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

Victor Sanh\textsuperscript{1}, Thomas Wolf\textsuperscript{1}, Sebastian Ruder\textsuperscript{2,3}

\textsuperscript{1} Hugging Face, New York, United States
\textsuperscript{2} Insight Research Centre, National University of Ireland, Galway, Ireland
\textsuperscript{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:

\[ \text{king} - \text{man} + \text{woman} = \text{queen} \]

(Mikolov et al., 2013)

Recent works on **sentence embeddings**.
Introduction
The quest for universal embeddings

Quest for “universal embeddings” which could be used across domains and are not task specific. (cf. Conneau et al., 2017)
Introduction
The quest for universal embeddings

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

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

Weakly related tasks encoding several aspects of a sentence.

Improve generalization in a low-resource context

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

2 Motivations
   • The tasks
   • Relatedness of tasks

3 The model
   • A hierarchical model
   • The training procedure

4 Results
   • Overall Performance
   • What did the embeddings learn?
   • Multi-Task Learning accelerates the training
Plan

1 Introduction

2 Motivations
   - The tasks
   - Relatedness of tasks

3 The model
   - A hierarchical model
   - The training procedure

4 Results
   - Overall Performance
   - What did the embeddings learn?
   - Multi-Task Learning accelerates the training
Motivations
The tasks—Named Entity Recognition (NER)

**Named entity**: real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name.
Motivations
The tasks—Named Entity Recognition (NER)

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

**Tasks:** Identify and classify named entities
**Input:** Sentence
**Output:** Named entities and their types in the sentence

[Homer Simpson]*PERS* lives in [Springfield]*LOC* with his wife and his three kids.
Motivations
The tasks—Entity Mention Detection (EMD)

**Mention**: an utterance of a real-world object, person, location, product, etc. It is not necessarily a proper name.
Motivations
The tasks—Entity Mention Detection (EMD)

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

**Tasks**: Identify and classify entity mentions
**Input**: Sentence
**Output**: Entity mentions and their types in the sentence

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


Motivations
The tasks—Coreference Resolution (CR)

**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.
Motivations
The tasks—Coreference Resolution (CR)

*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.

**Tasks**: Cluster the coreferent spans

**Input**: One or a few sentences

**Output**: Clusters of the coreferent spans

My mom tasted the cake. She liked it.
Motivations
The tasks—Relation Extraction (RE)

**Tasks:** Extract the semantic relations between the mentions

**Input:** A sentence

**Output:** Relations and their types

\[
\text{Homer Simpson is the head of the power plant.}
\]

relation_type: works_for
Motivations
Relatedness of tasks

- **Input:** X works for Y
  RE: \{work, X, Y\}
  X \(?\) = Person ; Y \(?\) = Organization or Person
Motivations
Relatedness of tasks

- **Input**: X works for Y
  RE: \{work, X, Y\}
  X ?= Person ; Y ?= 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 ?= Location
Motivations
Relatedness of tasks

- **Input**: X works for Y
  RE: \{work, X, Y\}
  X \equiv \text{Person} ; Y \equiv \text{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 \equiv \text{Location}

- **Input**: Dell announced a $500 millions net loss. The company is near bankruptcy.
  CR: (Dell, the company)
  Dell \equiv \text{Organization (and not an Person).}
Plan

1. Introduction

2. Motivations
   - The tasks
   - Relatedness of tasks

3. The model
   - A hierarchical model
   - The training procedure

4. Results
   - Overall Performance
   - What did the embeddings learn?
   - Multi-Task Learning accelerates the training
The model

Our contribution

- 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.
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.
The model

The different modules

- **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)
- **Linear Scorer and Mention Pair Scorer** (Lee et al., 2017)
The model
The training procedure

Requires:

- \( k \) tasks and \( k \) datasets
- Sampling probability distribution \((p_1, p_2, \ldots, p_k)\)

1: \textbf{while} \( \theta \) has not converged \textbf{do} 
2: \hspace{1cm} A. Sample a task \( j \sim (p_1, p_2, \ldots, p_k) \).
   B. Sample one batch from the \( j \)-th dataset
   C. Optimize toward the \( j \)-th task for one update (ADAM optimizer).
3: \textbf{end while}
The model

The training procedure

---

Requires:

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

1: while $\theta$ has not converged do
2: \hspace{1cm} A. Sample a task $j \sim (p_1, p_2, \ldots, p_k)$.
   \hspace{1cm} B. Sample one batch from the $j$-th dataset.
   \hspace{1cm} 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

1 Introduction

2 Motivations
   - The tasks
   - Relatedness of tasks

3 The model
   - A hierarchical model
   - The training procedure

4 Results
   - Overall Performance
   - What did the embeddings learn?
   - Multi-Task Learning accelerates the training
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 *.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Model</th>
<th>NER - $F_1$</th>
<th>EMD - $F_1$</th>
<th>RE - $F_1$</th>
<th>CR - Avg. $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>Full Model</td>
<td>87.36</td>
<td>85.69</td>
<td>61.30</td>
<td>64.78</td>
</tr>
<tr>
<td>(A-GM)</td>
<td>Full Model - GM</td>
<td>87.10</td>
<td><strong>87.24</strong></td>
<td><strong>62.69</strong></td>
<td><strong>70.29</strong>*</td>
</tr>
<tr>
<td>(B)</td>
<td>NER</td>
<td><strong>87.12</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(C)</td>
<td>EMD</td>
<td>-</td>
<td><strong>86.14</strong></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(D)</td>
<td>RE</td>
<td>-</td>
<td>-</td>
<td>55.99</td>
<td>-</td>
</tr>
<tr>
<td>(E)</td>
<td>CR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>65.67</td>
</tr>
<tr>
<td>(E-GM)</td>
<td>CR - GM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td><strong>69.38</strong>*</td>
</tr>
</tbody>
</table>

- State-of-the-art on Entity Mention Detection and Relation Extraction.
Table 1: Comparing single tasks and multi-task performances. For coreference, comparable figures are tagged with an *.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Model</th>
<th>NER - $F_1$</th>
<th>EMD - $F_1$</th>
<th>RE - $F_1$</th>
<th>CR - Avg. $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>Full Model</td>
<td>87.36</td>
<td>85.69</td>
<td>61.30</td>
<td>64.78</td>
</tr>
<tr>
<td>(A-GM)</td>
<td>Full Model - GM</td>
<td>87.10</td>
<td>87.24</td>
<td>62.69</td>
<td>70.29*</td>
</tr>
<tr>
<td>(B)</td>
<td>NER</td>
<td>87.12</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(C)</td>
<td>EMD</td>
<td>-</td>
<td>86.14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(D)</td>
<td>RE</td>
<td>-</td>
<td>-</td>
<td>55.99</td>
<td>-</td>
</tr>
<tr>
<td>(E)</td>
<td>CR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>65.67</td>
</tr>
<tr>
<td>(E-GM)</td>
<td>CR - GM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>69.38*</td>
</tr>
</tbody>
</table>

- 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).
Overall Performance
The benefits of using Multi-Task Learning—Adding more tasks

**Table 2: Adding more tasks to the model**

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

- RE can help both NER and EMD.
Overall Performance
The benefits of using Multi-Task Learning—Adding more tasks

Table 3: Adding more tasks to the model

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

- RE can help both NER and EMD.
- RE and CR can help NER.
Overall Performance
The benefits of using Multi-Task Learning—Adding more tasks

Table 4: Adding more tasks to the model

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

- 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.
Overall Performance

The hierarchy order

Table 5: Playing with the hierarchy order.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Model</th>
<th>NER - $F_1$</th>
<th>EMD - $F_1$</th>
<th>RE - $F_1$</th>
<th>CR - Avg. $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>Full Model</td>
<td>87.36</td>
<td>85.69</td>
<td>61.30</td>
<td>64.78</td>
</tr>
<tr>
<td>(L)</td>
<td>EMD + NER + RE + CR ($\Delta$)</td>
<td>-1.15</td>
<td>-0.55</td>
<td>-2.13</td>
<td>-0.61</td>
</tr>
<tr>
<td>(F)</td>
<td>NER + EMD</td>
<td>86.91</td>
<td>86.02</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(K)</td>
<td>EMD + NER ($\Delta$)</td>
<td>-0.48</td>
<td>-0.83</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- 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.
Overall Performance
Comparison to other canonical datasets

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

<table>
<thead>
<tr>
<th>Model</th>
<th>NER ($F_1$)</th>
<th>EMD ($F_1$)</th>
<th>RE ($F_1$)</th>
<th>CR ($F_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lample 2016</td>
<td>90.94</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Strubell 2017</td>
<td>90.54</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Peter 2018</td>
<td>92.22</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(A-CoNLL-2003)</td>
<td>91.63</td>
<td>86.53</td>
<td>60.83</td>
<td>70.14</td>
</tr>
<tr>
<td>Durret 2014</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>61.71</td>
</tr>
<tr>
<td>Lee 2017 (single)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>67.2</td>
</tr>
<tr>
<td>Lee 2017 (ensemble)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td><strong>68.8</strong></td>
</tr>
<tr>
<td>(A-CoNLL-2012)</td>
<td>86.90</td>
<td>85.04</td>
<td>61.07</td>
<td>62.48</td>
</tr>
</tbody>
</table>

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

Effects of the embeddings

Table 7: Ablation study on the embeddings.

<table>
<thead>
<tr>
<th>Model</th>
<th>NER ($F_1$)</th>
<th>EMD ($F_1$)</th>
<th>RE ($F_1$)</th>
<th>CR ($F_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glove + Char. embds + ELMo</td>
<td>87.10</td>
<td>87.24</td>
<td>62.69</td>
<td>70.29</td>
</tr>
<tr>
<td>Glove + Char. embds (∆)</td>
<td>-3.67</td>
<td>-4.11</td>
<td>-5.22</td>
<td>-3.85</td>
</tr>
<tr>
<td>Glove (∆)</td>
<td>-4.52</td>
<td>-0.13</td>
<td>-3.70</td>
<td>-2.18</td>
</tr>
</tbody>
</table>

- 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.
What did the embeddings learn?
Introduction to SentEval (Conneau et al., 2018)

- 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, syntactic, and semantic information) to evaluate the quality of sentence embeddings.
What did the embeddings learn?

SentEval results

![Diagram showing the process from input sentence to various tasks]

- **Input Sentence**
- **Word Representation**
  - GloVe
  - ElMo
  - CNN-extracted Char Features
- **Encoder**: Multi-layer BiLSTM
- **Named Entity Recognition**: Conditional Random Field
- **Entity Mention Detection**: Conditional Random Field
- **Coreference**: Mention + Pair Scorer
- **Relation Extraction**: Linear Scorer

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
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.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Surface Information</th>
<th>Syntactic Information</th>
<th>Semantic Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SentLen</td>
<td>WC</td>
<td>TreeDepth</td>
</tr>
<tr>
<td><strong>Word Embeddings</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bov-fastText (Conneau et al., 2018)</td>
<td>54.8</td>
<td>91.6</td>
<td>32.3</td>
</tr>
<tr>
<td>Our model ((g_e) - Max)</td>
<td>62.4</td>
<td>43.0</td>
<td>32.5</td>
</tr>
<tr>
<td>Our model ((g_e) - Average)</td>
<td>72.1</td>
<td>70.0</td>
<td>38.5</td>
</tr>
<tr>
<td><strong>BiLSTM-max encoders</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SkipThought (Conneau et al., 2018)</td>
<td>59.6</td>
<td>35.7</td>
<td>42.7</td>
</tr>
<tr>
<td>Our model (Encoder NER (g_{ner}))</td>
<td>50.7</td>
<td>3.24</td>
<td>19.5</td>
</tr>
<tr>
<td>Our model (Encoder EMD (g_{emd}))</td>
<td>43.3</td>
<td>1.8</td>
<td>19.3</td>
</tr>
<tr>
<td>Our model (Encoder RE (g_{re}))</td>
<td>56.8</td>
<td>1.2</td>
<td>19.3</td>
</tr>
<tr>
<td>Our model (Encoder CR (g_{cr}))</td>
<td>61.9</td>
<td>11.0</td>
<td>29.5</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Setup</th>
<th>Model</th>
<th>Time Δ</th>
<th>Performance Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B)</td>
<td>NER</td>
<td>-16%</td>
<td>-0.02</td>
</tr>
<tr>
<td>(C)</td>
<td>EMD</td>
<td>-44%</td>
<td>+1.14</td>
</tr>
<tr>
<td>(D)</td>
<td>RE</td>
<td>+78%</td>
<td>+6.76</td>
</tr>
<tr>
<td>(E-GM)</td>
<td>Coref-GM</td>
<td>-28%</td>
<td>+0.91</td>
</tr>
</tbody>
</table>

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