# Train a classifier on MASK and gold representations

Use the FEVER train set to train a binary fact-checking classifier as follows.

You can use the dev set in [7] for hyperparameters finetuning.

For a datapoint, get BERT contextual representations for both the sentence with the entity and the sentence masking the entity.

Example:

Representation masking the entity Lorelai Gilmore's father is named [MASK].

Representation with the entity Lorelai Gilmore's father is named Robert.

To obtain a vectorial representation, pick the last layer of the contextual embedding associated with the [MASK] token, in the first case, and with the entity token, in the second case.

The contextual representation can be obtained with LAMA, look at lama/get\_contextual\_embeddings.py

As an example, modify the code, line 11 to:

sentences = [

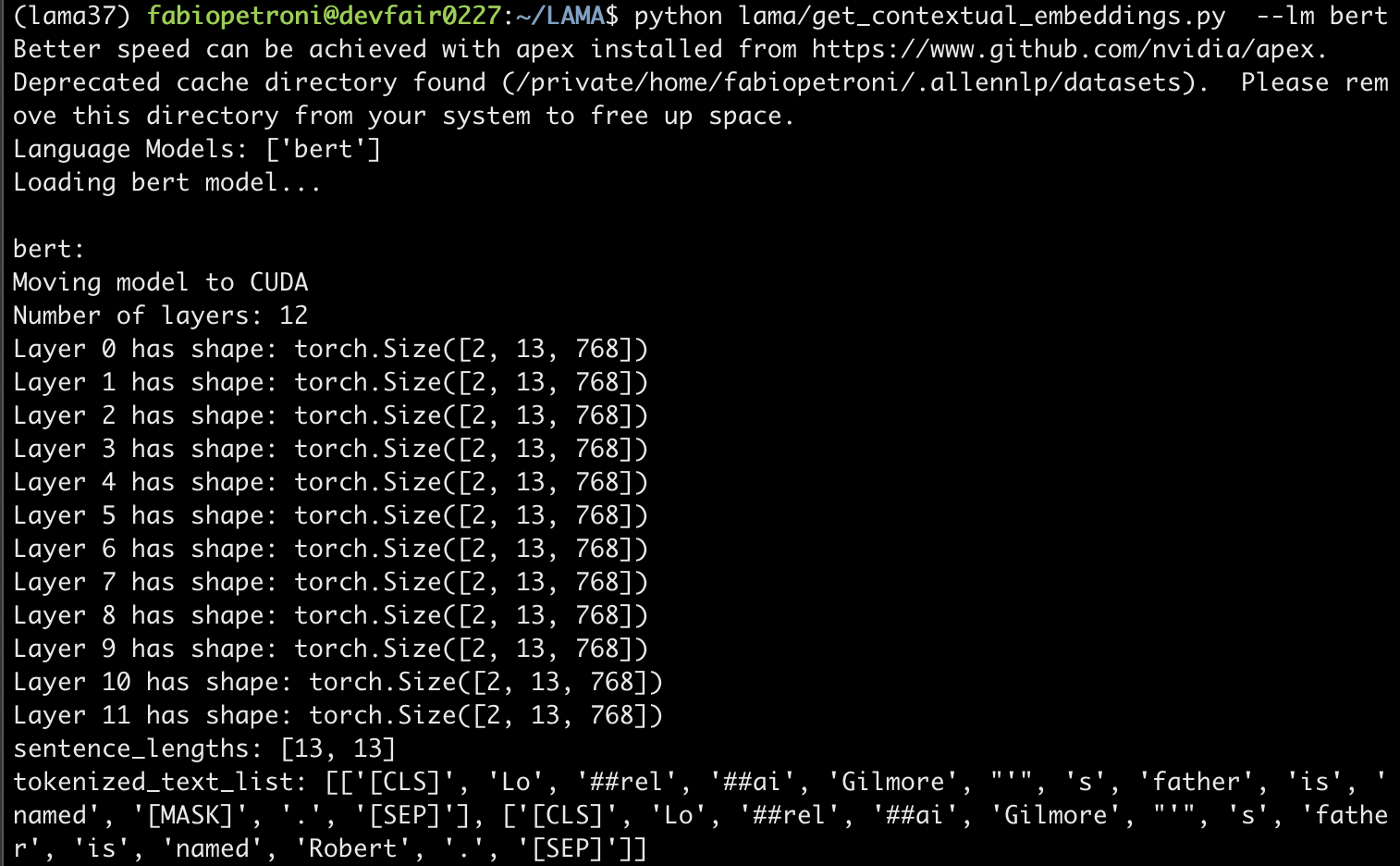
["Lorelai Gilmore's father is named [MASK]."],

["Lorelai Gilmore's father is named Robert."],

]

Run:

python lama/get\_contextual\_embeddings.py --lm "bert"



As representations you should use the vectors in the last layer (i.e., index 11), corresponding to the entity.

In the example, the entity is tokenized at index 11, see tokenized\_text\_list.

To access the representation you should use the contextual\_embeddings variable at line 27.

In particular

contextual\_embeddings[layer\_index=11][sentence\_index][token\_index]

This will give you a vector of dimension 768

So in this example,

**Representation masking the entity =** contextual\_embeddings[11][0][11]

**Representation with the entity =** contextual\_embeddings[11][1][11]

Use the concatenation of these two representations as input vectors and train a binary classifier, e.g., SVM, and get the accuracy performance on the dev set, as well as get predictions for **the official test set**. Remember: you can use the dev set to finetune the hyperparameters of the model (using e.g. GridSearch or RandomSearch).