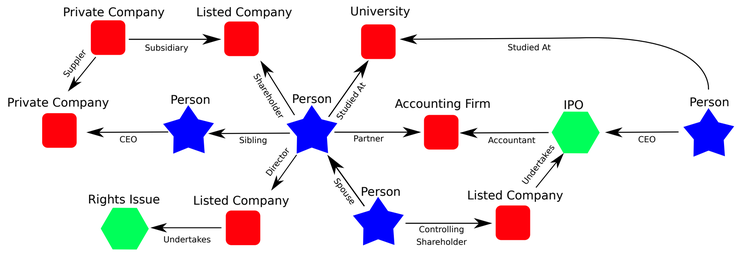
Named Entity Recognition from Experts using Deep Learning on GPUs

Dr Martin Andrews
Martin.Andrews@RedCatLabs.com

red cat labs

Commercial Motivation

A Singapore-based data provider currently updates a rich database of relationship information between people and companies by manually capturing and curating this data from publicly available regulatory filings. This data is then presented to its customers through an interactive graph of these entities connected by their interactions with each other.



Automating the entity and relationship extraction process is critical if this data provider is to expand into other markets quickly.

Existing state-of-the-art systems are constrained due to :

- Monolithic construction (difficult to extend)
- Licensing constraints
- Availability/confidentiality of training data

Objectives

Named Entity Recognition (NER) is an essential component of systems that process Natural Language documents, and the system to mine this unstructured legal data requires a specialized implementation with both high performance and high accuracy.

In this work, a new NER system is developed that uses the output of existing systems over large corpuses as its training set, ultimately enabling labelling with :

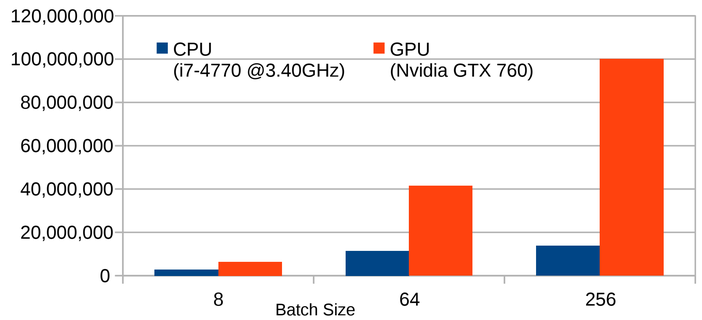
- Better F1 scores
- Higher labelling speeds
- No further dependence on the external software

Implementation

The Theano framework was used, with the (new) 'blocks' Deep Learning infrastructure, so that GPU code could be generated from a pure Python description of the problem. This code sample illustrates how elegantly these libraries (to which the author is a contributor) can describe deep networks :

```
import theano
import blocks

x = tensor.matrix('tokens', dtype="int32")
x_mask = tensor.matrix('tokens_mask', dtype=floatX)

lookup = LookupTable(vocab_size, embedding_dim)

x_extra = tensor.tensor3('extras', dtype=floatX)

rnn = Bidirectional(
  SimpleRecurrent(activation=Tanh(),
    dim = hidden_dim,
    weights_init = IsotropicGaussian(0.01),
    biases_init  = Constant(0.),
  ),
)

gather = Linear(name='hidden_to_output',
  input_dim    = hidden_dim*2,
  output_dim   = labels_size,
)

label_probs = Softmax().apply(labels_raw)
cost = CategoricalCrossEntropy().apply(y, label_probs)

algorithm = GradientDescent(
  cost=cost,
  parameters=ComputationGraph(cost).parameters,
  step_rule=AdaDelta(),
)

MainLoop(
  model=Model(cost),
  data_stream=data_stream.get_sentence_batches(),
  algorithm=algorithm,
).run()
```

Sentences Trained in 8 hours

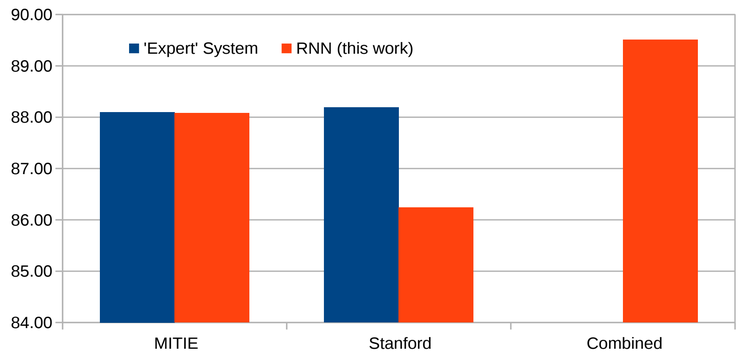


Results

The Recurrent Neural Network (RNN) system trained on the GPU was capable of :

- Learning the task from each Expert-labelled corpus
- Attaining performance close to each 'teacher' individually
- Surpassing both teachers with a combined training scheme

F1% score on CoNLL-2003 testb



Data sizes	*Bytes*	*Words*	*Sentences*
"Large Corpus"	1.0Gb	184,717,139	11,869,032
Training Set	3.3Mb	204,567	14,987
Development Set	827Kb	51,578	3,467
Test Set	748Kb	46,666	3,685

Labelling Performance	*Sentences per second*
Expert-MITIE	1,646
Expert-Stanford	48
RNN (all)	6,246

Future Work

The results for Deep Learning NER from Experts are very encouraging, and several extensions immediately suggest themselves :

- Retraining on more commercially-relevant corpuses
- Applying similar techniques to Relationship Extraction
- Extending NER to include learned character-based features

The last of these points is particularly relevant in the ASEAN area, where peoples' names are potentially easier to differentiate from the surrounding (English) text, reducing the need for providing a suitable local gazetteer.