Textual Entailment for Event Argument Extraction: Zero- and Few-Shot with Multi-Source Learning

Oscar Sainz[1], **Itziar Gonzalez-Dios**[1],
Oier Lopez de Lacalle[1], **Bonan Min**[2], and **Eneko Agirre**[1]

[1]HiTZ Basque Center for Language Technologies - Ixa NLP Group
University of the Basque Country UPV/EHU
[2]Raytheon BBN Technologies
oscar.sainz@ehu.eus

Abstract

Recent work has shown that NLP tasks such as Relation Extraction (RE) can be recasted as Textual Entailment tasks using verbalizations, with strong performance in zero-shot and few-shot settings thanks to pre-trained entailment models. The fact that relations in current RE datasets are easily verbalized casts doubts on whether entailment would be effective in more complex tasks. In this work we show that entailment is also effective in Event Argument Extraction (EAE), reducing the need of manual annotation to 50% and 20% in ACE and WikiEvents respectively, while achieving the same performance as with full training. More importantly, we show that recasting EAE as entailment alleviates the dependency on schemas, which has been a roadblock for transferring annotations between domains. Thanks to the entailment, the multi-source transfer between ACE and WikiEvents further reduces annotation down to 10% and 5% (respectively) of the full training without transfer. Our analysis shows that the key to good results is the use of several entailment datasets to pre-train the entailment model. Similar to previous approaches, our method requires a small amount of effort for manual verbalization: only less than 15 minutes per event argument type is needed, and comparable results can be achieved with users with different level of expertise.

1 Introduction

Building Information Extraction (IE) systems for real-world applications is very costly and has suffered from data-scarcity problems, due in part to the expertise and time required to annotate training data at a large scale with sufficient consistency, but also due to poor transfer between domains: IE annotations depend on the schema used in each domain, and moving to new domains requires new schemas, new annotation guidelines and the manual annotation of new data. In many cases, there is some information overlap between schemas, but performing transfer learning to leverage such overlap (i.e. learning from **multiple sources**) can be difficult: it often requires manually mapping labels between schemas, which is typically brittle, cumbersome and requires costly domain expertise (Kalfoglou and Schorlemmer, 2003).

In order to save annotation effort, recent work recasts IE tasks as Textual Entailment tasks (White et al., 2017; Poliak et al., 2018a; Levy et al., 2017; Sainz et al., 2021). For instance, Sainz et al. (2021) manually verbalize each relation type in the Relation Extraction (RE) dataset TACRED (Zhang et al., 2017) to generate hypotheses for each test example, and then apply an entailment model to output the relation type of the hypothesis with highest entailment probability. The entailment model is typically based on large language models pre-trained on entailment datasets such as MNLI (Williams et al., 2018). The approach obtains very strong results on zero-shot and few-shot scenarios, but we note that TACRED contains relations between two entities that are easily verbalizable,[1] casting doubts on whether entailment would be effective in more complex IE tasks. Event Argument Extraction (EAE) involves more complex contexts, higher ambiguity in the words that trigger events, and depends on the event type in addition to the relation (see Figure 1).

In this work, we present the first system for EAE that addresses the task as an entailment problem. We empirically show the robustness of the method on the zero-shot, few-shot and full training regimes, obtaining state-of-the-art results on ACE (Walker et al., 2006) and WikiEvents (Li et al., 2021b). In addition, we make the following contributions: (1) We show that our method reduces schema dependency, as it improves the performance on the WikiEvents results using additional ACE training data and vice versa with no extra manual work. (2)

[1]For instance, PER:DATE_OF_BIRTH can be verbalized as `{subj}'s birthday is on {obj}` in which subj and obj refers to the two text mentions involved in the relation.

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2439 - 2455

Ablation results show that training with several NLI datasets is significantly better than just using MNLI. (3) Our analysis of the manual work required for writing templates and annotating arguments sheds light in the sweet spot for future applications, and shows that template writing does not require much domain expertise as shown by the results using an independent novice template writer. We make the code, templates and models publicly available.[2]

2 Related Work

Textual Entailment Given a textual premise and a hypothesis, the task is to decide whether the premise entails or contradicts (or is neutral to) the hypothesis (Dagan et al., 2006). The current state-of-the-art uses large pre-trained Language Models (LM) (Lan et al., 2020; Liu et al., 2019; Conneau et al., 2020; Lewis et al., 2020; He et al., 2021) fine-tuned on manually annotated datasets such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), FEVER (Thorne et al., 2018) or ANLI (Nie et al., 2020). The task is also known as Natural Language Inference (NLI).

Prompt and Pivot task based learning has emerged as a candidate solution for data-scarcity problems (Le Scao and Rush, 2021; Min et al., 2021; Liu et al., 2021a). The use of discrete (Gao et al., 2021; Schick and Schütze, 2021a,b,c) or continuous (Liu et al., 2021b) prompts allowed language models to perform significantly better on many text classification tasks. Closely related to our approach, several works make use of a high-resource supervised task such as Question Answering or entailment as pivot tasks (Yin et al., 2019, 2020; Wang et al., 2021; Sainz and Rigau, 2021; McCann et al., 2018). In the case of entailment, Dagan et al. (2006) converted QA data to entailment manually and Demszky et al. (2018) did it automatically. Other semantic tasks such as Named Entity Recognition, Relation Extraction and Semantic Role Labelling have also been reformulated as entailment by automatically converting data into the entailment format (White et al., 2017; Poliak et al., 2018a; Levy et al., 2017; Sainz et al., 2021).

Multi-task learning reformulates multiple tasks to a single and common task via prompting large pre-trained language models, leveraging multiple data sources to improve each task of interest. Such approaches have shown improvements in supervised (Subramanian et al., 2018; Raffel et al., 2020; Aribandi et al., 2022) and zero-shot scenarios (Sanh et al., 2022; Wei et al., 2021a). While using the language modelling task as a pivot shows strong performance with very large language models, it is not clear that smaller models can benefit from this strategy in the same way. Wei et al. (2021a) and Mishra et al. (2022) obtained contradictory results. In a similar way, Question Answering has been proposed as a pivot task for multi-task learning but without promising results (McCann et al., 2018). In this work, we explore multi-source learning, where datasets from different or similar tasks are used to build a model for the target task.

Event Argument Extraction is a sub-task of Event Extraction. The goal is to identify arguments or fillers for a specific slot (a.k.a., role) in an event template. This task has been largely explored on the Message Understanding Conference (MUC, Grishman and Sundheim (1996)) and later on Automatic Content Evaluation (ACE). ACE focused mainly on sentence level evaluation due to the difficulty of the task at the time. Recently, new benchmarks such as RAMS (Ebner et al., 2020) and WikiEvents have emerged with the aim of addressing document level information extraction similar to MUC. However, most of the interest is still focused on the sentence level.

EAE has been recently addressed by end-to-end event extraction models (Wadden et al., 2019; Lin et al., 2020; Li et al., 2021a), instead of treating it as an independent task (Du and Cardie, 2020a), as we do, or as a subtask in a pipeline (Lyu et al., 2021). Lately, with the recent paradigm shift **to prompt design learning** (Min et al., 2021), several works reformulated the task as a Question Answering problem (Li et al., 2020; Feng et al., 2020; Du and Cardie, 2020b; Liu et al., 2020; Wei et al., 2021b; Lyu et al., 2021; Sulem et al., 2022) or as a Constrained Text Generation problem (Chen et al., 2020; Du et al., 2021; Li et al., 2021b) using predefined prompts, questions or templates. We instead reformulate the task as a textual entailment problem.

3 Approach

In order to cast EAE as an entailment task, we verbalize event argument instances using a set of intuitive and linguistically motivated templates to capture the event argument roles, and then per-

[2] https://github.com/osainz59/Ask2Transformers

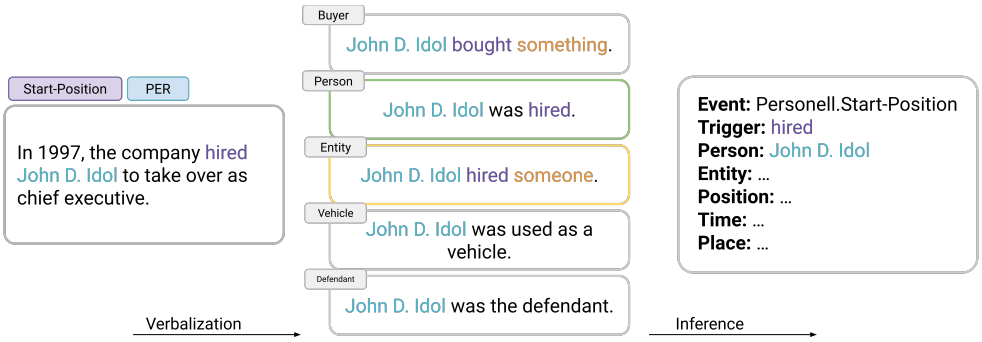

Figure 1: Entailment-based Event Argument Extraction. On the left, input information: the context, the event trigger (*hired*) and the argument candidate (*John D. Idol*), alongside the types of both. On the middle, some hypothesis verbalized using the templates: the green box is entailed, the yellow box matches the type constraint but it is not entailed, and the rest do not satisfy type constraints. On the right, the output with the inferred role (Person).

form inferences with entailment models. The entailment model can be additionally trained with EAE training data converted into the entailment format, similar to Sainz et al. (2021). Figure 1 shows the general workflow of the method. First, the possible roles are verbalized by means of predefined templates and the input, which comprises the context, trigger and argument candidate. Then, an entailment model is used to generate the entailment probability for each verbalization. To predict the role, the most probable hypothesis (verbalization) is chosen among the roles that satisfy the event-entity[3] constraints. A more detailed description of each component follows.

Label verbalization is attained using templates that combine the information of the instance and express a specific label. Different role verbalizations are shown in Figure 1. A verbalization is generated using templates that have been manually written based on the task guidelines of each dataset. The templates involve the candidate argument, and optionally the event trigger. In some cases, in order to produce a grammatical hypothesis, placeholders corresponding to the agent or theme are also introduced, which can be generic, e.g. *someone*, or dependent of the argument role, e.g. *defendant*. We defined several template types (see Table 1) to guide the creation of templates more systematically. In Section 5.1 we describe the process to create templates, and in Section 7 we analyse the differences between independent template developers and how this did not affect performance. The templates created for the ACE dataset are listed in Appendix C.

Entailment model. Given a premise and hypothesis, the model returns the probabilities of the hypothesis being entailed by, contradicted to or neutral to the premise. In principle, any model trained on the NLI task can be used.

Inference takes into account three key factors to output the role label for an argument candidate: the entailment probabilities of each verbalization, the type constraints of the specific role, and a threshold. Argument candidates which do not match the type constraints are discarded. From the rest, we return the role of the verbalized hypothesis with highest entailment probability, unless the probability is lower than the threshold, in which case we return the negative class.[4]

Training. Our entailment-based model can be applied without any training on the EAE task, in a zero-shot fashion, or, alternatively, the entailment model can be finetuned using training data from the EAE dataset. For this purpose, we convert the EAE training dataset into a NLI format, i.e we generate entailment, neutral and contradiction hypotheses heuristically from the data using the templates themselves. For each positive labeled example (a candidate that is an argument) we sample N_E entailment hypotheses using the templates that correspond to the correct label and N_N neutral hypotheses using templates from different roles. For each negative example (the candidate is not an argument of the event) we create N_C contradiction hypotheses using any template at random. N_E, N_N and N_C are considered hyperparameters of the training phase along with the hyperparameters of the neural network model such as learning-rate and

[3] In this context, entities also include values such as time or amounts.

[4] The class that represents that the argument candidate takes no part on the event.

Template type	Description	Example
{arg}	Templates with **implicit** information about the event. {arg} variable is the placeholder for the argument candidate.	The victim was {arg}.
{trg} → {arg}	Templates with **explicit** information about the event. The {trg} variable is the placeholder for the event trigger.	The {trg} occurred in {arg}.
{canonical(trg)} → {arg}	Templates with predefined canonical values for the {trg} variable.	{arg} was jailed.
{canonical(trg)}, placeholder → {arg}	Templates that makes use of agent or patient dummy placeholders in order to produce grammatical sentences.	The {arg} inspected something.

Table 1: The four main template categories used to create the role verbalizations.

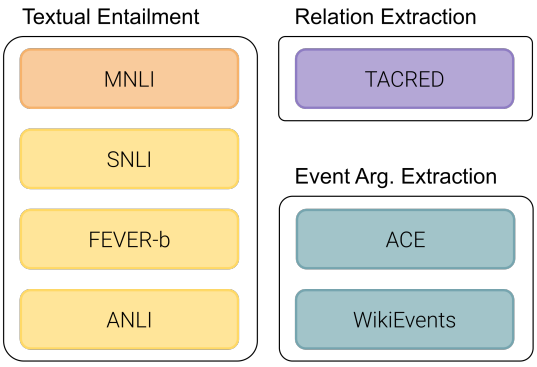

Figure 2: Datasets used by task category.

batch-size. In order to create challenging training examples for the negative class, we propose to use **constrained sampling**, based on the trigger-entity type constraints, where we create negative examples from candidates that satisfy the constraints. Preliminary experiments showed slight improvements with respect to regular sampling.

4 Entailment for Multi-source Learning

We hypothesize that two similar IE tasks can benefit from each other even if they do not share the same schema or domain. Although this hypothesis is very intuitive and it has been demonstrated on several works for tasks other than IE (see Multi-task learning on Section 2), actual IE models are limited by schema dependency, which makes it almost impossible to learn from datasets annotated with different IE schemas. One option is to perform a manual mapping between schemas, which is costly and often inaccurate (Kalfoglou and Schorlemmer, 2003). Our approach instead is domain and schema agnostic, and therefore allows to learning from multiple sources seamlessly. Given that the sources are recast into a single format in a common entailment formulation, it suffices to fine-tune the model in sequence across the sources.

To check our hypothesis we split tasks according to the following criteria: (1) IE sources like Relation Extraction that are different from EAE (e.g. TACRED), and (2) EAE sources using different schemas (e.g. WikiEvents and ACE). Figure 2 summarizes the tasks and datasets used in this work, including the four natural language understanding datasets.

5 Experimental Setup

In this section, we describe the methodology for template development, evaluation setting, the baselines used in our experiments, and the computation infrastructure specifications.

5.1 Methodology for verbalization

The templates used to generate the verbalizations were created based on the annotation guidelines of each dataset. During the creation, the template developers had access to the guidelines that describe each of the roles (which can include one or two examples) and a NLI model that the developer could use to verify whether the generated verbalizations of these examples were entailed by the model. The developer was allowed a maximum of 15 minutes per role, and spent 5 and 12 hours[5] to create the templates for ACE and WikiEvents respectively.

5.2 Evaluation

Datasets. We carried out our evaluation on two different EAE datasets: ACE (Walker et al., 2006) and WikiEvents (Li et al., 2021b). The ACE2005 dataset is a sentence-level Event Extraction dataset that contains entities, relations, event-triggers and arguments annotations on English, Chinese and

[5]Given that there is a total of 22 and 59 role types respectively, this is equivalent to an average of 13 and 12 minutes per role.

Train split	ACE # Pos	ACE Total	WikiEvents # Pos	WikiEvents Total
0%	-	-	-	-
1%	2.05	173	0.86	195
5%	11.36	843	4.09	966
10%	23.86	1736	8.26	1903
20%	45.00	3302	15.84	3578
100%	220.86	16502	79.68	18532

Table 2: Mean examples per role (pos) and total number of examples (positive and negative) across different training data splits and datasets.

Arabic texts. We worked only on the English EAE task. The WikiEvents dataset is instead more focused on document-level argument extraction task. Although the last is intended to be use as a document-level benchmark we focused on the sentence-level extraction[6] for two reasons: to maintain consistency with ACE dataset and because the nearest occurrence of the arguments are inside the sentence of the event trigger in almost all examples. For both ACE and WikiEvents, we split the training data into different amounts (0%, 1%, 5%, 10%, 20% and 100%) following Liu et al. (2020) to also evaluate our system on extreme data scarcity scenarios. Table 2 shows the amount of examples per split. The total amount refers to the addition of all positives and negatives trigger-candidate pairs.

Metrics. We have used the standard F1-Score, which is a common metric on IE tasks. Along with that, we propose the use of the Area Under the Curve (AUC) for better model comparison across all scenarios. The reported AUC scores are computed with all splits for the main results and just with 0%, 5% and 100% for the multi-source results, and therefore, they are not comparable.

5.3 Baselines and Models

Baselines. Our main point of comparison is our re-implementation of EM (Baldini Soares et al., 2019), as we can run it on the same few-shot splits as our system and allow for head-to-head comparison. EM is a state-of-the-art (Zhou and Chen, 2021) model that uses RoBERTa$_{\text{LARGE}}$ as a backbone. In addition we also report results of the state-of-the-art models that have been run on our same experimental setup, having access to gold event-trigger and entity annotations. On ACE, we report the results of BERTEE and RCEE_ER, both reported at (Liu et al., 2020), which correspond to a BERT (Devlin et al., 2019) based baseline and a QA based pivot approach that leverages SQuAD (Rajpurkar et al., 2016) data. Unfortunately the data splits used by (Liu et al., 2020) are not available[7] and thus, only the results for zero-shot (i.e. 0% training data) and full training (i.e. 100% training data) are directly comparable. Regarding WikiEvents Gen-Arg (Li et al., 2021b) uses gold triggers, but not gold entity information, so we decided to report Coref-F1[8] which refers to the F1-Score of predicting at least one of the gold entity coreferential chain as argument.

NLI models used in this work are based on the RoBERTa$_{\text{large}}$ (Liu et al., 2019) checkpoint, and are available via HuggingFace Transformer's model repository (Wolf et al., 2020). The main results use a model trained on all MNLI, SNLI, FEVER and ANLI, and in the analysis we also report the results of a model using just MNLI (see Appendix A for more information, including hyperparameters used).

5.4 Infrastructure

All the experiments were done in a **single** RTX 2080ti (11Gb) with a 250W power consumption. The average training times are:[9] 0.36h/epoch for ACE, 0.52h/epoch for WikiEvents and 2.86 h/epoch for TACRED. In total, 464.56 hours (154.86 if only a single run is done) of computation time are required to reproduce **all** the experiments, that in our setting corresponds to 21.36 kgCO_2eq carbon footprint[10] (roughly equivalent to the CO_2 emitted by 88.2 km driven by an average car).

6 Results

Main results. Table 3 reports our NLI system, including the median F1-Score and the standard deviation across 3 different runs of our implementations NLI and EM. On ACE our system is best on all comparable results and overall as shown by the AUC score. On the case of WikiEvents, our

[6] We consider as model prediction errors the arguments that are outside the sentence, to be consistent with other systems evaluation.

[7] Personal communication.

[8] We used this to alleviate the noise introduced by not using the gold entity annotations, and therefore, make the comparison more fair.

[9] The time required for training the model depends linearly with the sampling rates of entailment, neutral and contradiction examples.

[10] Estimation based on `mlco2.github.io/impact/`

ACE							
Model	0%	1%	5%	10%	20%	100%	AUC
BERTEE	-	*2.20	*10.5	*19.3	*28.6	64.7	*40.73
EM	-	4.58 ±1.55	37.5 ±2.98	50.9 ±0.96	58.7 ±1.9	72.1 ±0.65	60.87
RCEE_ER	37.0	*49.8	*59.9	*65.1	*67.6	70.1	*67.47
NLI	**40.6**	**45.4** ±0.16	**57.1** ±0.93	**64.6** ±1.12	**69.8** ±0.58	**74.6** ±0.88	**70.00**
WikiEvents							
Model	0%	1%	5%	10%	20%	100%	AUC
EM	-	16.9 ±0.63	41.5 ±1.47	49.9 ±0.28	54.9 ±1.30	61.3 ±1.04	55.26
*Gen-Arg	-	2.4 ±1.66	30.5 ±4.12	48.1 ±1.42	55.7 ±1.35	65.1	56.15
NLI	**35.9**	**42.6** ±1.36	**52.2** ±1.40	**59.5** ±0.58	**65.4** ±0.62	**69.9** ±0.70	**65.45**

Table 3: Main results on different training data splits for our NLI model, EM baseline and state-of-the-art systems. * for results not directly comparable with ours. Bold for best among comparable results.

	ACE				WikiEvents			
Source	0%	5%	100%	AUC	0%	5%	100%	AUC
NLI	40.6	57.1 ±0.93	74.6 ±0.88	65.0	35.9	52.2 ±1.40	69.9 ±0.70	60.2
NLI $_{+ \text{WikiEvents}}$	**62.7**	**69.3** ±0.35	**74.9** ±0.58	**71.8**	-	-	-	-
NLI $_{+ \text{ACE}}$	-	-	-	-	**57.3**	65.2 ±0.41	**71.5** ±1.07	**68.0**
NLI $_{+ \text{RE}}$	44.5	56.3 ±0.79	73.9 ±0.05	64.4	38.2	55.0 ±1.38	69.2 ±0.59	61.3
NLI $_{+ \text{RE} + \text{WikiEvents}}$	**62.7**	65.9 ±0.30	74.0 ±0.49	69.7	-	-	-	-
NLI $_{+ \text{RE} + \text{ACE}}$	-	-	-	-	56.7	**66.4** ±0.95	69.8 ±2.68	67.8

Table 4: Multi-source learning results of the NLI model. The AUC score reported on this table is only computed with 0%, 5% and 100% points, and therefore, is not comparable with Table 3. RE is shorthand for TACRED.

system is the best in all cases. In both datasets the EM baseline is outperformed by the NLI system.

Multi-source results. Table 4 describes our multi-source learning results, where we use NLI+ to indicate systems that use additional sources for training. We report the median F1-Score across 3 runs for 0%, 5% and 100% scenarios and the corresponding AUC score on ACE and WikiEvents. The rows show the impact of transferring knowledge from the training part of different tasks (for more detailed per role analysis see Appendix B). The results show that the signal between EAE datasets (i.e. WikiEvents and ACE) is strong, yielding significant improvements in all scenarios. For instance, on zero-shot evaluation, the systems obtain the impressive scores of 62.7 and 57.3, close to 20 points of improvement.

Sequentially fine-tuning our NLI model in TACRED and then in our target task shows small improvements on low-resource scenarios (0% split for ACE, 0% and 5% splits for WikiEvents). Training on the three sources sequentially does not seem to yield further improvements.

Figure 3 shows the performance of our NLI and multi-source enhanced NLI+ systems along with the EM baseline (data from Tables 3 and 4). The curves show that our NLI+ systems only need 10% and 5% of the data (on ACE and WikiEvents, respectively) to outperform the EM baseline that uses 100% of the training data.

7 Analysis

After performing the main experiments we did some additional analysis.

The importance of using several NLI datasets. A perfect NLI model should, in theory, solve any task that is framed correctly as entailment. Of course, there is not "perfect" NLI model. In fact, current state-of-the-art NLI models tend to learn artifacts and lexical patterns (Gururangan et al., 2018; Poliak et al., 2018b; Tsuchiya, 2018; Glockner et al., 2018; Geva et al., 2019; McCoy et al.,

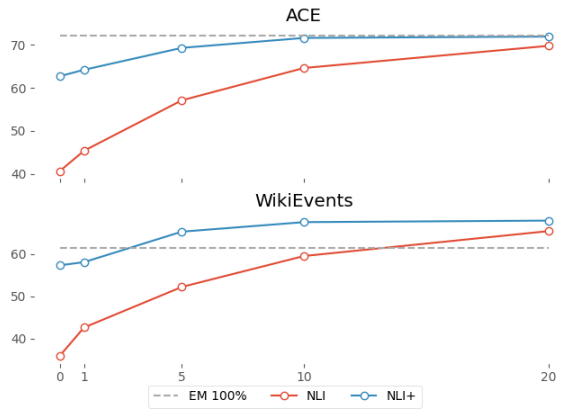

Figure 3: Comparison between the baseline EM model trained on 100% training, and our NLI and multi-source enhanced NLI+ models ($NLI_{+\ WikiEvents}$ and $NLI_{+\ ACE}$) with different training subsets.

Model $_{source}$	0%	5%	100%	AUC
		ACE		
NLI $_{MNLI\ only}$	31.4	46.0 ±0.55	62.8 ±2.83	53.6
NLI	**40.6**	**57.1** ±0.93	**74.6** ±0.88	**65.0**
		WikiEvents		
NLI $_{MNLI\ only}$	29.5	49.3 ±0.32	59.9 ±0.99	53.8
NLI	**35.9**	**52.2** ±1.40	**69.9** ±0.70	**60.2**
		TACRED		
NLI $_{MNLI\ only}$	55.6	64.1±0.20	71.0	67.2
NLI	**56.8**	**70.5**±0.62	**73.2**±0.65	**71.4**

Table 5: Ablation on NLI datasets used to-pretrain our NLI model on three datasets. NLI for our system using MNLI, FEVER, SNLI and ANLI (taken Table 3) and NLI $_{MNLI\ only}$ for our system when using MNLI only.

2019) instead of the task itself. Motivated by these issues, datasets like ANLI (Nie et al., 2020) were adversarially created to alleviate them. The lack of robustness of NLI models gets amplified when it comes to a cross-task evaluation. For instance, the model trained on MNLI achieves 90.2 accuracy on MNLI and 31.4, 29.5 and 55.6 F1-Score on ACE, WikiEvents and TACRED respectively (cf. Table 5). Adding FEVER, SNLI and ANLI to the training improves MNLI accuracy only 0.8 points to 91.0, but zero-shot scores on ACE, WikiEvents and TACRED improve +9.2, +6.4 and +1.2 respectively. In few-shot and full-training scenarios, the results also improve when using several NLI datasets. Our results suggest that new, more challenging NLI datasets, as well as NLI datasets automatically generated from other sources (as done in this work with WikiEvents and ACE) will yield more robust entailment models, and could further increase the performance of entailment-based EAE and IE.

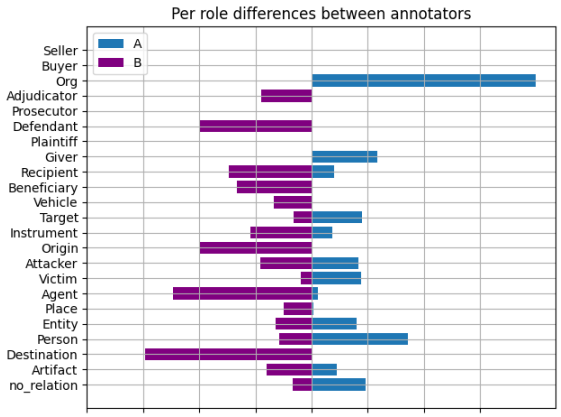

Figure 4: Recall differences between the main developer (A, right) and the linguist (B, left).

The impact of different template developers. In order to test the robustness of the templates, we enrolled a linguist with experience in NLP annotation but no prior contact with the project nor access to the original templates from the main developer. Under the same time and resource conditions, she was asked to write templates for the ACE dataset. The templates written by the main developer and the linguist vary in different ways: (1) the number of created templates per role and (2) the verbalization style, as the main developer tended to use finite and conjugated verbs while the linguist tended to use infinitives and lemmas. The templates of both are available in Appendix C.

To study the performance of the templates of each developer per role, Figure 4 shows the instances that a system correctly classified and the other system did not, and vice versa. The bars display the recall, as they are normalized by the frequencies of the roles. Missing bars on a row means that both performed the same on that role (e.g. Seller). When only a blue bar is shown (e.g. Org) it means that the main developer recovered arguments which the linguist did not, **and** there were no examples where the linguist recovered arguments that the developer did not. The same applies to situations where there is only purple bars. Roles with mixed results include examples where one or the other succeeded. As we can see, the approaches seem to be complementary, with the linguist having a higher recall with the roles that are more associated with classical semantic roles. Table 6 shows that in general, the templates of the linguist perform similarly to those of the main developer, except for 100% of the data, where the templates of the main developer were slightly better.

Developer	0%	1%	5%	10%	20%	100%
(A) Main	40.3	46.2	56.3	63.8	69.6	76.4
(B) Linguist	40.4	44.9	57.3	64.2	70.1	73.3
Δ F1	-0.1	+1.3	-1.0	-0.4	-0.5	+3.1

Table 6: Results for templates from two developers. Median F1 on the development set are reported.

Verbalizations vs. annotations Finally, we carried out an experiment to compare the time and effort requirements of annotation vs. writing the templates. To that end, the linguist re-annotated a small portion of ACE with the same information she had as she was creating the templates. That is, given the argument candidates for each event trigger in the document, she needs to decide whether the candidate was an argument and the type of the argument. She has access to the guidelines (similar to creating the templates), though she did not study them beforehand. Note also that she did the annotations **after** writing the templates, so she was already familiar with the slots. Under these conditions, she annotated 46 pairs (event trigger, potential argument candidate) in 30 minutes. Taking into account that ACE has 16.5000 such pairs, it would take approximately 180 hours to annotate ACE training part. Note that in practice, ACE requires much more time than our estimate to achieve the desired level of quality: the ACE annotation procedure involved double annotation and a second pass with a senior annotator (Doddington et al., 2004). For an analysis of the annotation procedure the interested reader is referred to Min and Grishman (2012).

Based on our estimation, 9 hours would allow an annotator to annotate 5% of the dataset which yields a 37.5 F1 (Figure 5), while 5 hours of template building yields 40.6 F1-Score in the zero-shot setting. With 18 hours 10% would be annotated and the F1-Score will be 50.9, while 5 hours of template building and 9 hours of annotations would yield 57. Figure 5 plots the performance according to manual hours on ACE, showing the huge gains provided by the initial 5 hours writing templates, plus the reuse of WikiEvents annotations. According to our experience, more hours on template building does not necessarily lead to improvements (contrary to annotation), so a **sweet spot for time investment** seems to be to firstly create templates, and then spend the remaining budget on annotating examples.

On another note, the linguist mentioned that writing templates is more natural and rewarding

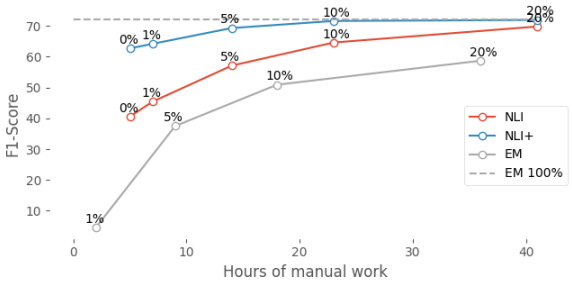

Figure 5: Performance on ACE according to our estimations of manual work in hours. We also indicate the percentage of training data used.

than annotating examples, which is more repetitive, stressful and tiresome. When writing templates, she was thinking in an abstract manner, trying to find generalizations, while she was paying attention to concrete cases when doing annotation.

8 Conclusions

This paper shows the entailment-based approach for event argument extraction is extremely effective in zero-shot, few-shot and full train scenarios both on ACE and WikiEvents, outperforming previous methods. First of all, recasting EAE as an entailment task allows it to reuse annotations from different event schemas, achieving large gains when transferring annotations between ACE and WikiEvents, and also some gains in the zero-shot performance when transferring annotations from a relation extraction model such as TACRED. Secondly, we show that using additional training entailment datasets improves results significantly over just using MNLI, not only on EAE but also on TACRED. Thirdly, we show that the relatively short time spent writing manual templates is much more effective than the time spent on doing annotations, with a sweet spot where the annotation effort is split between the two, with large savings in manual labour. Lastly, we show that an independent linguist is able to write templates with comparable performance without any special training. We think that our results and analysis support the potential of entailment models for other NLP tasks.

Our work paves the way for a new paradigm for IE, where the expert defines the schema using natural language and directly runs those specifications, annotating a handful of examples in the process, and allowing for quick trial-and-error iterations. Sainz et al. (2022) propose a user interface alongside this paradigm. More generally, inference capability could be extended, acquired and applied

from other tasks, in a research avenue where entailment and task performance improve in tandem.

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).

References

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. 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.

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, Online. Association for Computational Linguistics.

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.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

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.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation. In *Language Resources and Evaluation Conference (LREC)*, volume 2, pages 837–840. Lisbon.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding.

Xinya Du and Claire Cardie. 2020b. 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.

Xinya Du, Alexander Rush, and Claire Cardie. 2021. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

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

Yannis Kalfoglou and Marco Schorlemmer. 2003. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1):1–31.

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*.

Teven Le Scao and Alexander 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*, pages 2627–2636, Online. 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.

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.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021a. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks.

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*.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Bonan Min and Ralph Grishman. 2012. Compensating for Annotation Errors in Training a Relation Extractor. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 194–203.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018a. Towards a unified natural language inference framework to evaluate sentence representations. *CoRR*, abs/1804.08207.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018b. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Oscar Sainz, Haoling Qiu, Oier Lopez de Lacalle, Agirre Eneko, and Bonan Min. 2022. ZS4IE: A toolkit for zero-shot information extraction with simple verbalizations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, Online and Seattle, Washington. Association for Computational Linguistics.

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.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021a. 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. 2021b. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021c. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*.

Elior Sulem, Jamaal Hay, and Dan Roth. 2022. Yes, No or IDK: The challenge of unanswerable Yes/No questions. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Online and Seattle, Washington. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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 (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021a. Finetuned language models are zero-shot learners.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021b. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4672–4682, Online. Association for Computational Linguistics.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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.

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*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction.

Hyperparameter	EM	NLI	$\text{NLI}_{\text{MNLI only}}$
N_E / N_N / N_C	-	2 / 5 / 5	2 / 5 / 5
Batch size		32	
Learning rate	1×10^{-5}	4×10^{-6}	1×10^{-5}
Seeds		{0, 24, 42}	
Epochs		25 (*50)	
Weight decay		0.01	

Table 7: Hyperparameters of the trained systems. * indicates the difference between full-train and few-shot scenarios.

A Hyperparameters

On this section we describe the hyperparameters we have used on our experiments. All the hyperparameters optimized on this work were optimized for the 100% split with the batch-size fixed to 32, and used on the rest. The Table 7 describes the hyperparameters used on EM, NLI and $\text{NLI}_{\text{MNLI only}}$ variants, for the NLI+ the same hyperparameters as NLI were used. We have found that the same exact hyperparameters were the best on ACE, WikiEvents and TACRED datasets. For the future, we plan to test new hyperparameter sets that uses bigger batch-sizes, as recent works (Aribandi et al., 2022) suggest to be optimal for multi-task and -source learning experiments.

The pre-trained NLI models used on this work can be downloaded from the HuggingFace Models repository: $\text{NLI}_{\text{MNLI only}}$ (`roberta-large-mnli`) and NLI (`ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli`).

The fine-tuned models derived from this work will be uploaded to HuggingFace Models repository. Check the GitHub repository for updated information.

B Multi-task in-depth analysis

The Figure 6 shows the per role absolute improvement obtained by training on different tasks over the 0% NLI system. Overall, we can see that training on ACE or WikiEvents improves almost all the roles and training on TACRED improves some and some others do not. A result that was unexpected is that there are few roles on WikiEvents that after training on WikiEvents become worse in contrary to training on ACE. This could be explained by the differences among the frequency distributions that the train, development and test sets of WikiEvents has. Moreover, there are some roles on WikiEvents that decreases in all training scenarios, this suggests us that sequential fine-tuning might be not the best option for this type of multi-source learning and therefore further ways should be explored.

C ACE templates from both developers

The next table contains the templates written by both developers for the ACE arguments. We follow the notation introduced in Section 5.1. In addition, we also consider information from the event, such as the type on different granularity levels, including `{trg_type}` for the trigger type (e.g. *Movement* from *Movement.Transport*) and `{trg_subtype}` for the subtype of the trigger, e.g. *Transport* from *Movement.Transport*).

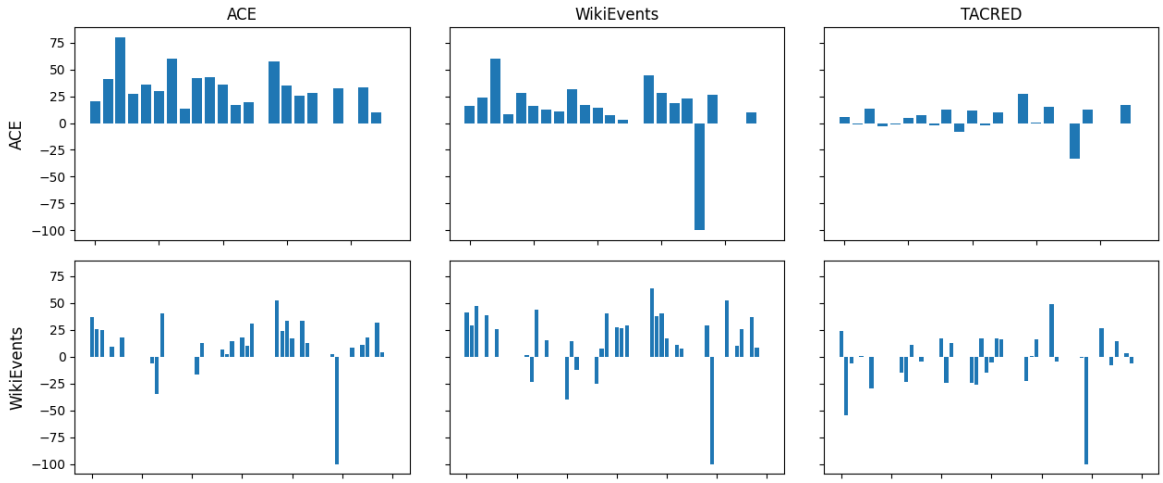

Figure 6: Absolute improvements over the NLI baseline using different tasks and sources. Rows indicates the testing data and columns the training data. Each bar indicates the F1-Score difference between the trained NLI system vs 0% NLI for a specific role.

Role	Main developer	Linguist
Adjudicator	{arg} tried the defendant. {arg} convicted the defendant. {arg} released the defendant. {arg} sentenced the defendant. {arg} acquitted the defendant.	{arg} convict someone. {arg} sentence someone. {arg} judge someone. {arg} fine someone. {arg} indict someone.
Agent	{arg} {trg} a person or organization.	{arg} do something. {arg} select something. {arg} carry out something. {arg} create something. {arg} give something.
Artifact	Someone {trg} the {arg}. Someone moved {arg}. Someone bought {arg}. Someone sold {arg}.	{arg} be an object. {arg} be a weapon.
Attacker	{arg} {trg} a person or organization.	{arg} assail someone. {arg} aggress someone. {arg} assault someone.
Beneficiary	The buyer bought to {arg} something.	{arg} get something . {arg} be beneficiary. {arg} benefit from something. {arg} obtain something.
Buyer	{arg} bought something.	{arg} buy something. {arg} possess something. {arg} own something.
Defendant	{arg} was the defendant.	{arg} be accused of something. {arg} be accused of a crime. {arg} be judged.
Destination	Someone {trg_subtype} to {arg}.	{trg_type} go to {arg}. {trg_type} finish in {arg}. {trg_type} move to {arg}. {arg} be a place. {arg} be a location.
Entity	{arg} attended the demonstration. {arg} met someone. {arg} fired someone. {arg} voted in the elections. {arg} released the defendant. {arg} was ordered to pay.	{arg} select something. {arg} carry out something. {arg} do something. {arg} create something. {arg} give something.
Giver	{arg} gave something to someone.	{arg} give something.
Instrument	Someone {trg_subtype} with {arg}.	{arg} be artifact. {arg} be object. {arg} be device. {arg} cause harm.
Org	{arg} organization declared bankruptcy.	{arg} be in bankruptcy.

(continued on the next page)

Role	Main developer	Linguist
	{arg} organization was dissolved. {arg} organization was merged. {arg} organization was launched.	{arg} be ended. {arg} be merged. {arg} be created. {arg} be company. {arg} be organization.
Origin	Someone {trg_subtype} from {arg}.	{arg} change location. {arg} be location. {trg_type} start in {arg}. {trg_type} move from {arg} .
Person	{arg} was {trg}.	{arg} be person. {arg} be living entity. {arg} be born. {arg} get married. {arg} be married. {arg} divorce. {arg}'s marriage ended. {arg} be hired. {arg} start a job. {arg} be fired. {arg} end a job. {arg} be nominated. {arg} be elected. {arg} be arrested. {arg} be jailed. {arg} be imprisoned. {arg} be released. {arg} be paroled. {arg} be executed. {arg} be extradited.
Place	{trg} occurred in {arg}.	{arg} be a place. {arg} be a location. {arg} be a placement.
Plaintiff	{arg} filed suit against someone.	{arg} bring a lawsuit against someone. {arg} bring a lawsuit against something. {arg} sue someone. {arg} sue something.
Prosecutor	{arg} indicted the defendant. {arg} charged the defendant.	{arg} prosecute. {arg} take somebody to court for a crime.
Recipient	{arg} received money from someone.	{arg} receive something. {arg} get something. {arg} get money.
Seller	{arg} sold something.	{arg} sell something.
Target	{arg} was {trg_subtype}.	{arg} be attacked. {trg_type}'s target be {arg}.
Vehicle	{arg} was used as a vehicle.	{arg} be a transport.

(continued on the next page)

Role	Main developer	Linguist
		{arg} be a vehicle. {arg} serve to move. {arg} serve to change location. {arg} serves as a means of transportation.
Victim	{arg} was {trg}.	{arg} be victim. {arg} be injured. {arg} be killed. {arg} be harmed. {arg} have a dead. {arg} have a tragedy.

The templates written by both developers for ACE.