COS 598C: Relation Extraction

Shunyu Yao & Zexuan Zhong

Information Extraction

Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR Corp., immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL Corp., said the increase took effect Thursday and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Denver to San Francisco.

- Entities and their relations are valuable information!

Named Entity Recognition (NER)

Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

Relation Extraction (RE)

Citing high fuel prices, [$_{ORG}$ **United Airlines**] said [$_{TIME}$ **Friday**] it has increased fares by [$_{MONEY}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{ORG}$ **American Airlines**], a unit of [$_{ORG}$ **AMR Corp.**], immediately matched the move, spokesman [$_{PER}$ **Tim Wagner**] said. [$_{ORG}$ **United**], a unit of [$_{ORG}$ **UAL Corp.**], said the increase took effect [$_{TIME}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{LOC}$ **Chicago**] to [$_{LOC}$ **Dallas**] and [$_{LOC}$ **Denver**] to [$_{LOC}$ **San Francisco**].

Entity 1	Relation	Entity 2
United	PartOf	UAL Corp.
Tim Wagner	OrgAff	American Airlines
…	…	…

Relation Extraction

- Relation extraction is a major task in the field of information extraction
- **Task definition 1**: Given a sentence with two annotated entities, classify their relation (or no relation)
- **Task definition 2**: Given a sentence, detect entities and all the relations between them
  - NER is required first
  - Entities can be pronouns, requiring coreference resolution
  - Relations can be pre-defined or discovered

Overview

1. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures [Miwa and Bansal, ACL’16]

   Annotated Entities $\longrightarrow$ End-to-end learned with relations

2. Matching the Blanks: Distributional Similarity for Relation Learning [Soares et al., ACL’19]

   Predefined Relations $\longrightarrow$ Pre-trained without annotations

Paper 1: [Miwa and Bansal, ACL'16]

End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures

Makoto Miwa
Toyota Technological Institute
Nagoya, 468-8511, Japan
makoto-miwa@toyota-ti.ac.jp

Mohit Bansal
Toyota Technological Institute at Chicago
Chicago, IL, 60637, USA
mbansal@ttic.edu

Motivation

1. Traditional systems do two separate tasks: named entity recognition (NER) and RE based on it.
    - **Problem**: relations and entity information interact!
    - **Example**: "*Toefting transferred to Bolton*": "*transferred" to* is a relational cue for the entity information that "*Toefting*" and "*Bolton"* are Person and Organization.

Motivation

1. Traditional systems do two separate tasks: named entity recognition (NER) and RE based on it.
    - **Problem**: relations and entity information interact!
    - **Example**: "*Toefting transferred to Bolton*": "*transferred" to* is a relational cue for the entity information that "*Toefting*" and "*Bolton"* are Person and Organization.
2. Previous RNN-based models focus on either word sequence or tree structure.
    - **Problem:** these two linguistic structures are complementary.

Motivation

1. Traditional systems do two separate tasks: named entity recognition (NER) and RE based on it.
    - **Problem**: relations and entity information interact!
    - **Example**: "*Toefting transferred to Bolton*": "*transferred" to* is a relational cue for the entity information that "*Toefting*" and "*Bolton*" are Person and Organization.
2. Previous RNN-based models focus on either word sequence or tree structure.
    - **Problem:** these two linguistic structures are complementary.

"We present a novel end-to-end model to extract relations between entities on both word sequence and dependency tree structures."

Model overview

Three components:

1. Embedding layer
2. Sequence layer
3. Dependency layer

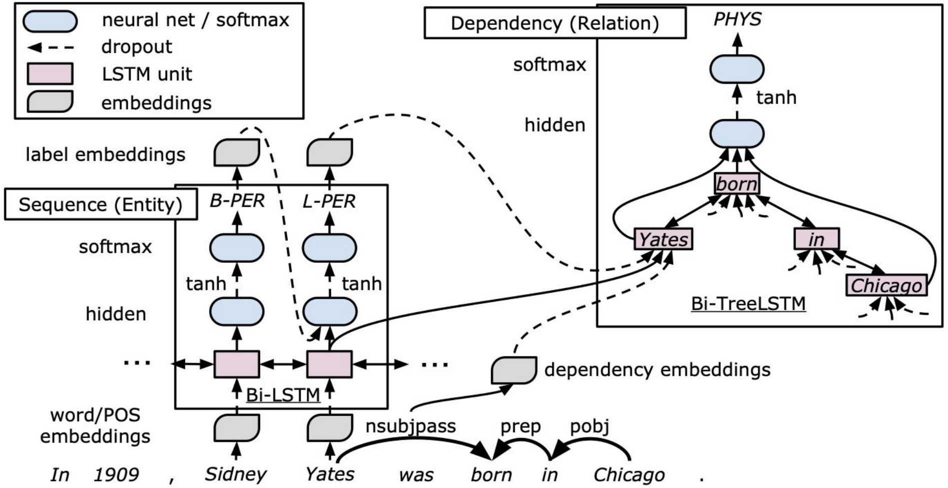
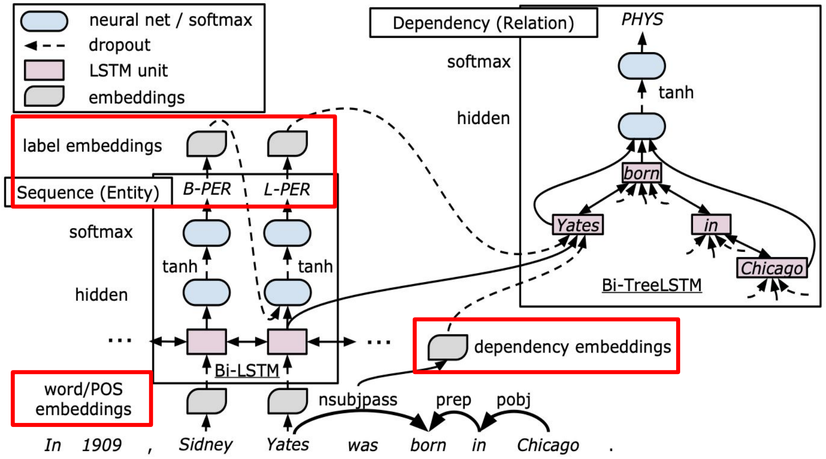


Fig. 1: Our incrementally-decoded end-to-end relation extraction model, with bidirectional sequential and bidirectional tree-structured LSTM-RNNs.

1. Embedding layer

Embedding type	Dim.	Label example
Word $v^{(w)}$	200	Yates
Part-of-speech $v^{(p)}$	25	NNP
Dependency type $v^{(d)}$	25	nsubjpass
Entity label $v^{(e)}$	25	L-PER

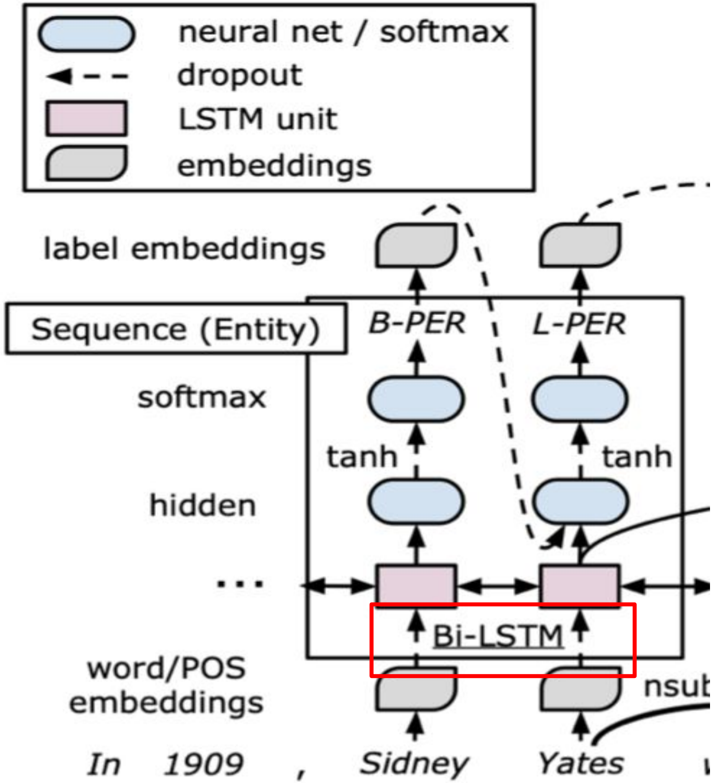


2. Sequence layer

- **Encoding**: Bi-LSTM
-

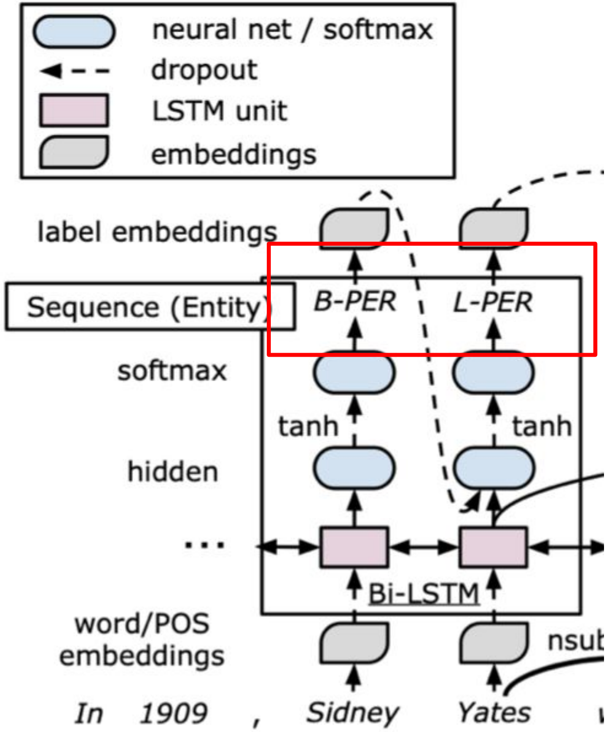
$$
\begin{aligned}
i_t &= \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right),\\
f_t &= \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right),\\
o_t &= \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right),\\
u_t &= \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right)\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot \tanh(c_t),
\end{aligned}
$$

$$s_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right].$$



2. Sequence layer

- **Encoding**: Bi-LSTM
- **Decoding**: entity detection as sequence labelin Assign an entity tag to each word using Begin, Inside, Last, Outside, Unit (BILOU) scheme
  - Joe works for New York Times in New York.
  - BILOU: Joe(U-PER) works(O) for(O) New(B-ORG) York(I-ORG) Times(L-ORG) in(O) New(B-LOC) York(L-LOC).
  - BIO: Joe(B-PER) works(O) for(O) New(B-ORG) York(I-ORG) Times(I-ORG) in(O) New(B-LOC) York(I-LOC).

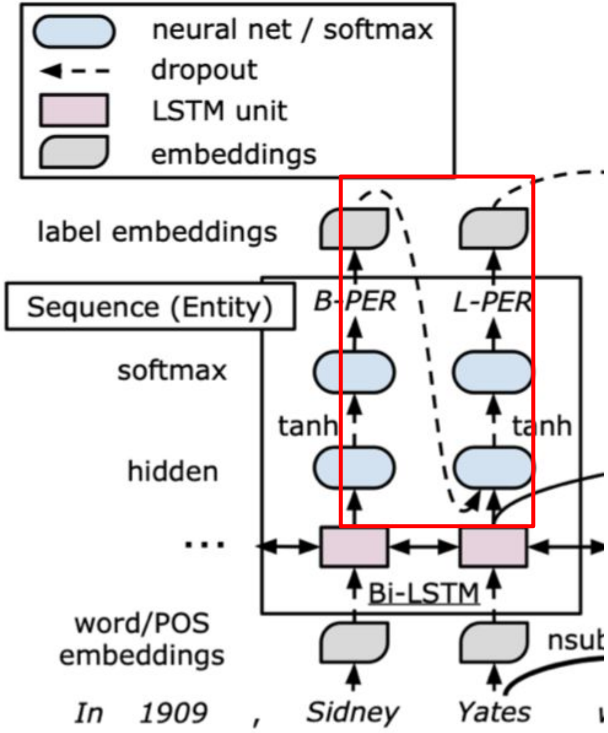


2. Sequence layer

- **Encoding**: Bi-LSTM
- **Decoding**: entity detection as sequence labelin
  Assign an entity tag to each word using Begin, Inside, Last, Outside, Unit (BILOU) scheme
  - Greedy left-to-right decoding conditioned on previous prediction

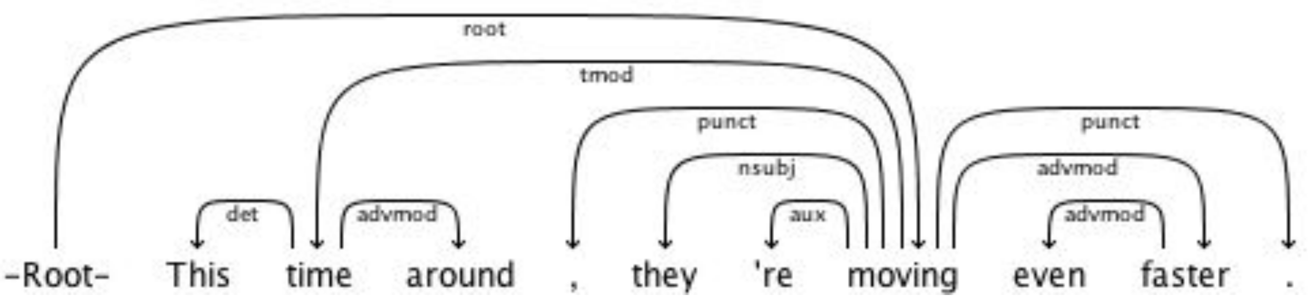
$$h_t^{(e)} = \tanh\left(W^{(e_h)}[s_t; \boxed{v_{t-1}^{(e)}}] + b^{(e_h)}\right) \quad (2)$$

$$y_t = \text{softmax}\left(W^{(e_y)} h_t^{(e)} + b^{(e_y)}\right) \quad (3)$$



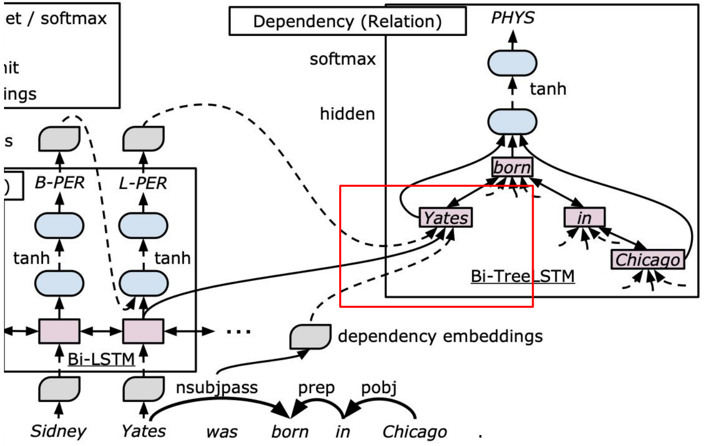
3. Dependency layer: what’s dependency?

- A dependency parser analyzes the **grammatical structure of a sentence**, establishing **relationships between "head" words and words which modify those heads**.
- The figure below shows a dependency parse of a short sentence. The arrow from the word *moving* to the word *faster* indicates that *faster* modifies *moving*, and the label *advmod* assigned to the arrow describes the exact nature of the dependency.



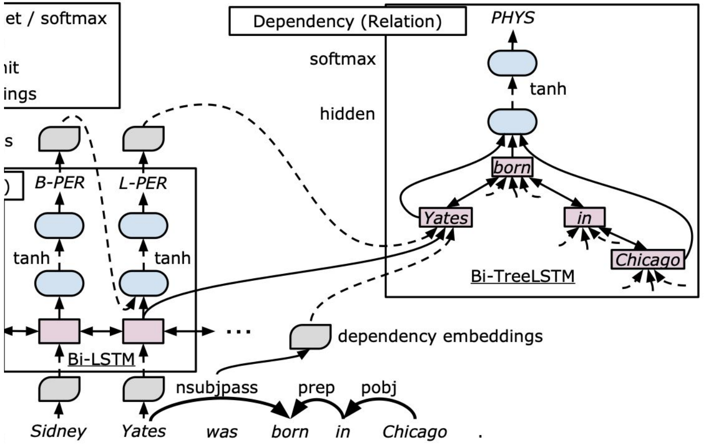
3. Dependency layer

- Input: $x_t = \left[s_t; v_t^{(d)}; v_t^{(e)}\right]$
  - Hidden state vector in the sequence layer s_t
  - Dependency type embedding $v_t^{(d)}$
  - Entity type embedding $v_t^{(e)}$
- Example: *Yates* on the right
  - Dependency label = “nsubjpass”
  - Entity type = “L-PER”



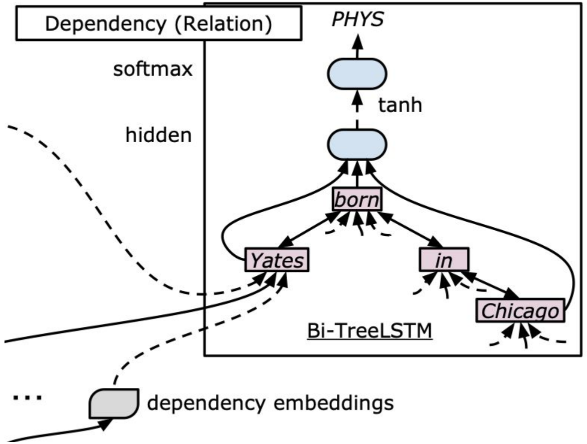
3. Dependency layer

- Input: $x_t = \left[s_t; v_t^{(d)}; v_t^{(e)}\right]$
- Incrementally build relation candidates using **all possible combinations** of the last words of detected entities, i.e., words with L or U labels in the BILOU scheme, during decoding.
- For **each pair** of relation candidates, consider their **shortest path** in the dependency tree, i.e. subtree from the lowest common ancestor (LCA)



3. Dependency layer

- Encoding: bi-directional tree-structure LSTM
  - Parent to children, and children to parent
  - C(t): children of t (variable #, different type)
  - m(l): type of l (in shortest path or not)



$$i_t = \sigma\left(W^{(i)}x_t + \sum_{l \in C(t)} U^{(i)}_{m(l)} h_{tl} + b^{(i)}\right), \quad (4)$$

$$f_{tk} = \sigma\left(W^{(f)}x_t + \sum_{l \in C(t)} U^{(f)}_{m(k)m(l)} h_{tl} + b^{(f)}\right),$$

$$o_t = \sigma\left(W^{(o)}x_t + \sum_{l \in C(t)} U^{(o)}_{m(l)} h_{tl} + b^{(o)}\right),$$

$$u_t = \tanh\left(W^{(u)}x_t + \sum_{l \in C(t)} U^{(u)}_{m(l)} h_{tl} + b^{(u)}\right),$$

$$c_t = i_t \odot u_t + \sum_{l \in C(t)} f_{tl} \odot c_{tl},$$

$$h_t = o_t \odot \tanh(c_t),$$

3. Dependency layer

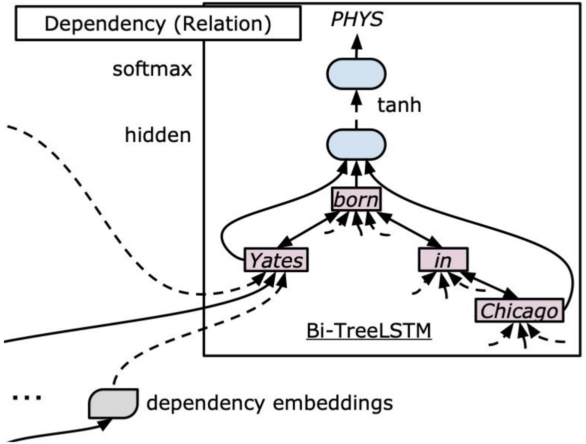
- Encoding: bi-directional tree-structure LSTM
- Relational classification:

$$d_p = [\uparrow h_{p_A}; \downarrow h_{p_1}; \downarrow h_{p_2}]$$

$$h_p^{(r)} = \tanh\left(W^{(r_h)} d_p + b^{(r_h)}\right) \quad (5)$$

$$y_p = \text{softmax}\left(W^{(r_y)} h_t^{(r)} + b^{(r_y)}\right) \quad (6)$$

- No relation is also an label



Training

- End-to-end training: dependency layer uses hidden state vector in the sequence layer & entity type embedding
- One problem in training: entity prediction is unreliable in the early stage, which makes learning relations impossible
    - Trick 1: **scheduled sampling**. Use gold entity labels with a decaying probability
    - Trick 2: **entity pre-training.** Pre-train entity detection before end-to-end training

Datasets: Automatic Content Extraction (ACE05,04)

- Recognition of <u>entities</u>, values, temporal expressions, <u>relations</u>, and events.
- ORG-AFF example: "...details about perks <u>Welch[PER]</u> received as part of his retirement package from <u>GE[ORG]</u>..."

Data Set		# sentences	# mentions	# relations
ACE'05	Train	7,273	26,470	4,779
	Dev	1,765	6,421	1,179
	Test	1,535	5,476	1,147
ACE'04		6,789	22,740	4,368

Table 1 ACE05 Entity Types and Subtypes

Type	Subtypes
FAC (Facility)	Airport, Building-Grounds, Path, Plant, Subarea-Facility
GPE (Geo-Political Entity[3])	Continent, County-or-District, GPE-Cluster, Nation, Population-Center, Special, State-or-Province
LOC (Location)	Address, Boundary, Celestial, Land-Region-Natural, Region-General, Region-International, Water-Body
ORG (Organization)	Commercial, Educational, Entertainment, Government, Media, Medical-Science, Non-Governmental, Religious, Sports
PER (Person)	Group, Indeterminate, Individual
VEH (Vehicle)	Air, Land, Subarea-Vehicle, Underspecified, Water
WEA (Weapon)	Biological, Blunt, Chemical, Exploding, Nuclear, Projectile, Sharp, Shooting, Underspecified

Table 6 ACE05 Relation Types and Subtypes
(Relations marked with an * are symmetric relations.)

Type	Subtype
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (Gen-affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONYMY*	*none*
ORG-AFF (Org-affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-whole)	Artifact, Geographical, Subsidiary
PER-SOC* (person-social)	Business, Family, Lasting-Personal
PHYS* (physical)	Located, Near

Datasets: SemEval-2010 Task 8

- 9 relation types + *Other* (negative relation); See below
- 8,000 training and 2,717 test sentences, each sentence annotated with a relation between two given nomials
    - Only annotate relation-related nominals, so can evaluate relation classification part along

Content-Container (CC). An object is physically stored in a delineated area of space. Example: *a bottle full of honey was weighed*

Entity-Origin (EO). An entity is coming or is derived from an origin (e.g., position or material). Example: *letters from foreign countries*

Entity-Destination (ED). An entity is moving towards a destination. Example: *the boy went to bed*

Component-Whole (CW). An object is a component of a larger whole. Example: *my apartment has a large kitchen*

Member-Collection (MC). A member forms a nonfunctional part of a collection. Example: *there are many trees in the forest*

Message-Topic (MT). A message, written or spoken, is about a topic. Example: *the lecture was about semantics*

Cause-Effect (CE). An event or object leads to an effect. Example: *those cancers were caused by radiation exposures*

Instrument-Agency (IA). An agent uses an instrument. Example: *phone operator*

Product-Producer (PP). A producer causes a product to exist. Example: *a factory manufactures suits*

Metrics

- The primary micro F1-score, precision and recall on both entity and relation extraction
  - Precision: #(true positive) / #(positive)
  - Recall: #(true positive) / #(true)
  - F1: harmonic mean of precision & recall
- Classification can be tricky
  - ✔️ entity correct when its type and the region of its head are correct
  - ✔️ relation correct when its type and argument entities are correct
  - ❌ treat all non-negative relations on wrong entities as false positives

Results: ACE05 and ACE04

- On ACE05 and ACE04, what does it mean that P is lower, R is higher?
- [Li and Ji, ACL 2014]: “Compared to human annotators, the bottleneck of automatic approaches is the low recall of relation extraction.”

Corpus	Settings	Entity			Relation		
		P	R	F1	P	R	F1
ACE05	Our Model (SPTree)	0.829	**0.839**	**0.834**	0.572	**0.540**	**0.556**
	Li and Ji (2014)	**0.852**	0.769	0.808	**0.654**	0.398	0.495
ACE04	Our Model (SPTree)	0.808	**0.829**	**0.818**	0.487	**0.481**	**0.484**
	Li and Ji (2014)	**0.835**	0.762	0.797	**0.608**	0.361	0.453

Table 1: Comparison with the state-of-the-art on the ACE05 test set and ACE04 dataset.

Results: SemEval-2010 Task 8

- Performances are similar

Settings	Macro-F1
No External Knowledge Resources	
Our Model (SPTree)	**0.844**
dos Santos et al. (2015)	0.841
Xu et al. (2015a)	0.840
+WordNet	
Our Model (SPTree + WordNet)	0.855
Xu et al. (2015a)	**0.856**
Xu et al. (2015b)	0.837

Table 4: Comparison with state-of-the-art models on SemEval-2010 Task 8 test-set.

Ablation study

- Entity pre-training is the most important
- Two-stage training (-Shared) does not harm performance much

Settings	Entity			Relation		
	P	R	F1	P	R	F1
Our Model (SPTree)	0.815	0.821	0.818	0.506	0.529	0.518
−Entity pretraining (EP)	0.793	0.798	0.796	0.494	0.491	0.492*
−Scheduled sampling (SS)	0.812	0.818	0.815	0.522	0.490	0.505
−Label embeddings (LE)	0.811	0.821	0.816	0.512	0.499	0.505
−Shared parameters (Shared)	0.796	0.820	0.808	0.541	0.482	0.510
−EP, SS	0.781	0.804	0.792	0.509	0.479	0.494*
−EP, SS, LE, Shared	0.800	0.815	0.807	0.520	0.452	0.484**

Table 2: Ablation tests on the ACE05 development dataset. * denotes significance at $p<0.05$, ** denotes $p<0.01$.

Ablation study (cont.)

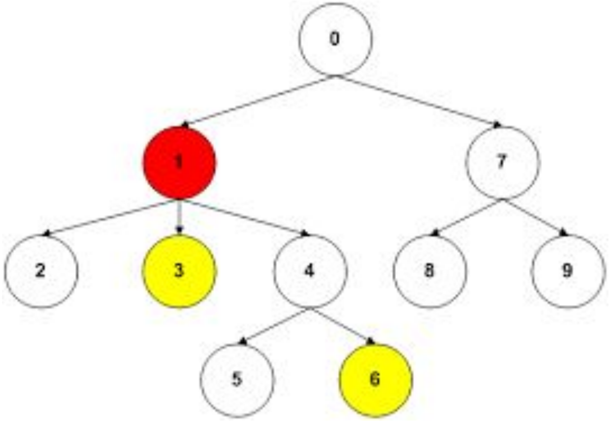
- -Pair: remove entity-related information from the sequence layer
- Need sequence layer or its information

Settings	Macro-F1
SPTree	0.851
−Hidden layer	0.839
−Sequence layer	0.840
−Pair	0.844
−Pair, Sequence layer	$\boxed{0.827*}$
Stanford PCFG	0.844
+WordNet	0.854
Left-to-right candidates	0.843
Neg. sampling (Xu et al., 2015a)	0.848

Table 6: Model setting ablations on SemEval-2010 development set.

Tree structure & LSTM study

- Tree structures:
  - **SPTree**: shortest path *(3->1->4->6)*
  - **SubTree**: subtree from LCA *(1...6)*
  - **FullTree**: full dependency tree *(0...9)*
  - **-SP**: for SubTree and FullTree, do not distinguish nodes in SPTree (i.e. one node type instead of two)
- Tree LSTM variants on SPTree:
  - **SPSeq**: bidirectional LSTMs on the shortest path, with input from the sequence layer concatenated with embeddings for the surrounding dependency types and directions. *(3<->1<->4<->6)*
  - **SPXu**: two LSTMs for the left and right subpaths of the shortest path *(3->1 and 6->4->1)*



Tree structure & LSTM study

- “...for end-to-end relation extraction, **selecting the appropriate tree structure representation of the input** (i.e., the shortest path) is more important than the choice of the LSTM-RNN structure on that input (i.e., sequential versus tree-based).”

Settings	Entity			Relation		
	P	R	F1	P	R	F1
SPTree	0.815	0.821	0.818	0.506	0.529	0.518
SubTree	0.812	0.818	0.815	0.525	0.506	0.515
FullTree	0.806	0.816	0.811	0.536	0.507	0.521
SubTree (-SP)	0.803	0.816	0.810	0.533	0.495	0.514
FullTree (-SP)	0.804	0.817	0.811	0.517	0.470	0.492*
Child-Sum	0.806	0.819	0.8122	0.514	0.499	0.506
SPSeq	0.801	0.813	0.807	0.500	0.523	0.511
SPXu	0.809	0.818	0.813	0.494	0.522	0.508

Table 3: Comparison of LSTM-RNN structures on the ACE05 development dataset.

Settings	Macro-F1
SPTree	0.851
SubTree	0.839
FullTree	0.829*
SubTree (-SP)	0.840
FullTree (-SP)	0.828*
Child-Sum	0.838
SPSeq	0.844
SPXu	0.847

Table 5: Comparison of LSTM-RNN structures on SemEval-2010 Task 8 development set.

Discussion

- **Message**: end-to-end entity+relation extraction, sequence+tree structure.
- **Limits**?

Discussion

- **Questions**:
  - Why is it a good idea to train entity detection and relation classification jointly (instead of training each component separately)?
  - Why is it a good idea to leverage *both* sequence structure and tree structure in modeling?
- **Comments**?
  - Is end-to-end training important?
  - Still rely on entity+relation supervision annotation
  - Computational cost of dependency layer?

Paper 2: [Soares et al., ACL'19]

Matching the Blanks: Distributional Similarity for Relation Learning

Livio Baldini Soares **Nicholas FitzGerald** **Jeffrey Ling*** **Tom Kwiatkowski**
Google Research
{liviobs,nfitz,jeffreyling,tomkwiat}@google.com

Inspiration: Extension of Distributional Hypothesis

Harris' Distributional Hypothesis: ***Words that occurred in the same contexts tend to be similar.***

Extension of Harris' Distributional Hypothesis: ***Relation statements that share the same two entities tend to express similar relations.***

[BLANK], inspired by Cale's earlier cover, recorded one of the most acclaimed versions of "[BLANK]"

[BLANK]'s rendition of "[BLANK]" has been called "one of the great songs" by Time, and is included on Rolling Stone's list of "The 500 Greatest Songs of All Time".

Figure 1: "Matching the blanks" example where both relation statements share the same two entities.

Main Goal: Learning Relation Representations

Given: a relation statement (a triple $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$)

- A sequence of tokens $\mathbf{x} = [x_0 \ldots x_n]$, where $x_0 = [\text{CLS}]$ $x_n = [\text{SEP}]$
- Entity mentions (spans): $\mathbf{s}_1 = (i, j)$ $\mathbf{s}_2 = (k, l)$

Goal: a function $\mathbf{h}_r = f_\theta(\mathbf{r})$, which maps a relation statement to a vector

Main Goal: Learning Relation Representations

Given: a relation statement (a triple $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$)

- A sequence of tokens $\mathbf{x} = [x_0 \ldots x_n]$, where $x_0 = [\mathrm{CLS}]$ $x_n = [\mathrm{SEP}]$
- Entity mentions (spans): $\mathbf{s}_1 = (i, j)$ $\mathbf{s}_2 = (k, l)$

Goal: a function $\mathbf{h}_r = f_\theta(\mathbf{r})$, which maps a relation statement to a vector

Once having the representations, we can

- Build a relation classifier on the top of relation representations
- Define similarity between relations by taking inner product of representations

Relation Classification and Extraction Tasks

Two types of relation extraction tasks:

- Supervised relation extraction (SemEval2010 Task8, KBP-37, TACRED)
    - Train a classification model on the training set
    - Directly use the trained classifier to predict relation

Relation Classification and Extraction Tasks

Two types of relation extraction tasks:

- Supervised relation extraction (SemEval2010 Task8, KBP-37, TACRED)
  - Train a classification model on the training set
  - Directly use the trained classifier to predict relation
- Few-shot relation matching (FewRel)
  - Relations on testing set do not appear on the training set
  - A few example statements for each relation are provided
  - During inference: pick “the most similar” provided statement to an input statement

Supervised Relation Extraction

Dataset	Input Example	Output Example
SemEval2010 Task8	People have been moving back into downtown.	Entity-Destination
KBP-37	The University of Central Arkansas is Arkansas's premiere dramatic school.	stateorprovince_of_headquarters
TACRED	He received an undergraduate degree from Morgan State University in 1950 and applied for admission to graduate school at the University of Maryland.	no_relation

Few-shot Relation Matching (FewRel)

Training: Use training data to train a similarity function

Testing: Few-shot relation classification

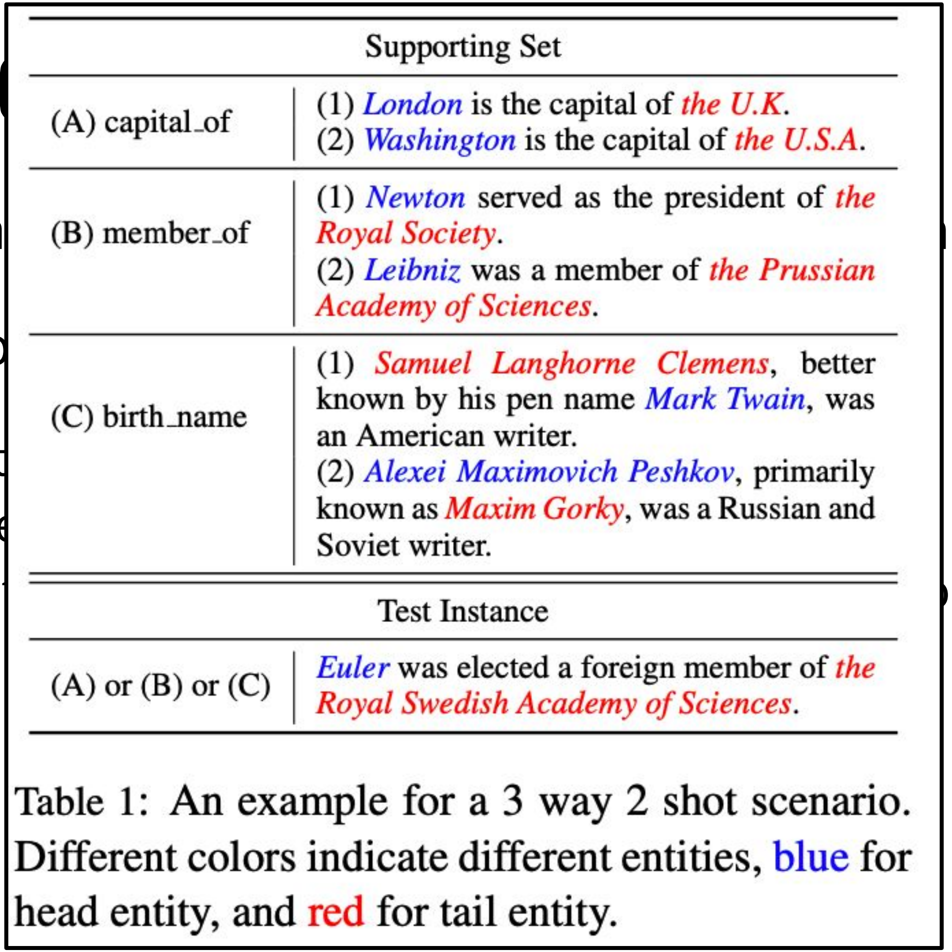
- Focus on relations that do not appear during training
- A few examples for each relation are provided
- “*N* Way *M* Shot”: N relations, M examples for each relation

Few-shot R

Training: Use tra n

Testing: Few-sho

- Focus on relat
- A few example
- “*N* Way *M* Sho on



Supporting Set	
(A) capital_of	(1) *London* is the capital of *the U.K*. (2) *Washington* is the capital of *the U.S.A*.
(B) member_of	(1) *Newton* served as the president of *the Royal Society*. (2) *Leibniz* was a member of *the Prussian Academy of Sciences*.
(C) birth_name	(1) *Samuel Langhorne Clemens*, better known by his pen name *Mark Twain*, was an American writer. (2) *Alexei Maximovich Peshkov*, primarily known as *Maxim Gorky*, was a Russian and Soviet writer.
Test Instance	
(A) or (B) or (C)	*Euler* was elected a foreign member of *the Royal Swedish Academy of Sciences*.

Table 1: An example for a 3 way 2 shot scenario. Different colors indicate different entities, blue for head entity, and red for tail entity.

Contributions

- Investigate different architectures for the relation encoder f_θ
  - Try different architectures built on top of BERT
  - Train and evaluate on relation extraction benchmarks
- Show that f_θ can be pre-trained from entity linked text of Wikipedia
  - Create training data from Wikipedia and train a relation encoder
  - Achieve the state of the art on FewRel, SemEval2010 Task8, KBP-37, and TACRED

1. Relation Representations

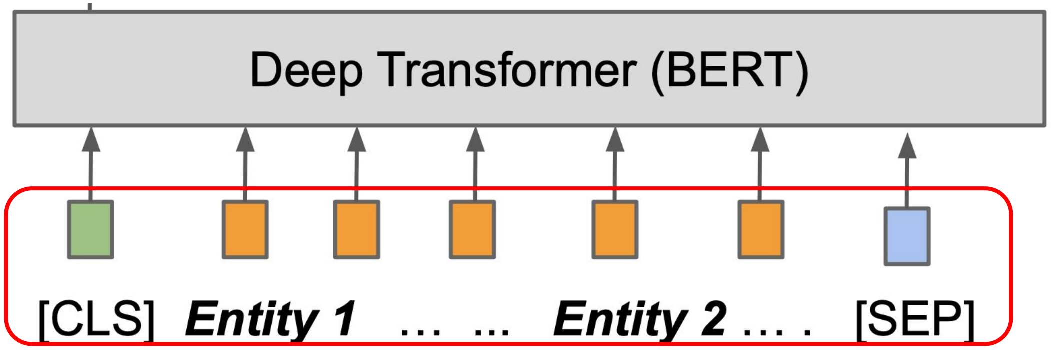
Given a sentence with annotated entities, how to use BERT to output a vector representation?

- How to feed a relation statement into BERT?
- How to get a representation vector based on the outputs of BERT?

1. Relation Representations (architecture)

How to get representations on top of BERT given a relation statement?

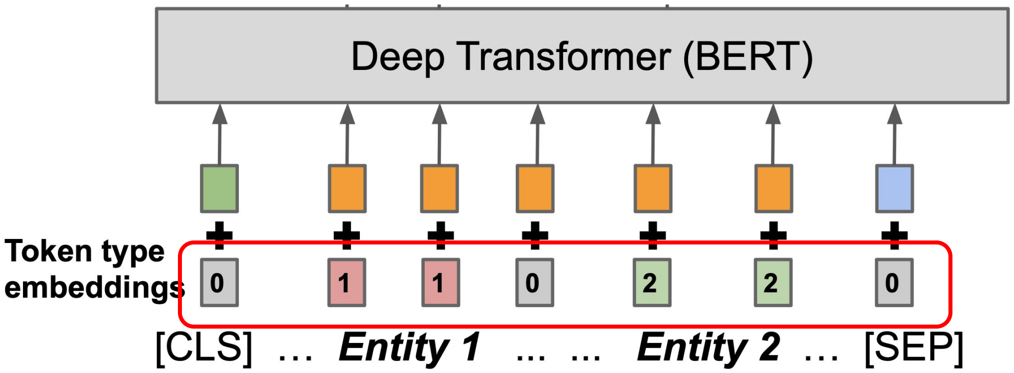
- Input Format
    - **Standard Input**



1. Relation Representations (architecture)

How to get representations on top of BERT given a relation statement?

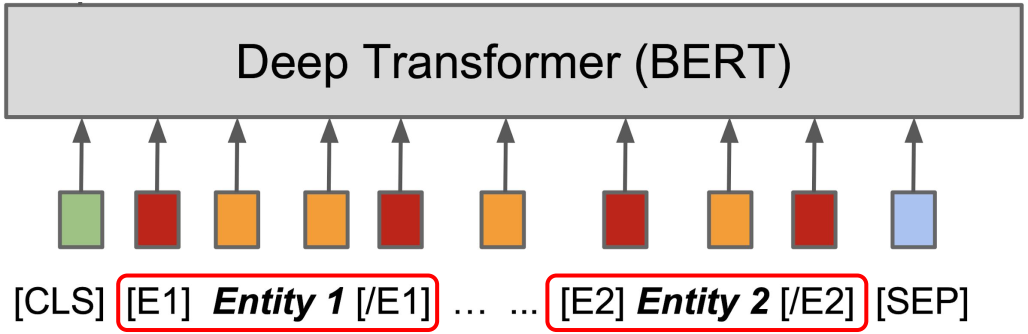
- Input Format
  - Standard Input
  - **Positional Embeddings**



1. Relation Representations (architecture)

How to get representations on top of BERT given a relation statement?

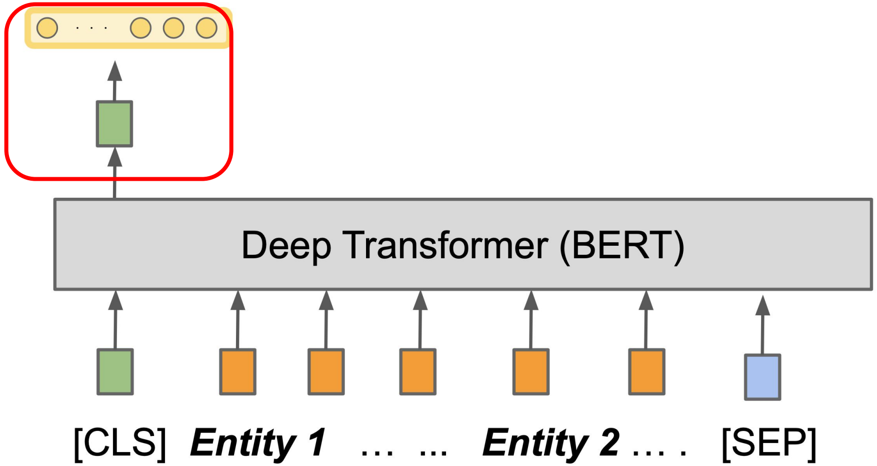
- Input Format
  - Standard Input
  - Positional Embeddings
  - **Entity marker tokens**



1. Relation Representations (architecture)

How to get representations on top of BERT given a relation statement?

- Output Representation
  - **[CLS] token**

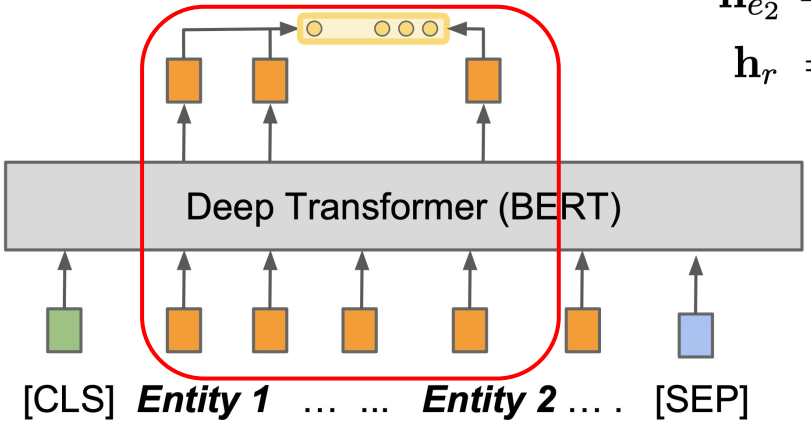


1. Relation Representations (architecture)

How to get representations on top of BERT given a relation statement?

- Output Representation
  - [CLS] token
  - **Entity mention pooling**

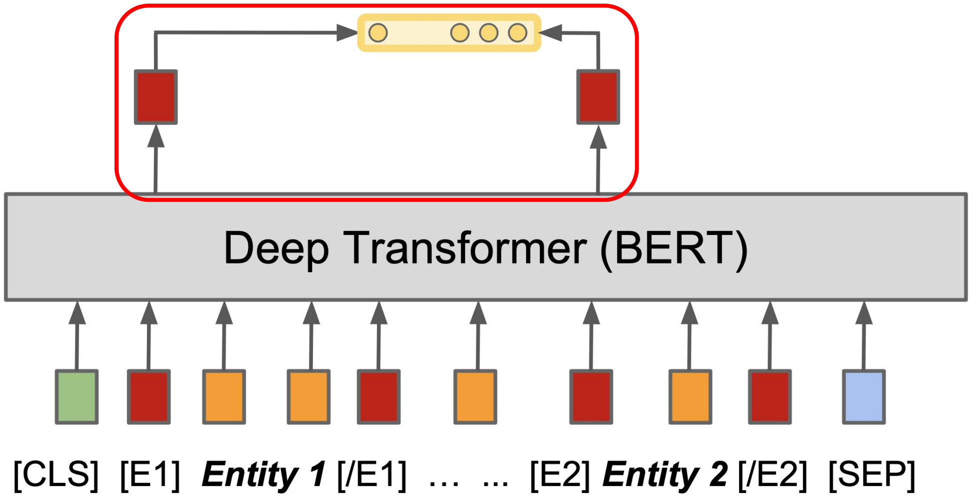
$$\mathbf{h}_{e_1} = \text{MAXPOOL}([\mathbf{h}_i ... \mathbf{h}_{j-1}])$$
$$\mathbf{h}_{e_2} = \text{MAXPOOL}([\mathbf{h}_k ... \mathbf{h}_{l-1}])$$
$$\mathbf{h}_r = \langle \mathbf{h}_{e_1} | \mathbf{h}_{e_2} \rangle$$



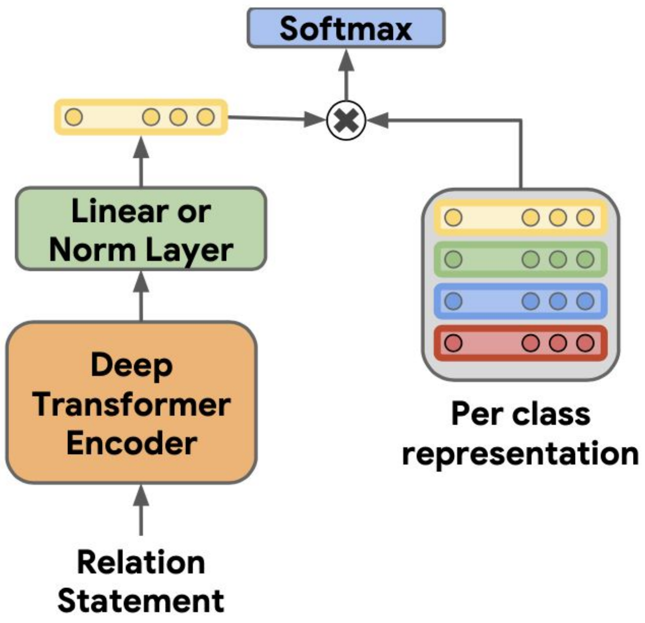
1. Relation Representations (architecture)

How to get representations on top of BERT given a relation statement?

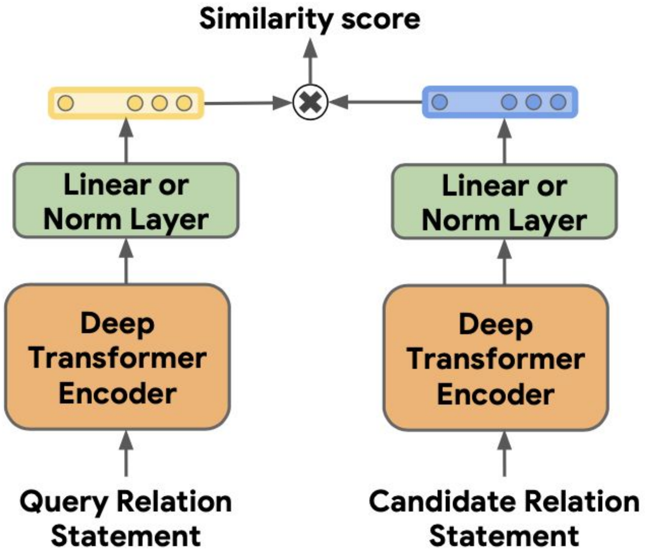
- Output Representation
  - [CLS] token
  - Entity mention pooling
  - **Entity start state**



Supervised Classification



Few-Shot Classification



1. Relation Representations: Results

		SemEval 2010 Task 8		KBP37		TACRED		FewRel 5-way-1-shot
# training annotated examples		8,000 (6,500 for dev)		15,916		68,120		44,800
# relation types		19		37		42		100
		Dev F1	Test F1	Dev F1	Test F1	Dev F1	Test F1	Dev Acc.
Wang et al. (2016)*		–	88.0	–	–	–	–	–
Zhang and Wang (2015)*		–	79.6	–	58.8	–	–	–
Bilan and Roth (2018)*		–	84.8	–	–	–	68.2	–
Han et al. (2018)		–	–	–	–	–	–	71.6
Input type	**Output type**							
STANDARD	[CLS]	71.6	–	41.3	–	23.4	–	85.2
STANDARD	MENTION POOL.	78.8	–	48.3	–	66.7	–	87.5
POSITIONAL EMB.	MENTION POOL.	79.1	–	32.5	–	63.9	–	87.5
ENTITY MARKERS	[CLS]	81.2	–	68.7	–	65.7	–	85.2
ENTITY MARKERS	MENTION POOL.	80.4	–	68.2	–	69.5	–	87.6
ENTITY MARKERS	ENTITY START	**82.1**	**89.2**	**70**	**68.3**	**70.1**	**70.1**	**88.9**

1. Relation Representations: Results

	SemEval 2010 Task 8	KBP37	TACRED	FewRel 5-way-1-shot
# training annotated examples	8,000 (6,500 for dev)	15,916	68,120	44,800
# relation types	19	37	42	100
				ev Acc.
V				–
Zha				–
Bi				–
				71.6

Input								
STAND								85.2
STAND								87.5
POSITION								87.5
ENTITY MARKERS	[CLS]	81.2	–	68.7	–	65.7	–	85.2
ENTITY MARKERS	MENTION POOL.	80.4	–	68.2	–	69.5	–	87.6
ENTITY MARKERS	ENTITY START	**82.1**	**89.2**	**70**	**68.3**	**70.1**	**70.1**	**88.9**

*The model using “**entity markers**” input format and “**entity start**” output representation achieves the best scores on all datasets!*

They use this setting for the remainder of the paper.

2. Pre-train Relation Representation Encoder

Hypothesis: $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$ and $\mathbf{r}' = (\mathbf{x}', \mathbf{s}_1', \mathbf{s}_2')$ encode the same relation if s1 and s1' refer to the same entity, s2 and s2' refer to the same entity.

Key Idea: If two relation statements, r and r', encode the same relation, the inner product $f_\theta(\mathbf{r})^\top f_\theta(\mathbf{r}')$ should be high.

2. Pre-train Relation Representation Encoder

Hypothesis: $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$ and $\mathbf{r}' = (\mathbf{x}', \mathbf{s}_1', \mathbf{s}_2')$ encode the same relation if s1 and s1' refer to the same entity, s2 and s2' refer to the same entity.

Key Idea: If two relation statements, r and r', encode the same relation, the inner product $f_\theta(\mathbf{r})^\top f_\theta(\mathbf{r}')$ should be high.

Pre-training:

- Get relation statements pairs (pos and neg) from entity linked text
- Train a relation representation encoder on those pairs

Pre-training Setup

Let $\mathcal{D} = [(\mathbf{r}^0, e_1^0, e_2^0) \dots (\mathbf{r}^N, e_1^N, e_2^N)]$ be a corpus of relation statements that have been linked to two entities $e_1^i \in \mathcal{E}$ and $e_2^i \in \mathcal{E}$, and $\mathbf{r}^i = (\mathbf{x}^i, \mathbf{s}_1^i, \mathbf{s}_2^i)$.

Define a binary classifier:

$$p(l = 1|\mathbf{r}, \mathbf{r}') = \frac{1}{1 + \exp f_\theta(\mathbf{r})^\top f_\theta(\mathbf{r}')}$$

Training loss:

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|^2} \sum_{(\mathbf{r},e_1,e_2)\in\mathcal{D}} \sum_{(\mathbf{r}',e_1',e_2')\in\mathcal{D}} \delta_{e_1,e_1'}\delta_{e_2,e_2'} \cdot \log p(l = 1|\mathbf{r}, \mathbf{r}') + (1 - \delta_{e_1,e_1'}\delta_{e_2,e_2'}) \cdot \log(1 - p(l = 1|\mathbf{r}, \mathbf{r}')) \quad (1)$$

$\delta_{e,e'} = 1$ iff $e = e'$, otherwise $\delta_{e,e'} = 0$

Pre-training Setup

Let $\mathcal{D} = [(\mathbf{r}^0, e_1^0, e_2^0) \ldots (\mathbf{r}^N, e_1^N, e_2^N)]$ be a corpus of relation statements that have been linked to two entities $e_1^i \in \mathcal{E}$ and $e_2^i \in \mathcal{E}$, and $\mathbf{r}^i = (\mathbf{x}^i, \mathbf{s}_1^i, \mathbf{s}_2^i)$.

Define a binary c

In practice, it is not possible to consider every negative pairs!

Training loss:

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|^2} \sum_{(\mathbf{r},e_1,e_2)\in\mathcal{D}} \sum_{(\mathbf{r}',e_1',e_2')\in\mathcal{D}} \delta_{e_1,e_1'} \delta_{e_2,e_2'} \cdot \log p(l = 1|\mathbf{r}, \mathbf{r}') + \boxed{(1 - \delta_{e_1,e_1'} \delta_{e_2,e_2'}) \cdot \log(1 - p(l = 1|\mathbf{r}, \mathbf{r}'))} \quad (1)$$

$\delta_{e,e'} = 1$ iff $e = e'$, otherwise $\delta_{e,e'} = 0$

Strong Negative Pairs

Sample a set of negatives:

- Randomly sample from the set of all relation statement pairs
- Randomly sample from the set of relation statements that share just a single entity (strong negative pairs)

$\mathbf{r}_A$	In 1976, e_1 (then of Bell Labs) published e_2, the first of his books on programming inspired by the Unix operating system.
$\mathbf{r}_B$	The "e_2" series spread the essence of "C/Unix thinking" with makeovers for Fortran and Pascal. e_1's Ratfor was eventually put in the public domain.
$\mathbf{r}_C$	e_1 worked at Bell Labs alongside e_3 creators Ken Thompson and Dennis Ritchie.
Mentions	e_1 = Brian Kernighan, e_2 = Software Tools, e_3 = Unix

Introducing Blanks

Entity linking system can perfectly minimize the loss, i.e., if the entities are the same, then predict as a positive sample.

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|^2} \sum_{(\mathbf{r},e_1,e_2)\in\mathcal{D}} \sum_{(\mathbf{r}',e_1',e_2')\in\mathcal{D}} \tag{1}$$

$$\delta_{e_1,e_1'}\delta_{e_2,e_2'} \cdot \log p(l = 1|\mathbf{r}, \mathbf{r}')+$$

$$(1 - \delta_{e_1,e_1'}\delta_{e_2,e_2'}) \cdot \log(1 - p(l = 1|\mathbf{r}, \mathbf{r}'))$$

Introducing Blanks

Entity linking system can perfectly minimize the loss, i.e., if the entities are the same, then predict as a positive sample.

However, entity linking system does not learn meaningful relation representations!

Solution (introducing blanks): replace entity mentions by [BLANK] symbol with probability α (α=0.7).

$$\tilde{\mathcal{D}} = [(\tilde{\mathbf{r}}^0, e_1^0, e_2^0) \ldots (\tilde{\mathbf{r}}^N, e_1^N, e_2^N)]$$

Pre-Training Data Collections

Corpus: English Wikipedia

Entity Linking System: Google Cloud Natural Language API

Positive pairs: All pairs of relation statements that contain the same entity pairs

Negative pairs: Randomly samples from the set of all pairs + randomly samples from the set of pairs that share just a single entity.

 # relation statement pairs: 600 million, 50% pos and 50% neg

Evaluation: Few-shot Relation Matching (FewRel)

1. BERT and BERT+MTB outperforms SOTA without seeing training data
2. BERT+MTB largely outperforms BERT in the unsupervised setting (0 train sample)

5 way 1 shot						
# examples per type	0	5	20	80	320	700
Prot.Net. (CNN)	–	–	–	–	–	$\boxed{71.6}$
BERT_{EM}	$\boxed{72.9}$	81.6	85.1	86.9	88.8	88.9
BERT_{EM}+MTB	$\boxed{80.4}$	85.5	88.4	89.6	89.6	90.1
10 way 1 shot						
# examples per type	0	5	20	80	320	700
Prot.Net. (CNN)	–	–	–	–	–	$\boxed{58.8}$
BERT_{EM}	$\boxed{62.3}$	72.8	76.9	79.0	81.4	82.8
BERT_{EM}+MTB	$\boxed{71.5}$	78.1	81.2	82.9	83.7	83.4

Evaluation: Few-shot Relation Matching (FewRel)

1. BERT and BERT+MTB outperforms SOTA without seeing training data
2. BERT+MTB largely outperforms BERT in the unsupervised setting (0 train sample)
3. BERT+MTB needs only 6% of data to match BERT trained on all data (MTB helps reduce annotation effort)

5 way 1 shot						
# examples per type	0	5	20	80	320	700
Prot.Net. (CNN)	–	–	–	–	–	71.6
BERT_{EM}	72.9	81.6	85.1	86.9	88.8	88.9
BERT_{EM}+MTB	80.4	85.5	88.4	89.6	89.6	90.1

10 way 1 shot						
# examples per type	0	5	20	80	320	700
Prot.Net. (CNN)	–	–	–	–	–	58.8
BERT_{EM}	62.3	72.8	76.9	79.0	81.4	82.8
BERT_{EM}+MTB	71.5	78.1	81.2	82.9	83.7	83.4

Evaluation: Few-shot Relation Matching (FewRel)

1. BERT and BERT+MTB outperforms SOTA without seeing training data
2. BERT+MTB largely outperforms BERT in the unsupervised setting (0 train sample)
3. BERT+MTB needs only 6% of data to match BERT trained on all data (MTB helps reduce annotation effort)
4. BERT+MTB outperforms the human upper bound on FewRel

	5-way 1-shot	**5-way 5-shot**	**10-way 1-shot**	**10-way 5-shot**
Proto Net	69.2	84.79	56.44	75.55
BERT_{EM}+MTB	**93.9**	**97.1**	**89.2**	**94.3**
Human	92.22	–	85.88	–

Evaluation: Supervised Relation Extraction

1. BERT+MTB outperforms previous SOTA approaches on three datasets

	SemEval 2010	**KBP37**	**TACRED**
SOTA	84.8	58.8	68.2
BERT_{EM}	89.2	68.3	70.1
BERT_{EM}+MTB	**89.5**	**69.3**	**71.5**

Evaluation: Supervised Relation Extraction

1. BERT+MTB outperforms previous SOTA approaches on three datasets

2. In low-resource cases, MTB training is even more effective, i.e., there is a larger gap between BERT and BERT+MTB.

% of training set	**1%**	**10%**	**20%**	**50%**	**100%**
SemEval 2010 Task 8					
BERT_{EM}	28.6	66.9	75.5	80.3	82.1
BERT_{EM}+MTB	31.2	70.8	76.2	80.4	82.7
KBP-37					
BERT_{EM}	40.1	63.6	65.4	67.8	69.5
BERT_{EM}+MTB	44.2	66.3	67.2	68.8	70.3
TACRED					
BERT_{EM}	32.8	59.6	65.6	69.0	70.1
BERT_{EM}+MTB	43.4	64.8	67.2	69.9	70.6

Discussion

Take-away message: external corpus (e.g., Wikipedia) can serve as an extra training set to pre-train the model. The pre-trained model can work well in low-resource cases.

Question: Do you think the pre-training method is a strong pre-training method or not? Any limitation?

Discussion

Take-away message: external corpus (e.g., Wikipedia) can serve as an extra training set to pre-train the model. The pre-trained model can work well in low-resource cases.

Limitations / Future Directions:

- Stronger training signals? (instead of only 0/1 classification)
  - From Wikidata, we can get a lot of (e1, r, e2) triples. ⇒ We can actually know the relation types given a pair of entities!
- ...