**Results for training the REL Model for the week 10th Oct – 14th Oct 2022**

The data preprocessing and testing of the .spacy files were being executed on Google Colaboratory.

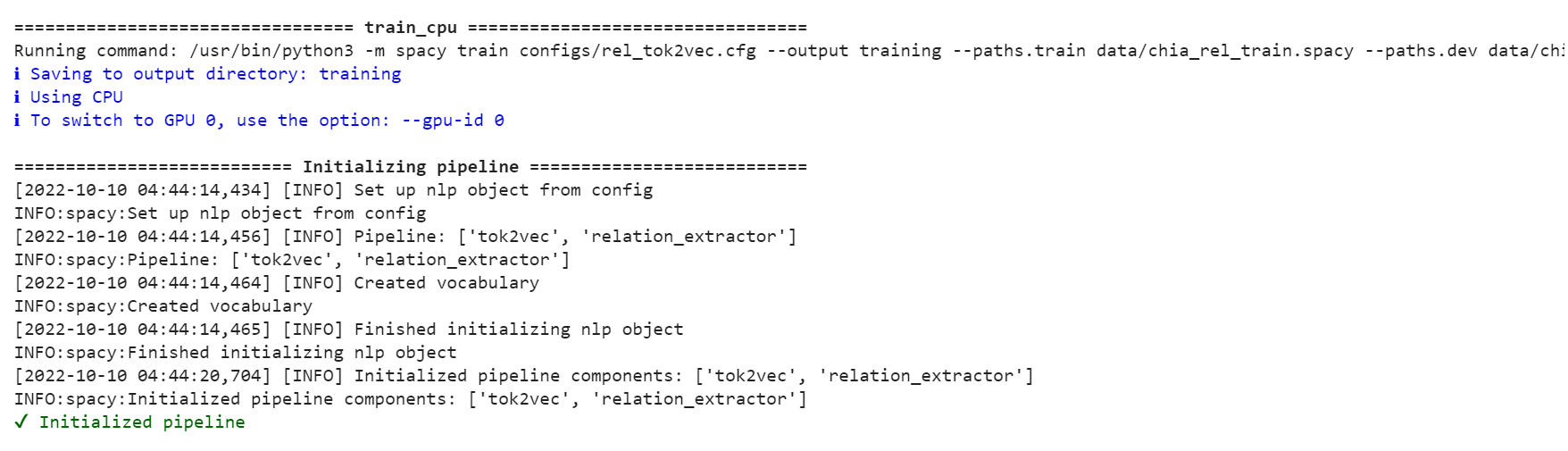
The REL model was trained on tok2vec pipeline and 3 different transformers.

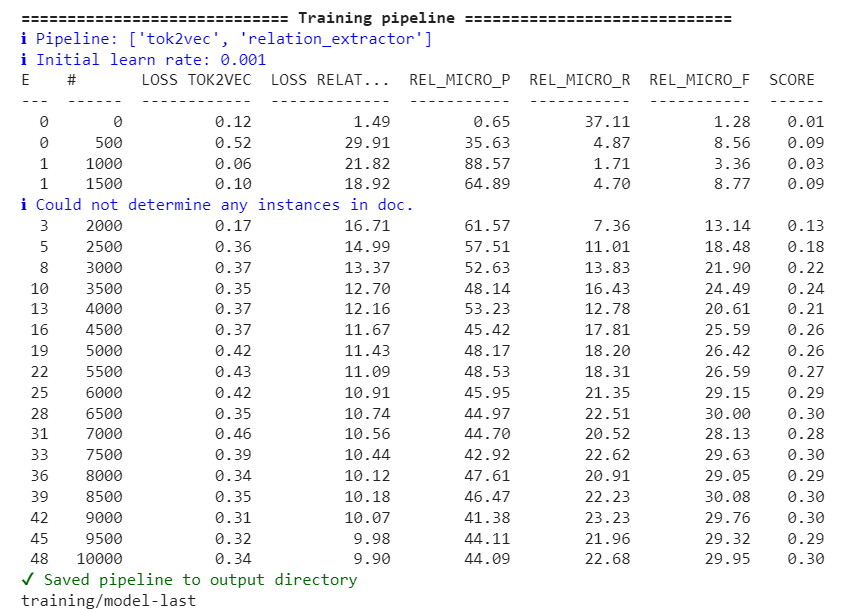
1. bert-base-uncased
2. bluebert\_pubmed\_uncased\_L-12\_H-768\_A-12
3. bert-base-cased
4. roberta-base

The top 5 relations which we must focus are ['Has\_value', 'Has\_temporal', 'Has\_qualifier', 'Subsumes', 'Has\_scope'].

1. **The results of training REL model on top 5 relations:**

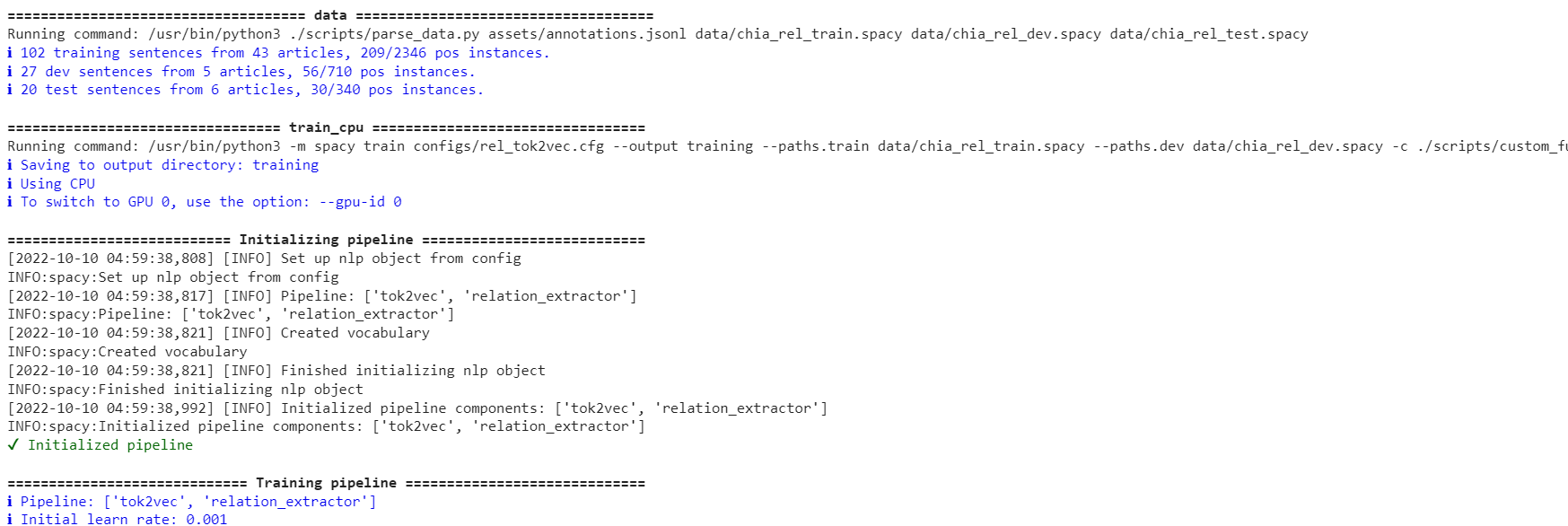
**a)** **Using CPU (tok2vec)**: When we run the command python -m spacy project run train\_cpu, we get the results as follows:

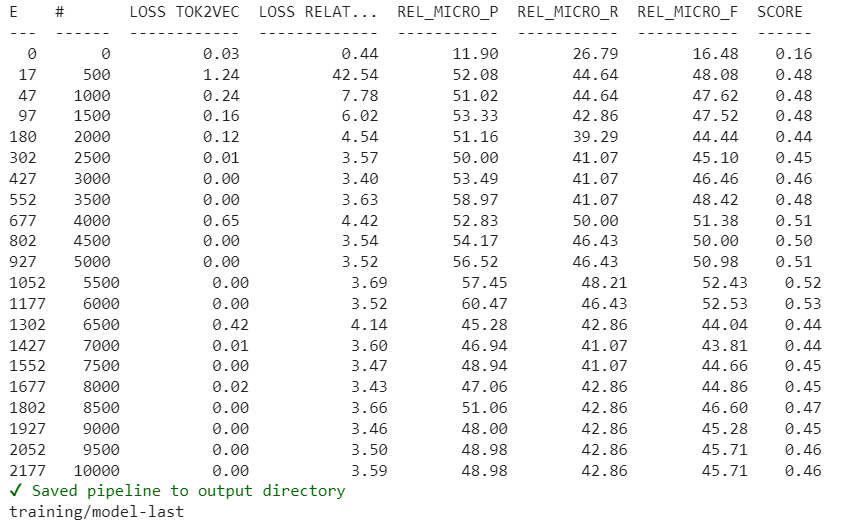


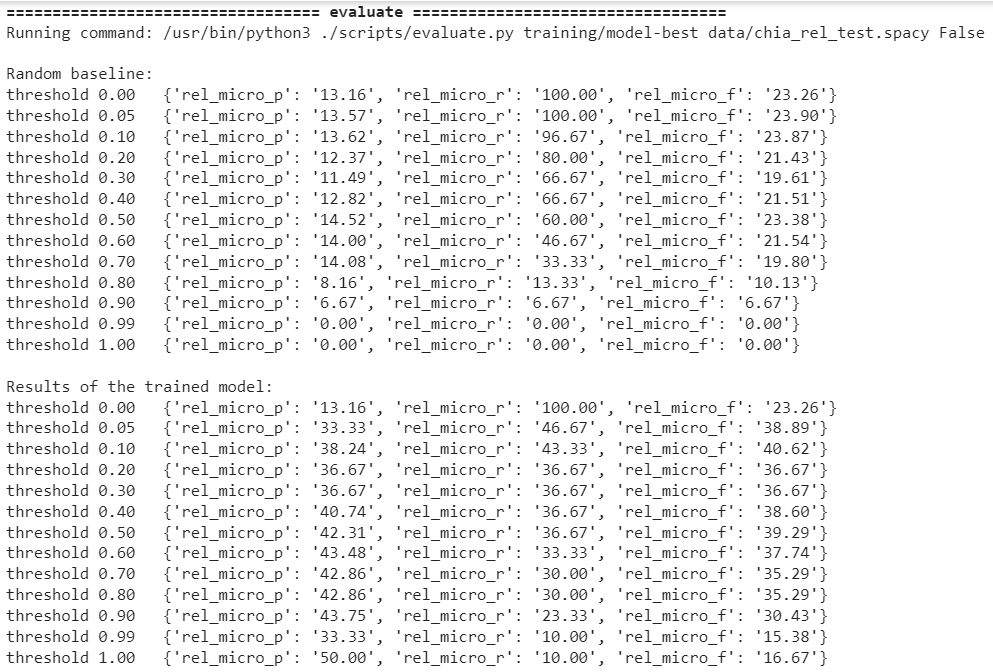


**The highest precision score for tok2vec pipeline is 88% with 48 epochs. The highest F1-score is 30%.**

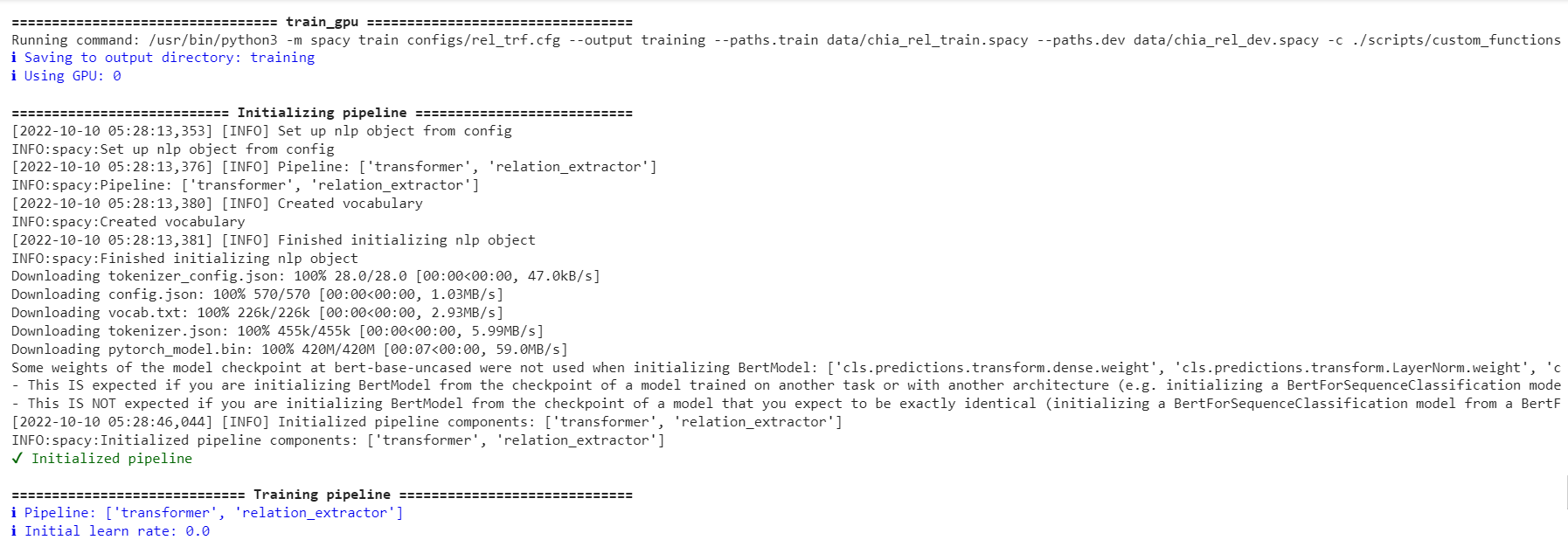
When we run the command python -m spacy project run all, we get the results as follows:

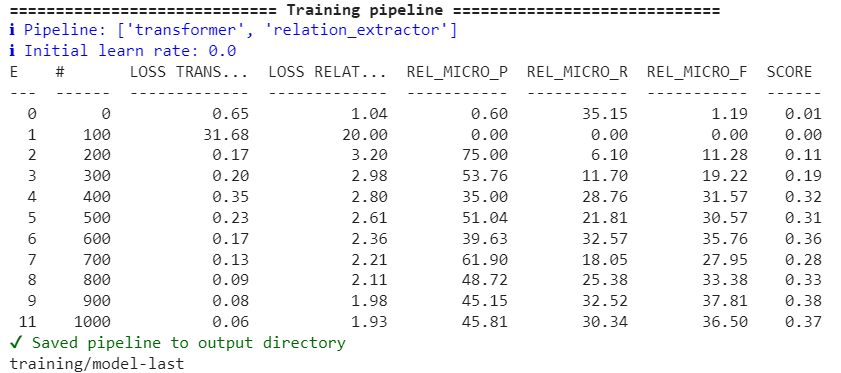






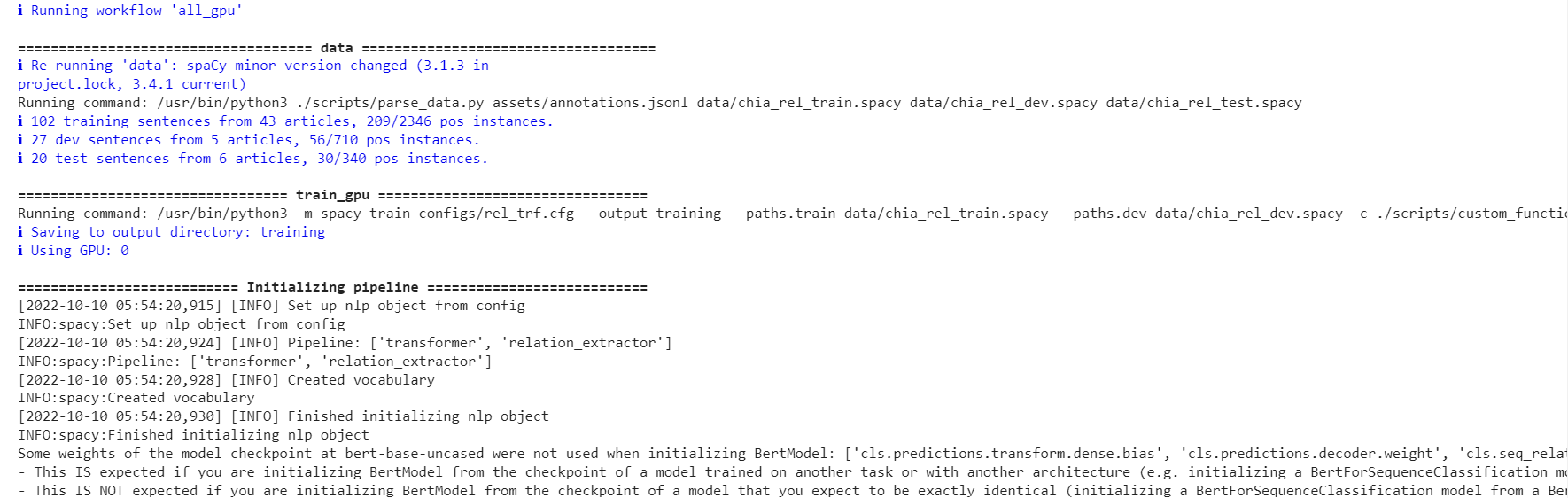
**b) Using GPU (Transformers):** **I) bert-base-uncased**: When we run the command python -m spacy project run train\_gpu, we get the results as follows:

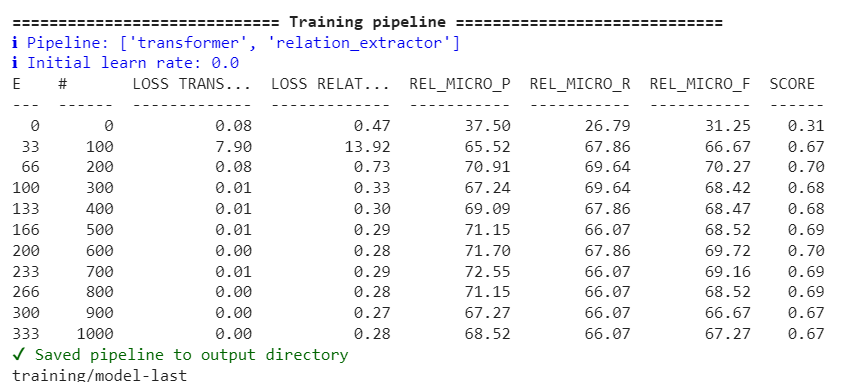


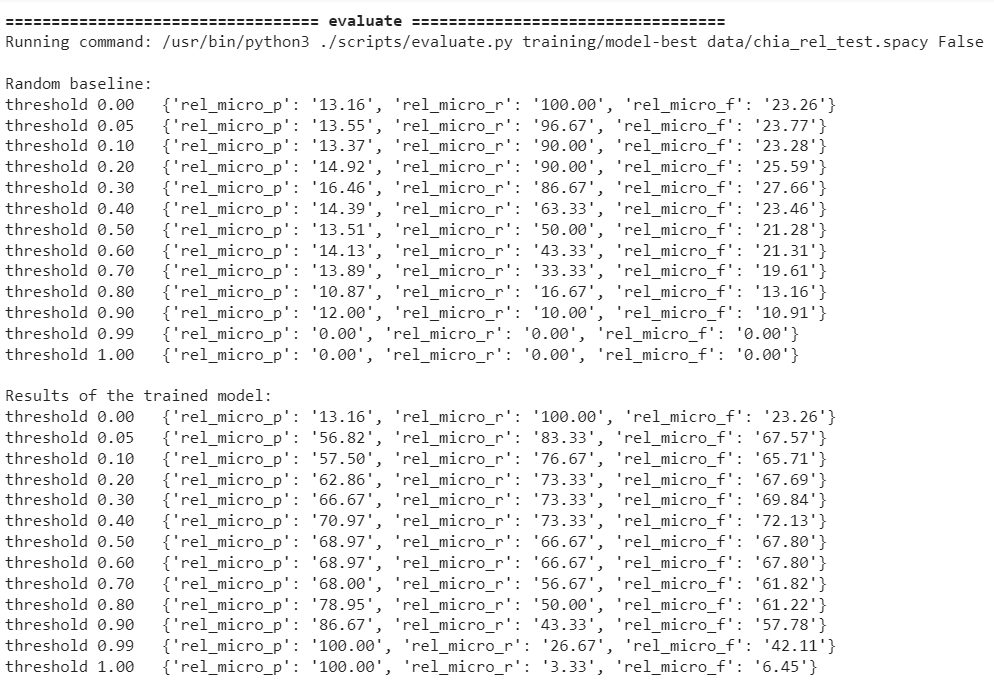


**The highest precision score for this transformer is 75% with 11 epochs. The highest F1-score is 38%.**

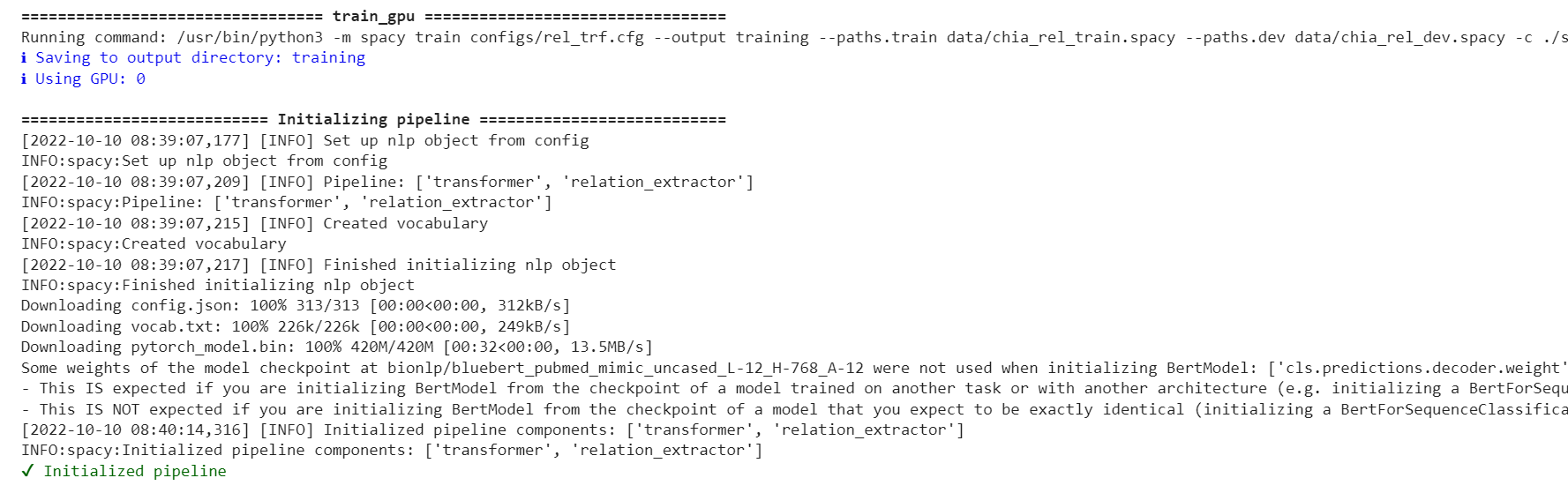
When we run the command python -m spacy project run all\_gpu, we get the results as follows:

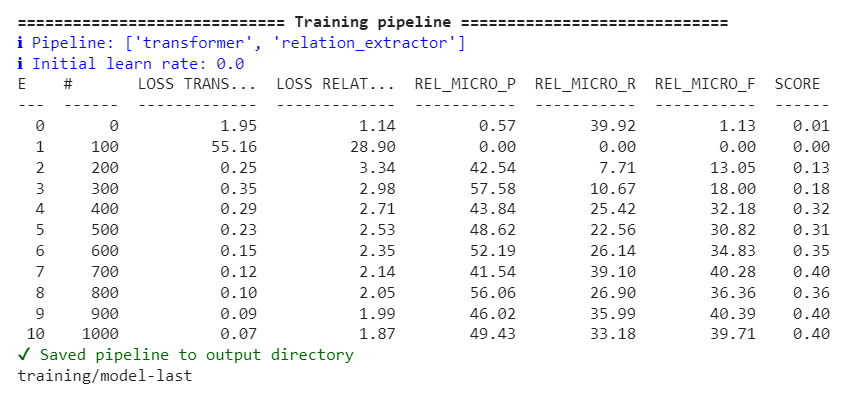






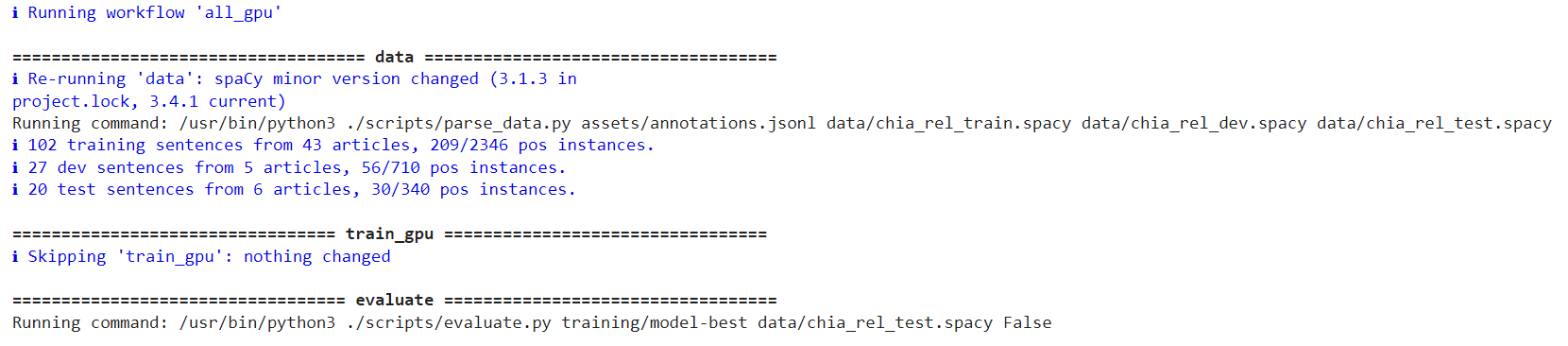
**II)** **bionlp/bluebert\_pubmed\_mimic\_uncased\_L-12\_H-768\_A-12**: When we run the command python -m spacy project run train\_gpu, we get the results as follows:

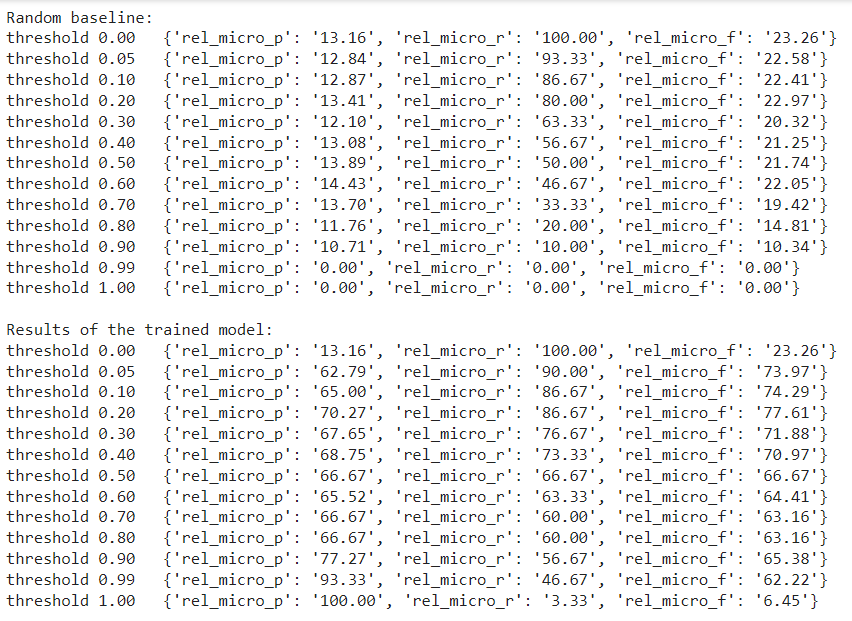




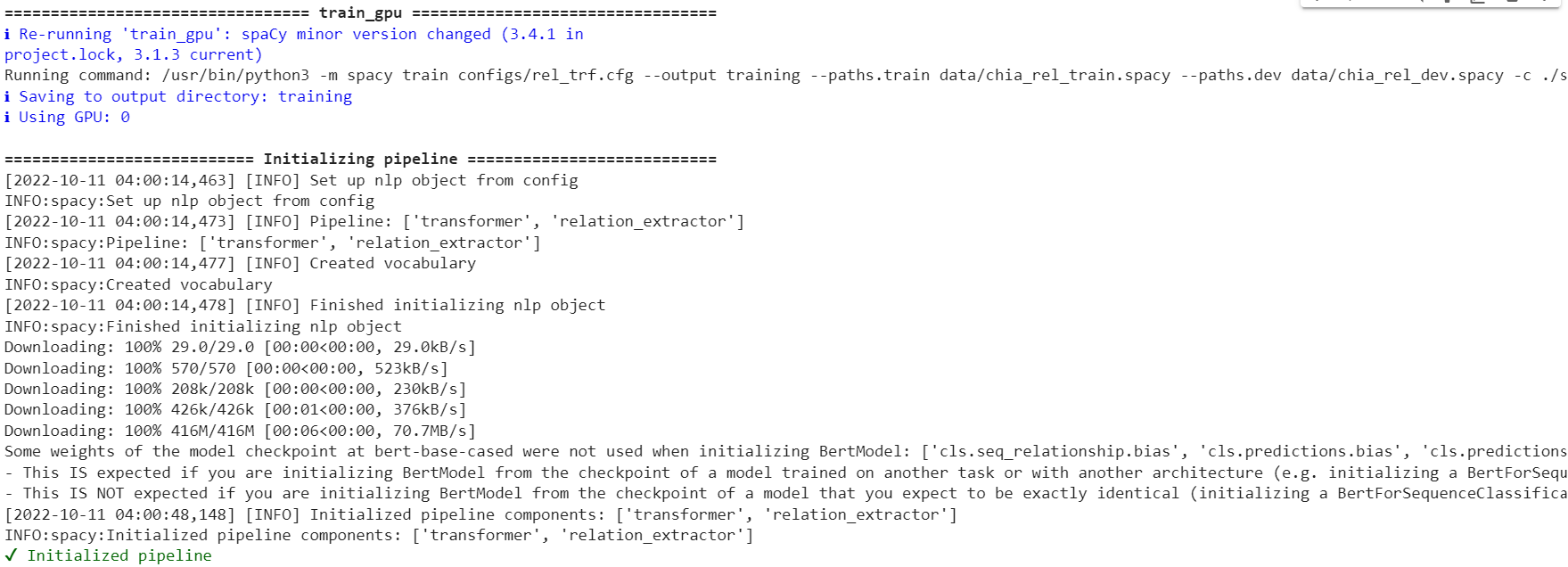
**The highest precision score for this transformer is 57% with 10 epochs. The highest F1-score is 40%.**

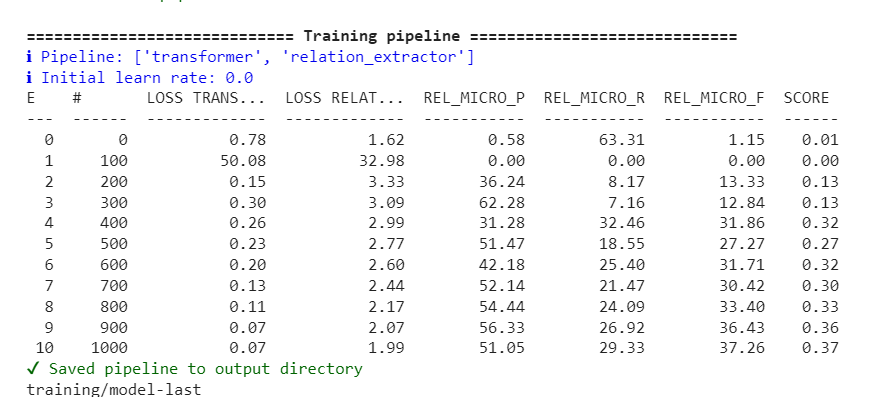
When we run the command python -m spacy project run all\_gpu, we get the results as follows:





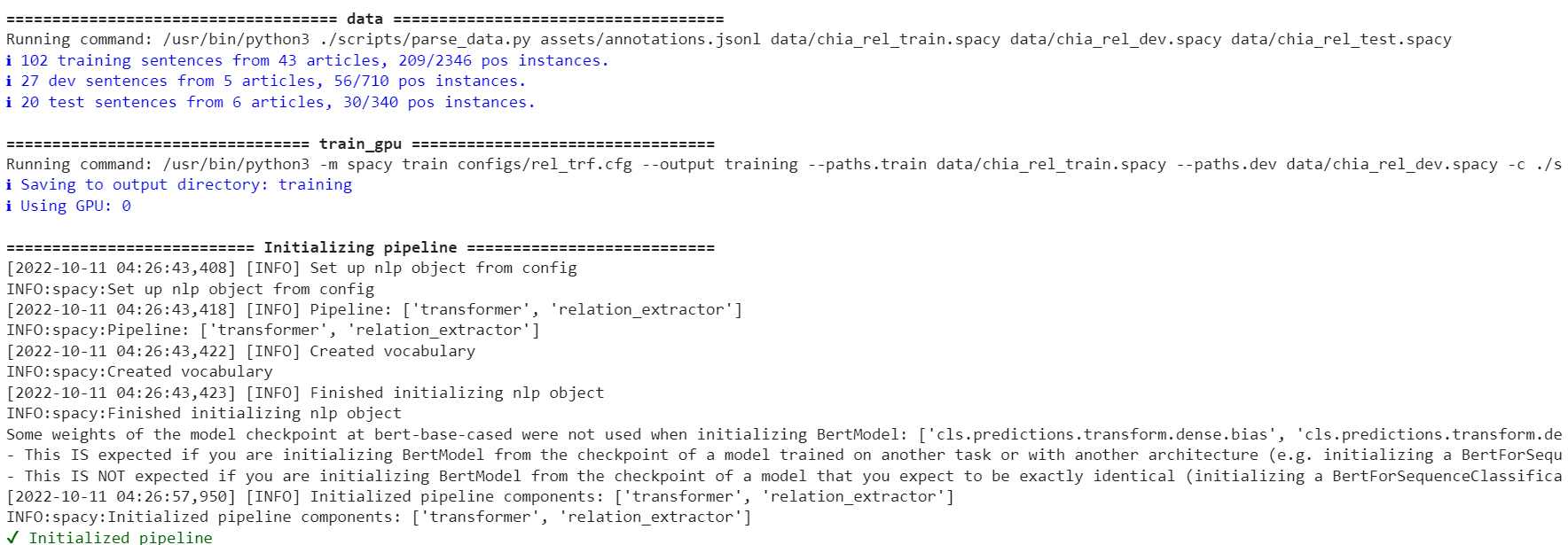
**III)** **bert-base-cased**: When we run the command python -m spacy project run train\_gpu, we get the results as follows:

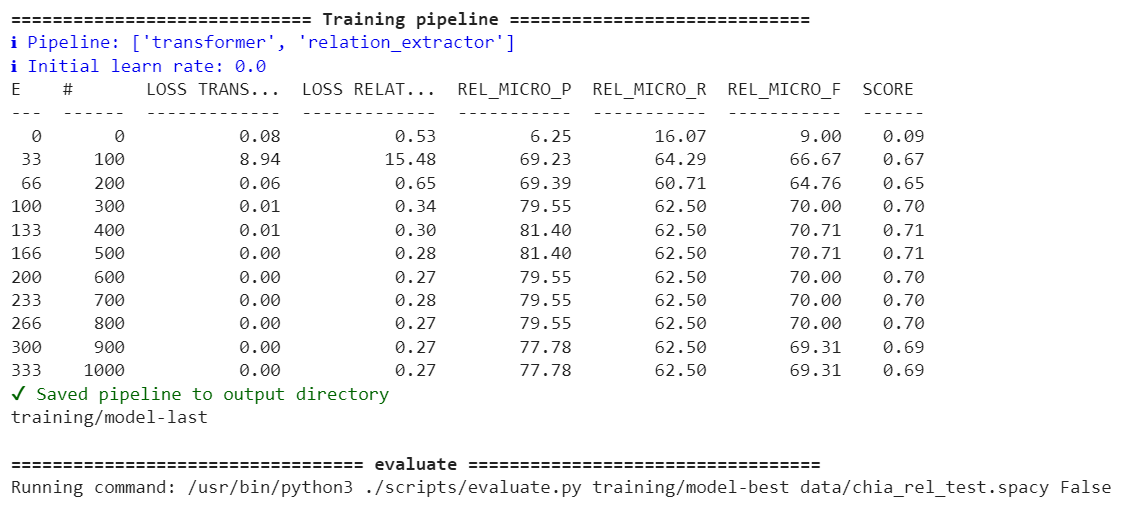


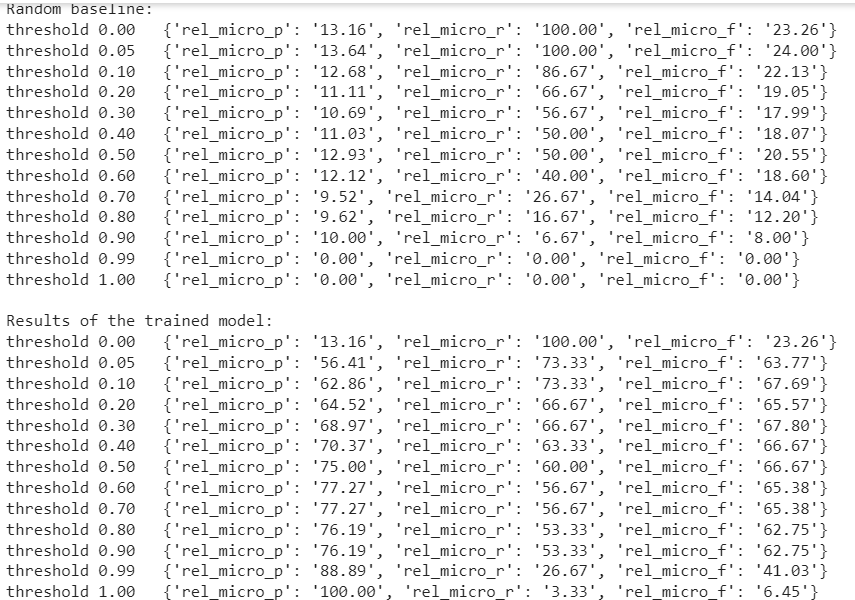


**The highest precision score for this transformer is 62% with 10 epochs. The highest F1-score is 37%.**

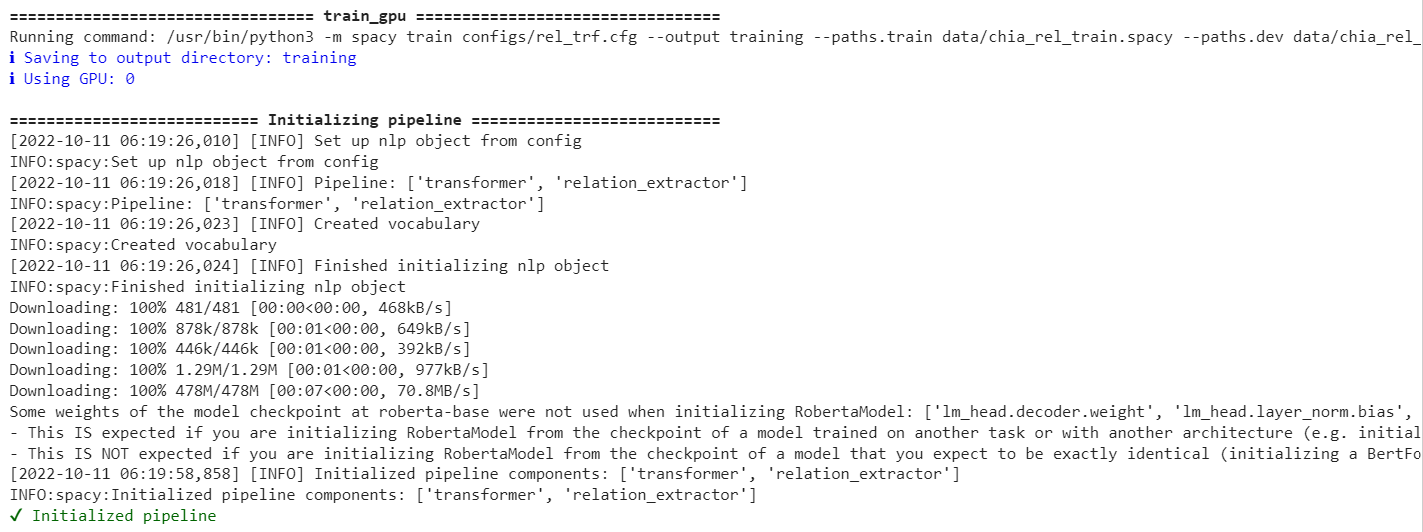
When we run the command python -m spacy project run all\_gpu, we get the results as follows:

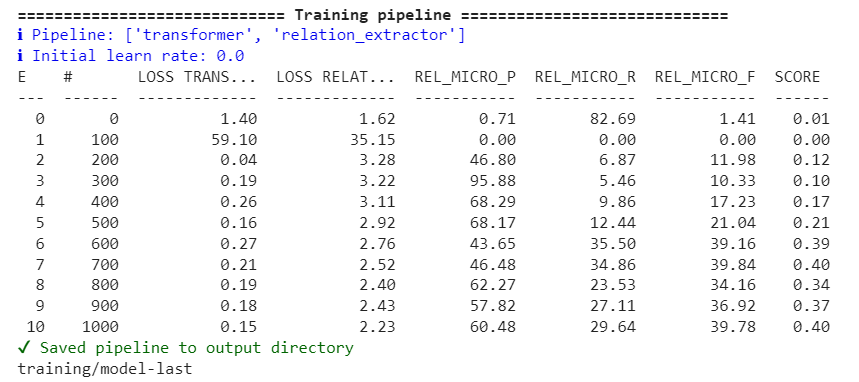






**IV)** **roberta-base**: When we run the command python -m spacy project run train\_gpu, we get the results as follows:





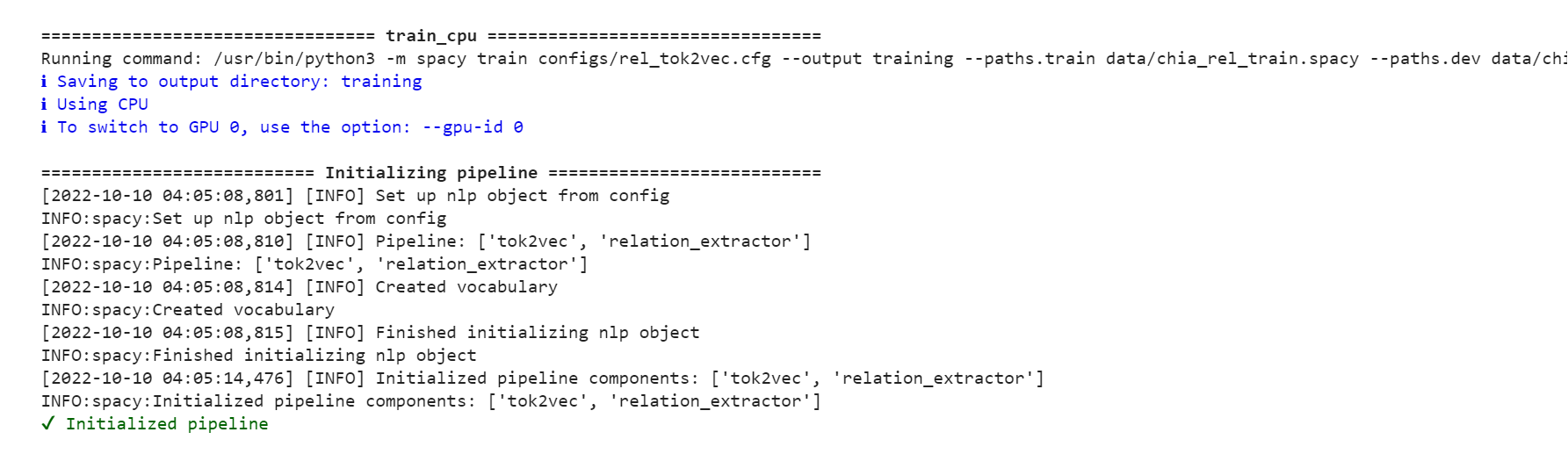
**The highest precision score for this transformer is 95% with 10 epochs. The highest F1-score is 40%.**

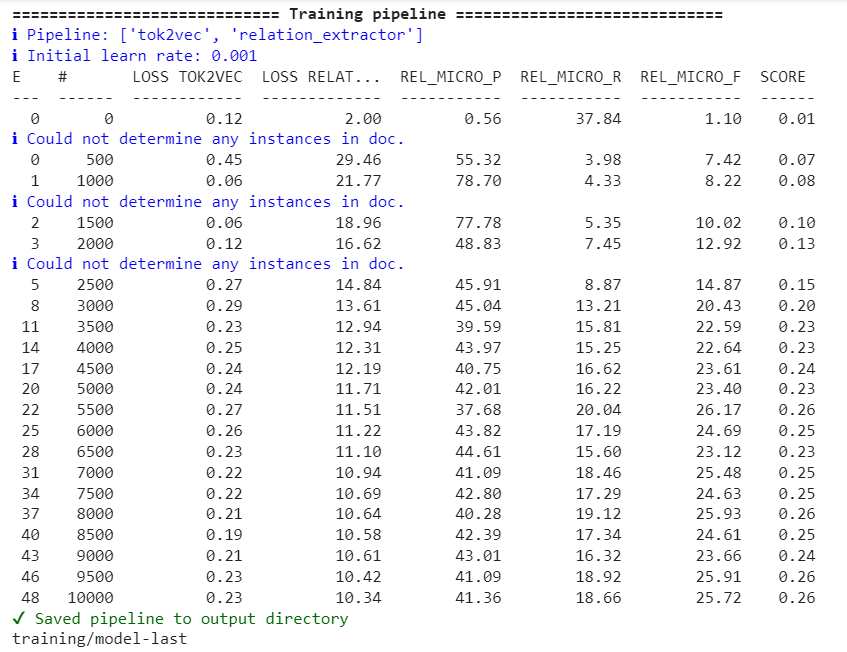
**Future Work**: The precision score of the trained REL model needs to be improved and will try increasing up to 80%.

**Note**: The training of REL model was initially performed on top 10 relations. Since the model performance did not improve, so we reduced our approach to just top 5 relations and ignored certain equivalence relations like AND, OR.

1. **The results of training REL model on top 10 relations:**

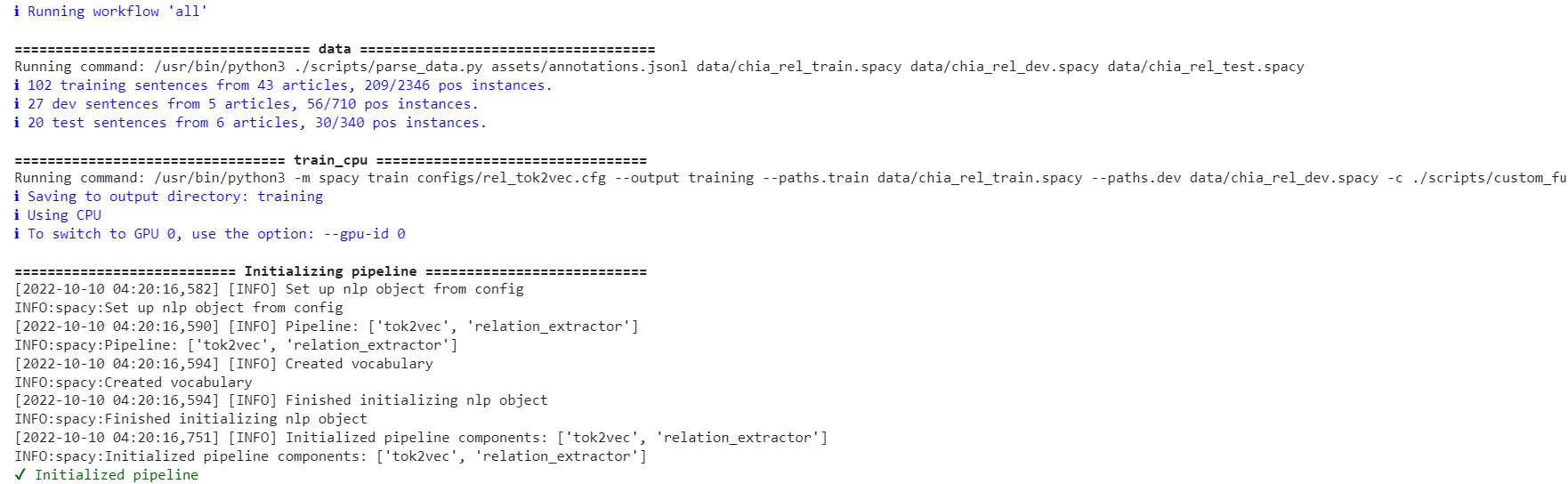
**Using CPU (tok2vec):** When we run the command python -m spacy project run train\_cpu, we get the results as follows:

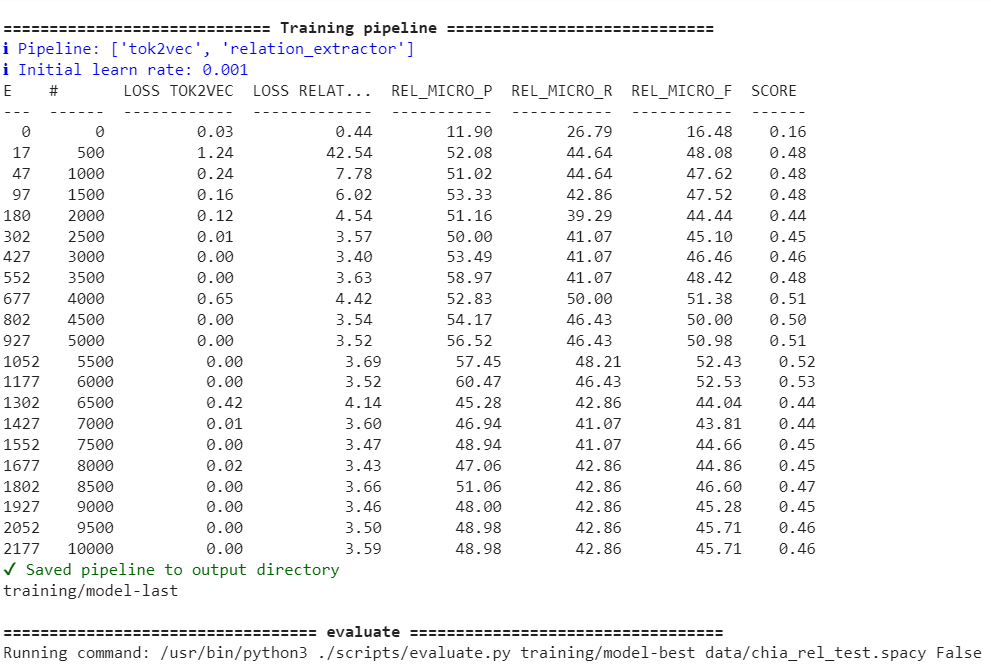
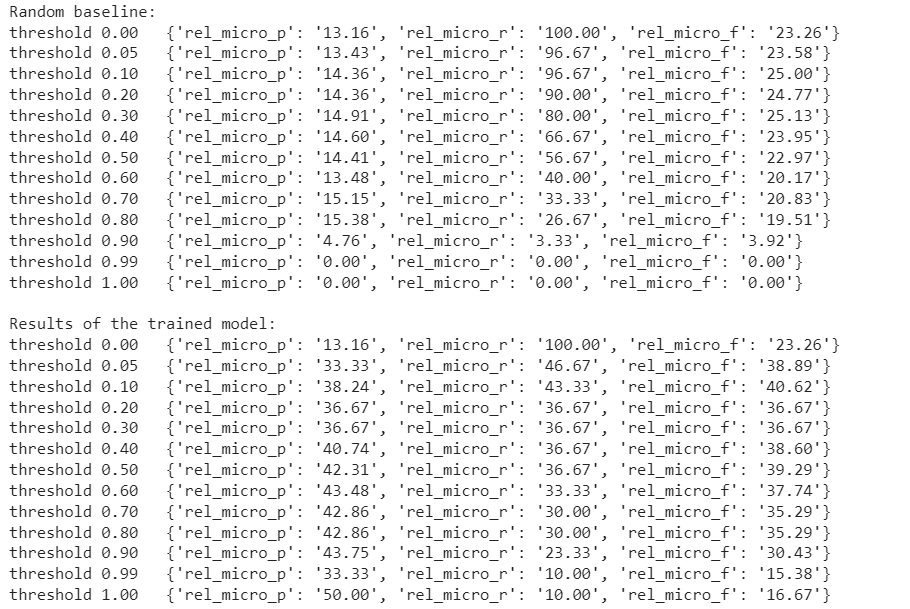




**The highest precision score for tok2vec pipeline is 78% with 48 epochs. The highest F1-score is 26%.**

When we run the command python -m spacy project run all, we get the results as follows:



The next step was the saved REL models were being used for packaging purpose.

To perform the packaging on the REL models using Spacy follow these steps:

1. Use the rel\_component repository from spacy explosions.
2. Inside this directory, create 2 folders **models** and **rel\_models**.
3. Place the trained Chia REL model folder chia\_roberta\_5\_rel inside the **models** directory.
4. Inside the chia\_roberta\_5\_rel directory, open the meta.json file and edit **"name":"chia\_rel\_pipeline\_trf".**
5. Inside command prompt, run this command to package the spacy model

**python -m spacy package models/chia\_roberta\_5\_rel** **rel\_models - -code list of comma separated custom component .py files.**

**e.g.** 1. python -m spacy package models/chia\_roberta\_5\_rel/ rel-models/ --code scripts/rel\_pipe.py, scripts/rel\_model.py

1. This creates the **en\_chia\_rel\_pipeline\_trf-0.0.0** inside the **rel\_models** directory.
2. Traverse to the directory **rel\_models/en\_chia\_rel\_pipeline\_trf-0.0.0** and run the command: **python setup.py sdist**
3. It creates **dist** folder and inside it created **en\_chia\_rel\_pipeline\_trf-0.0.0.tar.gz** file.
4. Traverse to **dist** folder and run the command:

**python -m** **pip install** **en\_chia\_rel\_pipeline\_trf-0.0.0.tar.gz**

1. It will successfully install the trained Chia REL spacy model inside the system.
2. Validate it using the command: **python -m spacy validate**.

Tried out Serving the saved REL model using FastAPI.

**To be done in the next week 17th Oct – 21st Oct 2022**:

Currently exploring some biomedical datasets like BioREL and some of the problem statements in NLP which I can work on in the next week as a part of data preprocessing and improving the ML model during training the REL model (preferably using PyTorch models).