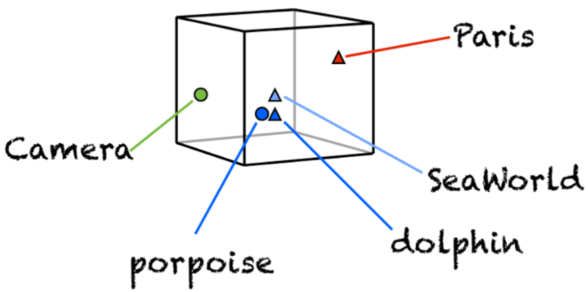
A Hierarchical Multi-Task Approach for Learning Embeddings from Semantic Tasks

Victor Sanh[1], **Thomas Wolf**[1], **Sebastian Ruder**[2,3]

1 Hugging Face, New York, United States
2 Insight Research Centre, National University of Ireland, Galway, Ireland
3 Aylien Ltd., Dublin, Ireland

March 1, 2019



Modern Natural Language Processing rely on **word embeddings**.

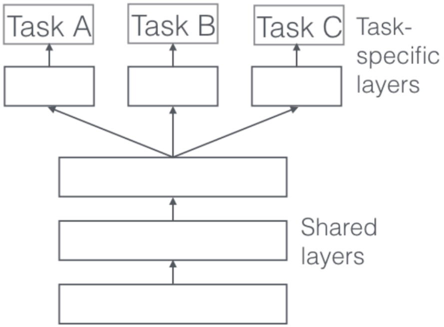
Widely used because they give text representations (almost) for **free** (no need for labeled data).

"Algebra-like" properties:
$king$ - $man + woman = queen$
(Mikolov et al., 2013)

Recent works on **sentence embeddings**.

Quest for **"universal embeddings"** which could be **used across domains** and are **not task specific**. (cf. Conneau et al., 2017)

Quest for **"universal embeddings"** which could be **used across domains** and are **not task specific**. (cf. Conneau et al., 2017)

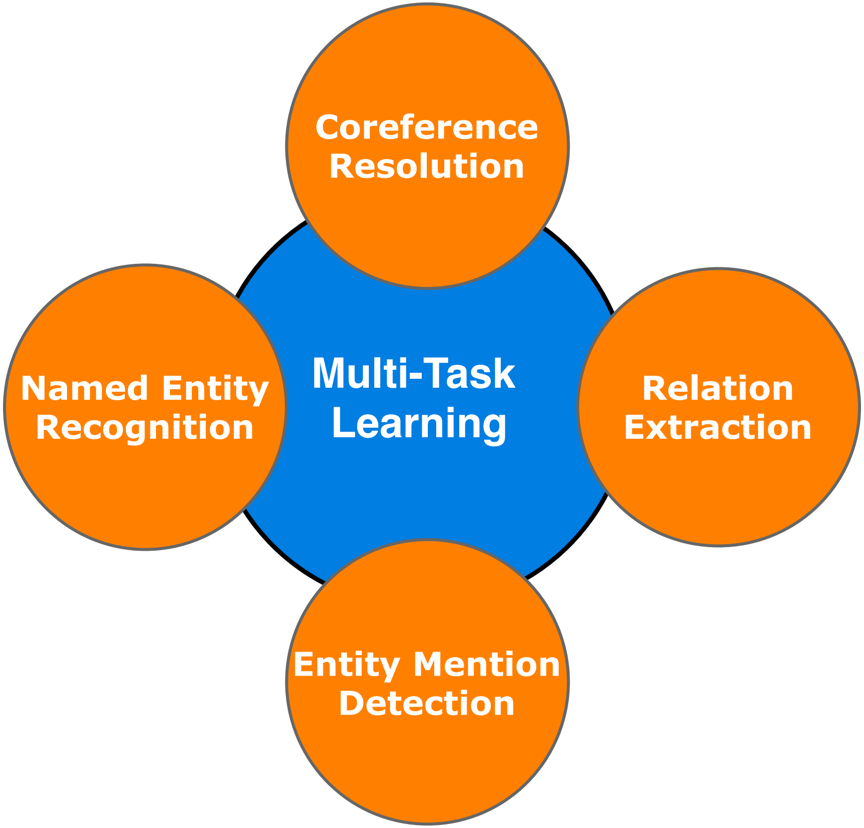


Architecture used in *Learning General Purpose Sentence Representations via Multi-task Learning (Subramanian et al., 2018)*
Source: ruder.io

Shared representation (encoder) followed by task-specific layers.

Weakly related tasks encoding several aspects of a sentence.

Improve generalization in a low-resource context


Coreference Resolution
Named Entity Recognition
Multi-Task Learning
Relation Extraction
Entity Mention Detection

Outline

Plan

***Named entity**: real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name.*

***Named entity**: real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name.*

Taks: Identify and classify named entities
Input: Sentence
Ouput: Named entities and their types in the sentence

[Homer Simpson]$_{PERS}$ lives in [Springfield]$_{LOC}$ with his wife and his three kids.

Mention: *an utterance of a real-world object, person, location, product, etc. It is not necessarily a proper name.*

Mention: *an utterance of a real-world object, person, location, product, etc. It is not necessarily a proper name.*

Taks: Identify and classify entity mentions
Input: Sentence
Ouput: Entity mentions and their types in the sentence

[The men]$_{PERS}$ held on [the sinking vessel]$_{VEH}$ until [the ship]$_{VEH}$ was able to reach them from [Corsica]$_{LOC}$.

Coreference: *the fact that two or more expressions in a text – like pronouns or nouns – link to the same person or thing in the world.*

***Coreference**: the fact that two or more expressions in a text – like pronouns or nouns – link to the same person or thing in the world.*

Taks: Cluster the coreferent spans
Input: One or a few sentences
Ouput: Clusters of the coreferent spans

My mom tasted the cake. She liked it.

Taks: Extract the semantic relations between the mentions
Input: A sentence
Ouput: Relations and their types

$$\overbrace{Homer\ Simpson}^{ARG1}\ is\ the\ head\ of\ \overbrace{the\ power\ plant}^{ARG2}.$$

relation_type: works_for

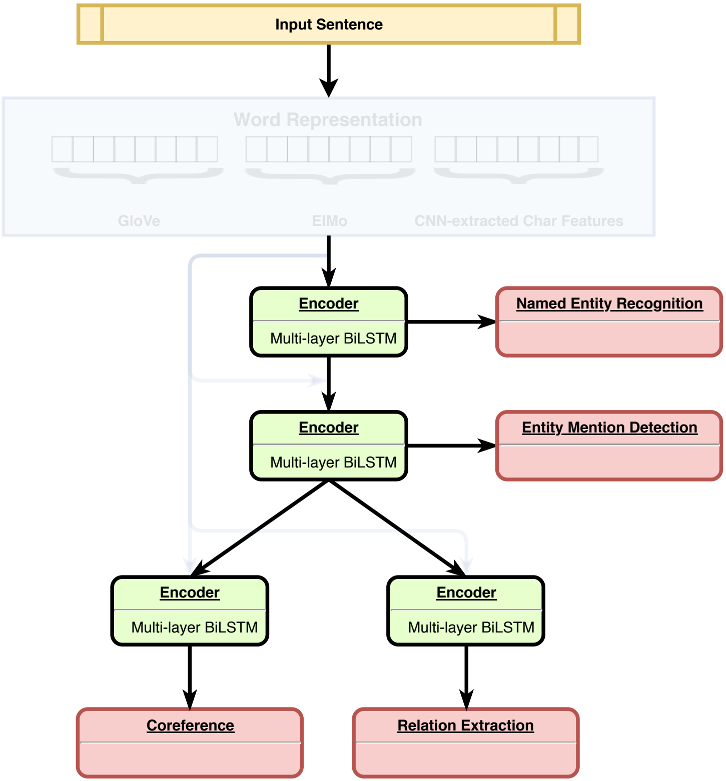
- **Input**: X works for Y
  RE: $\{work,\ X,\ Y\}$
  X $\stackrel{?}{=}$ Person ; Y $\stackrel{?}{=}$ Organization or Person

- **Input**: X works for Y
  RE: $\{work,\ X,\ Y\}$
  X $\stackrel{?}{=}$ Person ; Y $\stackrel{?}{=}$ Organization or Person
- **Input**: I love Melbourne. I've lived three years in this city.
  CR: *(Melbourne, this city)* ; RE: *live_in, I, this city*
  Melbourne $\stackrel{?}{=}$ Location

- **Input**: X works for Y
  RE: $\{work,\ X,\ Y\}$
  X $\stackrel{?}{=}$ Person ; Y $\stackrel{?}{=}$ Organization or Person
- **Input**: I love Melbourne. I've lived three years in this city.
  CR: *(Melbourne, this city)* ; RE: *live_in, I, this city*
  Melbourne $\stackrel{?}{=}$ Location
- **Input**: Dell announced a $500 millions net loss. The company is near bankruptcy.
  CR: *(Dell, the company)*
  Dell $\stackrel{?}{=}$ Organization (and not an Person).

Plan

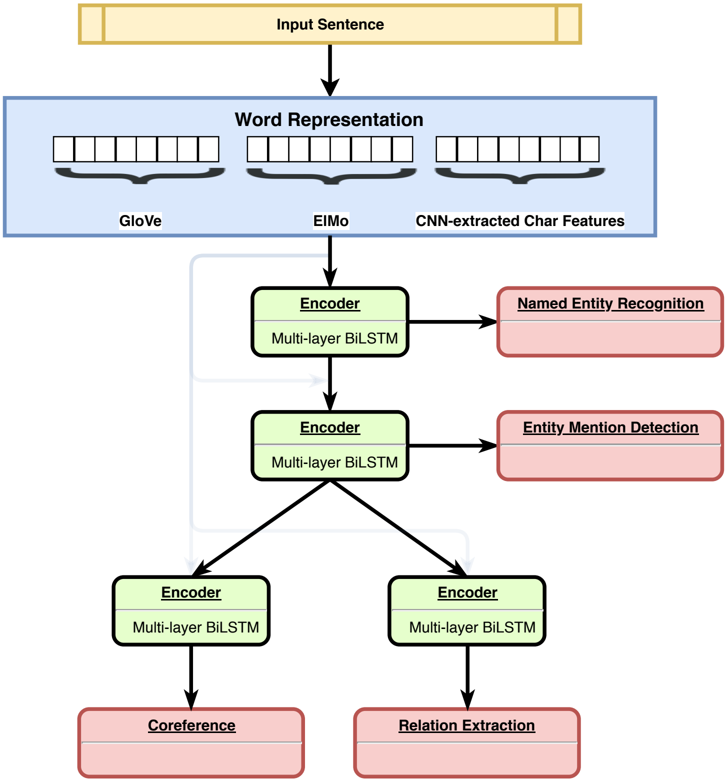
- We propose a **hierarchical multi-task model** that doesn't rely on any external linguistic tool (parsers...)
- We introduce a **new sampling strategy** for multi-task learning (*proportional sampling*)
- **State-of-the-art results** on three different tasks (NER, EMD, RE)
- **Analysis of the influence of multi-task learning** (embeddings and training speed)



- Several prior works do not take into account the **linguistic hierarchies between tasks**.
- **"Low-level" tasks** are supervised at lower layers of the model, and more complex (**"higher-level"**) tasks at higher layers.

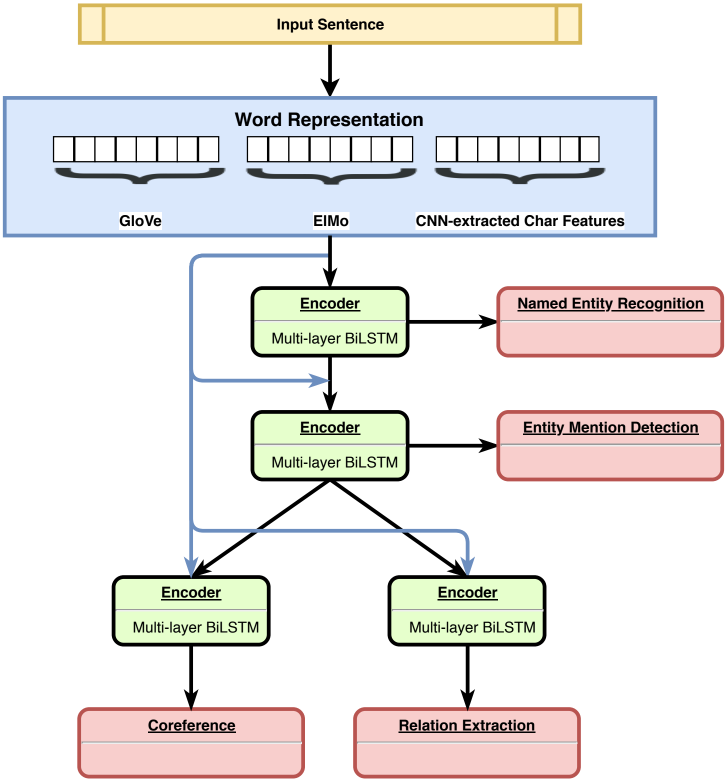
The model

Base word embeddings

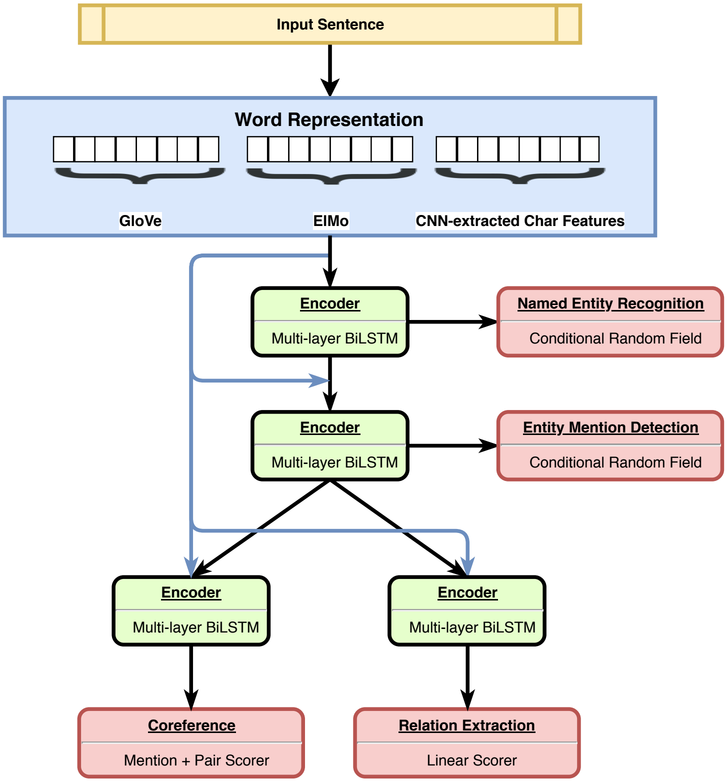


We use three types of embeddings:

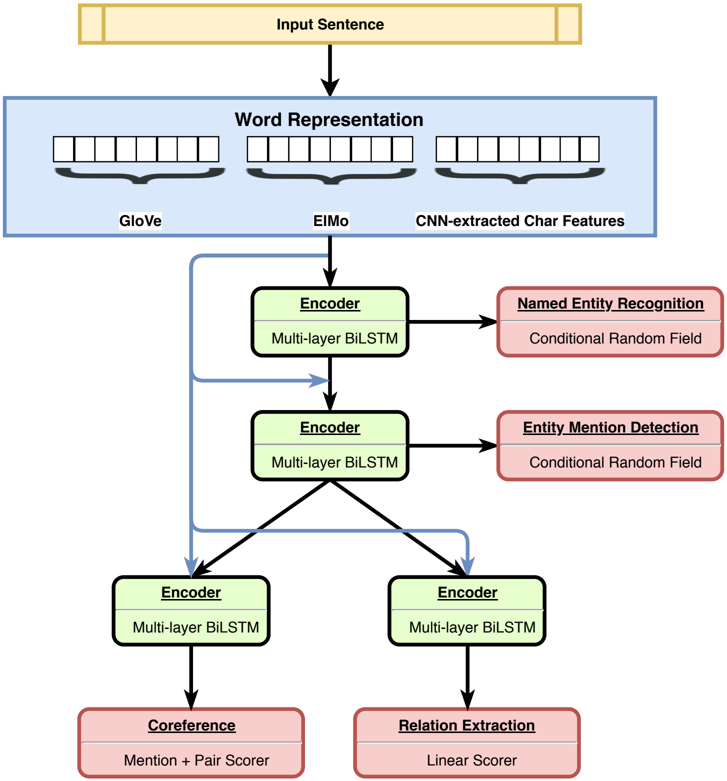
- Pre-trained GloVe word embeddings (fine-tuned)
- ELMo contextualized word embeddings (frozen)
- Learned character-level word embeddings



- Short-cut connections were introduced by Hashimoto et al. (2017)
- All the layers can benefit from the same shared base representation.



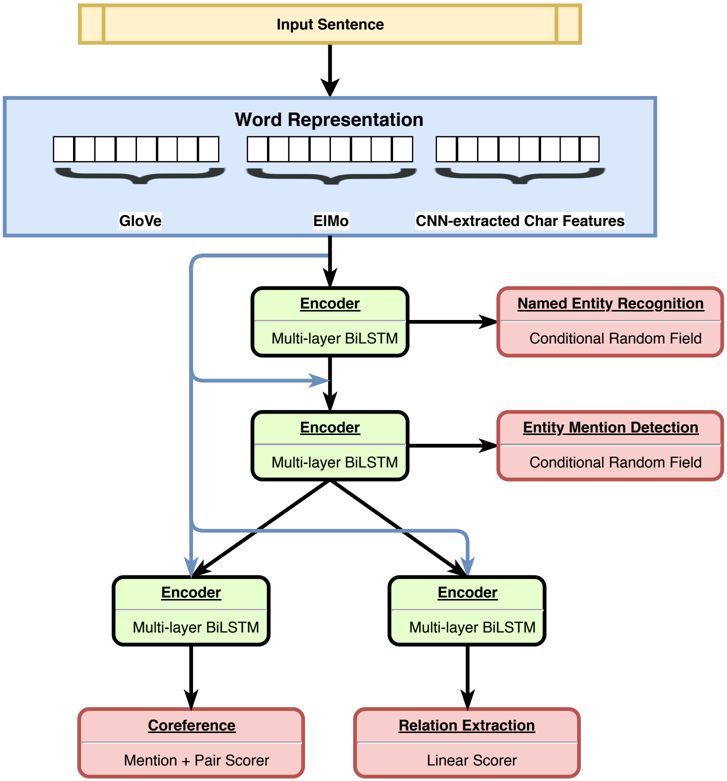
- **Conditional Random Field** (Lafferty et al., 2001) for NER and EMD (formulated as a sequence tagging task)
- **Linear Scorer followed by a sigmoid activation** for RE (Bekoulis et al., 2017)
- **Liner Scorer and Mention Pair Scorer** (Lee et al., 2017)



Requires:

- k tasks and k datasets
- Sampling probability distribution $(p_1, p_2, ...p_k)$

1: **while** θ has not converged **do**
2: A. Sample a task $j \sim (p_1, p_2, ...p_k)$.
B. Sample one batch from the j-th dataset
C. Optimize toward the j-th task for one update (ADAM optimizer).
3: **end while**



Requires:

- k tasks and k datasets
- Sampling probability distribution $(p_1, p_2, ...p_k)$

1: **while** θ has not converged **do**
2: A. Sample a task $j \sim (p_1, p_2, ...p_k)$.
B. Sample one batch from the j-th dataset
C. Optimize toward the j-th task for one update (ADAM optimizer).
3: **end while**

Example of proportional sampling:
Task 1: 10 batches, Task 2: 30 batches
$\Rightarrow p_1 = 0.25$; $p_2 = 0.75$

Plan

Overall Performance

The benefits of using Multi-Task Learning—Single task VS. Multi-Task

Table 1: Comparing single tasks and multi-task performances. For coreference, comparable figures are tagged with an *.

Setup	Model	NER - F_1	EMD - F_1	RE - F_1	CR - Avg. F_1
	Strubell (2017)	86.99	-	-	-
	Katiyar (2017)	-	82.6	55.9	-
	Miwa (2016)	-	83.4	55.6	-
	Li (2014)	-	80.8	52.1	-
	Durrett (2014)	-	-	-	76.16*
(A)	Full Model	87.36	85.69	61.30	64.78
(A-GM)	Full Model - GM	87.10	**87.24**	**62.69**	70.29*
(B)	NER	87.12	-	-	-
(C)	EMD	-	86.14	-	-
(D)	RE	-	-	55.99	-
(E)	CR	-	-	-	65.67
(E-GM)	CR - GM	-	-	-	69.38*

- State-of-the-art on Entity Mention Detection and Relation Extraction.

Overall Performance

The benefits of using Multi-Task Learning—Single task VS. Multi-Task

Table 1: Comparing single tasks and multi-task performances. For coreference, comparable figures are tagged with an *.

Setup	Model	NER - F_1	EMD - F_1	RE - F_1	CR - Avg. F_1
	Strubell (2017)	86.99	-	-	-
	Katiyar (2017)	-	82.6	55.9	-
	Miwa (2016)	-	83.4	55.6	-
	Li (2014)	-	80.8	52.1	-
	Durrett (2014)	-	-	-	76.16*
(A)	Full Model	87.36	85.69	61.30	64.78
(A-GM)	Full Model - GM	87.10	**87.24**	**62.69**	70.29*
(B)	NER	87.12	-	-	-
(C)	EMD	-	86.14	-	-
(D)	RE	-	-	55.99	-
(E)	CR	-	-	-	65.67
(E-GM)	CR - GM	-	-	-	69.38*

- State-of-the-art on Entity Mention Detection and Relation Extraction.
- Multi-task (almost) always outperforms a single task setting.
- Strongest gap is observed on Relation Extraction (+6 F_1 points).

Table 2: Adding more tasks to the model

Setup	Model	NER - F_1	EMD - F_1	RE - F_1	CR - Avg. F_1
(A)	Full Model	87.36	85.69	61.30	64.78
(A-GM)	Full Model - GM	87.10	**87.24**	**62.69**	70.29*
(B)	NER	87.12	-	-	-
(C)	EMD	-	86.14	-	-
(D)	RE	-	-	55.99	-
(E)	CR	-	-	-	65.67
(E-GM)	CR - GM	-	-	-	69.38*
(F)	NER + EMD	$\boxed{86.91}$	$\boxed{86.02}$	-	-
(G)	EMD + RE	-	85.50	60.49	-
(H)	EMD + CR	-	85.65	-	63.02
(I)	NER + EMD + RE	$\boxed{\mathbf{87.51}}$	$\boxed{86.26}$	60.18	-
(J)	NER + EMD + CR	87.50	85.87	-	66.64

- RE can help both NER and EMD.

Table 3: Adding more tasks to the model

Setup	Model	NER - F_1	EMD - F_1	RE - F_1	CR - Avg. F_1
(A)	Full Model	$\boxed{87.36}$	85.69	61.30	64.78
(A-GM)	Full Model - GM	87.10	**87.24**	**62.69**	70.29*
(B)	NER	87.12	-	-	-
(C)	EMD	-	86.14	-	-
(D)	RE	-	-	55.99	-
(E)	CR	-	-	-	65.67
(E-GM)	CR - GM	-	-	-	69.38*
(F)	NER + EMD	$\boxed{86.91}$	86.02	-	-
(G)	EMD + RE	-	85.50	60.49	-
(H)	EMD + CR	-	85.65	-	63.02
(I)	NER + EMD + RE	**87.51**	86.26	60.18	-
(J)	NER + EMD + CR	87.50	85.87	-	66.64

- RE can help both NER and EMD.
- RE and CR can help NER.

Table 4: Adding more tasks to the model

Setup	Model	NER - F_1	EMD - F_1	RE - F_1	CR - Avg. F_1
(A)	Full Model	87.36	85.69	61.30	64.78
(A-GM)	Full Model - GM	87.10	**87.24**	**62.69**	70.29*
(B)	NER	87.12	-	-	-
(C)	EMD	-	86.14	-	-
(D)	RE	-	-	55.99	-
(E)	CR	-	-	-	65.67
(E-GM)	CR - GM	-	-	-	69.38*
(F)	NER + EMD	86.91	86.02	-	-
(G)	EMD + RE	-	85.50	60.49	-
(H)	EMD + CR	-	85.65	-	63.02
(I)	NER + EMD + RE	**87.51**	86.26	60.18	-
(J)	NER + EMD + CR	87.50	85.87	-	66.64

- RE can help both NER and EMD.
- RE and CR can help NER.
- CR can help NER.

The information flowing from higher levels helps lower levels learn better representation.

Table 5: Playing with the hierarchy order.

Setup	Model	NER - F_1	EMD - F_1	RE - F_1	CR - Avg. F_1
(A)	Full Model	87.36	85.69	61.30	64.78
(L)	EMD + NER + RE + CR (Δ)	-1.15	-0.55	-2.13	-0.61
(F)	NER + EMD	86.91	86.02	-	-
(K)	EMD + NER (Δ)	-0.48	-0.83	-	-

- Switching the order between NER and EMD.
- Drop of performance for all tasks.
- It suggests that the hierarchy should follow the difficulty of the tasks.

Table 6: Comparison to other canonical datasets on NER (CoNLL-2003) and coreference (CoNLL-2012).

Model	NER (F_1)	EMD (F_1)	RE (F_1)	CR (F_1)
Lample 2016	90.94	-	-	-
Strubell 2017	90.54	-	-	-
Peter 2018	**92.22**	-	-	-
(A-CoNLL-2003)	91.63	86.53	60.83	70.14
Durret 2014	-	-	-	61.71
Lee 2017 (single)	-	-	-	67.2
Lee 2017 (ensemble)	-	-	-	**68.8**
(A-CoNLL-2012)	86.90	85.04	61.07	62.48

Performances are **mostly independent of the dataset**: similar performances when changing datasets.

Table 7: Ablation study on the embeddings.

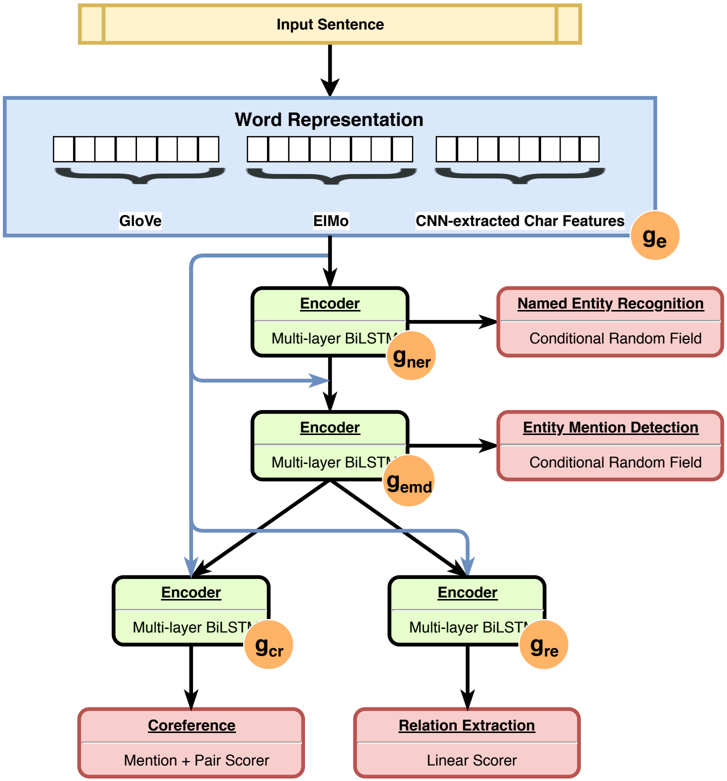
Model	NER (F_1)	EMD (F_1)	RE (F_1)	CR (F_1)
Glove + Char. embds + ELMo	87.10	87.24	62.69	70.29
Glove + Char. embds (Δ)	-3.67	-4.11	-5.22	-3.85
Glove (Δ)	-4.52	-0.13	-3.70	-2.18

- Removing ELMo leads to a $\sim$4 F_1 points drop on each task.
- Strong impact of character-level embeddings (morphological features) especially on NER, RE and CR.

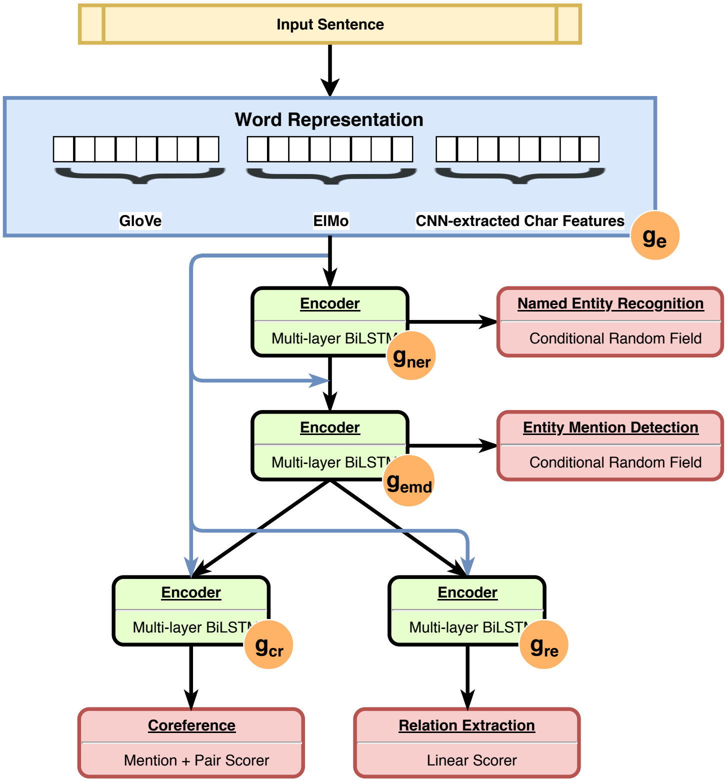
- High scores on a specific task suggests that the encoders/embeddings learned **relevant linguistic information** for the task.
- Sometimes it is **hard to analyze on what kind of linguistic features a model rely**.
- Conneau et al. (2018) introduce **10 "elementary" tasks** (called *probing tasks*) that focus on specific linguistic aspects of a sentence (surface, syntatic, and semantic information) to **evaluate the quality of sentence embeddings**.

What did the embeddings learn?

SentEval results



What did the embeddings learn?
SentEval results



From word embeddings to sentence embeddings:

- For word embeddings: average or max pooling (Arora et al., 2017)
- For encoders: max pooling (Conneau et al, 2018)

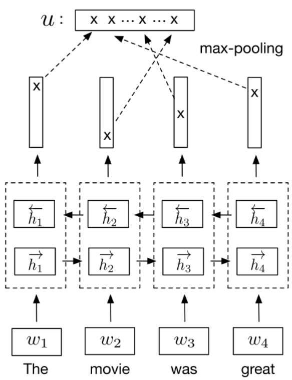


Figure 2: Max pooling the hidden states of biLSTM encoder

What did the embeddings learn?

SentEval results

Table 8: SentEval Probing task accuracies.

Tasks	Surface Information		Syntatic Information			Semantic Information				
	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
Word Embeddings										
Bov-fastText (Conneau et al., 2018)	54.8	91.6	32.3	63.1	50.8	87.8	81.9	79.3	50.3	52.7
Our model (g_e) - Max	62.4	43.0	32.5	76.3	74.5	88.1	85.7	82.7	54.7	56.9
Our model (g_e) - Average	72.1	70.0	38.5	79.9	81.4	89.7	88.5	86.5	57.4	63.0
BiLSTM-max encoders										
SkipThought (Conneau et al., 2018)	59.6	35.7	42.7	70.5	73.4	90.1	83.3	79.0	70.3	70.1
Our model (Encoder NER g_{ner})	50.7	3.24	19.5	34.2	57.2	66.6	63.5	61.6	50.7	52.0
Our model (Encoder EMD g_{emd})	43.3	1.8	19.3	30.0	56.3	64.0	60.1	57.9	51.3	50.4
Our model (Encoder RE g_{re})	56.8	1.2	19.3	24.5	53.9	62.3	60.8	57.1	50.4	52.2
Our model (Encoder CR g_{cr})	61.9	11.0	29.5	55.9	70.0	82.8	83.0	76.5	53.3	58.7

- The base representation (g_e) is already extremely rich.
- Significant discrepancies between the results of the word embeddings g_e and the encoder representations (g_{ner}, g_{emd}, g_{re}, and g_{cr}).
- CR encoder (g_{cr}) always have the best performances among all encoders.

Table 9: Speed of training: Difference in number of updates necessary before convergence. Multi-task VS. Single task.

Setup	Model	Time Δ	Performance Δ
(B)	NER	-16%	-0.02
(C)	EMD	-44%	+1.14
(D)	RE	+78%	+6.76
(E-GM)	Coref-GM	-28%	+0.91

We compare the training speed (in terms of number of updates before convergence) in the multi-task setting and the single task setting.

Multi-task learning **accelerates the training** while **improving the generalization power**.

Conclusion

- We introduced a **hierarchically supervised multi-task learning model** focused on **semantic tasks**.
- We achieved **state-of-the-art results** on Named Entity Recognition, Entity Mention Detection an Relation Extraction.
- We introduced a **simple training strategy** (proportional sampling).
- We analyzed the **influence of a multi-task learning setting** and the type of **information encoded** in the model.

Conclusion

- We introduced a **hierarchically supervised multi-task learning model** focused on **semantic tasks**.
- We achieved **state-of-the-art results** on Named Entity Recognition, Entity Mention Detection an Relation Extraction.
- We introduced a **simple training strategy** (proportional sampling).
- We analyzed the **influence of a multi-task learning setting** and the type of **information encoded** in the model.