

Frustratingly Easy Performance Improvements for Low-resource Setups: A Tale on BERT and Segment Embeddings

Rob van der Goot, Max Müller-Eberstein, Barbara Plank

Hey BERT, can you analyze: "Ik heb een banaan in mijn oor"?

!?!?

What if we change something arbitrary, like its segment embeddings?, maybe he will be smarter!

What are segment embeddings?

Well, they are used as input in BERT-based models to signal to which segment a subword belongs.



More concretely, they are just looked up embeddings, which are summed to the input word embedding. For tasks containing multiple segments, they can signal this difference to the model.

Ah cool!, Lets try alternations of segment embeddings in low-resource setups. Shall we try to just use segment ID's of 1?

	[CLS]	first	?	[SEP]	second	!	[SEP]
POS	0	1	2	3	4	5	6
ORIGINAL	0	0	0	0	1	1	1
1s	1	1	1	1	1	1	1
SEG							

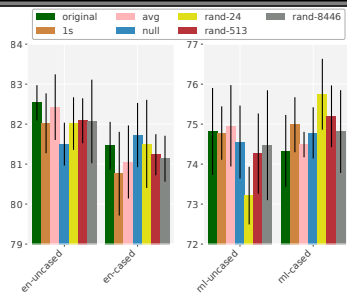
How about using the averages, or just making them empty (0.0) ?

	[CLS]	first	?	[SEP]	second	!	[SEP]
POS	0	1	2	3	4	5	6
ORIGINAL	0	0	0	0	1	1	1
1s	1	1	1	1	1	1	1
AVG	0	0	0	0	0	0	0
NULL	0	0	0	0	0	0	0
SEG							

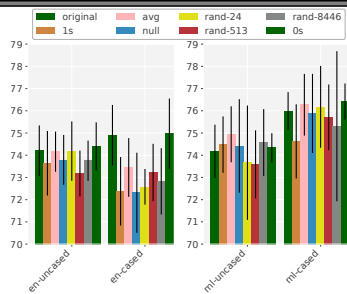
We could also randomly initialize them?, or just use ID 0 for all sub-words!

	[CLS]	first	?	[SEP]	second	!	[SEP]
POS	0	1	2	3	4	5	6
ORIGINAL	0	0	0	0	1	1	1
1s	1	1	1	1	1	1	1
AVG	0	0	0	0	0	0	0
NULL	0	0	0	0	0	0	0
RAND	0	0	0	0	0	0	0
0s	0	0	0	0	0	0	0
SEG							

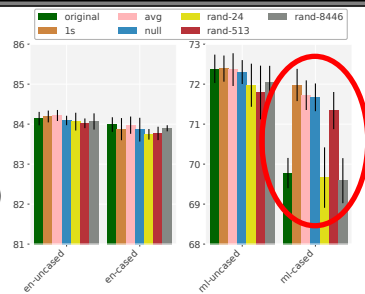
Results glue single sentence tasks (low-resource)



Results glue sentence pair tasks (low-resource)



Results UD parsing (low-resource)



For monolingual BERT the differences are small...

But if we use multilingual BERT, the default of using segment ID 0 performs very poorly for word-level annotations!

We gain 2.5 LAS points on average by simply swapping them to 1!