ZS4IE: A toolkit for Zero-Shot Information Extraction with simple Verbalizations

Oscar Sainz[1,*], **Haoling Qiu**[2,*],
Oier Lopez de Lacalle[1], **Eneko Agirre**[1], and **Bonan Min**[2]

[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, haoling.qiu@raytheon.com

Abstract

The current workflow for Information Extraction (IE) analysts involves the definition of the entities/relations of interest and a training corpus with annotated examples. In this demonstration we introduce a new workflow where the analyst directly verbalizes the entities/relations, which are then used by a Textual Entailment model to perform zero-shot IE. We present the design and implementation of a toolkit with a user interface, as well as experiments on four IE tasks that show that the system achieves very good performance at zero-shot learning using only 5–15 minutes per type of a user's effort. Our demonstration system is open-sourced at https://github.com/BBN-E/ZS4IE. A demonstration video is available at https://vimeo.com/676138340.

1 Introduction

Information Extraction (IE) systems are very costly to build. The current **define-then-annotate-and-train** workflow uses supervised machine learning, where the analyst first defines the schema with the entities and relations of interest and then builds a training corpus with annotated examples. Unfortunately, each new domain and schema requires starting from scratch, as there is very little transfer between domains.

We present an alternative **verbalize-while-defining workflow** where the analyst defines the schema interactively in a user interface using natural language verbalizations of the target entity and relation types. Figure 1 shows sample verbalization templates for a simple schema involving an employee relation and a passing away event, as well as a sample output annotated with the schema. The annotation of the EMPLOYEEOF relation requires performing Named Entity Recognition (NER) (Tjong Kim Sang and De Meulder, 2003) and Relation Extraction (RE) (Zhang et al., 2017), while annotating the LIFE.DIE event involves NER, Event Extraction (EE), and Event Argument Extraction (EAE) (Walker et al., 2006). Our toolkit is able to perform those four IE tasks using a single user interface, allowing the analyst to easily model and test the schema without the need to annotate examples.

Our toolkit leans on recent work which has successfully recast several IE tasks as Textual Entailment (TE) tasks (White et al., 2017; Poliak et al., 2018; Levy et al., 2017; Sainz et al., 2021). For instance, Sainz et al. (2021) model relation types between entity pairs using type-specific verbalization templates that describe the relation, generates a verbalization (hypothesis) automatically using those templates and then uses a pre-trained TE model to predict if the premise (the sentence where the pair appears) entails the hypothesis, therefore leading to a prediction of the relation or "no relation".

In this paper we thus present ZS4IE, a toolkit for zero-shot IE. We show that the four mainstream IE tasks mentioned above can be reformulated as TE problems, and that it is possible to achieve strong zero-shot performances leveraging pre-trained TE models and a small amount of templates curated by the user. Our toolkit allows a novice user to curate templates for each new types of entities, relations, events, and event argument roles, and validate their effectiveness online over any example. We also present strong results on widely used datasets with only 5-15 minutes per type of a user's effort.

2 Related Work

Textual Entailment has been shown to be a reasonable proxy for classification tasks like topic or sentiment analysis (Yin et al., 2019; Sainz and Rigau, 2021; Zhong et al., 2021). To reformulate a classification problem as TE, it often starts with defining templates to describe each class label, leading to a natural language text (a "verbalization" of a hypothesis) for each possible label. Inference is

[*]Denotes equal contribution.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations, pages 27 - 38
July 10-15, 2022

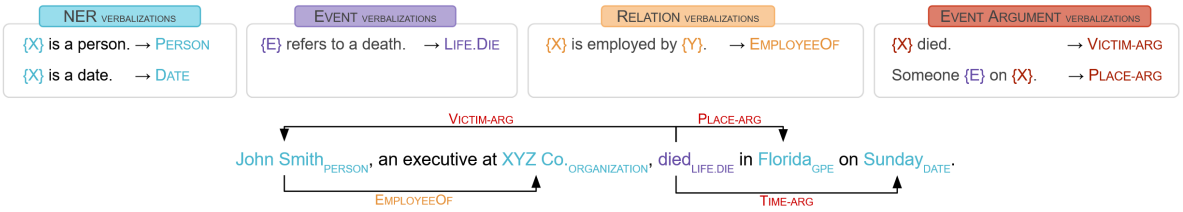


Figure 1: Verbalization templates for a sample schema involving four tasks (from left to right, NER, EE, RE, EAE), with example output (bottom). The schema contains a EMPLOYEEOF relation between PERSON and ORGANIZATION entities and a LIFE.DIE event with three argument types (VICTIM, PLACE and TIME) and PERSON, DATE and GPE entities as fillers. Due to space constraints, at most two verbalizations per task shown.

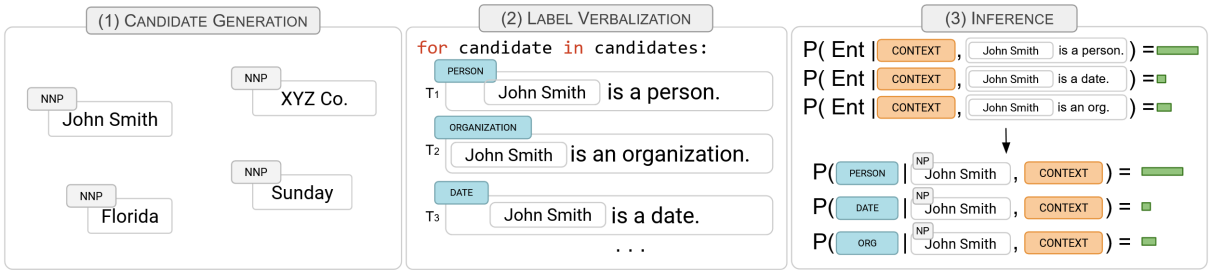


Figure 2: Three steps for entailment-based NER. The steps for the other IE tasks is analogous.

performed by selecting the most probable candidate hypothesis entailing the premise. TE is usually implemented with pre-trained language model fine-tuned on TE datasets, such as MNLI (Williams et al., 2018), SNLI (Bowman et al., 2015), FEVER (Thorne et al., 2018), ANLI (Nie et al., 2020) or XNLI (Conneau et al., 2018). The results on classification have been particularly strong for zero-shot and few-shot learning, with Wang et al. (2021b) hypothesizing that entailment is a true language understanding task, where a model that performs entailment well is likely to succeed on similarly-framed tasks.

Sainz et al. (2021) reformulated relation extraction as a TE task surpassing the state-of-the-art in zero- and few-short learning. A similar approach was previously explored by Obamuyide and Vlachos (2018), using TE models that are not based on pre-trained language models. Similar to TE, (Clark et al., 2019) performs yes/no Question Answering, in which a model is asked about the veracity of some fact given a passage. Lyu et al. (2021) recast the zero-shot event extraction as a TE task, using TE model to check whether a piece of text is about a type of event. Lastly, Sainz et al. (2022) showed that TE allows to leverage the knowledge from other tasks and schemas.

3 IE via Textual Entailment

We first describe how to recast each of the IE tasks (NER, RE, EE, EAE) as TE independently, and leave the workflow between the tasks for the next section. At a high level, the zero-shot TE reformulation consists of three steps: candidate generation, label verbalization and TE inference (Figure 2 illustrates the steps for NER). The first step, candidate generation, identifies text spans (e.g., proper nouns for NER) or span pairs (a pair of entity mentions for relation extraction) in the input sentence as the focus of the prediction. Taking a text span (or span pair) as input, the label verbalization step applies a verbalization template to generate a *hypothesis*, which is a natural language sentence describing the span (or span pair) being an instance of a type of entity, relation, event, or event argument. The verbalization generates hypothesis for each of the target types. Finally, the TE inference step takes the original sentence (the *premise*) and each *hypothesis* as input, and uses a pre-trained TE model to predict if the *premise* entails, contradicts, or is neutral to the *hypothesis*. The type with the verbalization having the highest entailment probability is selected. We next describe each step in detail.

3.1 Candidate Generation

We describe the candidate generation for each of the task below.

Named Entity Recognition (NER): Candidates are extracted using specific patterns of PoS tags as returned by Stanza (Qi et al., 2020). For instance, for the simple example in Figure 1 it suffices to

select proper nouns (shown in Figure 2), which are easily extended with other PoS patterns if needed. The toolkit also allows the usage of a constituency parser (Kitaev and Klein, 2018).

Relation Extraction (RE): Each relation requires a pair of entities that satisfy specific type constraints, e.g. the EMPLOYEEOF relation requires a PERSON and an ORGANIZATION. A NER module is used to extract all candidate entities that follow the required entity types according to the target schema. The toolkit uses the TE based NER module, although it also allows usage of a supervised NER system (Qi et al., 2020).

Event (Trigger) Extraction (EE): The main goal of this task is to detect whether the input sentence contains a mention of any of the target event types in the schema, e.g. LIFE.DIE. This task can be formulated as a multi-label text classification task, and in this case the full sentence is the candidate. Alternatively, the textual span that most likely expresses the event (the so-called trigger) can be extracted. In this case, the candidates are generated using specific PoS tags, e.g. verbs like *died* (cf. Figure 1). Our toolkit allows both options.

Event Argument Extraction: Given a sentence containing an event type (as detected by EE above), the goal is to extract entity mentions that are fillers of the target arguments in the schema. For example, the schema in Figure 1 involves three target arguments. Each of the arguments requires specific entity types, e.g. PERSON for the VICTIM argument. The candidates of the required types are extracted using the same NER module as for RE.

3.2 Label Verbalization

For each of the IE tasks, the label verbalization process takes a sentence, a set of candidates and the set of target types (e.g. NER types), and generates a natural language text (the hypothesis) describing the existence of the type in the sentence (the premise) using verbalization templates. Each candidate is a span (or pair of spans) that can belong to a specific type (e.g. being a PERSON in NER). Therefore, the textual verbalization is generated to express each potential type for the span or the pair of spans. For the NER and event extraction tasks, each verbalization expresses one potential entity (or event type) for the target candidate. For the relation and event argument extraction tasks, the verbalization template combines the information from the text spans of the candidate pair and produces a text that expresses a relation (or event argument role). The analyst just needs to write the verbalization templates for each target type, and they are applied to the candidates to generate the hypothesis, as shown in the second step in Figure 2 for NER.

Figure 1 shows sample TE verbalization templates for entity, relation, event, and event argument types corresponding to the 4 IE tasks, as well as sample example as output. The templates for **NER** and **event extraction** (leftmost part of the figure) are applied over a single candidate as extracted in the previous step (the candidate entity or event trigger, respectively). Note that for event extraction it is also possible to produce hypothesis using templates with no slots, e.g. "A person died" for LIFE.DIE. In the case of **relation extraction**, the verbalization templates contain two slots for the two entity spans potentially holding the relation. Finally, templates for **event argument extraction** can be more varied. The figure shows two examples: a template using a single slot for the candidate filler, and a template which, in addition to the filler slot, uses the trigger ("died" in this case, for PLACE).

3.3 Inference

Given a premise (the original sentence) and a hypothesis (an verbalization generated by label verbalization templates), we use a pre-trained TE model to decide whether the hypothesis is entailed by, contradicted with, or is neutral to the premise. In principle, any model trained on an entailment dataset can be used. The inference is mainly determined by three key factors: the TE probabilities for the verbalizations of all templates for all labels, the type-specific input span constraints, and a threshold that decides if the probability is high enough to consider the candidate a positive instance. The type-specific input span constraints are enforced to make sure we don't have candidates that violates the constraints. We return the class label of the hypothesis with highest entailment probability. If none of the hypothesis is higher than the threshold, we return the negative class, that is the class that represents that there is not a valid entity, relation, event, or event argument role type for the input candidate. The threshold for minimal entailment probability is set by default to 0.5.

4 ZS4IE toolkit

ZS4IE comprises a pipeline and a user interface.

4.1 The ZS4IE Pipeline

As described in Section 3.1 and illustrated in Figure 3, there are inter-task dependencies between the four IE tasks (e.g., relation extraction requires that entity mentions have already been tagged in the input sentence). Some task also require external NLP tools for generating candidates. To address these issues and to allow maximal flexibility for the users, we support the following two workflows.

The End-to-End (E2E) Mode: This mode will run the ZS4IE modules in a pipeline: we allow the users to start from raw text, and perform customization (e.g., develop templates for new types of interest) for all four IE tasks. The user has to follow the inter-task dependencies as illustrated in Figure 3: the user must finish NER customization before moving on to relation extraction or the event argument extraction task, because the later two tasks needs NER to generate their input candidates. Similarly, the user must finish customization for the event trigger classification task, before working on the event argument extraction task.

The end-to-end pipeline also runs a customizable pre-processing step including a POS tagger and a constituency parser, before any of the later modules.

The Task Mode: In this mode, the user can choose to work on each of the four IE tasks independently. In order to address the inter-dependencies, the user can choose to run an independent NER module instead, as part of the pre-processing step. The user interface allows the user to tag any spans for entity or event trigger types, before running customization for the more complex tasks such as relation extraction or event argument extraction. This option allows to explore additional entity and event trigger types before actually implementing them

4.2 User Interface (UI)

Figure 4 shows the User Interface. It allows the user to add new types of entities, relations, events and event argument roles, and then develop templates (along with input type constraints for each type). Figure 5 shows the NER extraction results on an user-input sentence. It also displays the likelihood scores produced by the TE model of those

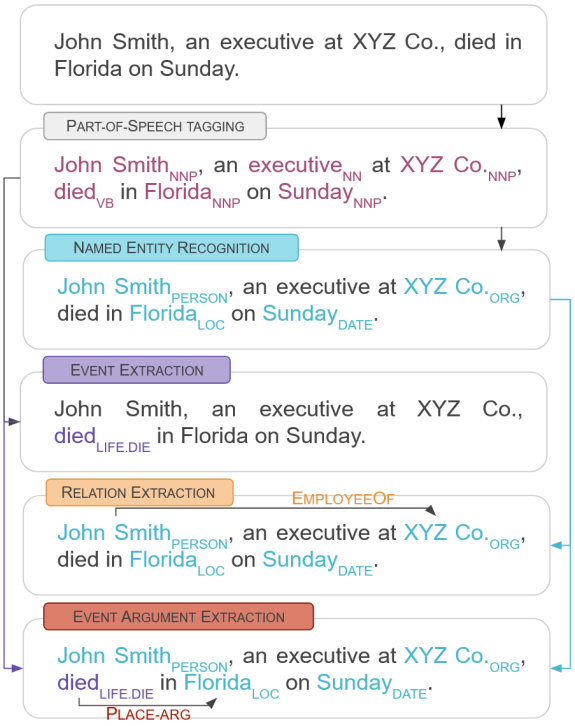


Figure 3: An illustration of the dependencies between the four IE tasks.

templates that are above the threshold, to allow the user to validate templates.

To show why it extracts each entity, it displays a ranked list of likely entity types, the template that led to that type, along with the entailment probability produced by the pre-trained TE model. The user can click on "+" and "-" sign next to each extraction to label its correctness. Our system will track the total number of extractions and and accuracy for each task, each type and each template, to allow the user to quickly validate the effectiveness of the templates and to spot any low-precision template.

Supplying Input Text: The user can supply a text snippet, one at a time, to test writing templates. As described in Section 4.1, when using the task mode, the user can label spans in the input text for the more complex relation extraction and event argument extraction tasks, so that the text already has the right entity or event trigger spans and types to begin with.

Develop Templates for New Types: The user can add new types of entities, relations, events, and event argument role. For each type, the user can create templates along with the input span type constraints, and then run inference interactively on the input text, to see whether these templates

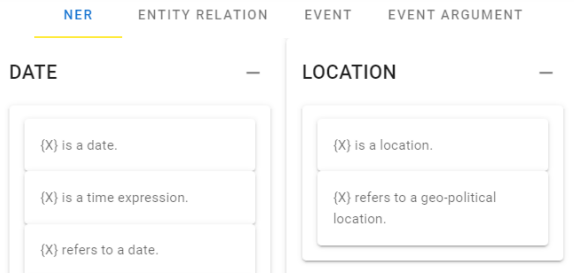


Figure 4: The UI for curating templates for types of interests for NER, relation extraction, event extraction and event argument extraction tasks. The NER tab is partially shown with two types.

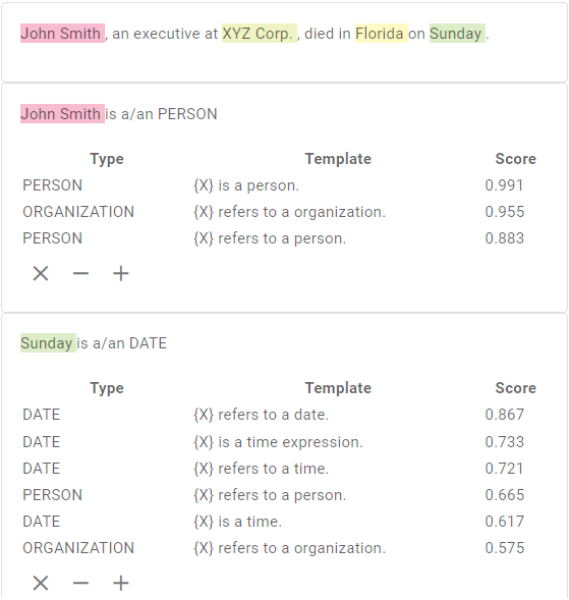


Figure 5: The UI for displaying NER extraction results on an user-input sentence. We show the extractions and the likelihood scores of the templates above the threshold (e.g. $\mathcal{T} = 0.5$).

can be used for extract the instances. The user can label the correctness of the extracted instances, resulting a small development dataset (the *dev* set) to help measuring the precision and relative recall for each template, and to tune the threshold for the TE inference.

Display Metrics: The UI displays the accuracy and yield for each template and each type in real-time, to allow the user to monitor the progress and make adjustments on the fly.

More screenshots and details of our UI are describe in Appendix A.

5 Experiments

We evaluated our system using publicly available datasets. We use CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) for NER evaluation, TACRED (Zhang et al., 2017) for RE, and ACE for EE and EAE (Walker et al., 2006). We evaluate each task independently (not as a pipeline) to make as comparable as possible to existing zero-shot systems. In order to apply our toolkit we made some adaptations as follows: We consider only proper nouns as candidates for NER, and we ignore the MISC label because it is not properly defined in the task [1]. We evaluate EE as event classification, where the task is to output the events mentioned in the sentence without extracting the trigger words, as we found that deciding which is the trigger word is in many cases an arbitrary decision [2]. In the case of RE we used the templates from (Sainz et al., 2021), which are publicly available. We will release the templates used on the experiments as additional material along with the paper. The analysts spent between 5-15 minutes per type, depending on the task, with NER and EE being the fastest.

Table 1 shows the zero-shot results for NER, RE, EE, and EAE tasks. We report the results of three entailment models: RoBERTa (Liu et al., 2019) trained on MNLI, RoBERTa* trained on MNLI, SNLI, FEVER and ANLI; and DeBERTa (He et al., 2021) trained on MNLI. The main results (top three rows) use the default threshold ($\mathcal{T} = 0.5$), we selected the $\mathcal{T}$ blindly, without checking any development result.

The results show strong zero-shot performance. Note that there is no best entailment model, suggesting that there still exists margin for improvement. However, we see that RoBERTa* performs relatively well in all scenarios except EE (see Section 6 for further discussion).

The table also shows in the middle three rows the results where we optimize the threshold on development. The results improve in most of the cases, and allow comparison to other zero-shot systems which sometimes optimize a threshold in development data.

Furthermore, we compare our system with zero-shot task specific approaches from other authors when available. For RE, Wang et al. (2021a) propose a text-to-triple translation method that given a text and a set of entities returns the existing relations. For EE, Lyu et al. (2021) propose, similar

[1] More specifically, we re-labeled the MISC instances to O label.

[2] Note that EAE can be addressed without an explicit mention of the trigger since we used templates that do not require the trigger

Model	NER			RE			EE			EAE			AVG
	Pre	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	F1
RoBERTa	53.3	54.5	53.9	32.8	75.5	45.7	**23.8**	**63.0**	**34.5**	20.5	60.9	30.7	46.7
RoBERTa*	**73.5**	**76.3**	**74.9**	36.8	76.7	49.8	23.5	60.8	33.9	**30.1**	**63.2**	**40.8**	**49.0**
DeBERTa	58.0	50.2	53.8	**40.3**	**77.7**	**53.0**	12.9	60.3	21.2	20.0	31.9	24.6	45.1
RoBERTa (+ $\mathcal{T}$ opt)	49.3	61.8	↑ 54.9	56.1	55.8	↑ 55.9	32.0	52.9	↑ 39.9	25.8	40.1	↑ 31.4	↑ 50.9
RoBERTa* (+ $\mathcal{T}$ opt)	71.9	77.8	↓ 74.8	54.2	59.5	↑ 56.8	25.1	58.6	↑ 35.1	31.1	58.3	↓ 40.6	↑ 51.9
DeBERTa (+ $\mathcal{T}$ opt)	56.3	63.1	↑ 59.5	66.3	59.7	↑ 62.8	13.0	55.8	↓ 21.1	28.9	17.5	↓ 21.8	↑ 51.3
Other authors	-	-	-	-	-	49.2	36.2†	69.1†	47.5†	38.2	35.8	37.0	-

Table 1: Results for NER, RE, EE and EAE experiments results. Three top rows for zero-shot systems with default parameters. Middle rows for threshold optimized on development. The best scores among our results obtained with default thresholds are marked in **bold**. The † indicates non-comparable results due to additional SRL preprocessing.

to us, the use of an entailment model, but in their case the input sentence is split in clauses according to the output of a Semantic Role Labelling system. In order to compare their results with ours, we only use the event types, not the trigger information[3]. The results from our system can be seen as an ablation where we do not make use of any SRL preprocessing. For EAE, Liu et al. (2020) perform zero-shot EAE by recasting the task as QA. Some of these approaches also optimize a threshold on development data, although it is not always clear. We show that our toolkit with default threshold obtains excellent results despite being an all-in-one method.

6 Discussion

Towards post-editing on IE. Our internal evaluation suggest that verbalizing-while-defining workflow can have similar impact as post-editing machine translated text, where human translators obtain quality translations with less effort (Toral et al., 2018). The idea of this new framework will bring down the effort required to create larger and higher quality datasets. Current IE system are subject to a predefined schema and are useless to classify new types of entities, relations and events. The use interface of ZS4IE brings to the annotators the opportunity of defining the schema interactively and manually annotating the dataset with the help of the entailment model. In the future we would like to use the manual annotations to fine-tune the TE model, which would further improve the performance, as shown by the excellent few-shot results of Sainz et al. (2021).

Implicit events extraction. During the development of the EE verbalizations we found out that the entailment model is prone to predict implicit events that are implied by other events. For example, an event type of JUSTICE:JAIL implies an event of JUSTICE:CONVICT where as the same time it implies event type of JUSTICE:TRIAL-HEARING. As the entailment models are not specifically trained for a particular IE task (e.g. EE) they are not limited to the extraction of **explicit** mentions of types (e.g. event types) annotated in the dataset. We think that this phenomenon might have penalized the RoBERTa* model on the EE task, as ACE dataset only contains annotations of explicit events. On the contrary, rather than a limitation of our approach, we believe that this is a positive feature that can be exploited by the users.

7 Conclusions

The ZS4IE toolkit allows a novice user to model complex IE schemas, curating simple yet effective templates for a target schema with new types of entities, relations, events, and event arguments. Empirical validation showed that reformulating the IE tasks as an entailment problem is easy and effective, as spending only 5-15 minutes per type allows to achieve very strong zero-shot performance. ZS4IE brings to the users the opportunity of defining the desired schema on the fly. In addition it allows to annotate examples, similar to post editing MT output. Rather than being a finalized toolkit, we envision several exciting directions, such as including further NLP tasks, allowing the user to select custom pre-processing steps for candidate generation and allowing the user to interactively improve the system annotating examples that are used to fine-tune the TE model.

More generally, we would like to extend the inference capability of our models, perhaps acquired from other tasks or schemas (Sainz et al., 2022),

[3]Output kindly provided by the authors.

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

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.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. 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.

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.

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.

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.

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.

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

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Agirre Eneko. 2022. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In *Findings of the Association for Computational Linguistics: NAACL-HLT 2022*, Online and Seattle, Washington. 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 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.

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.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Antonio Toral, Martijn Wieling, and Andy Way. 2018. Post-editing effort of a novel with statistical and neural machine translation. *Frontiers in Digital Humanities*, 5:9.

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

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021a. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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.

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.

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.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A User Interface

We present more details on our user interface (UI) in this section. Our system supports all 4 IE tasks into a single integrated interface.

Template development. Figure 6a shows the main template development UI, in which each tab on the top represents one of the entity, relation, event, and event argument tasks. The user switch between tasks by simply clicking on a different tab (the tabs for the other 3 tasks are shown in Figure 6b, 6c, and 6d, respectively).

Take the NER task as an example (Figure 6a), it shows an overview of all entity types along with the templates defined for each type (e.g., "X is a person" for the type PERSON, in which "X" is a placeholder that can be replaced with a noun phrase "New York City"). If the user clicks on the edit button (the pen-shaped button), the pop-up window for adding a new entity type (the right-hand side figure in Figure 6a) shows up. The user can add a template by clicking on "+" sign, and then input the template to the left (the user can repeat this several times to add more templates). The user can remove a template by clicking on "-". The user can also click on the big "+" card to the left to add a new entity type.

Template development for the relation extraction task is similar to NER, except for two differences: first, as shown in Figure 6b (right), we can further add a set of "allowed type" pairs, that are the set of

entity pairs each relation is defined over. For example, the "per:date_of_death" relation is only valid between a pair of PERSON and DATE mentions. Our UI allows the user to specify the "LeftEntityType" (left entity type) and the "RightEntityType" (right entity type) for each relation type under "allowed type". These type constraints are show on the top box for each relation card on the left figure in Figure 6b (e.g., "PERSON->DATE" under "per:date_of_death"). Second, a relation involves a pair of entity mentions. Therefore, each pattern has two placeholders, "X" and "Y", which can be replaced with two entity candidates that are likely to participate in the relationship.

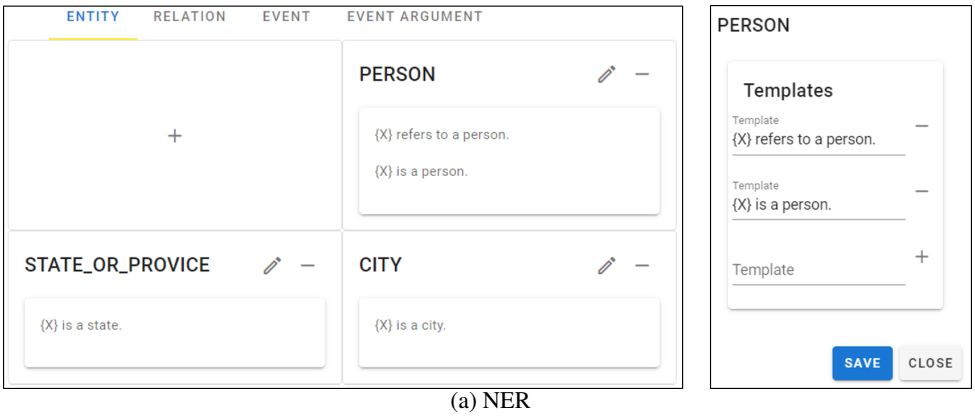
Template development for the event extraction task (Figure 6c) is also similar to NER, except that the template may not contain any trigger. For example, "Someone died" is a template for the "Death" event (Figure 6c). This template would allow the TE approach to classify whether an extent (e.g., a sentence) expresses a type of event.

Template development for the event argument extraction task (Figure 6d) is similar to relation extraction, except that the template can include either two placeholders "X" and "Y" in which "X" is an event trigger and "Y" is an event argument candidate filler (an entity), or only one placeholder "Y" which is the event argument candidate filler. The later would require the template to implicitly describes the event type as well (for example, "Someone died in Y" for the LOCATION event argument role in Figure 6d).

Template validation. We developed an interactive workflow to allow the user to quickly develop templates and validate their effectiveness in our TE-based framework. To support this workflow, our UI allows the user to run inference over any free text supplied by the user herself/himself. For simplicity, we omit the UI where we allow the user type in free text. We show the UI that displays the extraction output on the free text, that also allows the user to label the correcness of the extractions. Based on those labeled examples, the UI also automatically calculate a few metrics to help the user to find the effectiveness of the templates curated so far.

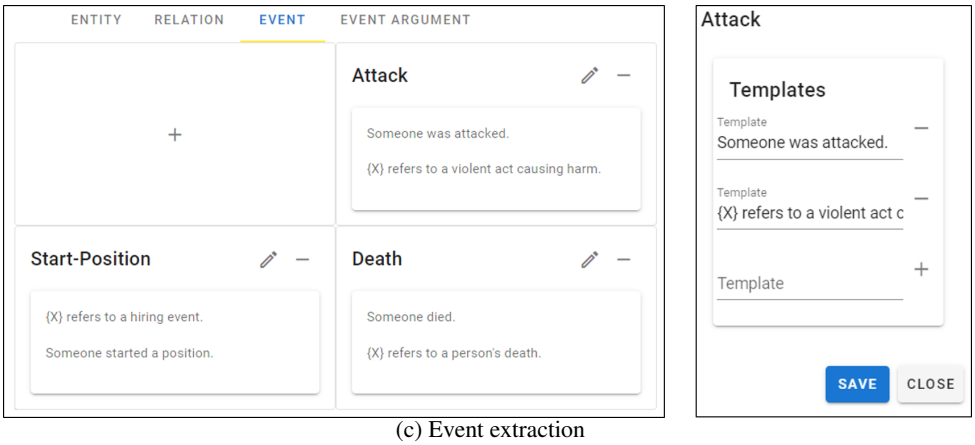
Figure 7 shows the UI for displaying NER extraction outputs (left) and automatically calculated metrics (right). Taken the user-supplied sentence "John Smith, an executive at XYZ Corp., died in Florida on Sunday" as input, the UI on the left-hand side shows the extracted named entities. It shows extractions such as "John Smith is a/an PERSON", "Sunday is a/an DATE", and so on. To provide rationale for each extraction, it displays a rank list of possible entity types, the template led to that type, along with the entailment probability produced by the pre-trained TE model. The user can click on "+" and "-" sign next to each extraction to label its correctness. In Figure 7, all extractions are green (labeled by the user as correct) except that "Florida is a/an CITY" is in red (labeled as incorrect by the user). Based on these user-labeled extractions, the system calculated a number of metrics to facilitate template validation: the total number of extracted named entities (shown under "total"), the number of correct and incorrect extractions under "correct" and "incorrect", respectively (the accuracy number is also shown in the parenthesis next to "correct") for the overall task, each type, and each pattern. The right-hand side UI in Figure 7 displays these metrics, and allows the user to sort patterns/types by each of the metric. The user can quickly identify some templates are low-precision (e.g., "X is a location" for the entity type CITY), and can revise them to improve precision.

Figure 8a, 8b, and 8c shows the UI for displaying extraction results for the relation extraction, event extraction, and event argument extraction, respectively. Similar to the NER task. Similarly, our system also includes metric board (the metrics above) for the other 3 IE tasks. To view the metric boards for these tasks, please refer to our demonstration video.



(a) NER

(b) Relation extraction



(c) Event extraction

(d) Event argument extraction

Figure 6: The UI for developing templates for the 4 IE tasks. For each task, we show the overall UI on the left, and the pop-up window for adding a new entity type PERSON on the right.

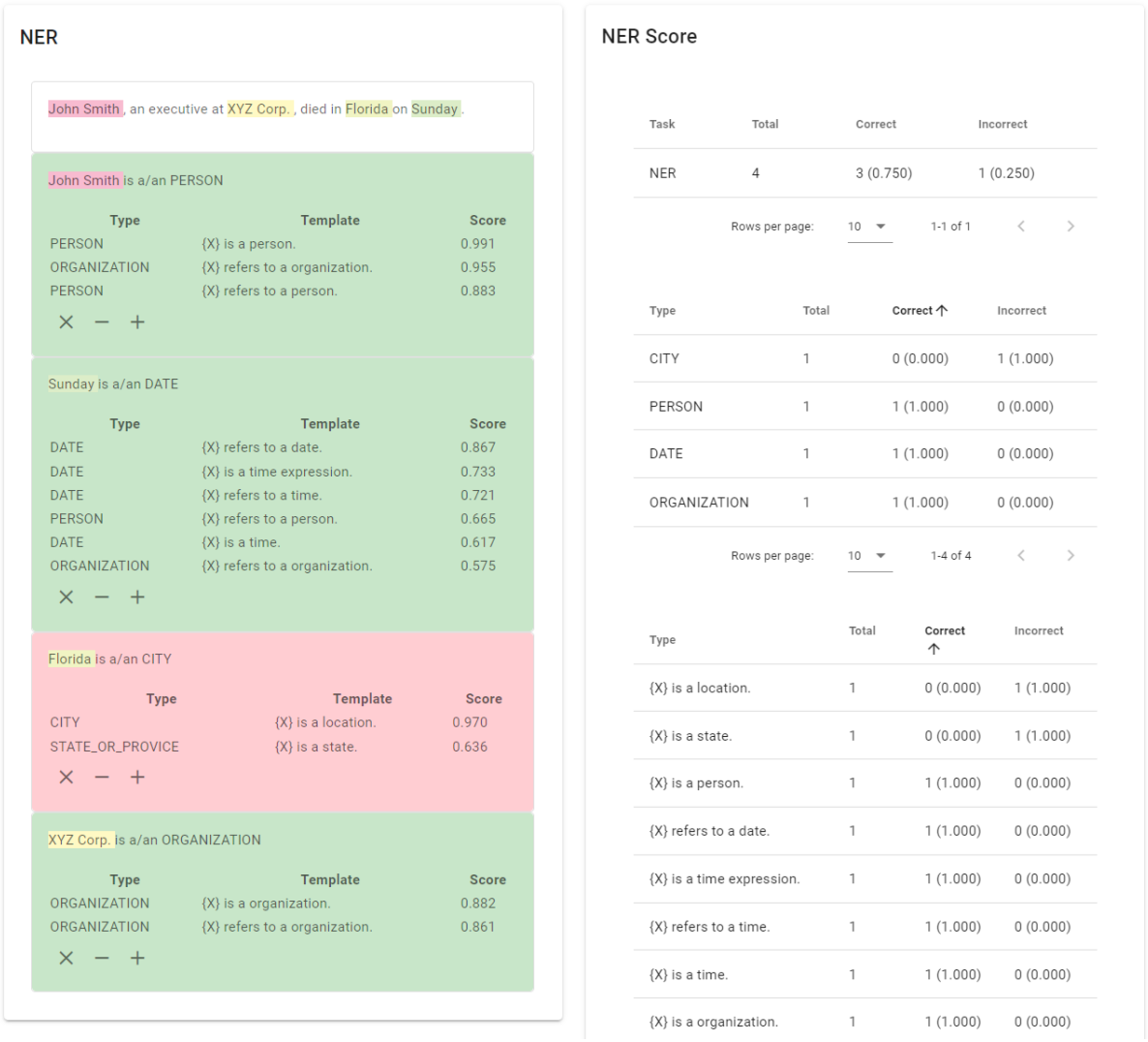
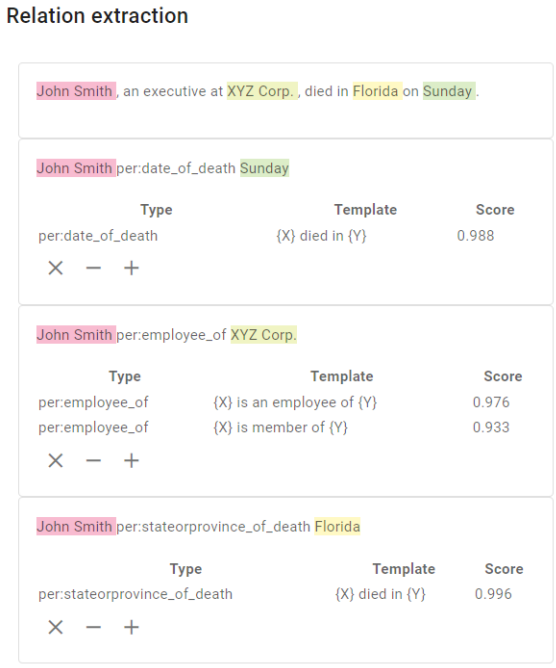
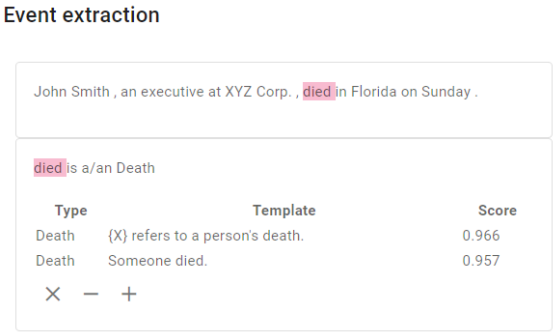


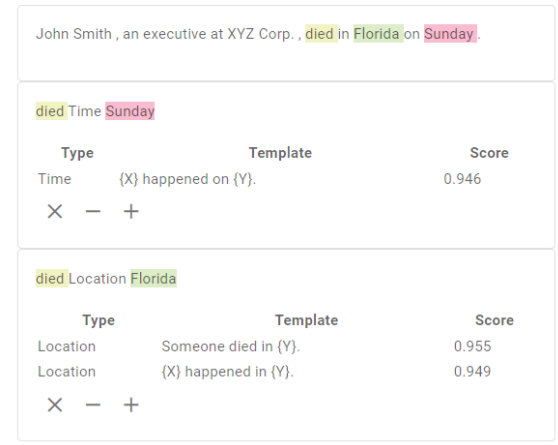
Figure 7: The UI for displaying NER extraction outputs (left) and automatically calculated metrics (right). The left-hand side shows the named entities extracted from an user-input sentence (shown on the top). The user can click on "+" and "-" sign next to each extraction to label its correctness. The right-hand side shows the total number of extracted named entities (total), the number of correct and incorrect extractions (the accuracy number is also shown in the parenthesis next to “correct”) for the overall task, each type, and each pattern. These metrics are calculated based on the set of user labels.



(a) Relation extraction



(b) Event extraction



(c) Event argument extraction

Figure 8: The UI for displaying extractions for relation extraction, event extraction, and event argument extraction, respectively. The user an click on the "+" or "-" sign next to each extraction to label the extraction as correct or incorrect.