Michael Gardner

Information Extraction

Assignment 3 – Relation Classification

Runtime parameters for all reported scores:

Train file:	semevalTrainReal.tsv
Dev file:	semevalDev.tsv
Lr:	0.00005
Batch size:	16
Max epochs:	20

Implemented cmd line additions:

Model:	dan, lstm, cnn
Llm_choice:	"distilbert-base-uncased", "bert-base-uncased", "openai-gpt"
Test-file:	whether to predict the test file; using checkpointed best model

This assignment, I built upon the models from previous assignments and presumed I would use the LSTM, which was the best performer there. The simple DANClassifier model which came in the starter code, with no alterations achieved a benchmark around 25% val accuracy and it was a bare minimum to beat. The base bidirectional LSTM came in with 23%. The experiment progressed from there to involve the Truncation.

Truncation is to say, the target sentence is shaved down to (entity1:entity2), inclusive so we can gain some knowledge from the target entity words themselves. The base LSTM improved with truncation, and we reached 37% accuracy and 0.27 F1. Other experiments with the entity-tags functionality didn't show any improvement over truncation, so it was shelved.

So the next upgrade was to stop using random vocab indexes and move on to embeddings. The code allows for llm to be turned on and llm_choice to be selected from distilbert, base bert, and gpt. Checkpointed models could also be stored to save their best performing valid_acc epoch.

ALL SCORES REFLECT THE DEV or "VALIDATION" SET.

	Val_acc	macro_f1	train_loss	sec/epoch
LSTM Truncated:	0.373	0.270	1.0	2.5s
Distilbert LSTM Truncated:	0.621	0.600	0.68	18s
BERTbase LSTM Truncated:	0.621	0.601	0.56	26s
GPT LSTM Truncated:	0.661	0.667	0.25	25s

Validation officially passed 60% baseline! And GPT is running away with it!

But how do we improve from here. My hypothesis in the model proposal was that truncation would show improvement and we've born that out at the base LSTM… but what about with embeddings. I needed to check.

	Val_acc	macro_f1	train_loss	sec/epoch
LSTM:	0.229	0.067	2.37	3s
Distilbert LSTM:	0.715	0.709	0.07	20s
BERTbase LSTM:	0.740	0.745	0.10	31s
GPT LSTM:	0.695	0.696	0.13	33s

That blew all my expectations away. GPT with full context stagnates, but BERT gains? Surpassed the 72% mark? If that's true, maybe we need the CNN from assignment 2; would that improve things?

The implementation of the CNN and LSTM from A2 each did take some alterations to accept this data, including excluding the masking functionality in favor the variable padding implemented in dataset, and the dataset code had to handle "input_ids" separately from tokens.

So, the CNN, using 100 filters of widths 4,3,2…

	Val_acc	macro_f1	train_loss	sec/epoch
Distilbert CNN:	0.7275	0.7323	0.0468	20s
BERTbase CNN:	0.7450	0.7444	0.0270	34s
GPT CNN:	0.7090	0.7064	0. 885	33s

BERT embeddings shine once more. I'd call those BERT results for the LSTM and CNN within a statistical margin of error, which was also the case in A2, neck-and-neck. But the distilbert version saw moderate improvement, so we can say the CNN makes gains over the LSTM, no matter the embedding method. Everything is hovering around 74%.

And in rerunning to get test predictions, we only managed to get 74.2% on the CNN, but edged a little higher with the LSTM to tie at 74.4%. So, both test predictions from those checkpoints have been included.

python main.py --batch-size 16 --max-epochs 20 --device cuda --train-file ../data/semevalTrainReal.tsv --dev-file ../data/semevalDev.tsv --lr 0.00005 --debug --llm --llm-choice bert-base-uncased --hidden-layer-sizes 100,100 --test-file ../data/semevalTest_without_keys.tsv --model lstm

LSTM BERT:

```
------- VALIDATION CLASSIFICATION REPORT -------------
                       precision    recall  f1-score   support

     Instrument-Agency       0.77      0.50      0.61        20
    Entity-Destination       0.93      0.91      0.92       207
       Component-Whole       0.79      0.74      0.76       113
 Instrument-Agency-Inv       0.79      0.66      0.72       110
 Member-Collection-Inv       0.81      0.85      0.83       155
  Product-Producer-Inv       0.73      0.78      0.75       100
         Entity-Origin       0.73      0.78      0.75       138
ntity-Destination-Inv       1.00      1.00      1.00         0
      Cause-Effect-Inv       0.79      0.87      0.83       173
     Member-Collection       0.32      0.84      0.46        19
   Component-Whole-Inv       0.73      0.71      0.72        99
     Entity-Origin-Inv       0.76      0.67      0.71        42
                 Other       0.61      0.48      0.53       355
         Message-Topic       0.79      0.83      0.81       115
      Product-Producer       0.85      0.74      0.79       100
          Cause-Effect       0.76      0.81      0.78        84
     Content-Container       0.68      0.90      0.77        88
 Content-Container-Inv       0.84      0.84      0.84        37
     Message-Topic-Inv       0.55      0.80      0.65        45

             micro avg       0.74      0.74      0.74      2000
             macro avg       0.75      0.77      0.75      2000
          weighted avg       0.75      0.74      0.74      2000
```

python main.py --batch-size 16 --max-epochs 20 --device cuda --train-file ../data/semevalTrainReal.tsv --dev-file ../data/semevalDev.tsv --lr 0.00005 --debug --llm --llm-choice bert-base-uncased --hidden-layer-sizes 4,3,2 --test-file ../data/semevalTest_without_keys.tsv --model cnn

CNN BERT:

```
-------- VALIDATION CLASSIFICATION REPORT -------------
                       precision    recall  f1-score   support

  Product-Producer-Inv       0.63      0.71      0.67       100
 Member-Collection-Inv       0.79      0.90      0.84       155
Entity-Destination-Inv       1.00      1.00      1.00         0
       Component-Whole       0.68      0.78      0.72       113
     Message-Topic-Inv       0.58      0.80      0.67        45
         Message-Topic       0.81      0.83      0.82       115
   Component-Whole-Inv       0.67      0.74      0.70        99
      Product-Producer       0.82      0.75      0.78       100
                 Other       0.72      0.35      0.47       355
          Cause-Effect       0.77      0.79      0.78        84
     Member-Collection       0.61      0.74      0.67        19
      Cause-Effect-Inv       0.81      0.88      0.84       173
    Entity-Destination       0.89      0.94      0.91       207
     Entity-Origin-Inv       0.67      0.76      0.71        42
 Content-Container-Inv       0.83      0.81      0.82        37
     Instrument-Agency       0.65      0.55      0.59        20
 Instrument-Agency-Inv       0.64      0.83      0.72       110
         Entity-Origin       0.73      0.81      0.77       138
     Content-Container       0.71      0.91      0.80        88

             micro avg       0.74      0.74      0.74      2000
             macro avg       0.74      0.78      0.75      2000
          weighted avg       0.74      0.74      0.73      2000
```