Label Verbalization and Entailment for Effective Zero- and Few-Shot Relation Extraction

Oscar Sainz **Oier Lopez de Lacalle** **Gorka Labaka**
Ander Barrena **Eneko Agirre**

HiTZ Basque Center for Language Technologies - Ixa NLP Group
University of the Basque Country (UPV/EHU)
{oscar.sainz, oier.lopezdelacalle, gorka.labaka, ander.barrena, e.agirre}@ehu.eus

Abstract

Relation extraction systems require large amounts of labeled examples which are costly to annotate. In this work we reformulate relation extraction as an entailment task, with simple, hand-made, verbalizations of relations produced in less than 15 minutes per relation. The system relies on a pretrained textual entailment engine which is run as-is (no training examples, zero-shot) or further fine-tuned on labeled examples (few-shot or fully trained). In our experiments on TACRED we attain 63% F1 zero-shot, 69% with 16 examples per relation (17% points better than the best supervised system on the same conditions), and only 4 points short of the state-of-the-art (which uses 20 times more training data). We also show that the performance can be improved significantly with larger entailment models, up to 12 points in zero-shot, giving the best results to date on TACRED when fully trained. The analysis shows that our few-shot systems are especially effective when discriminating between relations, and that the performance difference in low data regimes comes mainly from identifying no-relation cases.

1 Introduction

Given a context where two entities appear, the Relation Extraction (RE) task aims to predict the semantic relation (if any) holding between the two entities. Methods that fine-tune large pretrained language models (LM) with large amounts of labelled data have established the state of the art (Yamada et al., 2020). Nevertheless, due to differing languages, domains and the cost of human annotation, there is typically a very small number of labelled examples in real-world applications, and such models perform poorly (Schick and Schütze, 2021).

As an alternative, methods that only need a few examples (few-shot) or no examples (zero-shot) have emerged. For instance, *prompt based learning* proposes hand-made or automatically learned task and label verbalizations (Puri and Catanzaro, 2019; Schick and Schütze, 2021; Schick and Schütze, 2020) as an alternative to standard fine-tuning (Gao et al., 2020; Scao and Rush, 2021). In these methods, the prompts are input to the LM together with the example, and the language modelling objective is used in learning and inference. In a different direction, some authors reformulate the target task (e.g. document classification) as a *pivot task* (typically question answering or textual entailment), which allows the use of readily available question answering (or entailment) training data (Yin et al., 2019; Levy et al., 2017). In all cases, the underlying idea is to cast the target task into a formulation which allows us to exploit the knowledge implicit in pre-trained LM (prompt-based) or general-purpose question answering or entailment engines (pivot tasks).

Prompt-based approaches are very effective when the label verbalization is given by one or two words (e.g. text classification), as they can be easily predicted by language models, but strive in cases where the label requires a more elaborate description, as in RE. We thus **propose to reformulate RE as an entailment problem**, where the verbalizations of the relation label are used to produce a hypothesis to be confirmed by an off-the-shelf entailment engine.

In our work[1] we have manually constructed verbalization templates for a given set of relations. Given that some verbalizations might be ambiguous (between city of birth and country of birth, for instance) we complemented them with entity type constraints. In order to ensure that the manual work involved is limited and practical in real-world applications, we allowed at most 15 minutes of manual labor per relation. The verbalizations are used as-is for zero-shot RE, but we also recast labelled RE examples as entailment pairs and fine-tune the en-

[1] Code and splits available at: https://github.com/osainz59/Ask2Transformers

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1199–1212
November 7–11, 2021.

tailment engine for few-shot RE.

The results on the widely used TACRED (Zhang et al., 2017) RE dataset in zero- and few-shot scenarios are excellent, well over state-of-the-art systems using the same amount of data. In addition our method scales well with large pre-trained LMs and large amounts of training data, reporting the best results on TACRED to date.

2 Related Work

Textual Entailment. It was first presented by Dagan et al. (2006) and further developed by Bowman et al. (2015) who called it Natural Language Inference (NLI). Given a textual premise and hypothesis, the task is to decide whether the premise entails or contradicts (or is neutral to) the hypothesis. The current state-of-the-art uses large pre-trained LM fine-tuned in NLI datasets (Lan et al., 2020; Liu et al., 2019; Conneau et al., 2020; Lewis et al., 2020; He et al., 2021).

Relation Extraction. The best results to date on RE are obtained by fine-tuning large pre-trained language models equipped with a classification head. Joshi et al. (2020) pretrains a masked language model on random contiguous spans to learn span-boundaries and predict the entire masked span. LUKE (Yamada et al., 2020) further pretrains a LM predicting entities from Wikipedia, and using entity information as an additional input embedding layer. K-Adapter (Wang et al., 2020) fixes the parameters of the pretrained LM and use Adapters to infuse factual and linguistic knowledge from Wikipedia and dependency parsing.

TACRED (Zhang et al., 2017) is the largest and most widely used dataset for RE in English. It is derived from the TAC-KBP relation set, with labels obtained via crowdsourcing. Although alternate versions of TACRED have been published recently (Alt et al., 2020; Stoica et al., 2021), the state of the art is mainly tested in the original version.

Zero-Shot and Few-Shot learning. Brown et al. (2020) showed that task descriptions (*prompts*) can be fed into LMs for task-agnostic and few-shot performance. In addition, (Schick and Schütze, 2020; Schick and Schütze, 2021; Tam et al., 2021) extend the method and allow finetuning of LMs on a variety of tasks. Prompt-based prediction treats the downstream task as a (masked) language modeling problem, where the model directly generates a textual response to a given prompt. The manual generation of effective prompts is costly and requires domain expertise. Gao et al. (2020) provide an effective way to generate prompts for text classification tasks that surpasses the performance of hand picked ones. The approach uses few-shot training with a generative T5 model (Raffel et al., 2020) to learn to decode effective prompts. Similarly, Liu et al. (2021) automatically search prompts in a embedding space which can be simultaneously fine-tuned along with the pre-trained language model. Note that previous prompt-based models run their zero-shot models on a semi-supervised setting in which some amount of labeled data is given in training. Prompts can be easily generated for text classification. Other tasks require more elaborate templates (Goswami et al., 2020; Li et al., 2021) and currently no effective prompt-based methods for RE exist.

Besides prompt-based methods, the use of pivot tasks has been widely use for few/zero-shot learning. For instance, relation and event extraction have been cast as a question answering problem (Levy et al., 2017; Du and Cardie, 2020), associating each slot label to at least one natural language question. Closer to our work, NLI has been shown too to be a successful pivoting task for text classification (Yin et al., 2019, 2020; Wang et al., 2021; Sainz and Rigau, 2021). These works verbalize the labels, and apply an entailment engine to check whether the input text entails the label description.

In similar work to ours, the relation between entailment and RE was explored by Obamuyide and Vlachos (2018). In their work they present some preliminary experiments where they cast RE as entailment, but only evaluate performance as binary entailment, not as a RE task. As a consequence they do not have competing positive labels and avoid RE inference and the issue of detecting no-relation.

Partially vs. fullly unseen labels in RE. Existing zero/few-shot RE models usually see some labels during training (*label partially unseen*), which helps generalize to the unseen label (Levy et al., 2017; Obamuyide and Vlachos, 2018; Han et al., 2018; Chen and Li, 2021). These approaches do not fully address the data scarcity problem. In this work we address the more challenging *label fully unseen* scenario.

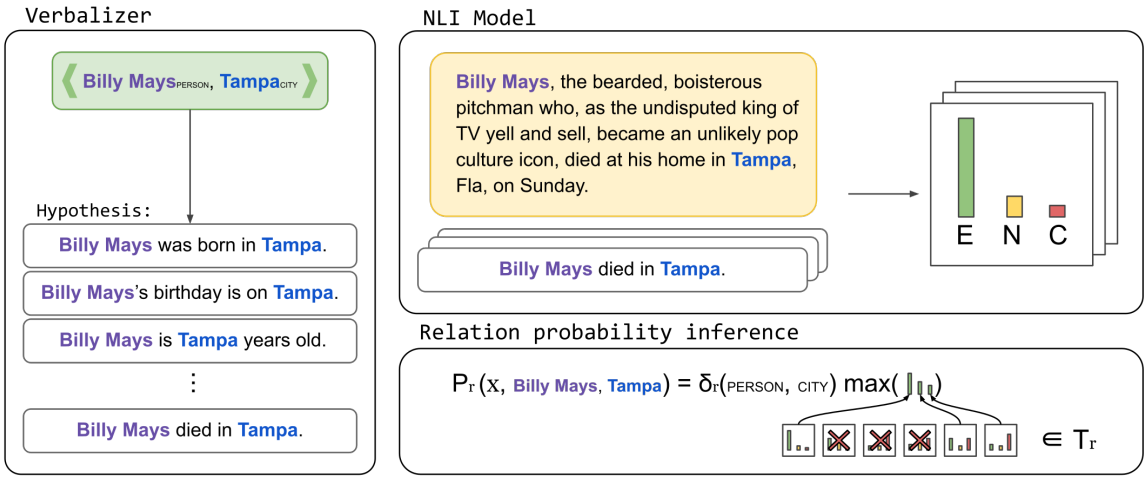

Figure 1: General workflow of our entailment-based RE approach.

3 Entailment for RE

In this section we describe our models for zero- and few-shot RE.

3.1 Zero-shot relation extraction

We reformulate RE as an entailment task: given the input text containing the two entity mentions as the premise and the verbalized description of a relation as hypothesis, the task is to infer if the premise entails the hypothesis according to the NLI model. Figure 1 illustrates the main 3 steps of our system. The first step is focused on relation verbalization to generate the set of hypotheses. In the second we run the NLI model[2] and obtain the entailment probability for each hypothesis. Finally, based on the probabilities and the entity types, we return the relation label that maximizes the probability of the hypothesis, including the NO-RELATION label.

Verbalizing relations as hypothesis. The hypotheses are automatically generated using a set of templates. Each template verbalizes the relation holding between two entity mentions. For instance, the relation PER:DATE_OF_BIRTH can be verbalized with the following template: `{subj}'s birthday is on {obj}`. More formally, given the text x that contains the mention of two entities (x_{e1}, x_{e2}) and template t, the hypothesis h is generated by VERBALIZE(t, x_{e1}, x_{e2}), which substitutes the `subj` and `obj` in the t with the entities x_{e1} and x_{e2}, respectively[3]. Figure 1 shows four verbalizations for the given entity pair.

A relation label can be verbalized by one or more templates. For instance, in addition to the previous template, PER:DATE_OF_BIRTH is also verbalized with `{subj} was born on {obj}`. At the same time, a template can verbalize more than one relation label. For example, `{subj} was born in {obj}` verbalizes PER:COUNTRY_OF_BIRTH and PER:CITY_OF_BIRTH. In order to cope with such ambiguous verbalizations, we added the entity type information to each relation, e.g. `COUNTRY` and `CITY` for each of the relations in the previous example. [4]

We defined a function δ_r for every relation $r \in R$ that checks the entity coherence between the template and the current relation label:

$$\delta_r(e_1, e_2) = \begin{cases} 1 & e_1 \in E_{r1} \wedge e_2 \in E_{r2} \\ 0 & \text{otherwise} \end{cases}$$

where e_1 and e_2 are the entity types of the first and second arguments, E_{r1} and E_{r2} are the set of allowed types for the first and second entities in relation r. This function is used at inference time, to discard relations that do not match the given types. Appendix C lists all templates and entity type restrictions used in this work.

NLI for inferring relations. In a second step we make use of the NLI model to infer the relation label. Given the text x containing two entities x_{e1}

[2] We describe the NLI models in Section 4.3

[3] Note that the entities are given in a fixed order, that is the relation needs to hold between x_{e1} and x_{e2} in that order; the reverse (x_{e2} and x_{e1}) would be a different example.

[4] Alternatively, one could think on more specific verbalizations, such as `{subj} was born in the city of {obj}` for PER:CITY_OF_BIRTH. In the checks done in the available 15 min. such specific verbalizations had very low recall and were not finally selected.

and x_{e2} the system returns the relation $\hat{r}$ from the set of possible relation labels R with the highest entailment probability as follows:

$$\hat{r} = \arg\max_{r \in R} \mathsf{P}_r(x, x_{e1}, x_{e2}) \quad (1)$$

The probability of each relation P_r is computed as the probability of the hypothesis that yields the maximum entailment probability (Eq. 2), among the set of possible hypothesis. In case the two entities do not match the required entity types, the probability would be zero.

$$\mathsf{P}_r(x, x_{e1}, x_{e2}) = \delta_r(e_1, e_2) \max_{t \in T_r} \mathsf{P}_{NLI}(x, hyp)$$
$$where\ hyp = \text{VERBALIZE}(t, x_{e1}, x_{e2}) \quad (2)$$

where P_{NLI} is the entailment probability between the input text and the hypothesis generated by the template verbalizer. Although entailment models return probabilities for entailment, contradiction and neutral, P_{NLI} just makes use of the entailment probability[5]. The right hand-side of Figure 1 shows the application of NLI models and how the probability for each relation, P_r, is computed.

Detection of no-relation. In supervised RE, the NO-RELATION case is taken as an additional label. In our case we examined two approaches.

In **template-based detection** we propose an additional template as if it was yet another relation label, and treated it as another positive relation in Eq. 1. The template for NO-RELATION: `{subj} and {obj} are not related`.

In **threshold-based detection** we apply a threshold $\mathcal{T}$ to P_r in Eq. 2. If none of the relations surpasses the threshold, then our system returns NO-RELATION. On the contrary, the model returns the relation label of highest probability (Eq. 1). When no development data is available, the threshold $\mathcal{T}$ is set to 0.5. Alternatively, we estimate $\mathcal{T}$ using the available development dataset, as described in the experimental part.

3.2 Few-Shot relation extraction

Our system is based on a NLI model which has been pretrained on annotated entailment pairs. When labeled relation examples exist, we can reformulate them as labelled NLI pairs, and use them to fine-tune the NLI model to the task at hand, that is, assigning highest entailment probability to the verbalizations of the correct relation, and assigning low entailment probabilities to the rest of the hypothesis (see Eq. 2).

Given a set of labelled relation examples, we use the following steps to produce labelled entailment pairs for fine-tuning the NLI model. 1) For each **positive** relation example we generate at least one **entailment** instance with the templates that describes the current relation. That is, we generate one or several premise-hypothesis pairs labelled as entailment. 2) For each **positive** relation example we generate one **neutral** premise-hypothesis instance, taken at random from the templates that do not represent the current relation. 3) For each **negative** relation example we generate one **contradiction** example, taken at random from the templates of the rest of relations.

If a template is used for the no-relation case, we do the following: First, for each **no-relation** example we generate one **entailment** example with the no-relation template. Then, for each **positive** relation example we generate one **contradiction** example using the no-relation template.

4 Experimental Setup

In this section we describe the dataset and scenarios we have used for evaluation, how we performed the verbalization process, the different pre-trained NLI models we have used and the state-of-the-art baselines that we compare with.

4.1 Dataset and scenarios

We designed three different low-resource scenarios based on the large-scale TACRED (Zhang et al., 2017) dataset. The full dataset consists of 42 relation labels, including the NO-RELATION label, and each example contains the information about the entity type, among other linguistic information. The scenarios are described in Table 1 and are formed by different splits of the original dataset. We applied a stratified sampling method to keep the original label distribution.

Zero-Shot. The aim of this scenario is the evaluation of the models when no data is available for training. We present two different situations on this scenario: 1) no data is available for development (0% split) and 2) a small development set is available with around 2 examples per relation (1%

[5]The probabilities for relations P_r defined in Eq. 2 are independent from each other, which, in a way, they could be easily extended to multi-label classification task.

Scenario	Split	Train (Gold) # Pos mean	Train (Gold) # Pos total	Train (Gold) # Neg total	Train (Silver) # Pos mean	Train (Silver) # Pos total	Train (Silver) # Neg total	Development # Pos mean	Development # Pos total	Development # Neg total
Full training	100%	317.4	13013	55112	-	-	-	132.6	5436	17195
Zero-Shot	No Dev	-	-	-	-	-	-	0	0	0
	1% Dev	-	-	-	-	-	-	1.9	54	173
Few-Shot	1%	3.6	130	552	-	-	-	1.9	54	173
	5%	16.3	651	2756	-	-	-	7.0	272	861
	10%	32.6	1302	5513	-	-	-	13.6	544	1721
Data Augment.	0%	0	0	0	246.3	9850	41205	1.9	54	173
	1%	3.6	130	552	246.3	9850	41205	1.9	54	173
	5%	16.3	651	2756	246.3	9850	41205	7.0	272	861
	10%	32.6	1302	5513	246.3	9850	41205	13.6	544	1721

Table 1: Statistics about the dataset scenarios based on TACRED used in the paper, including positive examples per relation, total amount of positive examples and the total amount of negative (no-relation) examples.

split)[6]. In this scenario the models are not allowed to train their own parameters but development data is used to adjust the hyperparameters.

Few-Shot. This scenario presents the challenge of solving the RE task with just a few examples per relation. We present three settings commonly used in few-shot learning (Gao et al., 2020) [7]: around 4 examples per relation (1% of the training data in TACRED), around 16 examples per relation (5%) and around 32 examples per relation (10%). We reduced the development set following the same ratio.

Full Training. In this setting we use all available training and development data.

Data Augmentation. In this scenario we want to test whether a silver dataset produced by running our systems on untagged data can be used to train a supervised relation extraction system (cf. Section 3). In this scenario 75% of the training data in TACRED is set aside as unlabeled data[8], and the rest of the training data is used in different splits (ranging from 1% to 10%). Under this setting we carried out two type of experiments: In the zero-shot experiments (0% in the table) we use our NLI based model to annotate the silver data and then fine-tune the RE model exclusively on the silver data. In the few-shot experiments the NLI model is first fine-tuned with the gold data, then used to annotate the silver data and finally the RE model is fine-tuned over both, silver and gold, annotations.

4.2 Hand-crafted relation templates

We manually created the templates to verbalize relation labels, based on the TAC-KBP guidelines which underlie the TACRED dataset. We limited the time for creating the templates of each relation to less than 15 minutes. Overall, we created 1-8 templates per relation (2 on average) (cf. Appendix C for full list).

The verbalization process consists of generating one or more templates that describe the relation and contain the placeholders `{subj}` and `{obj}`. The developer building the templates was given the task guidelines (brief description of the relation, including one or two examples and the type of the entities) and a NLI model (*roberta-large-mnli* checkpoint). For a given relation, he/she would create a template (or set of templates) and check whether the NLI model is able to output a high entailment probability for the template when applied on the guideline example(s). He/she could run this process for any new template that he/she could come up with. There was no strict threshold involved for selecting the templates, just the intuition of the developer. The spirit was to come up with simple templates quickly, and not to build numerous complex templates or to optimize entailment probabilities.

4.3 Pre-Trained NLI models

For our experiments we tried different NLI models that are publicly available with the Hugging Face Transformers (Wolf et al., 2020) python library.

[6]This setting is comparable to one where the examples in the guidelines are used as development.

[7]The commonly reported value in few-shot scenarios is 16 examples per label. We also added the 3-8 and 32 examples settings in the evaluation.

[8]We use part of the original TACRED dataset to produce silver data in order not to introduce noise coming from different documents and/or pre-processing steps.

NLI Model	# Param.	MNLI Acc.	No Dev ($\mathcal{T} = 0.5$) Pr.	Rec.	F1	1% Dev Pr.	Rec.	F1
$\text{ALBERT}_{\text{xxLarge}}$	223M	90.8	32.6	**79.5**	46.2	55.2	58.1	56.6 $\pm$1.4
RoBERTa	355M	90.2	32.8	75.5	45.7	58.5	53.1	55.6 $\pm$1.3
BART	406M	89.9	39.0	63.1	48.2	60.7	46.0	52.3 $\pm$1.8
$\text{DeBERTa}_{\text{xLarge}}$	900M	91.7	40.3	77.7	53.0	**66.3**	59.7	**62.8** $\pm$1.7
$\text{DeBERTa}_{\text{xxLarge}}$	1.5B	91.7	**46.6**	76.1	**57.8**	63.2	**59.8**	61.4 $\pm$1.0

Table 2: Zero-Shot scenario results (Precision, Recall and F1) for our system using several pre-trained NLI models in two settings: no development (default threshold $\mathcal{T}$=0.5), and small development (1% Dev.) for setting $\mathcal{T}$. In the leftmost columns we report the number of parameters and the accuracy in MNLI. For the 1% setting we report the median measures along with the F1 standard deviation in 100 runs.

We tested the following models which implement different architectures, sizes and pre-training objectives and were fine-tuned mainly over the MNLI (Williams et al., 2018) dataset[9]: ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019), BART (Lewis et al., 2020) and DeBERTa v2 (He et al., 2021). Table 2 reports the number of parameters of these models. Further details on models can be found in Appendix A.

For each of the scenarios we have tested different models. In zero-shot and full training scenarios we compare all the pre-trained models using the templates described in Section 4.2. For few-shot we used RoBERTa for comparability, as it was used in state-of-the-art systems (cf. Section 4.4), and DeBERTa which is the largest NLI model available on the HUB[10]. Finally, we only tested RoBERTa in data-augmentation experiments.

We ran 3 different runs on each of the experiments using different random seeds. In order to make a fair comparison with state-of-the-art systems (cf section 4.4.), we performed a hyperparameter exploration in the full training scenario, using the resulting configuration also in the zero/few-shot scenarios. We fixed the batch size at 32 for both RoBERTa and DeBERTa, and search the optimum learning-rate among $\{1e^{-6}, 4e^{-6}, 1e^{-5}\}$ on the development set. The best results were obtained using $4e^{-6}$ as learning-rate. For more detailed information refer to the Appendix B.

4.4 State-of-the-art RE models

We compared the NLI approach with the systems reporting the best results to date on TACRED: SpanBERT (Joshi et al., 2020), K-Adapter (Wang et al., 2020) and LUKE (Yamada et al., 2020) (cf. Section 2). In addition, we also report the results obtained by the vanilla RoBERTa baseline proposed by Wang et al. (2020) that serves as a reference for the improvements. We re-trained the different systems on each scenario setting using their publicly available implementations and best performing hyperparameters reported by the authors. All these models have a comparable number of parameters.

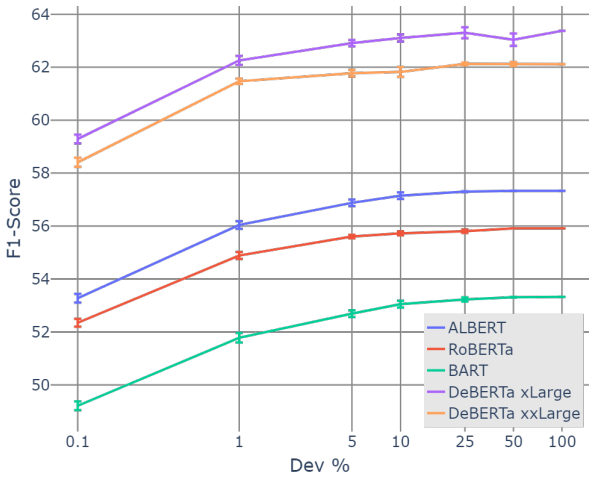

Figure 2: Zero-shot scenario results. Mean F1 and standard error scores when setting $\mathcal{T}$ on increasing number of development examples.

5 Results

5.1 Zero-Shot

Table 2 shows the results for different pre-trained NLI models, as well as the number of parameters and the MNLI *matched* accuracy. These results were obtained by using the threshold for negative relations, as we found that it works substantially better than the no-relation template alternative (cf. Section 3.1). For instance, RoBERTa yields an

[9]ALBERT was trained in some additional NLI datasets.
[10]https://huggingface.co/models

Model	1%			5%			10%		
	Pr.	Rec.	F1	Pr.	Rec.	F1	Prec.	Rec.	F1
SpanBERT	0.0	0.0	0.0 ±0.0	36.3	23.9	28.8 ±13.5	3.2	1.1	1.6 ±20.7
RoBERTa	56.8	4.1	7.7 ±3.6	52.8	34.6	41.8 ±3.3	61.0	50.3	55.1 ±0.8
K-Adapter	73.8	7.6	13.8 ±3.4	56.4	37.6	45.1 ±0.1	62.3	50.9	56.0 ±1.3
LUKE	61.5	9.9	17.0 ±5.9	57.1	47.0	51.6 ±0.4	60.6	60.6	60.6 ±0.4
$\text{NLI}_{\text{RoBERTa}}$ (ours)	56.6	55.6	56.1 ±0.0	60.4	68.3	64.1 ±0.2	**65.8**	69.9	67.8 ±0.2
$\text{NLI}_{\text{DeBERTa}}$ (ours)	**59.5**	**68.5**	**63.7** ±0.0	**64.1**	**74.8**	**69.0** ±0.2	62.4	**74.4**	**67.9** ±0.5

Table 3: Few-shot scenario results with 1%, 5% and 10% of training data. Precision, Recall and F1 score (standard deviation) of the median of 3 different runs are reported. Top four rows for third-party RE systems run by us.

F1 of 30.1[11] well below the 45.7 when using the default threshold ($\mathcal{T} = 0.5$). Overall we see an excellent zero-shot performance across all the models and settings proving that the approach is robust and model agnostic.

Regarding **pre-trained models**, the best F1 scores are obtained by the two DeBERTa v2 models, which also score the best on the MNLI dataset. Note that all the models achieve similar scores on MNLI, but small differences in MNLI result in large performance gaps when they come to RE, e.g. the 1.5 difference in MNLI between RoBERTa and DeBERTa becomes 7 points in No Dev. and 1% Dev. We think the larger differences in RE are due to the generalization ability of some of the larger models to domain and task differences.

The table includes the results for different values of the $\mathcal{T}$ hyperparameter. In the most challenging setting, with default $\mathcal{T}$, the results are worst, with at most 57.8 F1. However, using as few as 2 examples per relation in average (1% Dev. setting) the results improve significantly.

We performed further experiments using larger amounts of development data to tune $\mathcal{T}$. Figure 2 shows that, for all models, the most significant improvement occurs at the interval [0%, 1%) and that the interval [1%, 100%] is almost flat. The best results with all development data is 63.4%, only 0.6 points better than using 1% of development. These results show clearly that a small number of examples suffice to set an optimal threshold.

5.2 Few-Shot

Table 3 shows the results of competing RE systems and our systems on the few-shot scenario. We report the median and standard deviation across 3 different runs. The competing RE methods suffer a large performance drop, specially for the smallest training setting. For instance, the SpanBERT system (Joshi et al., 2020) has difficulties to converge, even with the 10% of data setting. Both K-Adapter (Wang et al., 2020) and LUKE (Yamada et al., 2020) improve over the RoBERTa system (Wang et al., 2020) in all three settings, but they are well below our $\text{NLI}_{\text{RoBERTa}}$ system, with improvements of 48, 22 and 13 points against the baseline in each setting. We also report our method based on $\text{DeBERTa}_{\text{xLarge}}$, which is specially effective in the smaller settings.

We would like to note that the zero-shot $\text{NLI}_{\text{RoBERTa}}$ system (1% Dev) is comparable in terms of F1 score to a vanilla RoBERTa trained with 10% of the training data. That is, 54 templates (10.5 hours, plus 23 development examples are roughly equivalent to 6800 annotated examples[12] for training (plus 2265 development) .

Model	Pr.	Rec.	F1
SpanBERT	70.8	70.9	70.8
RoBERTa	70.2	72.4	71.3
K-Adapter	70.1	74.0	72.0
LUKE	70.4	75.1	72.7
$\text{NLI}_{\text{RoBERTa}}$ (ours)	71.6	70.4	71.0
$\text{NLI}_{\text{DeBERTa}}$ (ours)	**72.5**	**75.3**	**73.9**

Table 4: Full training results (TACRED). Top four rows for third-party RE systems as reported by authors.

5.3 Full training

Some zero-shot and few-shot systems are not able to improve results when larger amounts of training data are available. Table 4 reports the results when the whole train and development datasets are used, which is comparable to official results

[11] Results ommitted from Table 2 for brevity.

[12] Unfortunately we could not find the time estimates for annotating examples.

Model	0%	1%	5%	10%
RoBERTa	-	7.7	41.8	55.1
+ Zero-Shot DA	**56.3**	**58.4**	58.8	59.7
+ Few-Shot DA	-	**58.4**	**64.9**	**67.7**

Table 5: Data Augmentation scenario results (F1) for different gold training sizes. Silver annotations by the zero-shot and few-shot $\text{NLI}_{\text{RoBERTa}}$ model.

Model	Scenario		P	PvsN
$\text{NLI}_{\text{DeBERTa}}$	Zero-Shot	No Dev	85.6	59.5
		1% Dev	85.6	67.7
	Few-Shot	5%	89.7	74.5
	Full train	-	92.2	77.8
LUKE	Few-Shot	5%	69.3	63.4
	Full train	-	90.2	77.3

Table 6: Performance of selected systems and scenarios on two metrics: the binary task of detecting a positive relation vs. no-relation (PvsN column, F1) and detecting the correct relation among positive cases (P, F1).

on TACRED. Focusing on our $\text{NLI}_{\text{RoBERTa}}$ system, and comparing it to the results in Table 3, we can see that it is able to effectively use the additional training data, improving from 67.9 to 71.0. When compared to a traditional RE system, it performs on a par to RoBERTa, and a little behind K-Adapter and LUKE, probably due to the infused knowledge which our model is not using. These results show that our model keeps improving with additional data and that it is competitive when larger amounts of training is available. The results of $\text{NLI}_{\text{DeBERTa}}$ show that our model can benefit from larger and more effective pre-trained NLI systems even in full training scenarios, and in fact achieves the best results to date on the TACRED dataset.

5.4 Data augmentation results

In this section we explore whether our NLI-based system can produce high-quality silver data which can be added to a small amount of gold data when training a traditional supervised RE system, e.g. the RoBERTa baseline (Wang et al., 2020). Table 5 reports the F1 results on the data augmentation scenario for different amounts of gold training data. Overall, we can see that both our zero-shot and few-shot methods[13] provide good quality silver data, as they improve significantly over the baseline in all settings. Although the zero-shot and few-shot methods yield the same result with 1% of training data, the few-shot model is better in the rest of training regimes, showing that it can effectively use the available training data in each case to provide better quality silver data. If we compare the results in this table with those of the respective NLI-based system with the same amount of gold training instances (Tables 2 and 3) we can see that the results are comparable, showing that our NLI-based system and a traditional RE system trained with silver annotations have comparable performance. A practical advantage of a traditional RE system trained with our silver data is that is easier to integrate on available pipelines, as one just needs to download the trained Transformer model. It also makes it easy to check additive improvements in the RE method.

[13]The zero-shot 1% Dev model is used in all data augmentation experiments, while the few-shot method changes to use the available data at each run (1%, 5% and 10%), both with RoBERTa

6 Analysis

Relation extraction can be analysed according to two auxiliary metrics: the binary task of detecting a positive relation vs. no-relation, and the multi-class problem of detecting which relation holds among positive cases (that is, discarding no-relation instances from test data). Table 6 shows the results of a selection of systems and scenarios. The first rows compare the performance of our best system, $\text{NLI}_{\text{DeBERTa}}$, across four scenarios, while the last two rows show the results for LUKE in two scenarios. The zero-shot No dev. system is very effective when discriminating the relation among positive examples (P column), only 7 points below the fully trained system, while it lags well behind when discriminating positive vs. negative, 18 points. The use of a small development data for tuning the $\mathcal{T}$ threshold closes the gap in PvsN, as expected, but the difference is still 10 points. All in all, these numbers show that our zero-shot system is very effective discriminating among positive examples, but that it still lags behind when detecting no-relation cases. Overall, the figures show the effectiveness of our methods in low data scenarios on both metrics.

Confusion analysis In supervised models some classes (relations) are better represented in training than others, usually due to data imbalance. Our system instead, represents each relations as a set of templates, which at least on a **zero-shot**

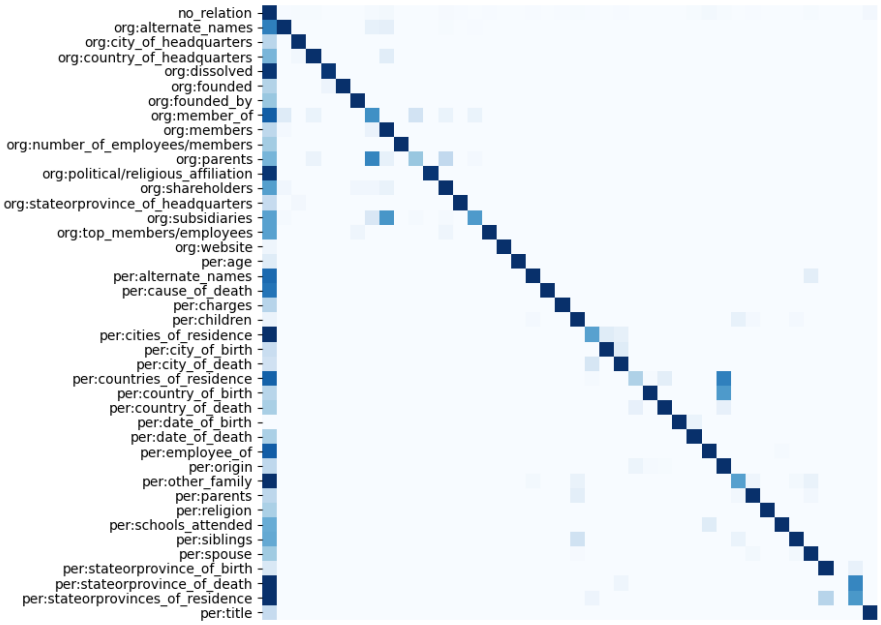

Figure 3: Confusion matrix of our $\text{NLI}_{\text{DeBERTa}}$ zero-shot system on the development dataset. The rows represent the true labels and the columns the predictions. The matrix is rowise normalized (recall in the diagonal).

scenario, should not be affected by data imbalance. The strong diagonal in the confusion matrix (Fig. 3) shows that our the model is able to discriminate properly between most of the relations (after all it achieves 85.6% accuracy, cf. Table 6), with exception of the no-relation column, which was expected. Regarding the confusion between actual relations, most of them are about **overlapping relations**, as expected. For instance, ORG:MEMBER_OF and ORG:PARENTS both involve some organization A being part or member of some other organization B, where ORG:MEMBERS is different from ORG:PARENTS in that correct fillers are distinct entities that are generally capable of autonomously ending their membership with the assigned organization[14]. Something similar occurs between ORG:MEMBERS and ORG:SUBSIDIARIES. Another reason for confusion happens when **two or more relations exist concurrently**, as in PER:ORIGIN, PER:COUNTRY_OF_BIRTH and PER:COUNTRY_OF_RESIDENCE. Finally, the model scores low on PER:OTHER_FAMILY, which is a bucket of many specific relations where only a handful were actually covered by the templates.

7 Conclusions

In this work we reformulate relation extraction as an entailment problem, and explore to what extent simple hand-made verbalizations are effective. The creation of templates is limited to 15 minutes per relation, and yet allows for excellent results in zero- and few-shot scenarios. Our method makes effective use of available labeled examples, and together with larger LMs produces the best results on TACRED to date. Our analysis indicates that the main performance difference against supervised models comes from discriminating no-relation examples, as the performance among positive examples equals that of the best supervised system using the full training data. We also show that our method can be used effectively as a data-augmentation method to provide additional labeled examples. For the future we would like to investigate better methods for detecting no-relation in zero-shot settings.

Acknowledgements

Oscar is funded by a PhD grant from the Basque Government (PRE_2020_1_0246). This work is based upon work partially supported via the IARPA BETTER Program contract No. 2019-19051600006 (ODNI, IARPA), and by the Basque Government (IXA excellence research group IT1343-19).

[14] Description extracted from the guidelines.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2021)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners.

Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. 2020. Unsupervised relation extraction from language models using constrained cloze completion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1263–1276, Online. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too.

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

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Oscar Sainz and German Rigau. 2021. Ask2Transformers: Zero-shot domain labelling with pretrained language models. In *Proceedings of the 11th Global Wordnet Conference*, pages 44–52, University of South Africa (UNISA). Global Wordnet Association.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *Computing Research Repository*, arXiv:2009.07118.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shotcomings of the tacred dataset. In *Proceedings of the Thirty-fifth AAAI Conference on Aritificial Intelligence 2021*.

Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239, Online. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

A Pre-Trained models

The pre-trained NLI models we have tested from the Transformers library are the next:

- ALBERT: *ynie/albert-xxlarge-v2-snli_mnli_fever_anli_R1_R2_R3-nli*
- RoBERTa: *roberta-large-mnli*
- BART: *facebook/bart-large-mnli*
- DeBERTa v2 xLarge: *microsoft/deberta-v2-xlarge-mnli*
- DeBERTa v2 xxLarge: *microsoft/deberta-v2-xxlarge-mnli*

B Experimental details

We carried out all the experiments on a single Titan V (16GB) except for the fine-tuning of DeBERTa, that has been done on a cluster of 4 Titan V100 (32GB). The average inference time for the zero and few-shot experiments is between 1h and 1.5h. The time needed for fine-tuning the NLI systems was at most 2.5h for RoBERTa and 5h for DeBERTa. All the experiments were done with mixed precision to speed up the overall runtime.

The whole hyperparameter settings used for fine-tuning $\text{NLI}_{\text{RoBERTa}}$ and $\text{NLI}_{\text{DeBERTa}}$ are listed below:

- **Train epochs:** 2
- **Warmup steps:** 1000
- **Learning-rate:** 4e-6
- **Batch-size:** 32
- **FP16 training**
- **Seeds:** {0, 24, 42}

Note that we are fine-tuning an already trained NLI system so we kept the number of epochs and learning-rate low. The rest of state-of-the-art systems were trained using the hyperparameters reported by the authors.

C TACRED templates

This section describes the templates used in the TACRED experiments. We performed all the experiments using the templates showed in Tables 1 (for PERSON relations) and 2 (for ORGANIZATION relations). These templates were manually created based on the TAC KBP Slot Descriptions[15] (annotation guidelines). Besides the templates, we also report the valid argument types that are accepted on each relation.

[15] https://tac.nist.gov/2014/KBP/ColdStart/guidelines/TAC_KBP_2014_Slot_Descriptions_V1.4.pdf

Relation	Templates	Valid argument types
per:alternate_names	{subj} is also known as {obj}	PERSON, MISC
per:date_of_birth	{subj}'s birthday is on {obj}	DATE
	{subj} was born on {obj}	
per:age	{subj} is {obj} years old	NUMBER, DURATION
per:country_of_birth	{subj} was born in {obj}	COUNTRY
per:stateorprovince_of_birth	{subj} was born in {obj}	STATE_OR_PROVINCE
per:city_of_birth	{subj} was born in {obj}	CITY, LOCATION
per:origin	{obj} is the nationality of {subj}	NATIONALITY, COUNTRY, LOCATION
per:date_of_death	{subj} died in {obj}	DATE
per:country_of_death	{subj} died in {obj}	COUNTRY
per:stateorprovince_of_death	{subj} died in {obj}	STATE_OR_PROVINCE
per:city_of_death	{subj} died in {obj}	CITY, LOCATION
per:cause_of_death	{obj} is the cause of {subj}'s death	CAUSE_OF_DEATH
per:countries_of_residence	{subj} lives in {obj}	COUNTRY, NATIONALITY
	{subj} has a legal order to stay in {obj}	
per:statesorprovinces_of_residence	{subj} lives in {obj}	STATE_OR_PROVINCE
	{subj} has a legal order to stay in {obj}	
per:city_of_residence	{subj} lives in {obj}	CITY, LOCATION
	{subj} has a legal order to stay in {obj}	
per:schools_attended	{subj} studied in {obj}	ORGANIZATION
	{subj} graduated from {obj}	
per:title	{subj} is a {obj}	TITLE
per:employee_of	{subj} is a member of {obj}	ORGANIZATION
per:religion	{subj} belongs to {obj}	RELIGION
	{obj} is the religion of {subj}	
	{subj} believe in {obj}	
per:spouse	{subj} is the spouse of {obj}	PERSON
	{subj} is the wife of {obj}	
	{subj} is the husband of {obj}	
per:children	{subj} is the parent of {obj}	PERSON
	{subj} is the mother of {obj}	
	{subj} is the father of {obj}	
	{obj} is the son of {subj}	
	{obj} is the daughter of {subj}	
per:parents	{obj} is the parent of {subj}	PERSON
	{obj} is the mother of {subj}	
	{obj} is the father of {subj}	
	{subj} is the son of {obj}	
	{subj} is the daughter of {obj}	
per:siblings	{subj} and {obj} are siblings	PERSON
	{subj} is brother of {obj}	
	{subj} is sister of {obj}	
per:other_family	{subj} and {obj} are family	PERSON
	{subj} is a brother in law of {obj}	
	{subj} is a sister in law of {obj}	
	{subj} is the cousin of {obj}	
	{subj} is the uncle of {obj}	
	{subj} is the aunt of {obj}	
	{subj} is the grandparent of {obj}	
	{subj} is the grandmother of {obj}	
	{subj} is the grandson of {obj}	
	{subj} is the granddaughter of {obj}	
per:charges	{subj} was convicted of {obj}	CRIMINAL_CHARGE
	{obj} are the charges of {subj}	

Table 1: Templates and valid arguments for PERSON relations.

Relation	Templates	Valid argument types
org:alternate_names	{subj} is also known as {obj}	ORGANIZATION, MISC
org:political/religious_affiliation	{subj} has political affiliation with {obj} {subj} has religious affiliation with {obj}	RELIGION, IDEOLOGY
org:top_memberts/employees	{obj} is a high level member of {subj} {obj} is chairman of {subj} {obj} is president of {subj} {obj} is director of {subj}	PERSON
org:number_of_employees/members	{subj} employs nearly {obj} people {subj} has about {obj} employees	NUMBER
org:members	{obj} is member of {subj} {obj} joined {subj}	ORGANIZATION, COUNTRY
org:subsidiaries	{obj} is a subsidiary of {subj} {obj} is a branch of {subj}	ORGANIZATION, LOCATION
org:parents	{subj} is a subsidiary of {obj} {subj} is a branch of {obj}	ORGANIZATION, COUNTRY
org:founded_by	{subj} was founded by {obj} {obj} founded {subj}	PERSON
org:founded	{subj} was founded in {obj} {subj} was formed in {obj}	DATE
org:dissolved	{subj} existed until {obj} {subj} disbanded in {obj} {subj} dissolved in {obj}	DATE
org:country_of_headquarters	{subj} has its headquarters in {obj} {subj} is located in {obj}	COUNTRY
org:stateorprovince_of_headquarters	{subj} has its headquarters in {obj} {subj} is located in {obj}	STATE_OR_PROVINCE
org:city_of_headquarters	{subj} has its headquarters in {obj} {subj} is located in {obj}	CITY, LOCATION
org:shareholders	{obj} holds shares in {subj}	ORGANIZATION, PERSON
org:website	{obj} is the URL of {subj} {obj} is the website of {subj}	URL

Table 2: Templates and valid arguments for ORGANIZATION relations.