Towards similarity and inclusion metrics for task comparison

Loïc Fosse

Orange Innovation, Lannion, France
Aix Marseille Univ., CNRS, LIS, Marseille, France

loic.fosse@orange.com

Introduction and motivations

Fine-tuning is the current dominating approach.

- for every task, there exists a fine-tuned model
- however, some tasks share some knowledge ⇒ we can re-use one task to do another

Our research question.

- How can we quantify, the shared knowledge between two tasks ?

Our proposal. use of task vectors for

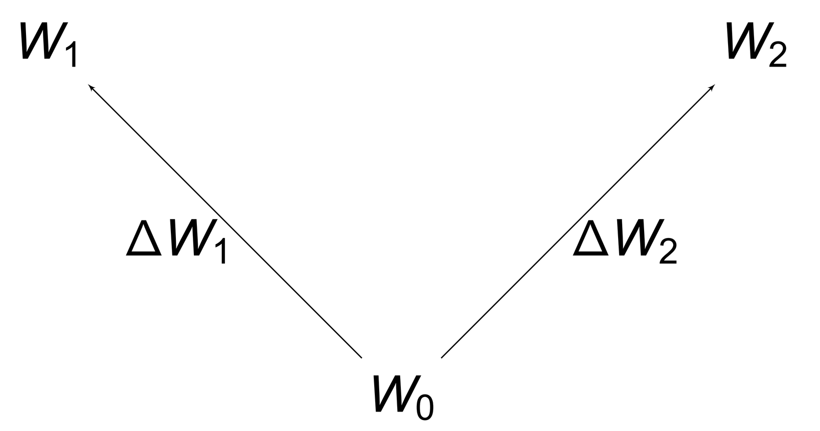
- Task similarity
- Task projection

Task vectors

- W_0 : pre-trained models
- W_t : model after fine-tuning on a task t

Task vector is defined by:

$$\Delta W_t = W_t - W_0$$



Link with Low Rank Adaptation

Let $W \in \mathbb{R}^{d\times d}$ a linear layer:

- **full-finetuning.** $W_t = W_0 + \Delta W$ avec $\Delta W \in \mathbb{R}^{d\times d}$
- **LoRA.** $W_t = W_0 + BA$ with $A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{d\times r}$ $(r << d)$

LoRA : low dimension estimation of task vectors

In practice:

- LoRA is performed on every linear layer (good for transformers)
- LoRA is performed on **queries** and **values**

Grassmann distance

$$d_G(\Delta W_1, \Delta W_2) = d(Im(\Delta W_1), Im(\Delta W_2))$$

Algorithm

1. $U_1 = orth(\Delta W_1)$ and $U_2 = orth(\Delta W_2)$
2. $G = U_1^t U_2$ (cosine similarity matrices)
3. $\sigma = sp(G)$
4. Grassmann $= \sum_{x\in\sigma} arccos(x)$

In our case the distance is calculated between low rank modules. In the definition, we can see that Grassmann distance is close to the cosine similarity.

Spectral projection

S.V.D Theorem gives us the following representation:

$$\Delta W_1 = US_1V^h$$

Interpretation:

- U, V: space for task 1
- S_1: how the space is used

We sick to represent ΔW_2 on ΔW_1 space ⇒ solve the following problem:

$$S_2 = \underset{S\in\mathcal{S}}{\arg\min}\ \|USV^h - \Delta W_2\|_F\ ,$$

Measure proximity between the different use of the spaces:

- $L_p(S_1, S_2) = \|S_1 - S_2\|_p = \left(\sum_i |S_1(i) - S_2(i)|^p\right)^{1/p}$ with $p \geq 1$
- $\mathrm{REC}(S_1, S_2) = \sum_i \min\left(\frac{S_1(i)}{\sum S_1}, \frac{S_2(i)}{\sum S_2}\right)$ (Bayesian error rate between spectrum)

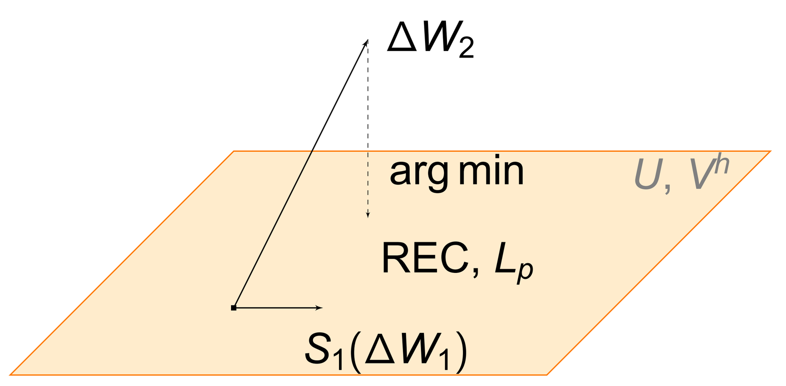


Figure: Spectral projection illustration.

Models

- Mistral 7B (Instruct and Base)
- Llama 3 8B (Instruct and Base)
- RoBERTa Base (on classification tasks)

Data: OnToNotes, linguistic gold annotations

Coreferences

in. In the following document, **keep sentences with longest coreference chains and replace coreference with anchor**. ### Document: First the news update. Here's David Coler. etc.

out. President Clinton is sending US mediator Dennis Ross back to the Middle East in yet another effort to make progress toward peace between Israel and the Palestinians before Mr. Clinton leaves office in two weeks. etc.

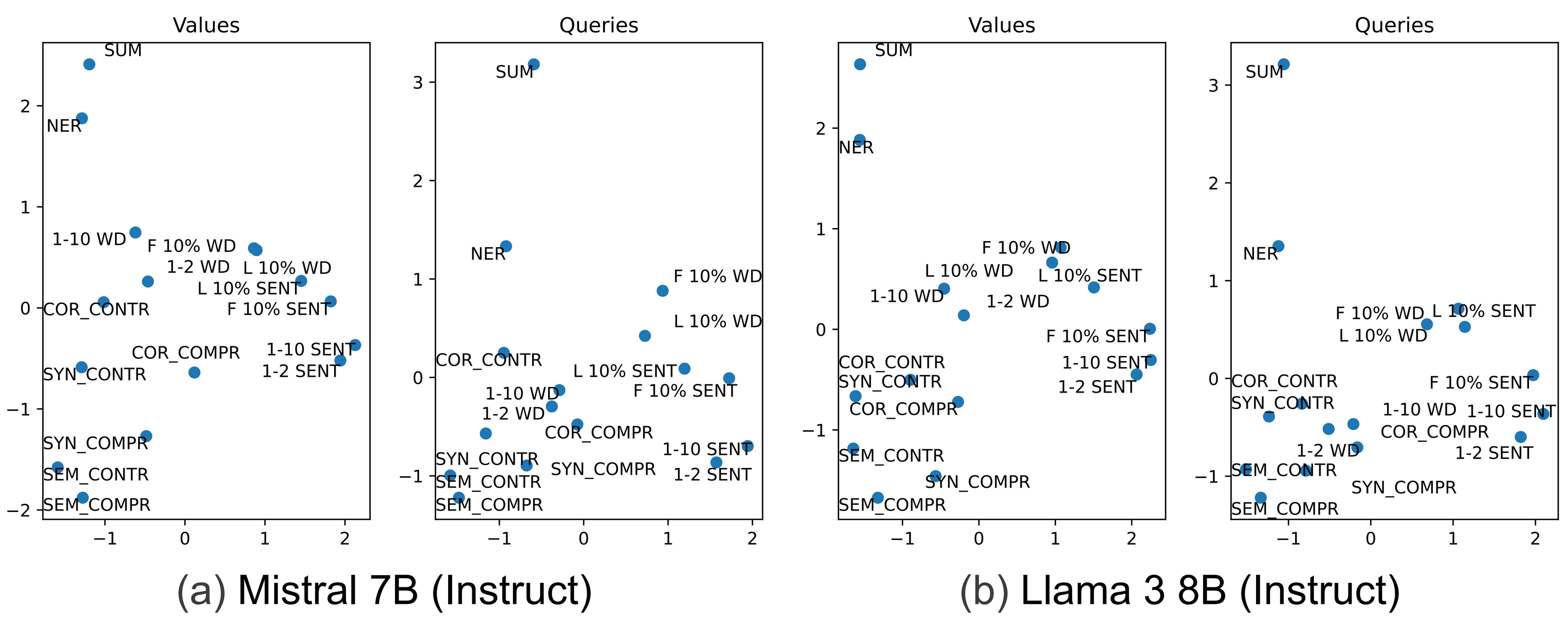
Semantic

in. In the following document, **build sentences with only ARG0 VERB ARG1 ARGM-TMP**. ### Document: First the news update. Here's David Coler. etc.

out. President Clinton sending US mediator. A senior White House official said Mr. etc.

Grassmann distance results

- $T(i, j, l)$: distance between task i and j on layer l
- $\bar{T}(i, j) = \frac{1}{L}\sum_l T(i, j, l)$ (mean pooling over the layers)
- PCA of $\bar{T}$



(a) Mistral 7B (Instruct) (b) Llama 3 8B (Instruct)

Figure: $PCA(\bar{T})$

Take Away:

- **Linguistic tasks.** cluster with Grassmann distance.
- **Counting tasks.** cluster with Grassmann distance.
- **Cosine similarity.** very close results.

Results of spectral projections

Chosen tasks:

- Summarization (SUM) : act as a general task.
- Named Entity Recognition (NER): supposed to be included in the summary.
- Select the first 10% sentences (10% SENT): act as a control task.

	Instruct		Base	
task	L_1	REC	L_1	REC
SUM→NER	0.271	0.644	0.493	0.667
SUM→10% SENT	0.277	0.608	0.501	0.602
10% SENT→NER	0.435	0.599	0.459	0.615
10% SENT→SUM	0.440	0.562	0.461	0.580
NER→10% SENT	0.660	0.519	0.676	0.557
NER→SUM	0.660	0.535	0.673	0.582

Table: Average L_1 and REC for combinations between, NER, SUM and 10% SENT

	Instruct		Base	
tasks	$H(S_1)$	$H(S_2)$	$H(S_1)$	$H(S_2)$
SUM→NER	0.852	0.839	0.868	0.841
NER→SUM	0.774	0.867	0.779	0.872

Table: Average (across layers and modules) of the Shannon entropy. The space of summary is much more diffused (more dimension are used)

Take away messages

- **A metric for task similarity.** Similar tasks provide similar task vectors. Other distances were tried (L_2, cos, Frobenius)
- **A way to project task vectors.** Summarization is more diffused than named entity recognition.
- **Following work.** Work more on the notion of inclusion.