Università degli Studi di Milano Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea Magistrale in Informatica

Neuro-Symbolic Fine-grained Entity Typing with KENN

Relatore: *Prof. Matteo Palmonari*
Co-relatore: *Dott. Manuel Vimercati*

Tesi di Laurea Magistrale di:
Christian Bernasconi
Matricola 816423

Anno Accademico 2020-2021

Background and Motivation

Background - From Entity Typing...

Entity Typing (ET):

- multilabel classification problem
- small type sets (e.g., *Person, Organization, Location*)

Example: *“The former President* ***<Barack Obama>*** *was born in Honolulu”*

left context | mention | right context

Type Set

Person | Organization | Location

Background - ...to Fine-grained Entity Typing

Entity Typing (ET):

- multilabel classification problem
- small type sets (e.g., *Person, Organization, Location*)

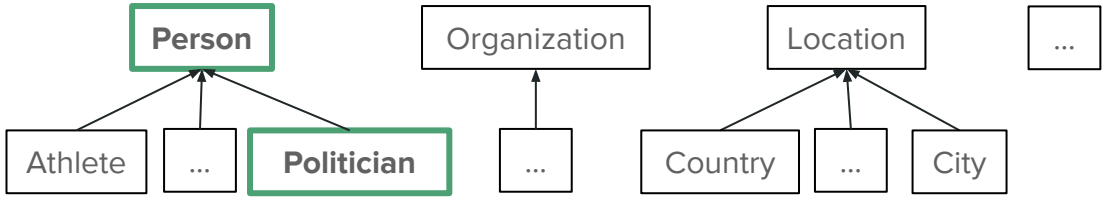
Fine-grained Entity Typing (FET):

- ET with larger and more complex type sets (e.g., *Person, Athlete, Actor, Politician, Cyclist, ...*)
- type sets usually organized hierarchically

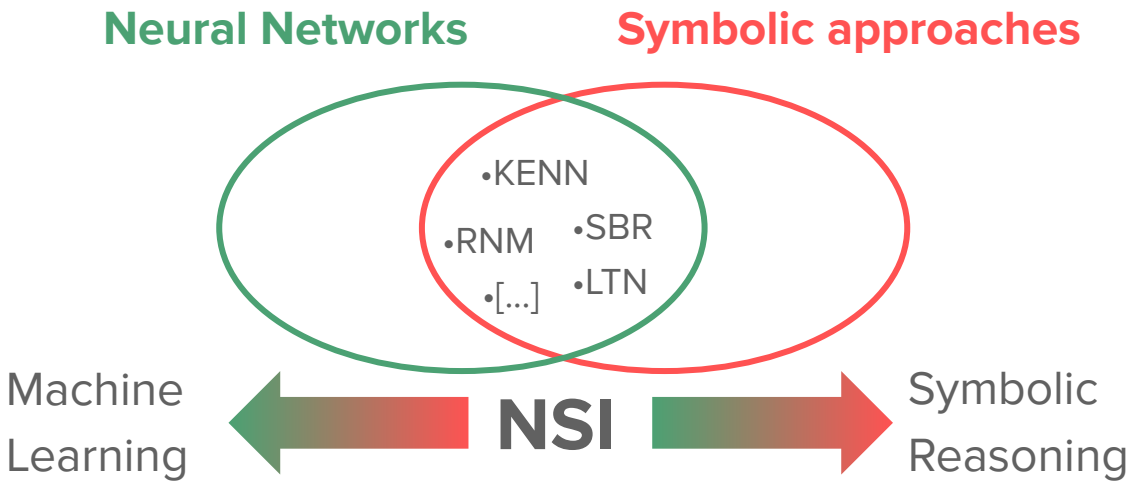
Example: *"The former President* ***<Barack Obama>*** *was born in Honolulu"*

left context | mention | right context

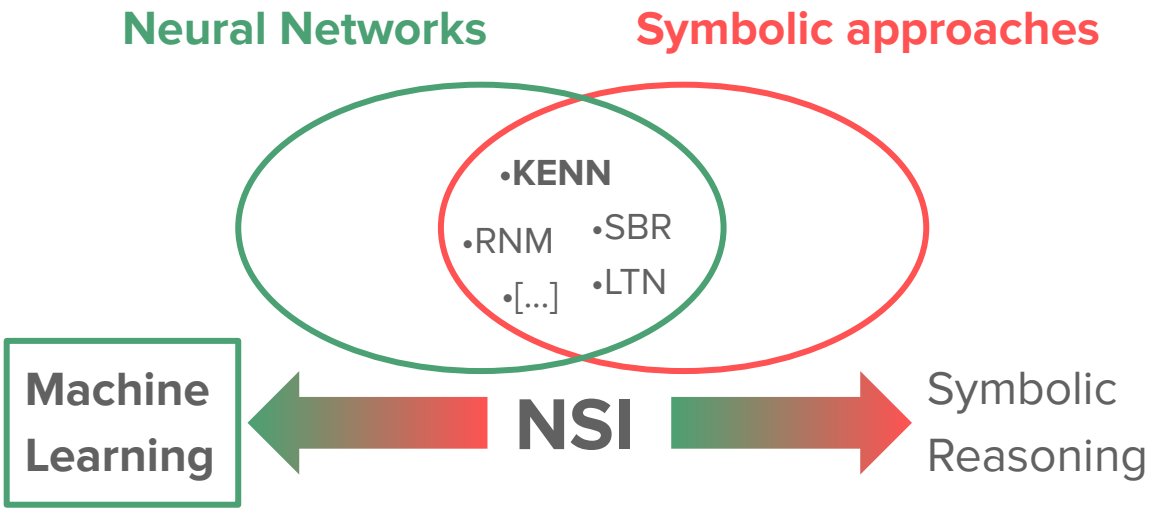
Type Set Hierarchy



Background - Neuro-Symbolic Integration (NSI)



Background - Neuro-Symbolic Integration (NSI)



KENN → injects knowledge into the learning process in **multilabel** scenarios through logical clauses

Motivation

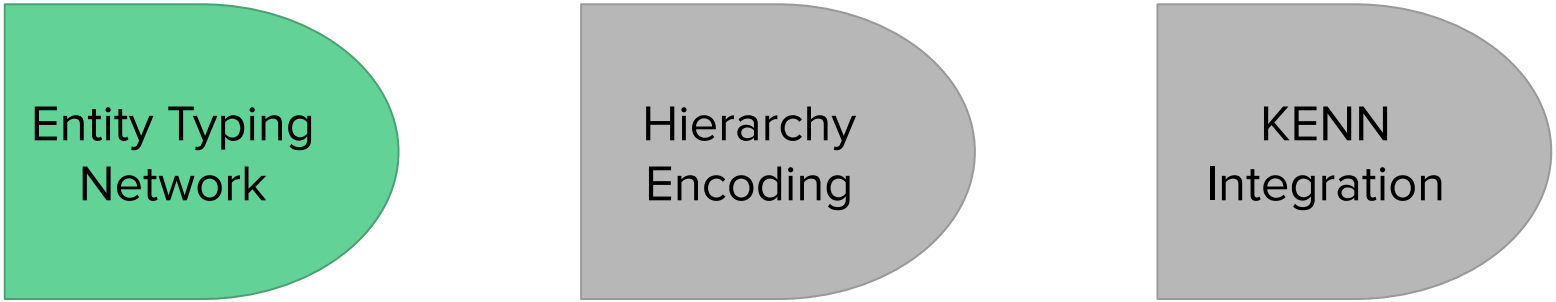
Goal: Use NSI to exploit hierarchical information in a FET problem

Research Questions:

- ***RQ1:*** *How can we encode hierarchical information through logic?*
- ***RQ2:*** *How does KENN behave with different encodings?*
- ***RQ3:*** *What are the benefits of using KENN in FET?*

Build a Neuro-Symbolic Network for FET

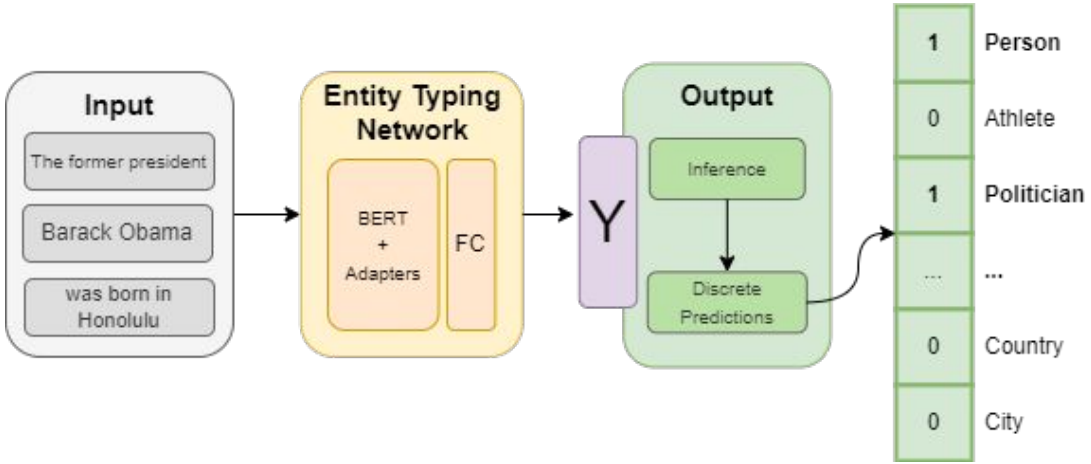
Build a Neuro-Symbolic Network for FET



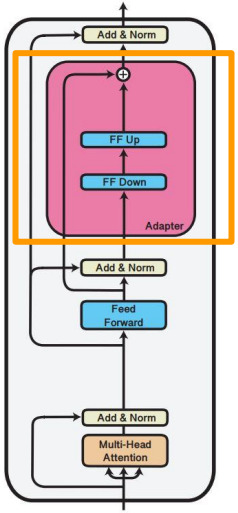
Entity Typing Network

- *Transformer-based* Language Model to encode the sentence (e.g., BERT)
  - Add ***Adapters*** and freeze Transformer's parameters
- Fully Connected Layer + Classification Layer

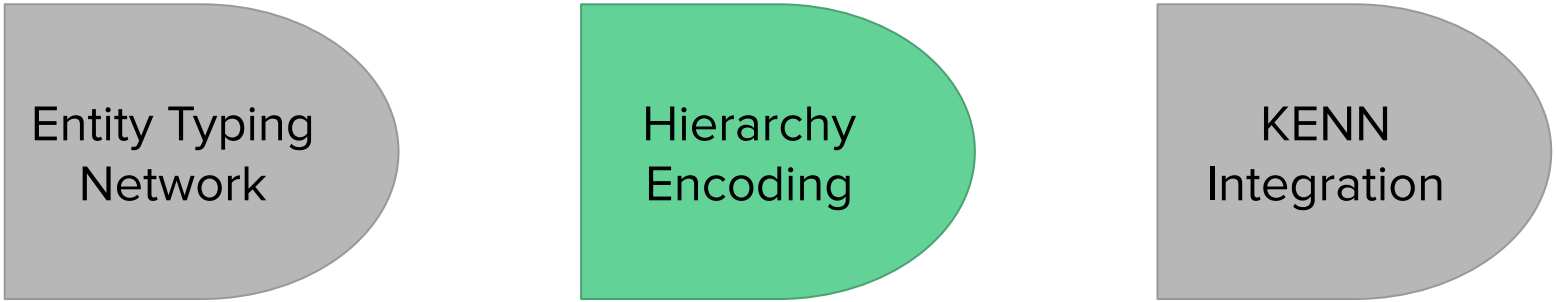
Following the literature, Adapters are introduced to reduce the number of parameters to optimize



Transformer encoder

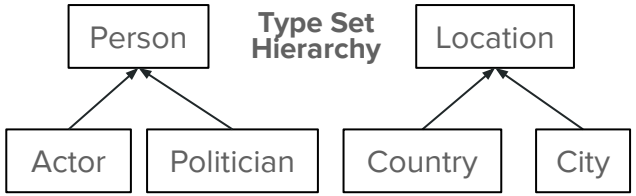


Build a Neuro-Symbolic Network for FET



Encoding a Hierarchy for FET

We can define different strategies to encode a First-Order Logic (FOL) Knowledge Base (KB) starting from a hierarchy:



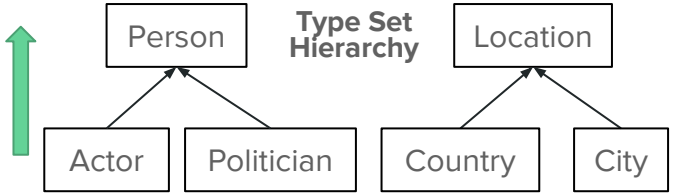
Encoding a Hierarchy for FET

We can define different strategies to encode a First-Order Logic (FOL) Knowledge Base (KB) starting from a hierarchy:

- **Bottom-Up:**

➔ propagate information from *subtype* to *supertype*

$\forall x.\ Subtype(x) \rightarrow Supertype(x)$



Bottom-Up KB:
Actor(x) ➔ Person(x)
Politician(x) ➔ Person(x)
Country(x) ➔ Location(x)
City(x) ➔ Location(x)

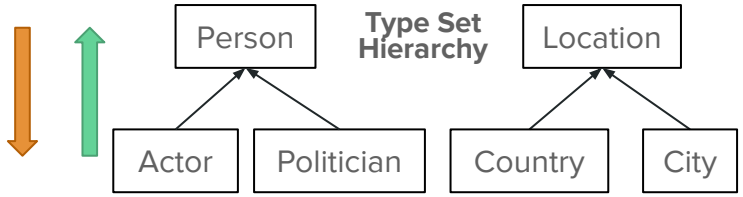
Encoding a Hierarchy for FET

We can define different strategies to encode a First-Order Logic (FOL) Knowledge Base (KB) starting from a hierarchy:

- **Bottom-Up**: *subtype* ➔ *supertype*
- **Top-Down:**

➔ propagate information from *supertype* to *subtypes*

$\forall x.\ Supertype(x) \rightarrow Subtype_1(x) \vee \ldots \vee Subtype_n(x)$



Bottom-Up KB:
Actor(x) ➔ Person(x)
Politician(x) ➔ Person(x)
Country(x) ➔ Location(x)
City(x) ➔ Location(x)

Top-Down KB:
Person(x) ➔ Actor(x) v Politician(x)
Location(x) ➔ Country(x) v City(x)

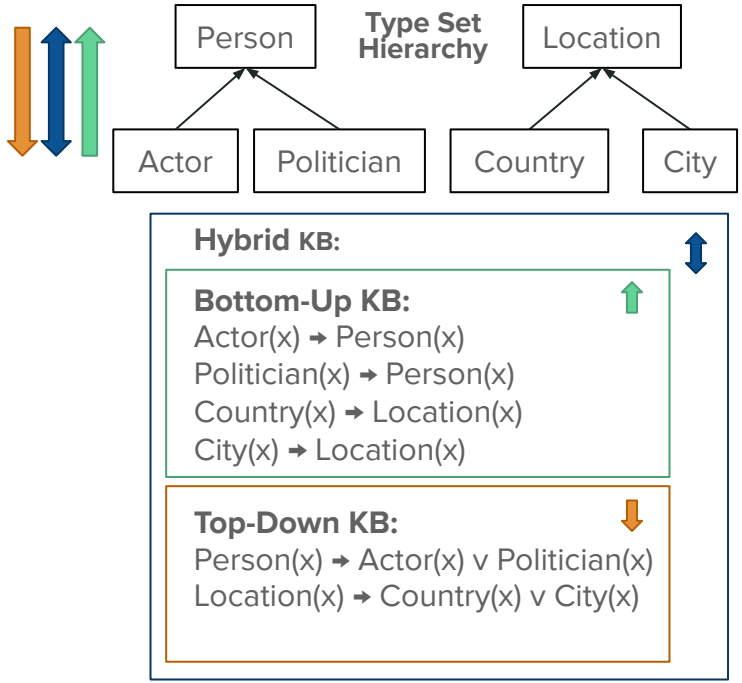
Encoding a Hierarchy for FET

We can define different strategies to encode a First-Order Logic (FOL) Knowledge Base (KB) starting from a hierarchy:

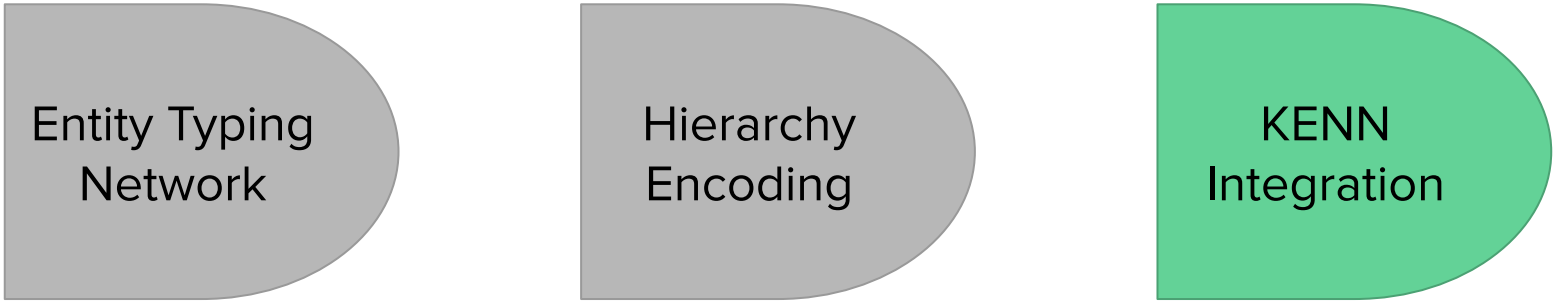
- **Bottom-Up**: *subtype ➔ supertype*
- **Top-Down:** *supertype ➔ subtypes*
- **Hybrid:**

➔ propagate information in both directions

Bottom-Up + Top-Down

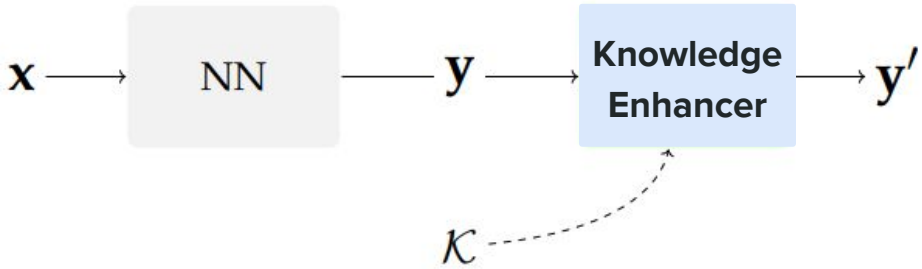


Build a Neuro-Symbolic Network for FET



KENN: Knowledge Enhanced Neural Networks

KENN is a NSI framework to inject prior knowledge into a neural network through an additional layer.

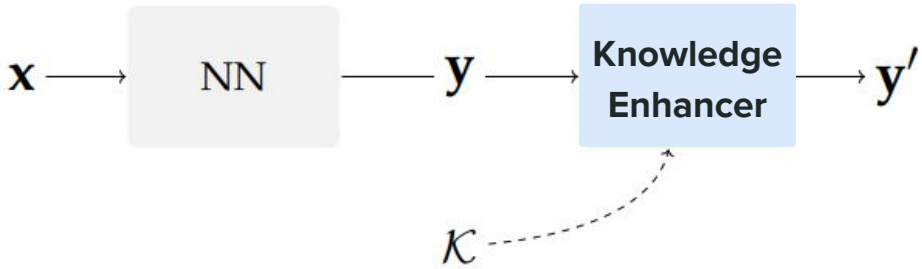


The knowledge is expressed in restricted **FOL** with **fuzzy** semantics:

- disjunction and negation
- unary and binary predicates
- function-free
- universal closure

KENN: Knowledge Enhanced Neural Networks

KENN is a NSI framework to inject prior knowledge into a neural network through an additional layer.



The knowledge is expressed in restricted **FOL** with **fuzzy** semantics:

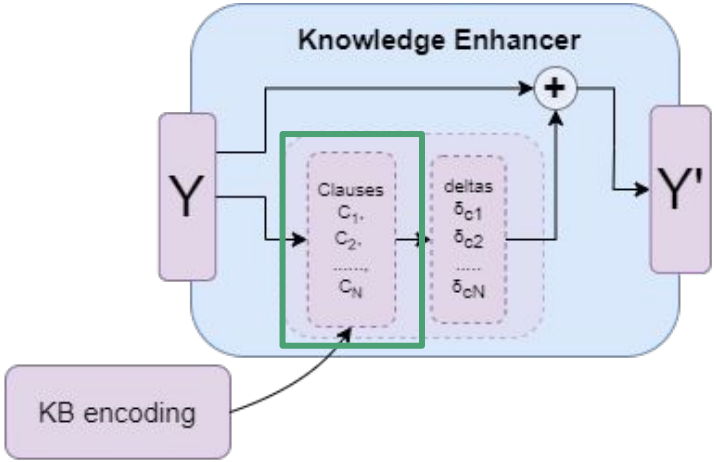
- disjunction and negation
- unary and binary predicates
- function-free
- universal closure

$$\forall x.\ Politician(x) \rightarrow Person(x)$$

$$\Downarrow$$

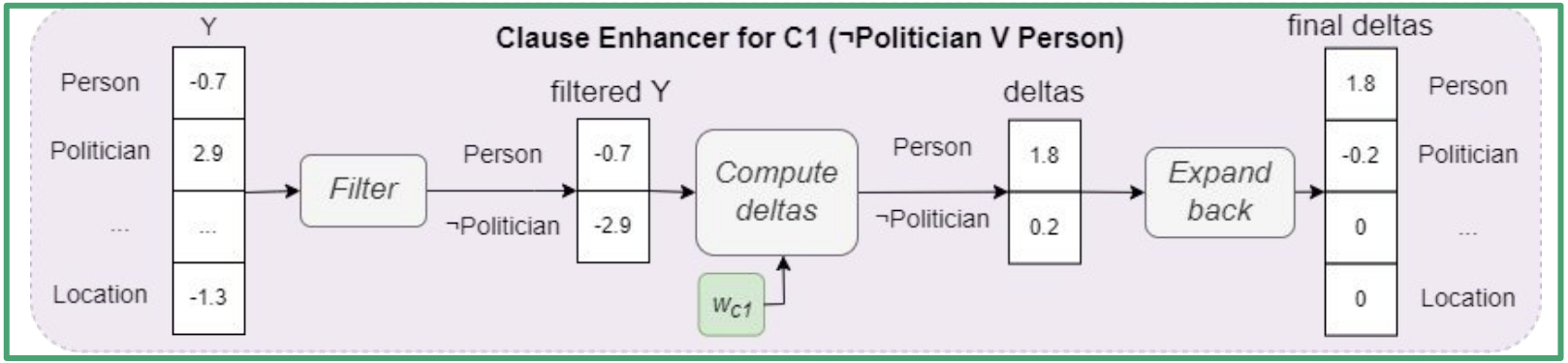
$$\neg Politician(x) \vee Person(x)$$

KENN - Knowledge Enhancement

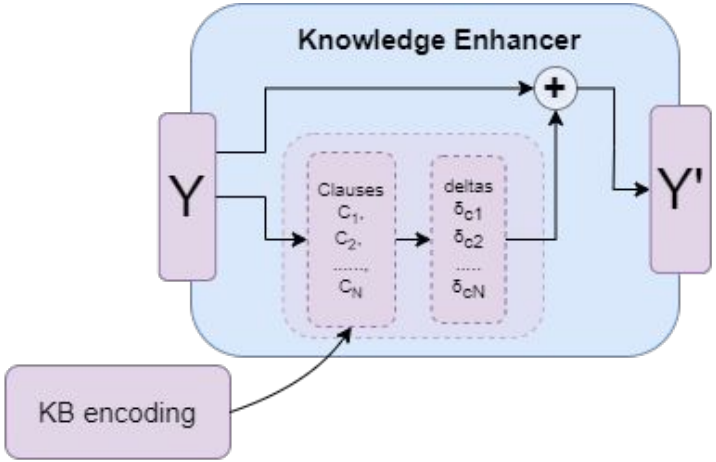


Basics:

1. Given a **KB** composed of *n* clauses, the **Knowledge Enhancer** instantiates *n* **Clause Enhancers** that are optimized independently

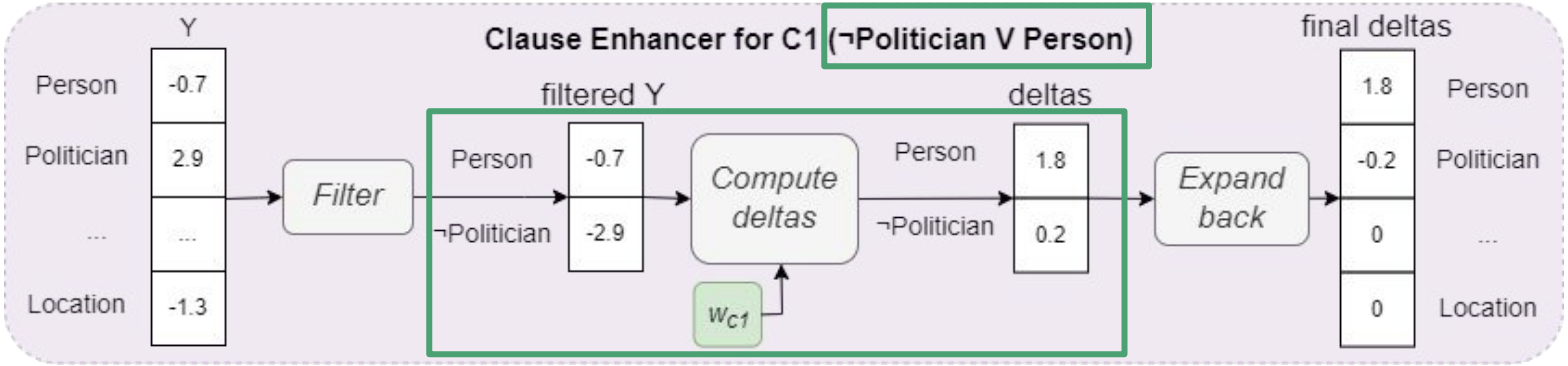


KENN - Knowledge Enhancement

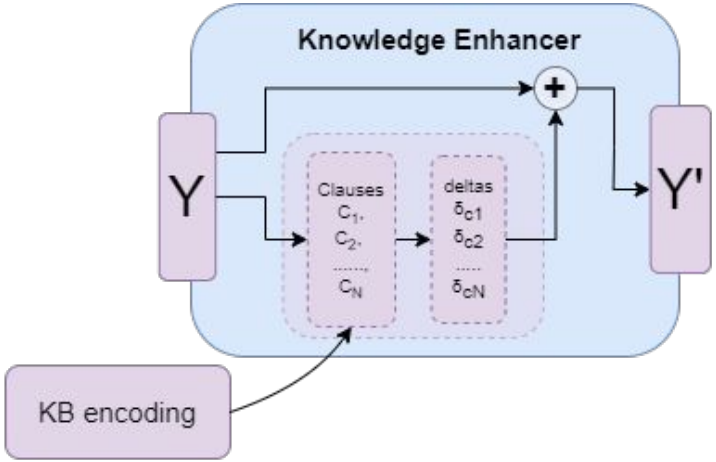


Basics:

2. Each **Clause Enhancer** produces variations (**deltas**) only for the types involved in the associated clause



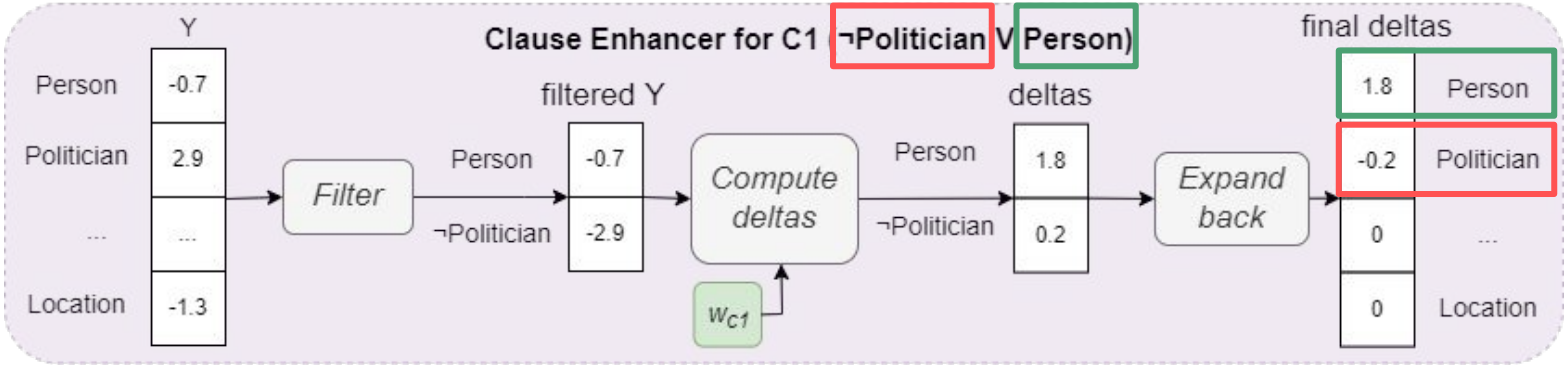
KENN - Knowledge Enhancement



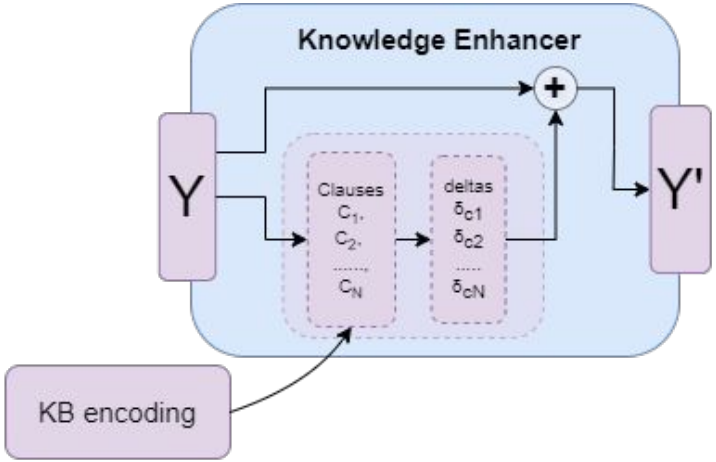
Basics:

2. Each **Clause Enhancer** produces variations (**deltas**) only for the types involved in the associated clause

   Note: *sign(delta) == sign(literal)*

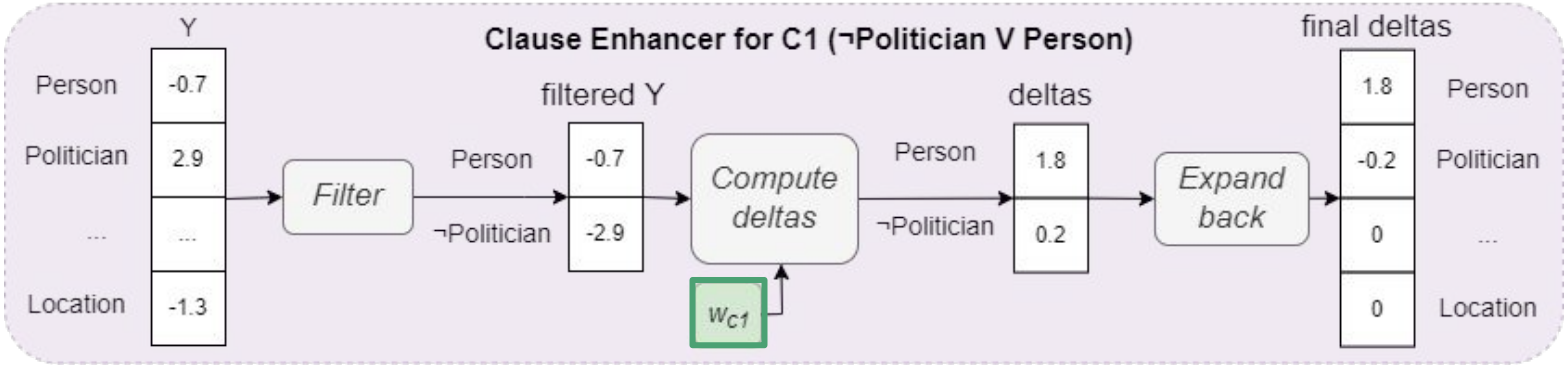


KENN - Knowledge Enhancement

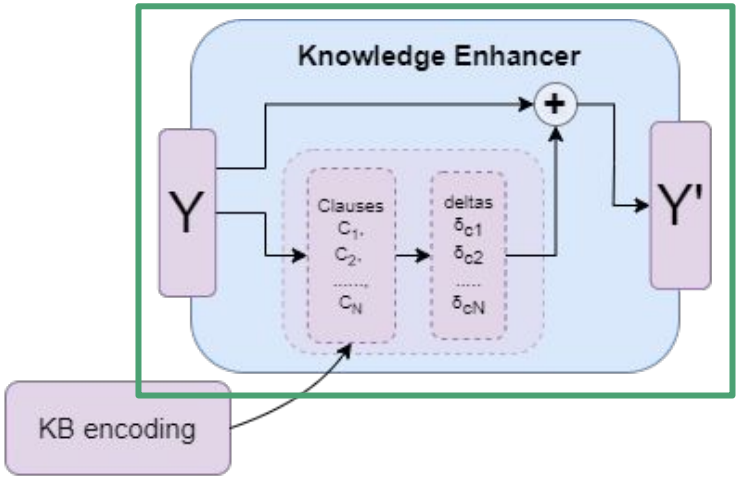


Basics:

3. The influence of each **Clause Enhancer** on *Y'* is regulated by a **clause weight**, which can be *fixed* or *learned*



KENN - Knowledge Enhancement



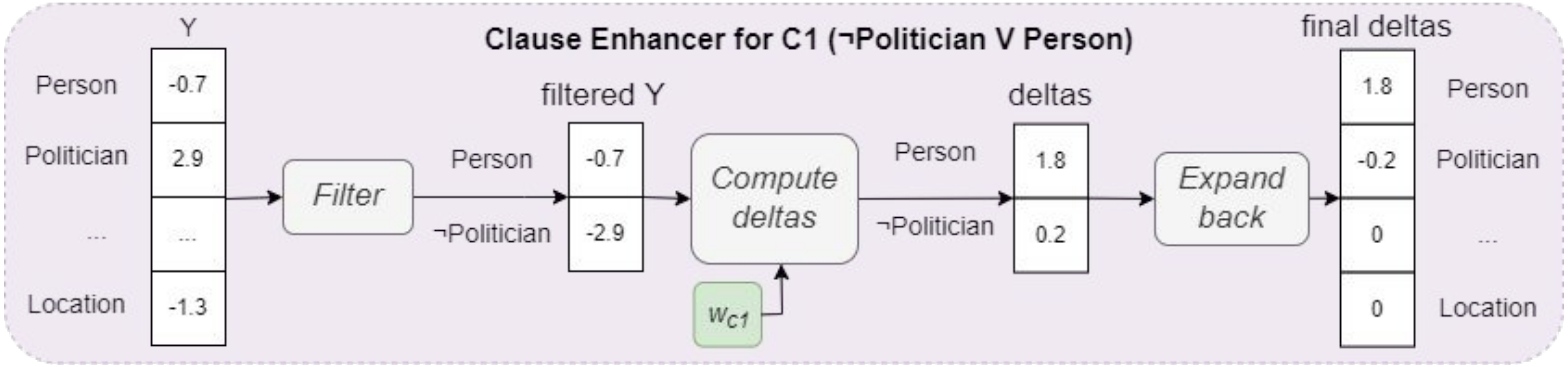
Basics:

4. Aggregate the deltas returned by all clause enhancers and sum them to *Y* to obtain *Y'*

Example:

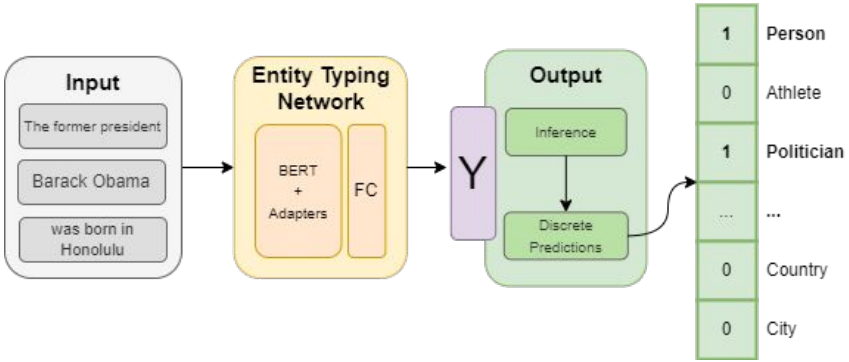
Person: *σ(Y) = 0.33* ⇒ *σ(Y') = 0.75*

Politician: *σ(Y) = 0.95* ⇒ *σ(Y') = 0.94*

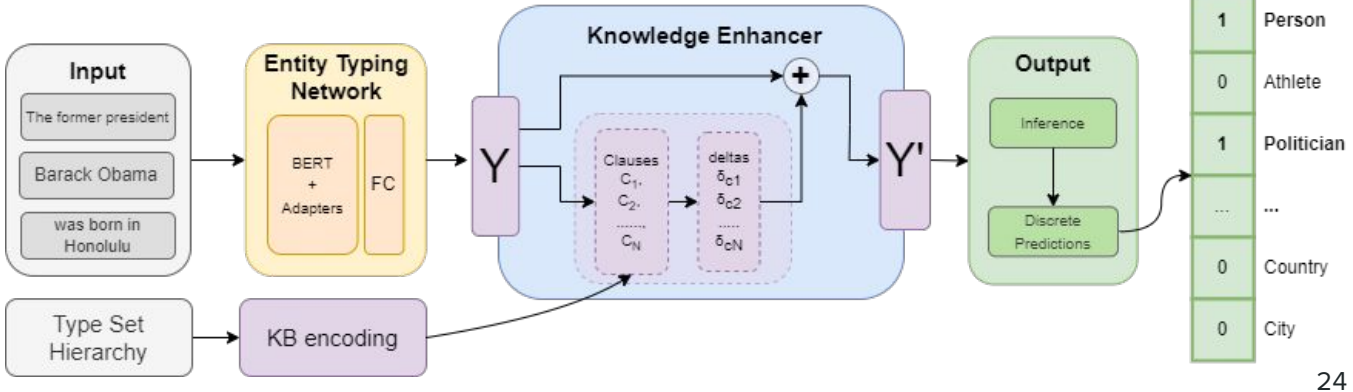


Neuro-Symbolic Fine-grained Entity Typing with KENN

Entity Typing Network



Neuro-Symbolic Entity Typing Network



Experiments on FIGER and BBN

Preliminary Experiments - Hyperparameters and KB

Experimental setup:

- **encoder:** DistilBERT
- **KB:** Bottom-Up, Top-Down, Hybrid
- **initial clause weight:** 0.5, 1.0, 2.0
- **non-learnable clause weights**

} 9 KENN configurations + baseline model

Results: KENN produces an initial boost regardless the hyperparameters and the KB

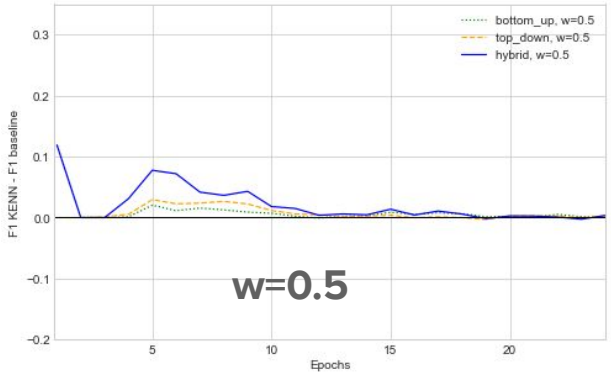
Preliminary Experiments - Hyperparameters and KB

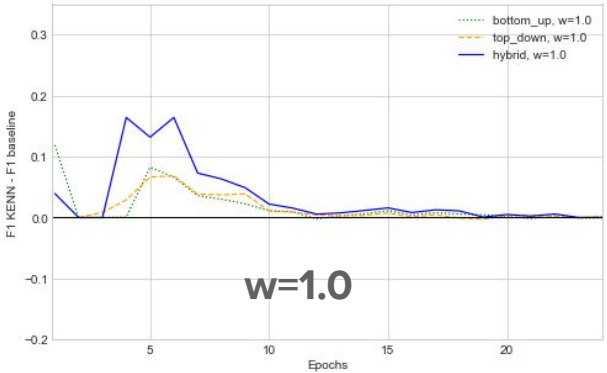
Experimental setup:

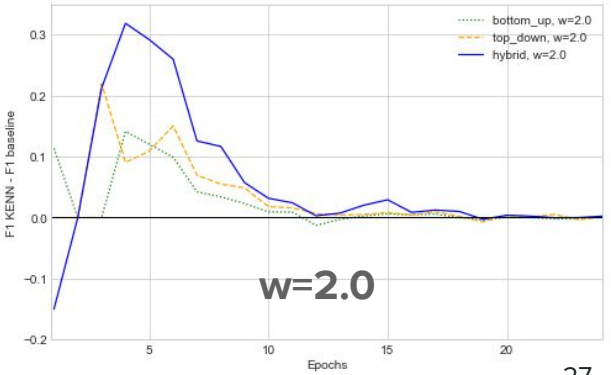
- **encoder:** DistilBERT
- **KB:** Bottom-Up, Top-Down, Hybrid
- **initial clause weight:** 0.5, 1.0, 2.0
- **non-learnable clause weights**

} 9 KENN configurations + baseline model

Results: KENN produces an initial boost regardless the hyperparameters and the KB







Analysis - Entity Typing Network Adaptation

Premises:

1. each *subtype* always co-occurs with its *supertype* (e.g., Politician=1 ➔ Person=1) ➔ **Forbidden Prediction (FP):** prediction that contradicts this pattern
2. the action of KENN is strictly dependent on the KB encoding

Number of FP before KENN for each KB:

KB	% FP	Effect on Supertype	Effect on Subtype
None	0.01	-	-
Bottom-Up	0.33	increase	decrease
Top-Down	0	decrease	increase
Hybrid	0.13	increase or decrease	increase or decrease

Analysis - Entity Typing Network Adaptation

Premises:

1. each *subtype* always co-occurs with its *supertype* (e.g., Politician=1 ➔ Person=1) ➔ **Forbidden Prediction (FP):** prediction that contradicts this pattern
2. the action of KENN is strictly dependent on the KB encoding

Number of FP before KENN for each KB:

KB	% FP	Effect on Supertype	Effect on Subtype
None	0.01	-	-
Bottom-Up	0.33	increase	decrease
Top-Down	0	decrease	increase
Hybrid	0.13	increase or decrease	increase or decrease

Observations:

1. Information about the hierarchy can be learned from data without KENN

Analysis - Entity Typing Network Adaptation

Premises:

1. each *subtype* always co-occurs with its *supertype* (e.g., Politician=1 ➔ Person=1)
   ➔ **Forbidden Prediction (FP):** prediction that contradicts this pattern
2. the action of KENN is strictly dependent on the KB encoding

Number of FP before KENN for each KB:

KB	% FP	Effect on Supertype	Effect on Subtype
None	0.01	-	-
Bottom-Up	0.33	increase	decrease
Top-Down	0	decrease	increase
Hybrid	0.13	increase or decrease	increase or decrease

Observations:

2. The **desired action** of KENN to fix an FP is to *increase supertype* and/or *decrease subtype*, so:
   - ➔ Bottom-Up can fix a FP
   - ➔ Hybrid can fix a FP
   - ➔ Top-Down cannot fix a FP

Analysis - Entity Typing Network Adaptation

Premises:

1. each *subtype* always co-occurs with its *supertype* (e.g., Politician=1 ➔ Person=1) ➔ **Forbidden Prediction (FP):** prediction that contradicts this pattern
2. the action of KENN is strictly dependent on the KB encoding

Number of FP before KENN for each KB:

KB	% FP	Effect on Supertype	Effect on Subtype
None	0.01	-	-
Bottom-Up	0.33	increase	decrease
Top-Down	0	decrease	increase
Hybrid	0.13	increase or decrease	increase or decrease

Observations:

3. While with Bottom-Up and Hybrid KBs the Entity Typing Network can rely on KENN to fix an FP, with Top-Down KB it has to avoid this situation by itself

Analysis - Entity Typing Network Adaptation

Premises:

1. each *subtype* always co-occurs with its *supertype* (e.g., Politician=1 ➔ Person=1) ➔ **Forbidden Prediction (FP):** prediction that contradicts this pattern
2. the action of KENN is strictly dependent on the KB encoding

Number of FP before KENN for each KB:

KB	% FP	Effect on Supertype	Effect on Subtype
None	0.01	-	-
Bottom-Up	0.33	increase	decrease
Top-Down	0	decrease	increase
Hybrid	0.13	increase or decrease	increase or decrease

Conclusion:

The Entity Typing Network adapts its predictions based on the *perceived* action of KENN

Model Refinement - Multiloss Training

Idea:

Force the Entity Typing Network to exploit the KB avoiding *adaptation*

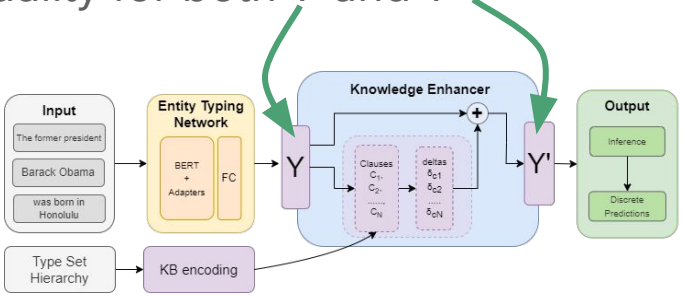
Intuition:

Optimize the whole network to preserve a discrete quality for both *Y* and *Y'*

➜ Multiloss function:

$$L(Y, Y', Y_t) = \alpha \cdot BCE(Y, Y_t) + (1 - \alpha) \cdot BCE(Y', Y_t)$$

*with α=0.5

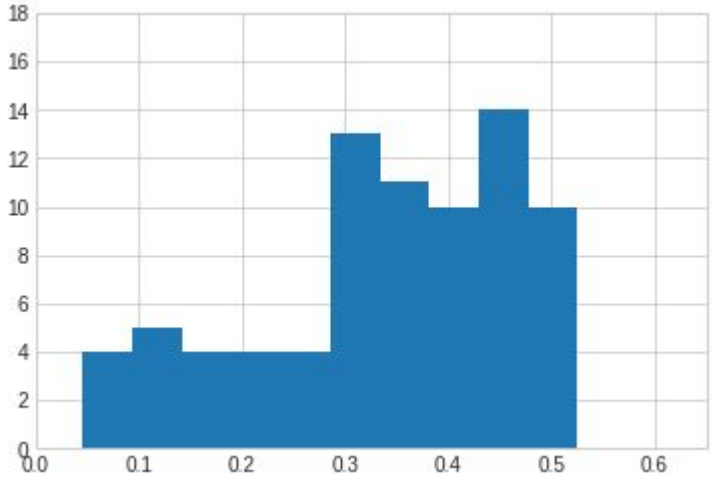


Experimental setup:

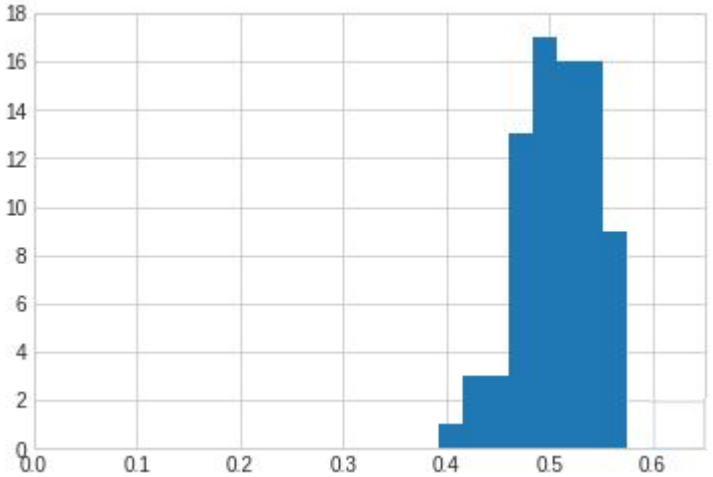
The clause weights have been initialized to *0.5* and set as *learnable parameters* to start with a soft influence and observe the weights evolution

Multiloss Training - Weight Changes

BCE multiloss



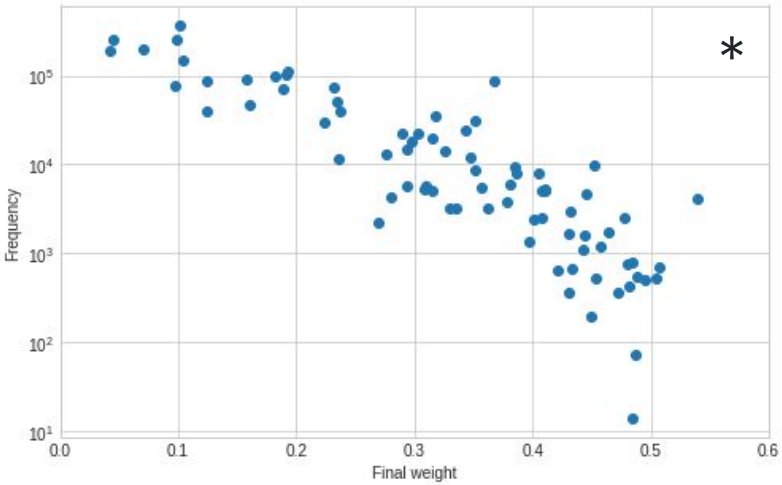
BCE → the model adapts



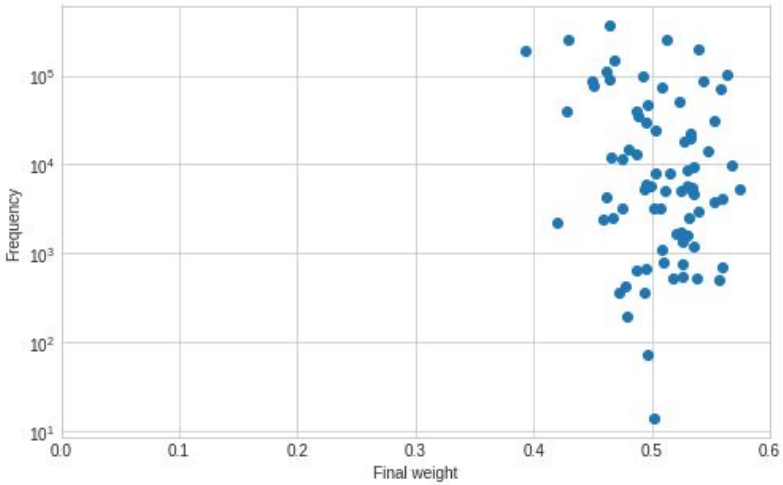
What are the clauses most penalized by the learning process?

Multiloss Training - Weight Changes

BCE multiloss



BCE → the model adapts



corr(weight, freq):

	Bottom-Up		Top-Down		Hybrid	
	DistilBERT	BERT	DistilBERT	BERT	DistilBERT	BERT
FIGER	-.77 *	-.76	-.68	-.74	-.63	-.68
BBN	-.70	-.73	-.47	-.55	-.73	-.74

Final Experiments - Test Evaluation

FIGER

Loss	Encoder	KB	Initial Clause Weight Value	Learnable or Fixed Weight	P	R	F1	
BCE	DistilBERT	None	None	None	**.7413**	.8080	.7732	~
BCE Multiloss	DistilBERT	Bottom-Up	.5	Learnable	.7391	.7929	.7651	
BCE Multiloss	DistilBERT	Top-Down	.5	Learnable	.7404	.8092	**.7733**	
BCE Multiloss	DistilBERT	Hybrid	.5	Learnable	.7316	**.8102**	.7689	
BCE	BERT	None	None	None	.7884	.8601	.8227	+.0151
BCE Multiloss	BERT	Bottom-Up	.5	Learnable	.7854	.8611	.8215	
BCE Multiloss	BERT	Top-Down	.5	Learnable	.7997	**.8796**	**.8378**	
BCE Multiloss	BERT	Hybrid	.5	Learnable	**.8016**	.8674	.8332	

BBN

Loss	Encoder	KB	Initial Clause Weight Value	Learnable or Fixed Weight	P	R	F1	
BCE	DistilBERT	None	None	None	.7183	.7913	.7530	+.0127
BCE Multiloss	DistilBERT	Bottom-Up	.5	Learnable	.7300	.8017	.7642	
BCE Multiloss	DistilBERT	Top-Down	.5	Learnable	**.7304**	**.8046**	**.7657**	
BCE Multiloss	DistilBERT	Hybrid	.5	Learnable	.7175	.7893	.7517	
BCE	BERT	None	None	None	.7428	.8197	.7793	+.0113
BCE Multiloss	BERT	Bottom-Up	.5	Learnable	.7417	.8149	.7766	
BCE Multiloss	BERT	Top-Down	.5	Learnable	**.7551**	**.8297**	**.7906**	
BCE Multiloss	BERT	Hybrid	.5	Learnable	.7531	.8248	.7876	

Final Experiments - Test Evaluation

*	Loss	Encoder	KB	Initial Clause Weight Value	Learnable or Fixed Weight	P	R	F1
				FIGER				
	BCE	BERT	None	None	None	.7884	.8601	.8227
→	BCE Multiloss	BERT	Top-Down	.5	Learnable	.7997	.8796	.8378
→	(Ren 2020) - best approach that injects hierarchy					-	-	.826
	(Li et al. 2021) - construct a graph with hierarchy and entities and use GCN and GAT					-	-	.877
	(Dai, Song, and Li 2020b) - best approach overall (EL + H + SRL)					-	-	.8909
				BBN				
	BCE	BERT	None	None	None	.7428	.8197	.7793
→	BCE Multiloss	BERT	Top-Down	.5	Learnable	.7551	.8297	.7906
→	(Chen, Chen, and Van Durme 2020) - best approach that injects hierarchy					-	-	.797
	(Li et al. 2021) - construct a graph with hierarchy and entities and use GCN and GAT					-	-	.876
	(Dai, Song, and Li 2020b) - best approach overall (EL + H + SRL)					-	-	.9147

+.012

-.006

Table 7: Comparison of: baseline, best approaches presented in this article, best approaches from sota based only on hierarchical information, and best approaches from sota based also on other information (Graph Convolution Network + Graph Attention Network or Entity Linking + Hypernym embedding + Semantic Role Labeling)

**from the article submitted to KR 2022 - "Neuro-Symbolic Integration for Fine-Grained Entity Typing"*

Final Experiments - Test Evaluation

*	Loss	Encoder	KB	Initial Clause Weight Value	Learnable or Fixed Weight	P	R	F1	
				FIGER					
	BCE	BERT	None	None	None	.7884	.8601	.8227	
→	BCE Multiloss	BERT	Top-Down	.5	Learnable	.7997	.8796	.8378	
	(Ren 2020) - best approach that injects hierarchy					-	-	.826	-.0531
	(Li et al. 2021) - construct a graph with hierarchy and entities and use GCN and GAT					-	-	.877	
→	(Dai, Song, and Li 2020b) - best approach overall (EL + H + SRL)					-	-	.8909	
				BBN					
	BCE	BERT	None	None	None	.7428	.8197	.7793	
→	BCE Multiloss	BERT	Top-Down	.5	Learnable	.7551	.8297	.7906	
	(Chen, Chen, and Van Durme 2020) - best approach that injects hierarchy					-	-	.797	-.1241
	(Li et al. 2021) - construct a graph with hierarchy and entities and use GCN and GAT					-	-	.876	
→	(Dai, Song, and Li 2020b) - best approach overall (EL + H + SRL)					-	-	.9147	

Table 7: Comparison of: baseline, best approaches presented in this article, best approaches from sota based only on hierarchical information, and best approaches from sota based also on other information (Graph Convolution Network + Graph Attention Network or Entity Linking + Hypernym embedding + Semantic Role Labeling)

**from the article submitted to KR 2022 - "Neuro-Symbolic Integration for Fine-Grained Entity Typing"*

Other Experiments and Analysis

- Definition of other KB encoding strategies
- Confidence scores analysis
  - pre-KENN vs post-KENN distributions
  - Finite State Machines
  - Sankey diagrams
- KENN with different encoders: DistilBERT vs BERT
- KENN with multiloss and variable clause weights
- How much additional resources are needed for KENN?

Conclusions

Research Questions:

- ***RQ1:*** *How can we encode hierarchical information through logic?*
  ➔ definition of different KB encoding strategies through FOL

- ***RQ2:*** *How does KENN behave with different encodings?*
  ➔ initial setup: the Entity Typing Network adapts regardless the KB
  ➔ multiloss setup:
    ➔ the Entity Typing Network does not adapt
    ➔ top-down seems to be the most advantageous KB

- ***RQ3:*** *What are the benefits of using KENN in FET?*
  ➔ it can help in scenarios with low-resources and/or small data
  1. *it accelerates the learning process (initial boost)*
  2. *it preserves high weights on the clauses involving rare types (multiloss)*
  3. *it has negligible costs*

Future Works

- use KENN to learn new types introduced after training
- enrich the KBs with additional information
  e.g., type disjointness: Person ➜ ¬Location
- combine KENN with literature techniques
- test other NSI approaches

Thank you for your attention!
Questions?

References

[FET]
Ling, X., & Weld, D. S. (2012, July). Fine-grained entity recognition. In Twenty-Sixth AAAI Conference on Artificial Intelligence.

[NSI]
Sarker, M. K., Zhou, L., Eberhart, A., & Hitzler, P. (2021). Neuro-symbolic artificial intelligence. AI Communications, (Preprint), 1-13.

[KENN]
Daniele, A., & Serafini, L. (2019, August). Knowledge enhanced neural networks. In Pacific Rim International Conference on Artificial Intelligence (pp. 542-554). Springer, Cham.
Daniele, A., & Serafini, L. (2020). Neural Networks Enhancement with Logical Knowledge. arXiv preprint arXiv:2009.06087.

[RNM]
Marra, G.; Diligenti, M.; Giannini, F.; Gori, M.; and Maggini, M. 2020. Relational neural machines.

[SBR]
Diligenti, M., Gori, M., & Sacca, C. (2017). Semantic-based regularization for learning and inference. Artificial Intelligence, 244, 143-165.

[LTN]
Badreddine, S.; d'Avila Garcez, A.; Serafini, L.; and Spranger, M. 2022. Logic tensor networks. Artificial Intelligence 303:103649.

[BERT]
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[DistilBERT]
Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

References

[Adapters]
Pfeiffer, J., Vulić, I., Gurevych, I., & Ruder, S. (2020). Mad-x: An adapter-based framework for multi-task cross-lingual transfer. arXiv preprint arXiv:2005.00052.
Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., & Gurevych, I. (2020). AdapterFusion: Non-destructive task composition for transfer learning. arXiv preprint arXiv:2005.00247.

[BBN]
Ren, X., He, W., Qu, M., Voss, C. R., Ji, H., & Han, J. (2016, August). Label noise reduction in entity typing by heterogeneous partial-label embedding. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1825-1834).

[FIGER]
Ling, X., & Weld, D. S. (2012, July). Fine-grained entity recognition. In Twenty-Sixth AAAI Conference on Artificial Intelligence.

[table sota comparison]

Ren, Q. (2020, June). Fine-grained entity typing with hierarchical inference. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (Vol. 1, pp. 2552-2558). IEEE.

Li, J., Chen, X., Wang, D., & Li, Y. Enhancing Label Representations with Relational Inductive Bias Constraint for Fine-Grained Entity Typing.

Dai, H., Song, Y., & Li, X. (2020, February). Exploiting semantic relations for fine-grained entity typing. In Automated Knowledge Base Construction.

Chen, T., Chen, Y., & Van Durme, B. (2020). Hierarchical entity typing via multi-level learning to rank. arXiv preprint arXiv:2004.02286.