IKASKETA-ADIBIDE URRIKO INFORMAZIO-ERAUZKETA

LOW-RESOURCE INFORMATION EXTRACTION

Oscar Sainz Jimenez

Supervised by **Eneko Agirre** and **Oier Lopez de Lacalle**

HiTZ Zentroa - Ixa taldea
Euskal Herriko Unibertsitatea UPV/EHU

PhD Dissertation
July 15, 2024

Universidad del País Vasco · Euskal Herriko Unibertsitatea

HiTZ
Hizkuntza Teknologiako Zentroa
Basque Center for Language Technology

OUTLINE

[1] Section 2.1 is going to be presented in Basque

Introduction

Motivation

The amount of information and knowledge about the world is growing at an unprecedented rate:

- News
- Social media
- Scientific publications
- ...

This knowledge, is transmitted **using natural language**, which:

- Is the natural way to communicate for humans.
- But, is not easily understood and processed by machines, which prefer some **structured representation**.

The field of Information Extraction (IE) aims to bridge this gap by:

- Automatically extracting structured information from unstructured text.
- Enabling machines to interpret the text.
- Enabling the creation of knowledge bases and representations.

However, current IE systems **are limited to a few schemas and domains**.

What do we do if we want to extract information from some texts, but:

- No available data to train a model **for our specific use case.**
- No resources —expertise, time or money— to create a large dataset.

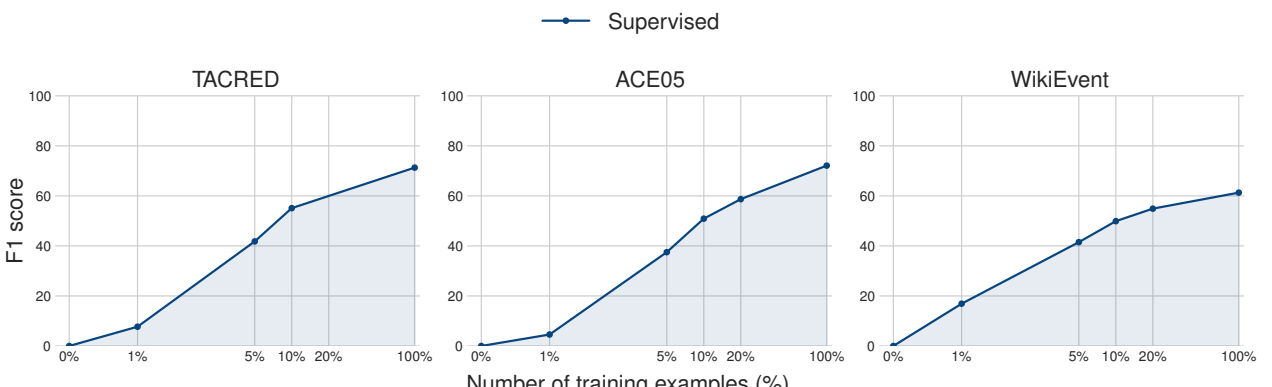


Figure. Performance of state-of-the-art IE models on different benchmarks for different amounts of training data.

Current state-of-the-art systems for Information Extraction

- Require large amounts of labeled data — do not work without data.
- Are tied to the schema of the training data.

Humans, however:

- Can perform IE tasks with task descriptions and a couple of examples (i.e. annotation guidelines).
- Can adapt to changes in the annotation schemas with minimal effort.

Main Research Question

Can we leverage the recent advances carried by Language Models to create Information Extraction systems that can adapt to new tasks and schemas **with minimal supervision**?

Introduction

Background – Information Extraction

Definition (Grishman, 2019)

- *IE is the automatic identification and classification of instances of user-specified entities, relations, and events from the text.*
- *The output must be structured.*
- *The specification may take the form of examples or textual descriptions.*
- *Equivalent texts should be mapped to the same output.*

Definition (Grishman, 2019)

- *IE is the automatic identification and classification of instances of user-specified entities, relations, and events from the text.*
- *The output must be structured.*
- *The specification may take the form of examples or textual descriptions.*
- *Equivalent texts should be mapped to the same output.*

Definition (Grishman, 2019)

- *IE is the automatic identification and classification of instances of user-specified entities, relations, and events from the text.*
- *The output must be structured.*
- *The specification may take the form of examples or textual descriptions.*
- *Equivalent texts should be mapped to the same output.*

Definition (Grishman, 2019)

- *IE is the automatic identification and classification of instances of user-specified entities, relations, and events from the text.*
- *The output must be structured.*
- *The specification may take the form of examples or textual descriptions.*
- *Equivalent texts should be mapped to the same output.*

Example

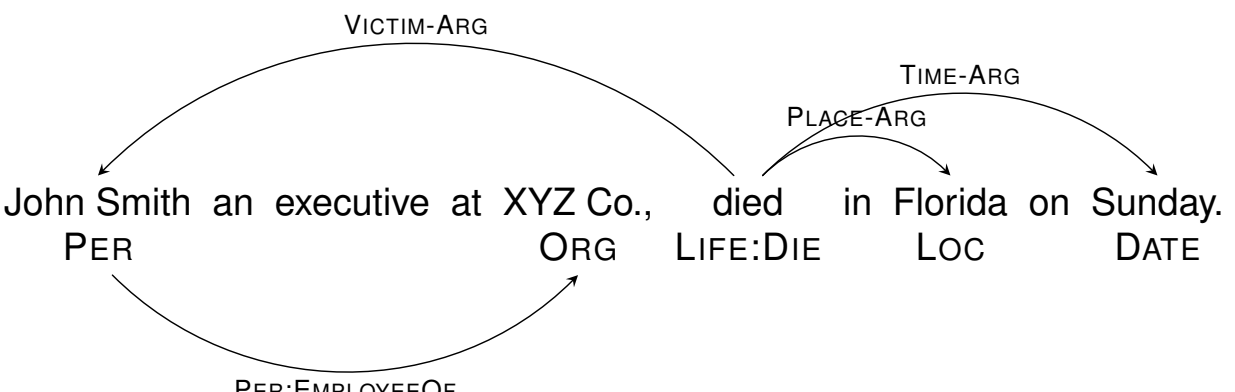


Figure. This example contains annotations for entities, relations, events, and their arguments.

Example

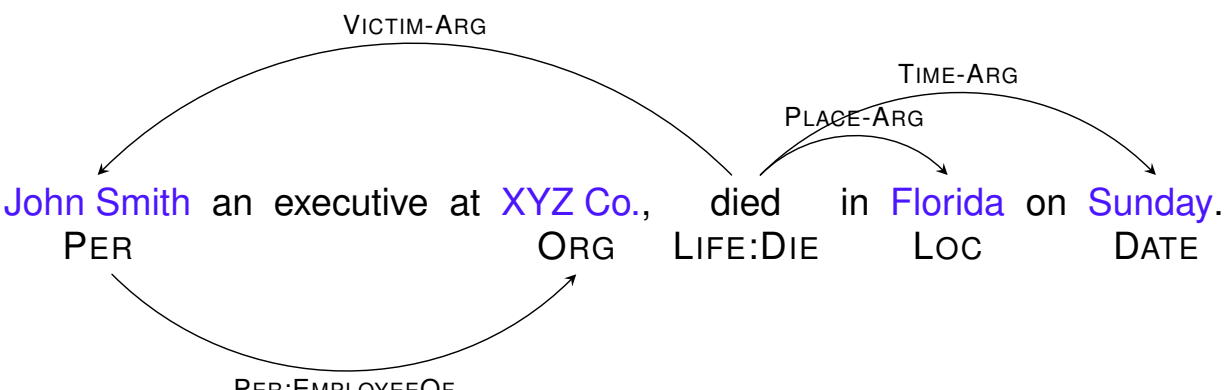


Figure. This example contains annotations for entities, relations, events, and their arguments.

Example

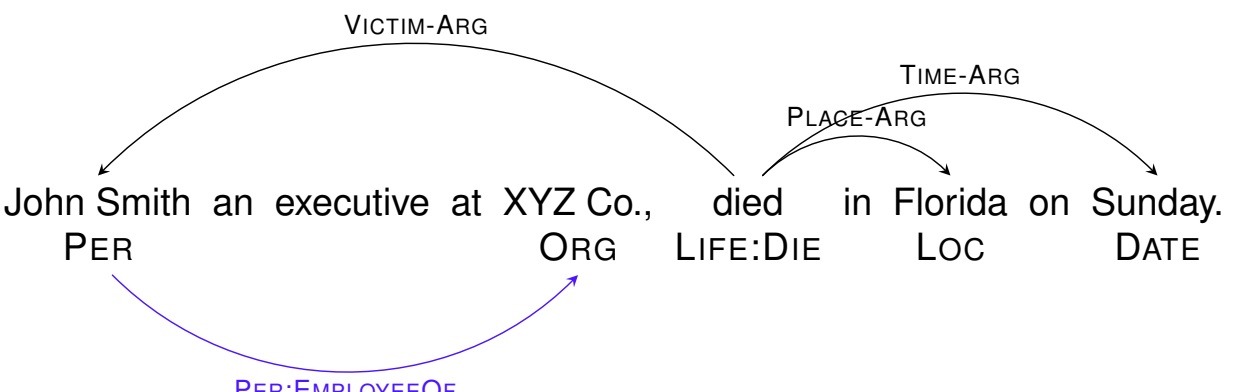


Figure. This example contains annotations for entities, relations, events, and their arguments.

Example

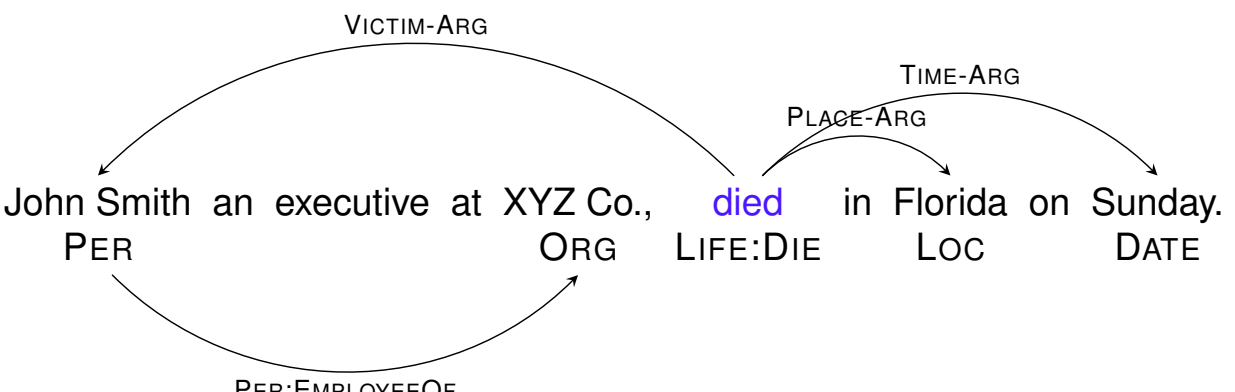


Figure. This example contains annotations for entities, relations, events, and their arguments.

Example

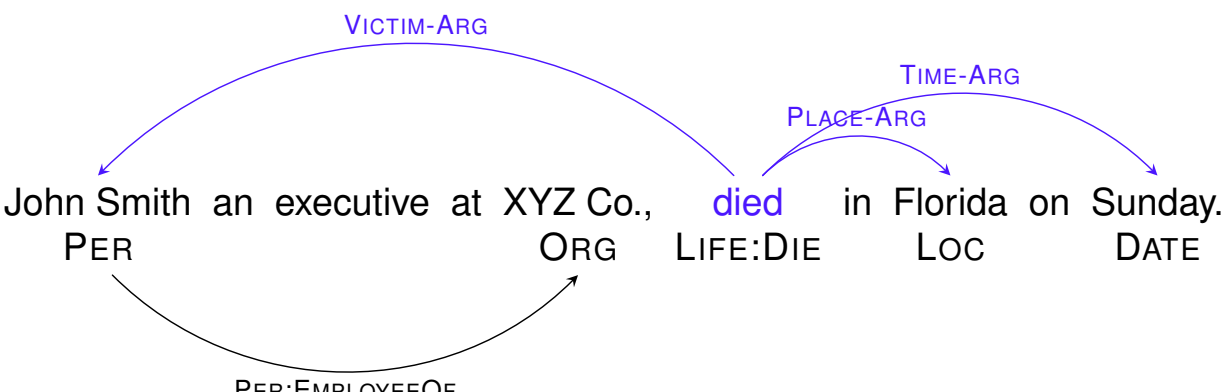


Figure. This example contains annotations for entities, relations, events, and their arguments.

Introduction

Background – Zero and Few-Shot Learning

Definition

- Zero-Shot Learning (ZSL) is a paradigm that aims to learn to recognize new classes without any training examples.
- In some cases, this is also applied to new tasks or domains.
- Few-Shot learning is a related paradigm where the model is given a few examples of the new classes, tasks, or domains.

Definition

- Zero-Shot Learning (ZSL) is a paradigm that aims to learn to recognize new classes without any training examples.
- In some cases, this is also applied to new tasks or domains.
- Few-Shot learning is a related paradigm where the model is given a few examples of the new classes, tasks, or domains.

Definition

- Zero-Shot Learning (ZSL) is a paradigm that aims to learn to recognize new classes without any training examples.
- In some cases, this is also applied to new tasks or domains.
- Few-Shot learning is a related paradigm where the model is given a few examples of the new classes, tasks, or domains.

ZSL in Natural Language Processing

Methods such as **In-Context Learning (ICL)**, **Pattern-Exploiting Training (PET)**, and **pivot-task based approaches** have been proven effective for Zero-Shot Learning methods.

Intuition: Formulate the task of interest as similar to the training objective of the LMs.

Introduction

Background – Textual Entailment

Definition (Dagan et al., 2006; de Marneffe et al., 2008)

- Textual entailment is defined as a directional relationship between pairs of text expressions called the premise *P* and the hypothesis *H*.
- Is said that *P* entails *H* if, typically, a human reading *P* would infer that *H* is true.
- Is said that *H* contradicts *P* if, a human reading *P* would infer that events in *H* are very unlikely to occur given *P*.

The task of Textual Entailment is usually formulated as a **three-way classification task**.

Definition (Dagan et al., 2006; de Marneffe et al., 2008)

- Textual entailment is defined as a directional relationship between pairs of text expressions called the premise *P* and the hypothesis *H*.
- Is said that *P* entails *H* if, typically, a human reading *P* would infer that *H* is true.
- Is said that *H* contradicts *P* if, a human reading *P* would infer that events in *H* are very unlikely to occur given *P*.

The task of Textual Entailment is usually formulated as a **three-way classification task**.

Definition (Dagan et al., 2006; de Marneffe et al., 2008)

- Textual entailment is defined as a directional relationship between pairs of text expressions called the premise P and the hypothesis H.
- Is said that P entails H if, typically, a human reading P would infer that H is true.
- Is said that H contradicts P if, a human reading P would infer that events in H are very unlikely to occur given P.

The task of Textual Entailment is usually formulated as a **three-way classification task**.

Example

Premise:	*A person on a horse jumps over a broken-down airplane.*
Entails:	*A person is outdoors, on a horse.*
Neutral:	*A person is training his horse for a competition.*
Contradicts:	*A person is at a diner, ordering an omelette.*

Figure. Example of Textual Entailment task.

Example

Premise:	*A person on a horse jumps over a broken-down airplane.*
Entails:	*A person is outdoors, on a horse.*
Neutral:	*A person is training his horse for a competition.*
Contradicts:	*A person is at a diner, ordering an omelette.*

Figure. Example of Textual Entailment task.

In this case, the premise entails the hypothesis because the hypothesis is **a general case of** the premise.

Example

Premise:	*A person on a horse jumps over a broken-down airplane.*
Entails:	*A person is outdoors, on a horse.*
Neutral:	*A person is training his horse for a competition.*
Contradicts:	*A person is at a diner, ordering an omelette.*

Figure. Example of Textual Entailment task.

The relation is neutral because the premise **lacks information** about the events in the hypothesis.

Example

Premise:	*A person on a horse jumps over a broken-down airplane.*
Entails:	*A person is outdoors, on a horse.*
Neutral:	*A person is training his horse for a competition.*
Contradicts:	*A person is at a diner, ordering an omelette.*

Figure. Example of Textual Entailment task.

The hypothesis contradicts the premise because the events in the hypothesis **are not compatible with** the events in the premise.

Why Textual Entailment as a pivot for ZSL?

In order to properly solve the task of recognizing textual entailment, a model must understand:

- Lexical semantics: a dog is an animal but not a cat.
- Predicate argument structure: I baked him a cake entails I baked a cake but not I baked him .
- Logic inference: I will drink coffee or water does not entail I will drink coffee but it does the other way around.
- World knowledge: there are Basque speakers in Donostia entails there are Basque speakers in Gipuzkoa .

Why Textual Entailment as a pivot for ZSL?

In order to properly solve the task of recognizing textual entailment, a model must understand:

- Lexical semantics: a dog is an animal but not a cat.
- Predicate argument structure: I baked him a cake entails I baked a cake but not I baked him.
- Logic inference: I will drink coffee or water does not entail I will drink coffee but it does the other way around.
- World knowledge: there are Basque speakers in Donostia entails there are Basque speakers in Gipuzkoa.

Why Textual Entailment as a pivot for ZSL?

In order to properly solve the task of recognizing textual entailment, a model must understand:

- Lexical semantics: a dog is an animal but not a cat.
- Predicate argument structure: I baked him a cake entails I baked a cake but not I baked him.
- Logic inference: I will drink coffee or water does not entail I will drink coffee but it does the other way around.
- World knowledge: there are Basque speakers in Donostia entails there are Basque speakers in Gipuzkoa.

Why Textual Entailment as a pivot for ZSL?

In order to properly solve the task of recognizing textual entailment, a model must understand:

- Lexical semantics: a dog is an animal but not a cat.
- Predicate argument structure: I baked him a cake entails I baked a cake but not I baked him.
- Logic inference: I will drink coffee or water does not entail I will drink coffee but it does the other way around.
- World knowledge: there are Basque speakers in Donostia entails there are Basque speakers in Gipuzkoa.

CONTRIBUTIONS

TEXTUAL ENTAILMENT FOR IE

Motivation

- Current state-of-the-art approaches require a lot of data to perform well.
- These approaches are tied to a single schema (defined by the data) by design.

Motivation

- Current state-of-the-art approaches require a lot of data to perform well.
- These approaches are tied to a single schema (defined by the data) by design.

Motivation

- Current state-of-the-art approaches require a lot of data to perform well.
- These approaches are tied to a single schema (defined by the data) by design.

How can we formulate Information Extraction to be flexible for new scenarios?

Approach

Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in **Tampa**, Fla, on Sunday.

context

Billy Mays PER , **Tampa** LOC

(e_1, e_2)

Approach

Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in **Tampa**, Fla, on Sunday.

context

Billy Mays PER , **Tampa** LOC

(e_1, e_2)

PER:CITY_OF_DEATH

Approach

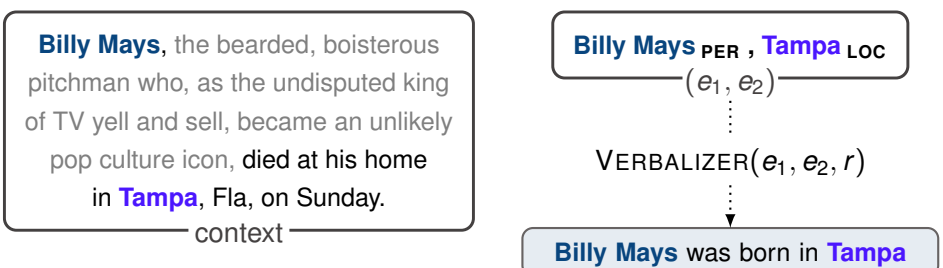
Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in **Tampa**, Fla, on Sunday.

context

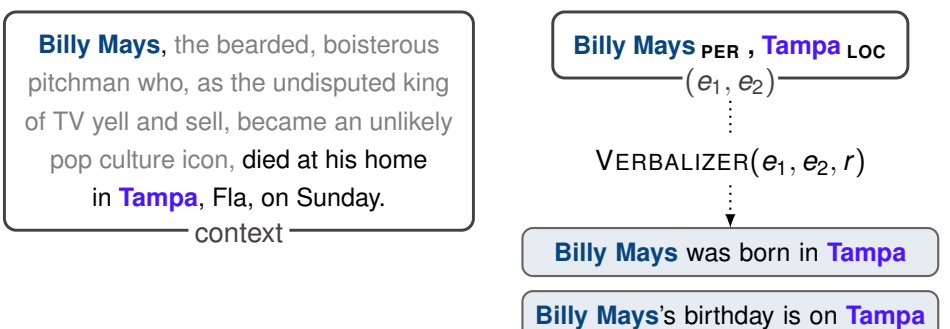
Billy Mays PER , **Tampa** LOC

(e_1, e_2)

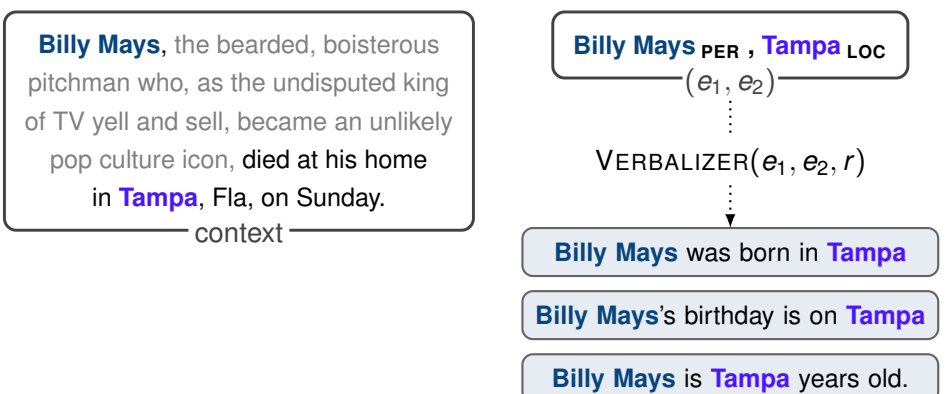
Approach


Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in Tampa, Fla, on Sunday.
context
Billy Mays PER , Tampa LOC
(e_1, e_2)
VERBALIZER(e_1, e_2, r)
Billy Mays was born in Tampa

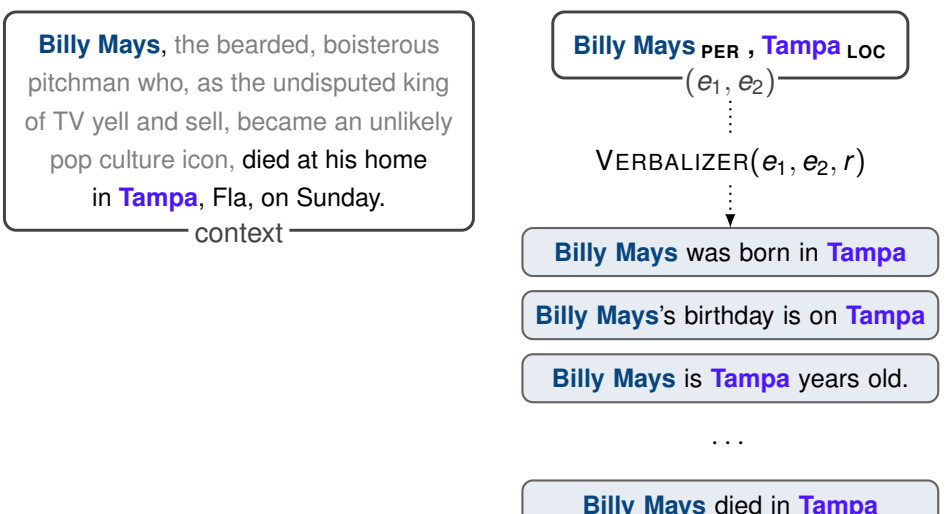
Approach


Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in Tampa, Fla, on Sunday.
context
Billy Mays PER , Tampa LOC
(e_1, e_2)
VERBALIZER(e_1, e_2, r)
Billy Mays was born in Tampa
Billy Mays's birthday is on Tampa

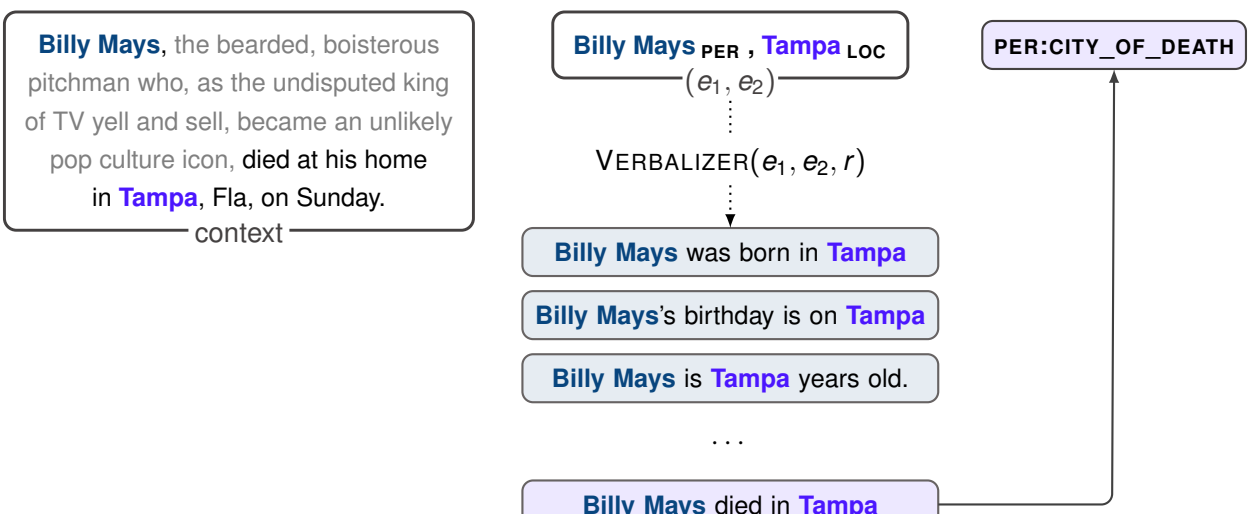
Approach


Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in Tampa, Fla, on Sunday.
context
Billy Mays PER , Tampa LOC
(e_1, e_2)
VERBALIZER(e_1, e_2, r)
Billy Mays was born in Tampa
Billy Mays's birthday is on Tampa
Billy Mays is Tampa years old.

Approach


Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in Tampa, Fla, on Sunday.
context
Billy Mays PER , Tampa LOC
(e_1, e_2)
VERBALIZER(e_1, e_2, r)
Billy Mays was born in Tampa
Billy Mays's birthday is on Tampa
Billy Mays is Tampa years old.
...
Billy Mays died in Tampa

Approach

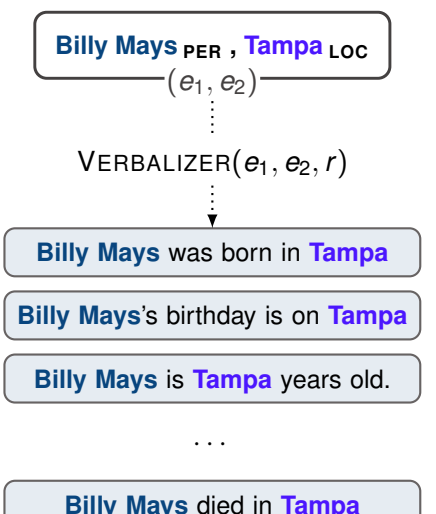

Billy Mays, the bearded, boisterous pitchman who, as the undisputed king of TV yell and sell, became an unlikely pop culture icon, died at his home in Tampa, Fla, on Sunday.
context
Billy Mays PER , Tampa LOC
(e_1, e_2)
VERBALIZER(e_1, e_2, r)
Billy Mays was born in Tampa
Billy Mays's birthday is on Tampa
Billy Mays is Tampa years old.
...
Billy Mays died in Tampa
PER:CITY_OF_DEATH

Label verbalization

- Label verbalization is the process of converting a label into a prototypical natural language sentence.
- These verbalizations will be later used as hypotheses $H_r = \text{VERBALIZER}(e_1, e_2, r)$

Relation	Templates	Valid argument types
per:alternate_names	{e1} is also known as {e2}	PERSON
per:date_of_birth	{e1}'s birthday is on {e2}	DATE
	{e1} was born on {e2}	
per:age	{e1} is {e2} years old	NUMBER, DURATION
per:country_of_birth	{e1} was born in {e2}	COUNTRY
per:stateorprovince_of_birth	{e1} was born in {e2}	STATEORPROVINCE

Table. Templates and valid argument types for some relations.



Inference

The probability for a given relation r to hold between two entities e_1 and e_2 is computed as:

$$P(r \mid x, e_1, e_2) = \max_{hyp \in H_r} P_\theta(\text{ent} \mid x, hyp) \tag{1}$$

where $P_\theta(ent \mid \cdot)$ is the entailment probability given by the textual entailment model and x is the context.

The relation is predicted as the one with the highest probability, among the relations that satisfy the type constraints:

$$y = \arg\max_{r \in R} \delta_r(e_1, e_2) \cdot P(r \mid x, e_1, e_2) \tag{2}$$

where $\delta_r(e_1, e_2)$ is a function that returns 1 if the entities are of the correct type for the relation r and 0 otherwise:

$$\delta_r(e_1, e_2) = \begin{cases} 1 & (e_1, e_2) \in E_r \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Inference

The probability for a given relation r to hold between two entities e_1 and e_2 is computed as:

$$P(r \mid x, e_1, e_2) = \max_{hyp \in H_r} P_\theta(\text{ent} \mid x, hyp) \tag{1}$$

where $P_\theta(ent \mid \cdot)$ is the entailment probability given by the textual entailment model and x is the context.

The relation is predicted as the one with the highest probability, among the relations that satisfy the type constraints:

$$y = \arg\max_{r \in R} \delta_r(e_1, e_2) \cdot P(r \mid x, e_1, e_2) \tag{2}$$

where $\delta_r(e_1, e_2)$ is a function that returns 1 if the entities are of the correct type for the relation r and 0 otherwise:

$$\delta_r(e_1, e_2) = \begin{cases} 1 & (e_1, e_2) \in E_r \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Inference

The probability for a given relation r to hold between two entities e_1 and e_2 is computed as:

$$P(r \mid x, e_1, e_2) = \max_{hyp \in H_r} P_\theta(\text{ent} \mid x, hyp) \tag{1}$$

where $P_\theta(ent \mid \cdot)$ is the entailment probability given by the textual entailment model and x is the context.

The relation is predicted as the one with the highest probability, among the relations that satisfy the type constraints:

$$y = \arg\max_{r \in R} \delta_r(e_1, e_2) \cdot P(r \mid x, e_1, e_2) \tag{2}$$

where $\delta_r(e_1, e_2)$ is a function that returns 1 if the entities are of the correct type for the relation r and 0 otherwise:

$$\delta_r(e_1, e_2) = \begin{cases} 1 & (e_1, e_2) \in E_r \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Inference

Information Extraction tasks usually define an extra class for the case where no relation holds between the entities. This class is usually called **no relation**, **none**, or **negative** class.

In order to predict the correct label considering the negative class we leverage a threshold τ:

$$\hat{r} = \begin{cases} y & P(y \mid x, e_1, e_2) >= \tau \\ none & \text{otherwise} \end{cases} \tag{4}$$

The threshold τ is usually set to 0.5, but it can be adjusted to maximize the F1 score on the validation set.

Inference

Information Extraction tasks usually define an extra class for the case where no relation holds between the entities. This class is usually called **no relation**, **none**, or **negative** class.

In order to predict the correct label considering the negative class we leverage a threshold τ:

$$\hat{r} = \begin{cases} y & P(y \mid x, e_1, e_2) >= \tau \\ none & \text{otherwise} \end{cases} \tag{4}$$

The threshold τ is usually set to 0.5, but it can be adjusted to maximize the F1 score on the validation set.

Research Questions

- How does this approach perform on different Information Extraction tasks in a Zero-Shot setting?
  - How does the performance vary with different models?
- How does the Textual Entailment approach scale with Information Extraction data?
  - How does it compare with traditional (state-of-the-art) methods?
- How does the amount and variety of textual entailment data affect the performance of the model?
- Can we transfer the knowledge learned from one schema to another?
- How does the performance vary with different verbalization styles?
- How is the effort-performance relation of creating verbalizations compared to annotating examples?

Experimental setup

- We compared two state-of-the-art models fine-tuned on Textual Entailment data:
  - RoBERTa (Liu et al., 2019)
  - DeBERTa (He et al., 2021)
- We implemented and created verbalizations for 4 tasks across 3 different datasets:
  - NER: CoNLL 2003 (Tjong Kim Sang & De Meulder, 2003)
  - RE: TACRED (Zhang et al., 2017)
  - EE and EAE: ACE05 (Walker et al., 2006)
- We evaluated the impact of optimizing the threshold τ on the performance.

Zero-Shot results

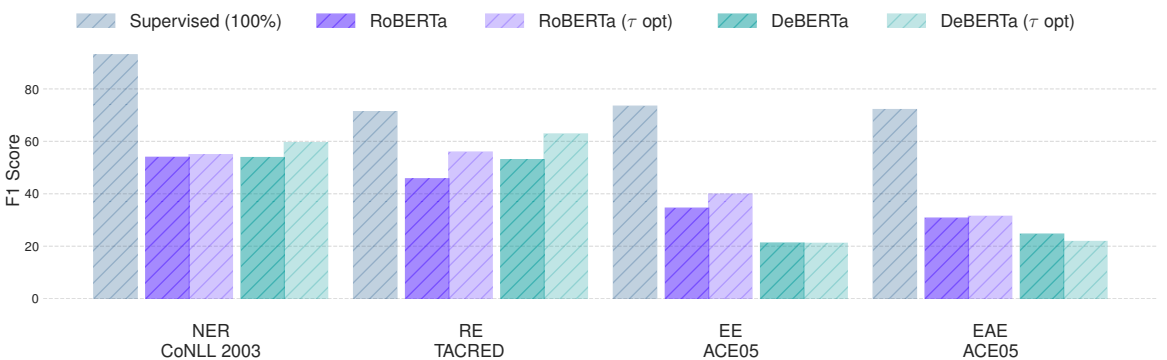


Figure. Zero-Shot results on several IE tasks.

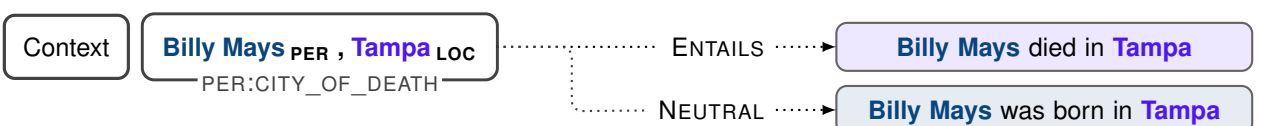
- There is no one best model for all cases (RoBERTa vs DeBERTa).
- Optimizing the threshold τ boosts performance in some cases but, it can also overfit.

Research Questions

- How does this approach perform on different Information Extraction tasks in a Zero-Shot setting?
  - How does the performance vary with different models?
- How does the Textual Entailment approach scale with Information Extraction data?
  - How does it compare with traditional (state-of-the-art) methods?
- How does the amount and variety of textual entailment data affect the performance of the model?
- Can we transfer the knowledge learned from one schema to another?
- How does the performance vary with different verbalization styles?
- How is the effort-performance relation of creating verbalizations compared to annotating examples?

Fine-tuning Textual Entailment models with Information Extraction data

Converting **positive** examples:



For **negative** examples:

Experimental setup

- We focused on the Relation Extraction and Event Argument Extraction tasks.
- We compared the performance of the approach with the state-of-the-art models for each of the tasks and a **strong baseline** (Baldini Soares et al., 2019).
- We fine-tuned each of the models with different amounts of data: 0%, 1%, 5%, 10%, 20% and 100%.
- Average across three runs is reported.

Few-Shot results

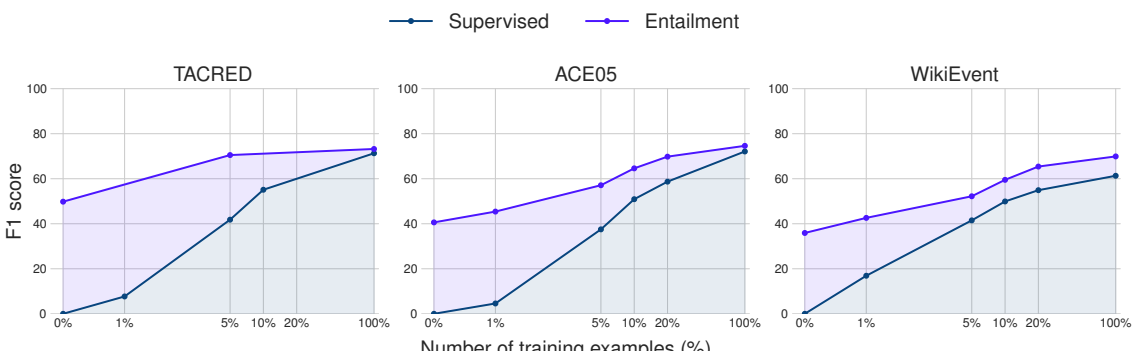


Figure. Few-Shot results on Relation Extraction and Event Argument Extraction.

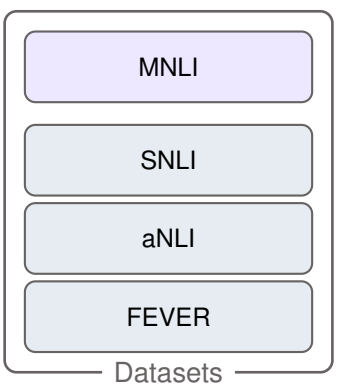
- The Textual Entailment approach outperforms the baseline in all stages, particularly in low-resource.
- For some cases (WikiEvents), the approach even improves by a large margin with 100% of the data.

Research Questions

- How does this approach perform on different Information Extraction tasks in a Zero-Shot setting?
  - How does the performance vary with different models?
- How does the Textual Entailment approach scale with Information Extraction data?
  - How does it compare with traditional (state-of-the-art) methods?
- How does the amount and variety of textual entailment data affect the performance of the model?
- Can we transfer the knowledge learned from one schema to another?
- How does the performance vary with different verbalization styles?
- How is the effort-performance relation of creating verbalizations compared to annotating examples?

Textual Entailment data

- MNLI: Multi-Genre Natural Language Inference.
- SNLI: Stanford Natural Language Inference.
- aNLI: Adversarial Natural Language Inference.
- Fever-NLI: Fact Extraction and VERification.



Experimental setup

We compared RoBERTa models fine-tuned on different textual entailment data:

- Entailment: a RoBERTa model fine-tuned on MNLI, SNLI, aNLI and Fever-NLI data.
- Entailment $_{\text{MNLI-only}}$: a RoBERTa model fine-tuned only on MNLI data.

Textual Entailment data results

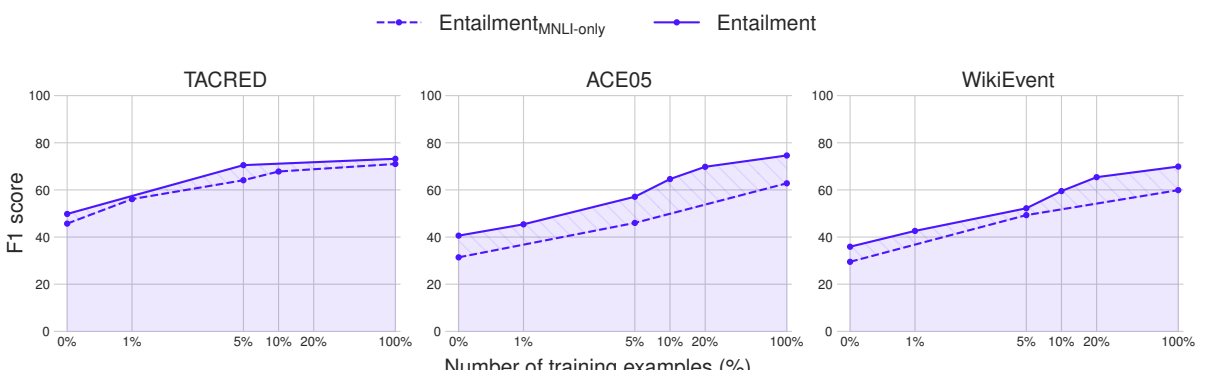


Figure. Results on Relation Extraction and Event Argument Extraction with different Textual Entailment data.

- The variety and quantity of data affect all scenarios, from low-resource to high-resource.

Research Questions

- How does this approach perform on different Information Extraction tasks in a Zero-Shot setting?
  - How does the performance vary with different models?
- How does the Textual Entailment approach scale with Information Extraction data?
  - How does it compare with traditional (state-of-the-art) methods?
- How does the amount and variety of textual entailment data affect the performance of the model?
- Can we transfer the knowledge learned from one schema to another?
- How does the performance vary with different verbalization styles?
- How is the effort-performance relation of creating verbalizations compared to annotating examples?

Experimental setup

We compared Entailment models fine-tuned on different Event Argument Extraction data:

- WikiEvents –> ACE: a model trained first on WikiEvents and then evaluated on ACE.
- ACE –> WikiEvents: a model trained first on ACE and then evaluated on WikiEvents.

Multi-source learning results

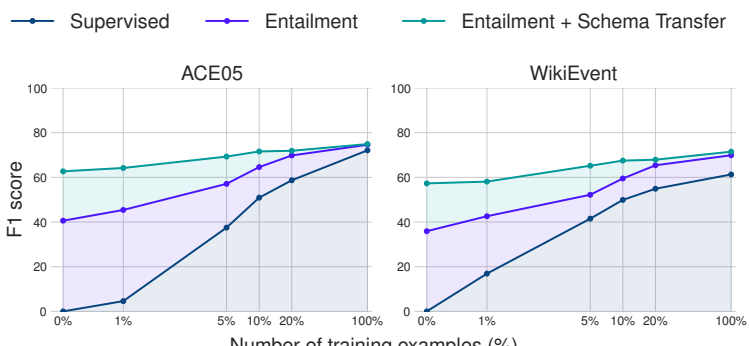


Figure. Schema transfer results across Event Argument Extraction tasks.

- The knowledge from one schema is successfully transferred to another schema without mapping.
- It particularly impacts the Zero-Shot results, where the model has no access to the target schema.

Research Questions

- How does this approach perform on different Information Extraction tasks in a Zero-Shot setting?
  - How does the performance vary with different models?
- How does the Textual Entailment approach scale with Information Extraction data?
  - How does it compare with traditional (state-of-the-art) methods?
- How does the amount and variety of textual entailment data affect the performance of the model?
- Can we transfer the knowledge learned from one schema to another?
- How does the performance vary with different verbalization styles?
- How is the effort-performance relation of creating verbalizations compared to annotating examples?

Verbalizations for different tasks

Named Entity Recognition	Relation Extraction	Event Extraction	Event Argument Extraction
`{X} is a person.`	`{X} is employed by {Y}.`	`{X} refers to a birth.`	`The victim was {X}`
`{X} is a date.`	`{X} and {Y} are siblings.`	`Someone got married.`	`The {event} occurred in {X}.`
`{X} is a {type}.`	`{Y} is the son of {X}.`	`{argument} was jailed.`	`{X} inspected something.`

The verbalization templates:

- must have variables for the spans —`{X}` and `{Y}`— to classify
- can be generic as `{X} is a {type}`, or specific as `{X} is a person`.
- can include additional information about the instance as `{argument} was jailed` or some placeholders as `Someone got married`.

Experimental setup

We compared the performance in Event Argument Extraction with different verbalization styles:

- We compared verbalizations created by:
  - a developer with **Computer Science** background and experience in Information Extraction.
  - a developer with **Linguistics** background and experience in annotation tasks.
- Developers were limited to 15 minutes per type.
- Developers were provided with the annotation guidelines, a couple of examples of the type and a small script to test the verbalizations individually with the examples.

Verbalization style comparison

- The average and standard deviation of the F1 scores across 3 runs are reported.
- There are **no major differences** in the performance between the two verbalization styles, except for the 100% scenario where the templates of the developer A are slightly better (3 F1 points).

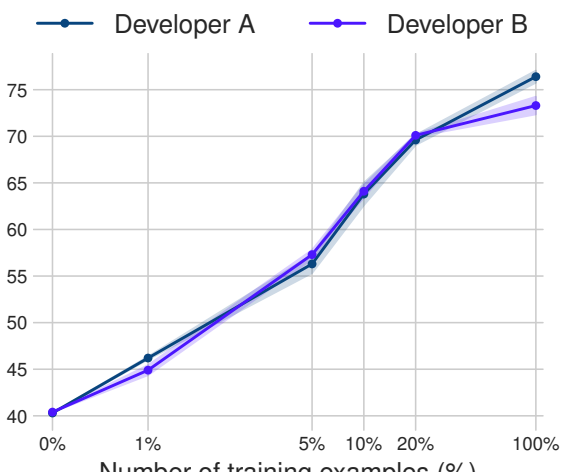


Figure. Performance difference on the development.

Research Questions

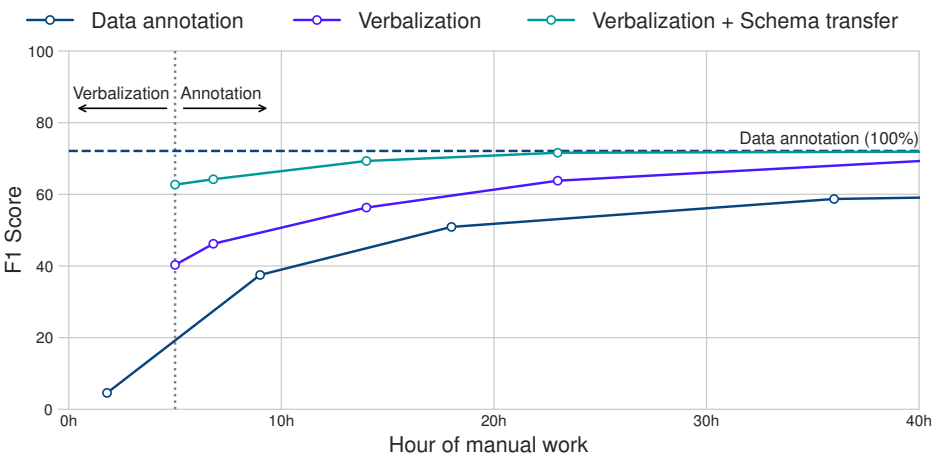
- How does this approach perform on different Information Extraction tasks in a Zero-Shot setting?
  - How does the performance vary with different models?
- How does the Textual Entailment approach scale with Information Extraction data?
  - How does it compare with traditional (state-of-the-art) methods?
- How does the amount and variety of textual entailment data affect the performance of the model?
- Can we transfer the knowledge learned from one schema to another?
- How does the performance vary with different verbalization styles?
- How is the effort performance ratio of creating verbalizations compared to annotating examples?

Experimental setup

We compared the performance in Event Argument Extraction of annotating examples vs creating verbalizations in terms of time:

- We performed a simulation of annotating a portion of the ACE05 dataset.
- Based on the time spent, we extrapolated the time to annotate the whole dataset.
- We measured the time spent in creating the verbalizations for the same dataset.

Effort-performance comparison



- Spending some initial time creating verbalizations is more efficient than annotating examples.
- At ∼23 hours of work, the verbalizations are more efficient than annotating examples.

Conclusions

- **Works out of the box**: the approach works without giving any example for training.
- **Few-shot is easy**: examples from the tasks can be automatically converted to Textual Entailment.
- **Multi-source learning**: the knowledge from one schema can be transferred to another schema without any mapping.
- **Time efficiency**: creating verbalizations is more time efficient than annotating examples.

Conclusions

- **Works out of the box**: the approach works without giving any example for training.
- **Few-shot is easy**: examples from the tasks can be automatically converted to Textual Entailment.
- **Multi-source learning**: the knowledge from one schema can be transferred to another schema without any mapping.
- **Time efficiency**: creating verbalizations is more time efficient than annotating examples.

Conclusions

- **Works out of the box**: the approach works without giving any example for training.
- **Few-shot is easy**: examples from the tasks can be automatically converted to Textual Entailment.
- **Multi-source learning**: the knowledge from one schema can be transferred to another schema without any mapping.
- **Time efficiency**: creating verbalizations is more time efficient than annotating examples.

Conclusions

- **Works out of the box**: the approach works without giving any example for training.
- **Few-shot is easy**: examples from the tasks can be automatically converted to Textual Entailment.
- **Multi-source learning**: the knowledge from one schema can be transferred to another schema without any mapping.
- **Time efficiency**: creating verbalizations is more time efficient than annotating examples.

Limitations

- **Need of span candidates**: the approach classifies spans or span pairs, therefore it requires a span candidate generation step.
- **Computation efficiency**: the approach requires to perform one inference per verbalization, which can be computationally expensive when the amount of verbalizations is large.
- **Lack of detail**: verbalizations are prototypical expressions of the type defined, but lack the details and exceptions that are present in the guidelines.
- **Definition disagreement**: type definition from one dataset to another can vary, and sometimes it cannot be expressed through simple verbalizations.

Limitations

- **Need of span candidates**: the approach classifies spans or span pairs, therefore it requires a span candidate generation step.
- **Computation efficiency**: the approach requires to perform one inference per verbalization, which can be computationally expensive when the amount of verbalizations is large.
- **Lack of detail**: verbalizations are prototypical expressions of the type defined, but lack the details and exceptions that are present in the guidelines.
- **Definition disagreement**: type definition from one dataset to another can vary, and sometimes it cannot be expressed through simple verbalizations.

Limitations

- **Need of span candidates**: the approach classifies spans or span pairs, therefore it requires a span candidate generation step.
- **Computation efficiency**: the approach requires to perform one inference per verbalization, which can be computationally expensive when the amount of verbalizations is large.
- **Lack of detail**: verbalizations are prototypical expressions of the type defined, but lack the details and exceptions that are present in the guidelines.
- **Definition disagreement**: type definition from one dataset to another can vary, and sometimes it cannot be expressed through simple verbalizations.

Limitations

- **Need of span candidates**: the approach classifies spans or span pairs, therefore it requires a span candidate generation step.
- **Computation efficiency**: the approach requires to perform one inference per verbalization, which can be computationally expensive when the amount of verbalizations is large.
- **Lack of detail**: verbalizations are prototypical expressions of the type defined, but lack the details and exceptions that are present in the guidelines.
- **Definition disagreement**: type definition from one dataset to another can vary, and sometimes it cannot be expressed through simple verbalizations.

CONTRIBUTIONS

INFORMATION EXTRACTION WITH LARGE LANGUAGE MODELS

Motivation

- Large Language Models (LLMs) have shown to be effective in several Natural Language Processing tasks, particularly in low-resource scenarios.
- Instruction tunning has shown that LLMs can learn to follow instructions to perform multiple tasks with a single model.
- But, current approaches for low-resource Information Extraction —based on generative LLMs or not— do not leverage detailed instructions about the task:
  - They fail when different datasets define a type differently.

Can we make use of detailed guidelines to improve the performance of LLMs in Information Extraction tasks?

Motivation

- Large Language Models (LLMs) have shown to be effective in several Natural Language Processing tasks, particularly in low-resource scenarios.
- Instruction tunning has shown that LLMs can learn to follow instructions to perform multiple tasks with a single model.
- But, current approaches for low-resource Information Extraction —based on generative LLMs or not— do not leverage detailed instructions about the task:
  - They fail when different datasets define a type differently.

Can we make use of detailed guidelines to improve the performance of LLMs in Information Extraction tasks?

Motivation

- Large Language Models (LLMs) have shown to be effective in several Natural Language Processing tasks, particularly in low-resource scenarios.
- Instruction tunning has shown that LLMs can learn to follow instructions to perform multiple tasks with a single model.
- But, current approaches for low-resource Information Extraction —based on generative LLMs or not— do not leverage detailed instructions about the task:
  - They fail when different datasets define a type differently.

Can we make use of detailed guidelines to improve the performance of LLMs in Information Extraction tasks?

Motivation

- Large Language Models (LLMs) have shown to be effective in several Natural Language Processing tasks, particularly in low-resource scenarios.
- Instruction tunning has shown that LLMs can learn to follow instructions to perform multiple tasks with a single model.
- But, current approaches for low-resource Information Extraction —based on generative LLMs or not— do not leverage detailed instructions about the task:
  - They fail when different datasets define a type differently.

Can we make use of detailed guidelines to improve the performance of LLMs in Information Extraction tasks?

Motivation

- Large Language Models (LLMs) have shown to be effective in several Natural Language Processing tasks, particularly in low-resource scenarios.
- Instruction tunning has shown that LLMs can learn to follow instructions to perform multiple tasks with a single model.
- But, current approaches for low-resource Information Extraction —based on generative LLMs or not— do not leverage detailed instructions about the task:
  - They fail when different datasets define a type differently.

Can we make use of detailed guidelines to improve the performance of LLMs in Information Extraction tasks?

Challenges

In order to teach LLMs to follow guidelines for Information Extraction tasks, we need to address two main challenges:

- How to properly represent the guideline definitions to be used by the LLM?
- How to avoid the LLM memorizing the tasks and not attending to the guidelines?

Input-output representation

- Labels are defined as Python classes.
- Guidelines are introduced as docstrings in the code.
- Representative candidates are introduced as comments.
- The text is introduced as a variable.
- The result is a list of instances.

```
@dataclass
class Metric(Entity):
  """Refers to evaluation metrics used to assess
  the performance of AI models and algorithms.
  Annotate specific metrics like Accuracy."""

  span:  str # Such as:  "mean squared error", ...

text = "The Information Extraction system was
evaluated using F1-Score."

result = [Metric(span="F1-score")]
```

Training regularization

LLMs can easily memorize the tasks by finding patterns in the input, and consecuently not attending to the guidelines. To avoid this, we propose the following regularizations:

- **Class order shuffling**: the order of the classes is shuffled for each instance.
- **Class dropout**: a random percentage of the classes is dropped for each instance and the output is changed accordingly.
- **Guideline paraphrasing**: the guidelines are paraphrased to avoid the model memorizing the definitions.
- **Representative candidate sampling**: the representative candidates are sampled from a pool of candidates.
- **Class name masking**: some class names are masked —replaced by `LABEL_1` — in the input.

Training regularization

LLMs can easily memorize the tasks by finding patterns in the input, and consecuently not attending to the guidelines. To avoid this, we propose the following regularizations:

- **Class order shuffling**: the order of the classes is shuffled for each instance.
- **Class dropout**: a random percentage of the classes is dropped for each instance and the output is changed accordingly.
- **Guideline paraphrasing**: the guidelines are paraphrased to avoid the model memorizing the definitions.
- **Representative candidate sampling**: the representative candidates are sampled from a pool of candidates.
- **Class name masking**: some class names are masked —replaced by `LABEL_1` — in the input.

Training regularization

LLMs can easily memorize the tasks by finding patterns in the input, and consecuently not attending to the guidelines. To avoid this, we propose the following regularizations:

- **Class order shuffling**: the order of the classes is shuffled for each instance.
- **Class dropout**: a random percentage of the classes is dropped for each instance and the output is changed accordingly.
- **Guideline paraphrasing**: the guidelines are paraphrased to avoid the model memorizing the definitions.
- **Representative candidate sampling**: the representative candidates are sampled from a pool of candidates.
- **Class name masking**: some class names are masked —replaced by `LABEL_1` — in the input.

Training regularization

LLMs can easily memorize the tasks by finding patterns in the input, and consecuently not attending to the guidelines. To avoid this, we propose the following regularizations:

- **Class order shuffling**: the order of the classes is shuffled for each instance.
- **Class dropout**: a random percentage of the classes is dropped for each instance and the output is changed accordingly.
- **Guideline paraphrasing**: the guidelines are paraphrased to avoid the model memorizing the definitions.
- **Representative candidate sampling**: the representative candidates are sampled from a pool of candidates.
- **Class name masking**: some class names are masked —replaced by `LABEL_1` — in the input.

Training regularization

LLMs can easily memorize the tasks by finding patterns in the input, and consecuently not attending to the guidelines. To avoid this, we propose the following regularizations:

- **Class order shuffling**: the order of the classes is shuffled for each instance.
- **Class dropout**: a random percentage of the classes is dropped for each instance and the output is changed accordingly.
- **Guideline paraphrasing**: the guidelines are paraphrased to avoid the model memorizing the definitions.
- **Representative candidate sampling**: the representative candidates are sampled from a pool of candidates.
- **Class name masking**: some class names are masked —replaced by `LABEL_1` — in the input.

Research Questions

- Are guidelines helpful when data is available? And when it is not?
- How do the guidelines affect seen and unseen labels?
- Where do the errors come from?
- Which are the remaining challenges?

Experimental Setup

- We fine-tuned CodeLlama models **with and without using guidelines**.
- We divided our pool of datasets into train and eval:
  - For training we used 9 datasets from *News* and *Biomedical* domains.
  - For evaluation we used 10 datasets from *News*, *Biomedical*, *Cybercrime*, *Wikipedia* and more domains.

Baseline

```
@dataclass
class VulnerabilityPatch(Event):

  mention: str
  cve: List[str]
  issues: List[str]
  platforms: List[str]
  vulnerability: List[str]
  releaser: List[str]
```

GoLLIE

```
@dataclass
class VulnerabilityPatch(Event):
  """A VulnerabilityPatch Event happens when a
  software company addresses a known vulnerability
  by releasing or describing an update."""
  mention: str
  """The text span that triggers the event.
  Such as: patch, fixed, addresses, implemented"""
  cve: List[str] # The vulnerability identifier
  issues: List[str] # What did the patch fix
  platforms: List[str] # The platforms that ...
  vulnerability: List[str] # The vulnerability
  releaser: List[str] # Entity releasing the patch
```

Train and evaluation data

- Some domains are kept for evaluation only.
- Datasets from different tasks are used:
  - Named Entity Recognition.
  - Relation Extraction.
  - Event Extraction.
  - Event Argument Extraction.
  - Slot Filling.

Dataset	Domain	NER	RE	EE	EAE	SF	Training	Evaluation
ACE05	News	✓	✓	✓	✓		✓	✓
BC5CDR	Biomedical	✓					✓	✓
CoNLL 2003	News	✓					✓	✓
DIANN	Biomedical	✓					✓	✓
NCBIDisease	Biomedical	✓					✓	✓
Ontonotes 5	News	✓					✓	✓
RAMS	News				✓		✓	✓
TACRED	News					✓	✓	✓
WNUT 2017	News	✓					✓	✓
BroadTwitter	Twitter	✓						✓
CASIE	Cybercrime			✓	✓			✓
CrossNER	*Many*	✓						✓
E3C	Biomedical	✓						✓
FabNER	Science	✓						✓
HarveyNER	Twitter	✓						✓
MIT Movie	Queries	✓						✓
MIT Restaurants	Queries	✓						✓
MultiNERD	Wikipedia	✓						✓
WikiEvents	Wikipedia	✓		✓	✓			✓

Table. Datasets used on the experiments.

Evaluation results

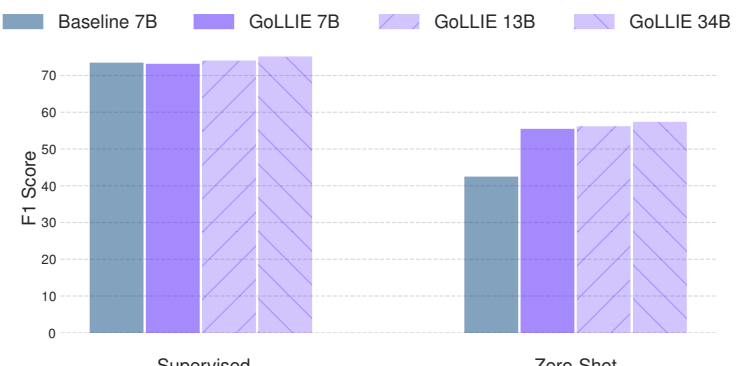


Figure. Supervised and Zero-Shot results.

- In the supervised setup, adding guidelines has little to no effect.
- In the Zero-Shot setup, however, adding guidelines improves the performance by 13 F1 points.

Research Questions

- Are guidelines helpful when data is available? And when it is not?
- How do the guidelines affect seen and unseen labels?
- Where do the errors come from?
- Which are the remaining challenges?

Experimental Setup

- We categorize the labels into **seen** and **unseen** based on the labels used for training.
- We recomputed the F1-score average according to that categorization.

Seen vs Unseen results

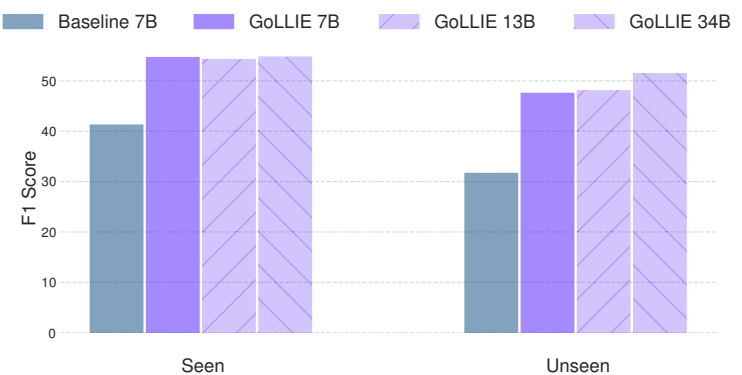


Figure. Seen vs Unseen Zero-Shot results.

- In the case of seen labels, the baseline still struggles mainly due to label definition shifts.
- For the unseen labels, GoLLIE is highly superior to the baseline.
- Interestingly, the gap between seen and unseen becomes smaller with the size of the model.

Research Questions

- Are guidelines helpful when data is available? And when it is not?
- How do the guidelines affect seen and unseen labels?
- Where do the errors come from?

Error analisis

Dataset	Label	Guideline	Baseline	GoLLIE
MultiNERD	Media	Titles of films, books, songs, albums, fictional characters, and languages.	13.6	69.1
CASIE	Vul. Patch	When a software company addresses a vulnerability by releasing an update.	27.7	70.5
Movie	Trailer	Refers to a short promotional video or preview of a movie.	00.0	76.4
AI	Task	Particular research task or problem within a specific AI research field.	02.7	63.9
MultiNERD	Time	Specific and well-defined time intervals, such as eras, historical periods, centuries, years and important days.	01.4	03.5
Movie	Plot	Recurring concept, event, or motif that plays a significant role in the development of a movie.	00.4	05.1
AI	Misc	Named entities that are not included in any other category.	01.1	05.2
Literature	Misc	Named entities that are not included in any other category.	03.7	30.8
Literature	Writer	Individual actively engaged in the creation of literary works.	04.2	65.1
Literature	Person	Person name that is not a writer.	33.5	49.4
Science	Scientist	A person who is studying or has expert knowledge of a natural science field.	02.1	05.8
Science	Person	Person name that is not a scientist.	46.1	45.9
Politics	Polit. Party	Organization that compete in a particular country's elections.	11.2	34.9

- Green rows represent helpful guidelines.

Error analisis

Dataset	Label	Guideline	Baseline	GoLLIE
MultiNERD	Media	Titles of films, books, songs, albums, fictional characters, and languages.	13.6	69.1
CASIE	Vul. Patch	When a software company addresses a vulnerability by releasing an update.	27.7	70.5
Movie	Trailer	Refers to a short promotional video or preview of a movie.	00.0	76.4
AI	Task	Particular research task or problem within a specific AI research field.	02.7	63.9
MultiNERD	Time	Specific and well-defined time intervals, such as eras, historical periods, centuries, years and important days.	01.4	03.5
Movie	Plot	Recurring concept, event, or motif that plays a significant role in the development of a movie.	00.4	05.1
AI	Misc	Named entities that are not included in any other category.	01.1	05.2
Literature	Misc	Named entities that are not included in any other category.	03.7	30.8
Literature	Writer	Individual actively engaged in the creation of literary works.	04.2	65.1
Literature	Person	Person name that is not a writer.	33.5	49.4
Science	Scientist	A person who is studying or has expert knowledge of a natural science field.	02.1	05.8
Science	Person	Person name that is not a scientist.	46.1	45.9
Politics	Polit. Party	Organization that compete in a particular country's elections.	11.2	34.9

- Blue rows represent datasets that do not follow their annotation guidelines.

Error analisis

Dataset	Label	Guideline	Baseline	GoLLIE
MultiNERD	Media	Titles of films, books, songs, albums, fictional characters, and languages.	13.6	69.1
CASIE	Vul. Patch	When a software company addresses a vulnerability by releasing an update.	27.7	70.5
Movie	Trailer	Refers to a short promotional video or preview of a movie.	00.0	76.4
AI	Task	Particular research task or problem within a specific AI research field.	02.7	63.9
MultiNERD	Time	Specific and well-defined time intervals, such as eras, historical periods, centuries, years and important days.	01.4	03.5
Movie	Plot	Recurring concept, event, or motif that plays a significant role in the development of a movie.	00.4	05.1
AI	Misc	Named entities that are not included in any other category.	01.1	05.2
Literature	Misc	Named entities that are not included in any other category.	03.7	30.8
Literature	Writer	Individual actively engaged in the creation of literary works.	04.2	65.1
Literature	Person	Person name that is not a writer.	33.5	49.4
Science	Scientist	A person who is studying or has expert knowledge of a natural science field.	02.1	05.8
Science	Person	Person name that is not a scientist.	46.1	45.9
Politics	Polit. Party	Organization that compete in a particular country's elections.	11.2	34.9

- Red rows represent labels that have poorly defined guidelines.

Conclusions and Future Work

In this thesis, we have made the following contributions:

- Zero- and Few-Shot Information Extraction is possible thanks to a textual representation.
- The Textual Entailment data is key for the performance.
- Knowledge transfer between schemas is possible without any mapping.
- It is robust to different verbalization styles, and it is more efficient than annotating examples.

Additionally, annotation guidelines are needed to:

- Help the model with unknown and unseen labels.
- Address the definition shift from different datasets.

In this thesis, we have made the following contributions:

- Zero- and Few-Shot Information Extraction is possible thanks to a textual representation.
- The Textual Entailment data is key for the performance.
- Knowledge transfer between schemas is possible without any mapping.
- It is robust to different verbalization styles, and it is more efficient than annotating examples.

Additionally, annotation guidelines are needed to:

- Help the model with unknown and unseen labels.
- Address the definition shift from different datasets.

Estimation of quality and gold standards

- The field is moving towards **dynamic and preference evaluations**.
- In Information Extraction gold-standards suffer from annotation errors and low-agreement rates.
- Maybe there can be better ways to evaluate the models.

Document level Information Extraction

- LLMs unlock the opportunity to work with large documents.
- The current approaches are limited to sentence-level or paragraph-level, which do not capture the whole picture of the task.
- Working on document level requires to **address challenges that are largely unexplored**.

Estimation of quality and gold standards

- The field is moving towards **dynamic and preference evaluations**.
- In Information Extraction gold-standards suffer from annotation errors and low-agreement rates.
- Maybe there can be better ways to evaluate the models.

Document level Information Extraction

- LLMs unlock the opportunity to work with large documents.
- The current approaches are limited to sentence-level or paragraph-level, which do not capture the whole picture of the task.
- Working on document level requires to **address challenges that are largely unexplored**.

Papers that are part of this thesis

- **Sainz O.**, Lopez de Lacalle O., Labaka G., Barrena A., and Agirre E. Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction. (EMNLP 2021)
- **Sainz O.**, Gonzalez-Dios I., Lopez de Lacalle O., Min B., and Agirre E. Textual Entailment for Event Argument Extraction: Zero- and Few-Shot with Multi-Source Learning. (NAACL-Findings 2022)
- **Sainz O.**, Qiu H., Lopez de Lacalle O., Agirre E., and Min B. ZS4IE: A toolkit for Zero-Shot Information Extraction with simple Verbalizations. (NAACL 2022)
- **Sainz O.**, García-Ferrero I., Agerri R., Lopez de Lacalle O., Rigau G., and Agirre E. GoLLIE: Annotation guidelines improve zero-shot information-extraction. (ICLR 2024)

Papers that are not part of this thesis

- **Sainz O.** and Rigau G. Ask2Transformers: Zero-shot domain labelling with pretrained language models. (GWC 2021)
- **Sainz O.,** Lopez de Lacalle O., Agirre E., and Rigau G. What do language models know about word senses? zero-shot WSD with language models and domain inventories. (GWC 2023)
- Min B., Ross H., Sulem E., Veyseh A.P.B., Nguyen T.H., **Sainz O.**, Agirre E., Heintz I., and Roth D. Recent advances in natural language processing via large pre-trained language models: A survey (ACM Computing Surveys 2023)
- García-Ferrero I., Campos J.A., **Sainz O.**, Salaberria A., and Roth D. IXA/cog-comp at SemEval-2023 task 2: Context-enriched multilingual named entity recognition using knowledge bases (SemEval 2023)
- **Sainz O.**, Campos J., García-Ferrero I., Etxaniz J., de Lacalle O.L., and Agirre E. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark (EMNLP-Findings 2023)
- Zubillaga M., **Sainz O.**, Estarrona A., Lopez de Lacalle O., Agirre A. Event Extraction in Basque: Typologically motivated Cross-Lingual Transfer-Learning Analysis (LREC-Coling 2024)
- Etxaniz J., **Sainz O.**, Perez N., Aldabe I., Rigau G., Agirre E., Ormazabal A., Artetxe M., Soroa A. Latxa: An Open Langauge Model and Evaluation Suite for Basque. (ACL 2024)

IKASKETA-ADIBIDE URRIKO INFORMAZIO-ERAUZKETA

LOW-RESOURCE INFORMATION EXTRACTION

Oscar Sainz Jimenez

Supervised by **Eneko Agirre** and **Oier Lopez de Lacalle**

HiTZ Zentroa - Ixa taldea
Euskal Herriko Unibertsitatea UPV/EHU

PhD Dissertation
July 15, 2024

Universidad del País Vasco Euskal Herriko Unibertsitatea

HiTZ
Hizkuntza Teknologiako Zentroa
Basque Center for Language Technology