sents	10,573	10,573	-	14,521
# ents	38,367	34,426	38,383	38,370
# rels	7,105	7,105	6,642	7,117

Table 3: Global datasets statistics in CoNLL04 and ACE05 as reported by different sources. More detailed statistics are available in Appendix.

5 A Small Empirical Study

Given these previous inconsistencies, we can legitimately wonder what is the impact of different evaluation settings on quantitative performance. However, it is also unrealistic to reimplement and test each and every paper in a same setting to establish a benchmark. Instead, we propose a small empirical study to quantify the impact of using the Boundaries setting instead of the Strict setting on the two main benchmarks: CoNLL04 and ACE05. We discard the Relaxed setting because it cannot evaluate true end-to-end RE without striclty taking argument detection into account. It is also limited to CoNLL04 and we have no example of misuse.

We will consider a limited set of models representative of the main Entity Filtering approach. And we seize this opportunity to perform two ablations that correspond to meaningful recent proposals and are missing in related work.

First, when looking at Table 2, it is difficult to draw general conclusions beyond the now established improvements due to LM pretraining. And in the absence of ablation studies on the matter[1], it is impossible to compare models using LM pretraining and anterior works. For example, in the novel work of Li et al. (2019), we cannot disentangle the quantitative effects of LM pretraining and the proposed MultiTurn QA.

Second, to our knowledge, no article compares the recent use of span-level NER instead of classical sequence tagging in end-to-end RE.

5.1 Dataset preprocessing and statistics

We use the standard preprocessing from (Miwa and Bansal, 2016) to preprocess ACE05[2].

For CoNLL04, we take the preprocessed dataset and train / dev / test split from (Eberts and Ulges, 2020)[3] and check it corresponds to the standard train / test split from (Gupta et al., 2016)[4]. We report global dataset statistics in Table 3.

5.2 Models

We propose to use the model from (Eberts and Ulges, 2020) as a baseline for our ablation study since it combines BERT finetuning and Span-level NER. We then perform two ablations: replacing BERT by a BiLSTM encoder with non-contextual

representations and substituting Span-level NER with BILOU sequence tagging.

Encoder : BiLSTM vs BERT We use BERT (Devlin et al., 2019) as LM pretraining baseline, expecting that the effects of ELMo (Peters et al., 2018) would be similar. As in related work, we use cased $BERT_{BASE}$ and finetune its weights. A word is represented by max-pooling of the last hidden layer representations of all its subwords.

For our non-contextual baseline, we take the previously ubiquitous BiLSTM encoder and choose a 384 hidden size in each direction so that the encoded representation matches BERT's. We feed this encoder with the concatenation of 300d GloVe 840B word embeddings (Pennington et al., 2014) and a reproduction of the charBiLSTM from (Lample et al., 2016) (100d char embeddings and hidden size 25 in each direction).

NER Decoder : BILOU vs Span In the sequence tagging version, we simply feed the previously encoded word representation $\mathbf{h}_i$ into a linear layer with a softmax to predict BILOU tags.

$$\hat{\mathbf{y}}_i^{seq} = \text{softmax}(W^{seq}.\mathbf{h}_i + \mathbf{b}^{seq}) \qquad (1)$$

For span-level NER, we follow (Eberts and Ulges, 2020). We only consider spans up to maximal length 10, which are represented by the max pooling of the representations of their tokens. An additional span width embedding $\mathbf{w}$ of dimension 25 is concatenated to this representation as in (Lee et al., 2017). The only difference with (Eberts and Ulges, 2020) is that they also concatenate the representation of the [CLS] token in all span representations to incorporate sentence-level information. We discard this specificity of BERT-like models. All these span-level representations are classified using a linear layer followed by a softmax to predict entity types (including None). We also use negative sampling by randomly selecting 100 negative spans during training.

$$\mathbf{h}(s) = \text{MaxPool}(\mathbf{h}_i, ...\mathbf{h}_{i+l-1}) \qquad (2)$$
$$\mathbf{e}(s) = [\mathbf{h}(s); \mathbf{w}(l)] \qquad (3)$$
$$\hat{\mathbf{y}}^{span}(s) = \text{softmax}(W^{span}.\mathbf{e}(s) + \mathbf{b}^{span}) \quad (4)$$

The NER loss $\mathcal{L}_{NER}$ is the cross-entropy over either BILOU tags or entity classes.

RE Decoder For the RE Decoder, we strictly follow (Eberts and Ulges, 2020). We first filter candidate entity pairs i.e. all the ordered pairs of entity

---

[1]Excepting in (Sanh et al., 2019) which ablates ELMo
[2]github.com/tticoin/LSTM-ER
[3]github.com/markus-eberts/spert
[4]github.com/pgcool/TF-MTRNN

μF1		CoNLL04						ACE05					
		NER		RE (S)		RE (B)		NER		RE (S)		RE (B)	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
BERT Span	train	$85.2_{1.9}$	$86.5_{1.4}$	$69.5_{1.9}$	$\mathbf{67.8}_{.6}$	$69.6_{2.0}$	$\mathbf{68.0}_{.5}$	$84.6_{.6}$	$86.2_{.4}$	$\mathbf{60.1}_{1.0}$	$\mathbf{59.6}_{1.0}$	$\mathbf{63.2}_{.9}$	$\mathbf{62.9}_{1.2}$
	+dev	-	$87.5_{.8}$	-	$\mathbf{70.1}_{1.2}$	-	$\mathbf{70.4}_{1.2}$	-	$86.5_{.4}$	-	$\mathbf{61.2}_{1.3}$	-	$\mathbf{64.2}_{1.3}$
BERT Seq	train	$\mathbf{86.4}_{1.0}$	$\mathbf{87.4}_{.8}$	$\mathbf{71.0}_{1.8}$	$\mathbf{68.3}_{1.9}$	$\mathbf{71.1}_{1.7}$	$\mathbf{68.5}_{1.8}$	$85.7_{.2}$	$87.0_{.3}$	$60.1_{.8}$	$59.7_{1.1}$	$62.6_{1.1}$	$62.9_{1.2}$
	+dev	-	$\mathbf{88.9}_{0.6}$	-	$70.0_{1.2}$	-	$70.2_{1.2}$	-	$87.4_{.3}$	-	$61.2_{1.1}$	-	$\mathbf{64.4}_{1.6}$
BiLSTM Span	train	$79.8_{1.6}$	$80.3_{1.2}$	$61.0_{1.2}$	$56.1_{1.4}$	$61.2_{1.1}$	$56.4_{1.4}$	$80.0_{.2}$	$81.3_{.4}$	$46.5_{.8}$	$49.4_{1.3}$	$49.3_{.9}$	$51.9_{1.3}$
	+dev	-	$82.7_{1.2}$	-	$58.2_{1.5}$	-	$58.5_{1.6}$	-	$82.2_{.3}$	-	$49.3_{.2}$	-	$51.9_{.6}$
BiLSTM Seq	train	$80.5_{.7}$	$82.0_{.3}$	$62.8_{.6}$	$60.6_{1.9}$	$63.3_{.9}$	$60.7_{1.8}$	$80.8_{.5}$	$82.5_{.4}$	$47.2_{.5}$	$50.3_{1.4}$	$49.3_{.5}$	$52.8_{1.4}$
	+dev	-	$82.6_{.9}$	-	$61.6_{1.8}$	-	$61.7_{1.6}$	-	$82.8_{.2}$	-	$50.1_{1.4}$	-	$52.9_{1.6}$

Table 4: Double ablation study of BERT and Span-level NER. We report the average of five runs and their standard deviation in subscript. For RE we consider both the Strict and Boundaries settings, RE Strict score is used as the criterion for early stopping.

mentions detected by the NER decoder. Then, for every pair, the input to the relation classifier is the concatenation of each span representation $\mathbf{e}(s_i)$ and a context representation $\mathbf{c}(s_1, s_2)$, the max pooling of all tokens strictly between the two spans[1]. Once again, this pair representation is fed to a linear classifier but with a sigmoid activation so that multiple relations could be predicted for each pair.

$$\mathbf{x}(s_1, s_2) = [\mathbf{e}(s_1); \mathbf{e}(s_2); \mathbf{c}(s_1, s_2)] \quad (5)$$

$$\hat{\mathbf{y}}^{rel}(s_1, s_2) = \sigma(W^{rel}.\mathbf{x}(s_1, s_2) + \mathbf{b}^{rel}) \quad (6)$$

$\mathcal{L}_{RE}$ is computed as the binary cross-entropy over relation classes. During training, we sample up to 100 random negative pairs of detected or ground truth spans, as in (Eberts and Ulges, 2020).

Joint Training As in most related work, we simply optimize for $\mathcal{L} = \mathcal{L}_{NER} + \mathcal{L}_{RE}$.

5.3 Experimental Setting

We implement these models with Pytorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2019). For all settings, we fix a dropout rate of 0.1 across the entire network, a 0.1 word dropout for Glove embeddings and a batch size of 8. We use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A preliminary grid search on CoNLL04 led us to select a learning rate of 10^{-5} when using BERT and 5.10^{-4} with the BiLSTM[2].

We perform early stopping with patience 5 on the dev set Strict RE μ F1 score with a minimum of 10 epochs and a maximum of 100. To compare to

related work on CoNLL04, we retrain on train+dev for the optimal number of epochs as determined by early stopping. We report aggregated results from five runs in Table 4.

5.4 Comparing Boundaries and Strict Setups

This humble study first quantifies the impact of using Boundaries instead of Strict evaluation to an overestimation of 2.5 to 3 F1 points on ACE05, which is far from negligible.

But it is also interesting to see that such a mistake has almost no impact on CoNLL04, which highlights an overlooked difference between the two datasets. A simple explanation is the reduced number of entity types (4 against 7) which reduces the chance to wrongly type an entity. But we can also notice the difference in the variety of argument types in each relation. Indeed, in CoNLL04 there is a bijective mapping between a relation type and the ordered types of its arguments; this minimal difference suggests that our models have mostly learned it. On the contrary on ACE05, this mapping is much more complex (e.g. the relation PART-WHOLE fits 9 pairs of types[3]) which explains the larger difference between metrics, whereas the NER F1 scores are comparable.

5.5 Comments on the Ablations

First, we fail to reproduce Eberts and Ulges (2020)'s results on CoNLL04 with our full BERT and Span NER setting. This is most likely due

---

[1] If there are none, $\mathbf{c}(s_1, s_2) = \mathbf{0}$

[2] Search in $\{10^{-6}, 5.10^{-6}, \mathbf{10^{-5}}, 5.10^{-5}, 10^{-4}\}$ with BERT and $\{10^{-4}, \mathbf{5.10^{-4}}, 10^{-3}, 5.10^{-3}, 10^{-2}\}$ otherwise.
[3] see additional details in Appendix

to differences in our hyperparameters and implementation. Furthermore, we generally observe an important variance over runs, especially for RE.

As expected, the empirical gains mainly come from using BERT, which allows the use of simpler decoders for both NER and RE. Indeed, although our non-contextual BILOU model matches (Bekoulis et al., 2018a) on CoNLL04, the results on ACE05 are overtaken by models using external syntactic information or more sophisticated decoders with a similar BiLSTM encoder.

Comparing the Span-level and sequence tagging approaches for NER is also interesting. Although an advantage of Span-level NER is the ability to detect overlapping mentions, its contribution to end-to-end RE has never been quantified to our knowledge. Our experiments suggest that we obtain better results on this task with the more classical sequence tagging approach.

6 How to Prevent Future Mistakes?

The accumulation of mistakes and invalid comparisons should raise questions to both authors and reviewers of end-to-end RE papers. How was it possible to make them in the first place and not to detect them in the second place? How can we reduce their chance to occur in the future?

6.1 Lack of Reproducibility

First, it is no secret that the lack of reproducibility is an issue in science in general and Machine Learning in particular, but we think this is a perfect illustration of its symptoms. Indeed, in the papers we studied, we only found comparisons to reported scores and rarely an attempt to reimplement previous work by different authors. This is perfectly understandable given the complexity of such a reproduction, in particular in the multitask learning setting of end-to-end RE and often without (documented) source code.

However, this boils down to comparing results obtained in different settings. We believe that simply evaluating an implementation of the most similar previous work enables to detect differences in metrics or datasets. But it also allows to properly assess the source of empirical gains (Lipton and Steinhardt, 2018) which could come from different hyperparameter settings (Melis et al., 2018) or in-depth changes in the model.

6.2 Need for More Complete Reports

Although it is often impossible to exactly reproduce previous results even when the source code is provided, we should at least expect that the evaluation setting is always strictly reproduced. This requires a complete explicit formulation of the evaluation metrics associated with a clear and unambiguous terminology, to which end we advocate for using (Bekoulis et al., 2018a)'s. Datasets preprocessing and statistics should also be reported to provide a sanity check. This should include at least the number of sentences, entity and relation mentions as well as the details of train / test partitions.

6.3 Towards a Unified Evaluation Setting

Finally, in order to reduce confusion, we should aim at unifying our evaluation settings. We propose to always at least report RE scores with the Strict criterion, which considers both the boundaries and types of arguments. This view matches the NER metrics and truly assess end-to-end RE performance. It also happens to be the most used in previous work.

The Boundaries setting proposes a complementary measure of performance more centered on the relation. The combination of Strict and Boundaries metrics can thus provide additional insights on the models, as discussed in section 5.4 where we deduce that models can learn the bijective mapping between argument and relation types in CoNLL04. However, we believe this discussion on their specificities often lacks in articles where both metrics are reported mostly in order to compare to previous works. Hence we can only encourage to also report a Boundaries score provided sufficient explanation and exploitation of both metrics.

On the contrary, in our opinion, the Relaxed evaluation, which does not account for argument boundaries, cannot evaluate end-to-end RE since it reduces NER to Entity Classification. Furthermore, some papers report the average of NER and RE metrics (Adel and Schütze, 2017; Giorgi et al., 2019), which we believe is also an incorrect metric since the NER performance is already measured in the RE score.

Using a unified setting would also ease cross-dataset analyses and help to better reflect their often overlooked specificities.

7 Conclusion

The multiplication of settings in the evaluation of end-to-end Relation Extraction makes the comparison to previous work difficult. Indeed, in this confusion, numerous articles present unfair comparisons, often overestimating the performance of their proposed model. Furthermore, this fragmentation of the community complicates the emergence of new proposals. Our critical literature review epitomizes the need for more rigorous reports of evaluation settings, including detailed datasets statistics. And we call for a unified end-to-end RE evaluation setting to prevent future mistakes and enable more meaningful cross-domain comparisons.

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Nguyen Bach and Sameer Badaskar. 2007. A Review of Relation Extraction. *Literature review for Language and Statistics II 2*, page 15.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018a. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836, Brussels, Belgium. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018b. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Ronan Collobert and Jason Weston. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-Level Model for Relation Extraction. pages 5308–5314.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Timothy Dozat and Christopher D Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *ICLR 2017*.

Markus Eberts and Adrian Ulges. 2020. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI)*.

John M Giorgi, Xindi Wang, Nicola Sahar, Won Young Shin, Gary D Bader, Bo Wang, Young Shin, Gary D Bader, and Bo Wang. 2019. End-to-end Named Entity Recognition and Relation Extraction using Pre-trained Language Models. *arXiv preprint arXiv:1912.13415*.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv:1508.01991*.

Rohit J. Kate and Raymond Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 203–212, Uppsala, Sweden. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations*.

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

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

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

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18:198.

Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Joint Models for Extracting Adverse Drug Events from Biomedical Text. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 2838–2844.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Zachary C. Lipton and Jacob Steinhardt. 2018. Troubling Trends in Machine Learning Scholarship. *arXiv:1807.03341 [cs, stat]*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabor Melis, Chris Dyer, and Phil Blunsom. 2018. On the State of the Art of Evaluation in Neural Language Models. In *International Conference on Learning Representations (ICLR)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2019. End-to-end neural relation extraction using deep biaffine attention. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11437 LNCS, pages 729–738.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury Google, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kpf Xamla, Edward Yang, Zach Devito, Martin Raison Nabla, Alykhan Tejani, Sasank Chilamkurthy, Qure Ai, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, pages 8026–8037.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke

Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2002. Probabilistic reasoning for entity & relation recognition. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6949–6956.

Changzhi Sun, Yuanbin Wu, Man Lan, Shiliang Sun, Wenting Wang, Kuang-Chih Lee, and Kewen Wu. 2018. Extracting entities and relations with joint minimum risk training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2265, Brussels, Belgium. Association for Computational Linguistics.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, Relation, and Event Extraction with Contextualized Span Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5788–5793.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rmi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Transformers: State-of-the-art Natural Language Processing. In *ArXiv, abs/1910.03771*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. 2017. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59–66.

A Additional Implementation Details

We used an Nvidia V100 server with 16BG VRAM for our experiments. They can be run with a single Nvidia GTX 1080 with 8GB VRAM with the same hyperparameters as experimented during prototyping. We report the average number of epochs and time for every configuration in Table 5. We report the number of parameters in our models in Table 6.

Model	CoNLL04		ACE05	
	Ep.	Time	Ep.	Time
BERT + Span	52	166	25	160
BERT + BILOU	16	20	22	50
BiLSTM + Span	20	52	17	100
BiLSTM + BILOU	14	7	14	18

Table 5: Average number of epochs before early stopping and corresponding runtime in minutes for a training with early stopping on the dev RE Strict μ F1 score.

Module	CoNLL04	ACE05
BERT Embedder	108 M	108 M
GloVe Embedder	2.6 M	5.6 M
charBiLSTM	34 k	35 k
BiLSTM Encoder	2.3 M	2.3 M
Span NER	4 k	7 k
BILOU NER	13 k	22 k
RE Decoder	12 k	14 k
BERT + Span	108 M	108 M
BERT + BILOU	108 M	108 M
BiLSTM + Span	5 M	8 M
BiLSTM + BILOU	5 M	8 M

Table 6: Number of parameters in the different modules of our models.

B Additional Datasets Statistics

We provide more detailed statistics on the two datasets we used for our experimental study in Tables 7 and 8. We believe that reporting the number of sentences, entity mentions and relation mentions per training partition is a minimum to enable sanity checks ensuring data integrity.

	Reference	Train	Dev	Test	Total
Sentences	(R&Y, 04)	-	-	-	1437
	(G, 16)	922	231	288	1441
	Ours	922	231	288	1441
Tokens	(A&S, 17)	23,711	6,119	7,384	37,274
	Ours	26,525	6,993	8,336	41,854
Entities	(R&Y, 04)	-	-	-	5,336
	(A&S, 17)	3,373	858	1,071	5,302
	Ours	3,377	893	1,079	5,349
Relations	(R&Y, 04)	-	-	-	2,040
	(A&S, 17)	1,270	351	422	2,043
	Ours	1,283	343	422	2,048

Table 7: Detailed statistics of our CoNLL04 dataset, as preprocessed by Eberts and Ulges (2020) [1]. We compare to previously reported statistics (Roth and Yih, 2004; Gupta et al., 2016; Adel and Schütze, 2017). The test sets from (Gupta et al., 2016), (Adel and Schütze, 2017) and (Eberts and Ulges, 2020) are supposedly the same but we observe differences. Only (Eberts and Ulges, 2020) released their complete training partition.

	Reference	Train	Dev	Test	Total
Documents	(L&J, 14)	351	80	80	511
	Ours	351	80	80	511
Sentences	(L&J, 14)	7,273	1,765	1,535	10,573
	Ours	10,051	2,420	2,050	14,521
Tokens	Ours	144,783	35,548	30,595	210,926
Entities	(L&J, 14)	26,470	6,421	5,476	38,367
	Ours	26,473	6,421	5,476	38,370
Relations	(L&J, 14)	4,779	1,179	1,147	7,105
	Ours	4,785	1,181	1,151	7,117

Table 8: Detailed statistics of our ACE05 dataset, following Miwa and Bansal (2016)'s preprocessing scripts[2]. We compare to previously reported statistics by (Li and Ji, 2014). The large difference in the number of sentences is likely due to a different sentence tokenizer.

---

[1]github.com/markus-eberts/spert
[2]github.com/tticoin/LSTM-ER

C   Additional Comparison of ACE05 and CoNLL04

ACE05 and CoNLL04 have key differences we propose to visualize with global statistics. First, in CoNLL04 every sentence contains at least two entity mentions and one relation while the majority of ACE05 contains no entities nor relations as depicted in Fig. 1.We can also notice that among sentences containing relations, a higher proportion of ACE05 contain several of them. Second, the variety of combinations between relation types and argument types makes RE on ACE05 much more difficult than on CoNLL04 (Fig. 2 and 3).
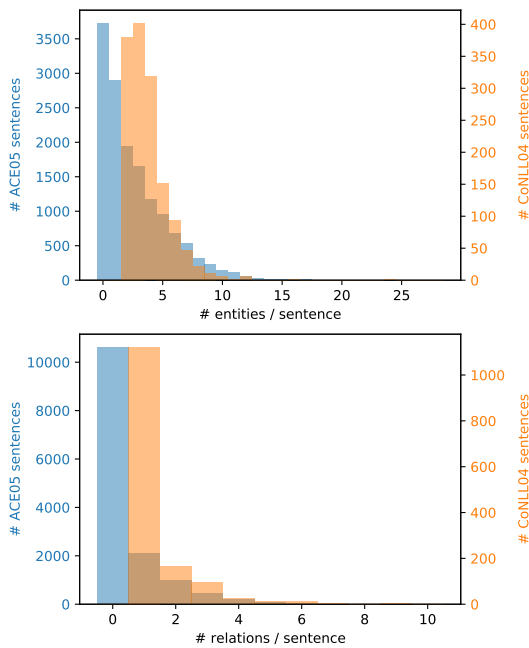

Figure 1: Distribution of the number of entity and relation mentions per sentence in ACE05 and CoNLL04.
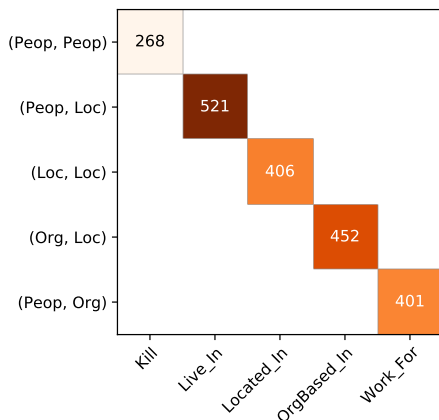

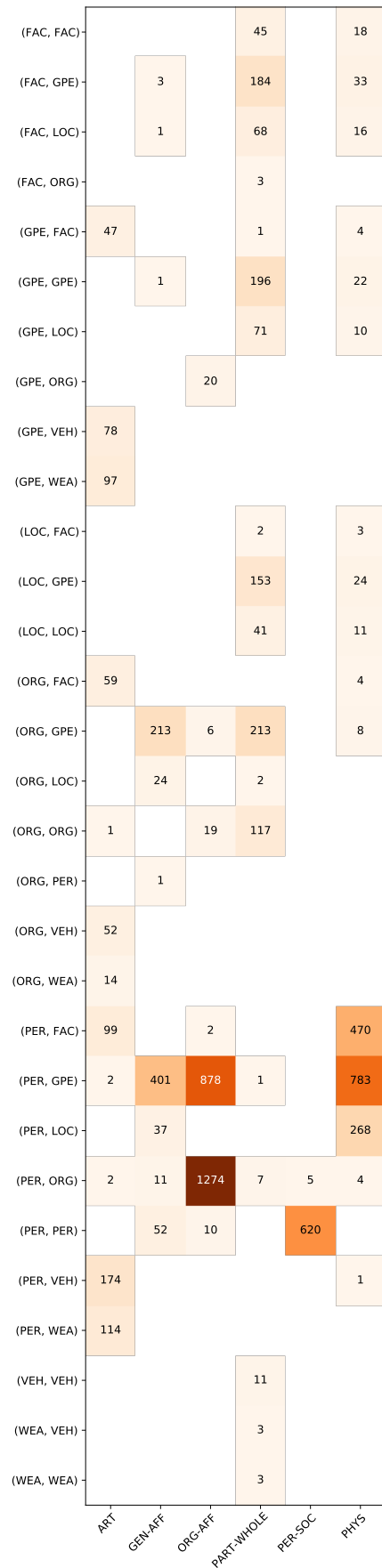Figure 2: Occurrences of each relation / argument types combination in CoNLL04.

Figure 3: Occurrences of each relation / argument types combination in ACE05.