# Text Summarization for Social Good

Aman Khullar
Minor Project under guidance of,
Dr. Deepika Kukreja

Department of Information Technology
Netaji Subhas University of Technology
31st October 2018

# Content

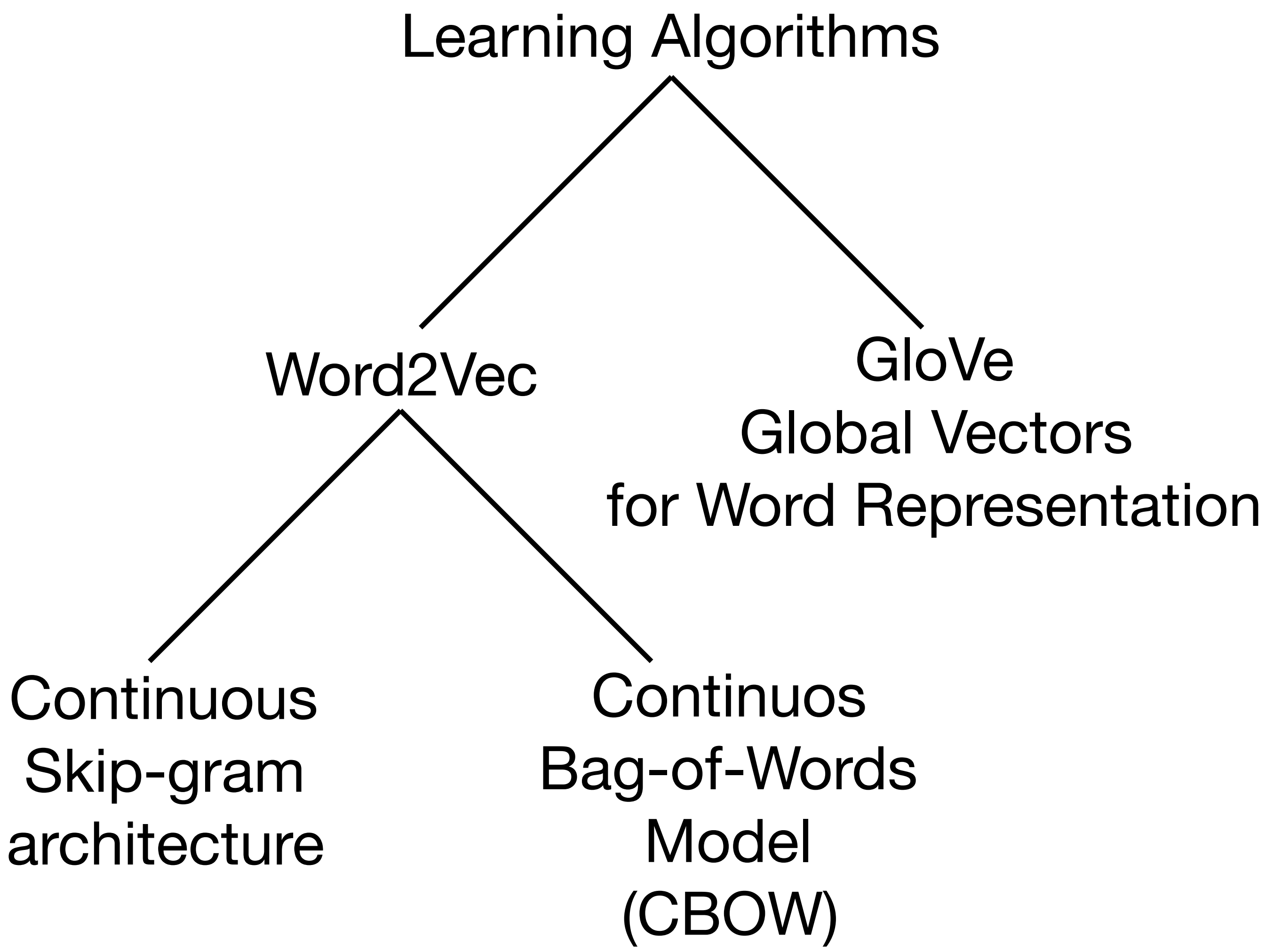
- Introduction
- Word Representation
- Dependency Parsing using Neural Networks
- RNNs, LSTMs and GRUs.
- Implementation of the model.

# Introduction

- The goal of this project is to produce a reading comprehension system
- Stanford Question Answering Dataset (SQuAD)
- Evaluation Metric - F1 and Exact Match(EM) scores

# Word Representation

- Represent words which are atomic units in continuous vector space representation.



Learning Algorithms

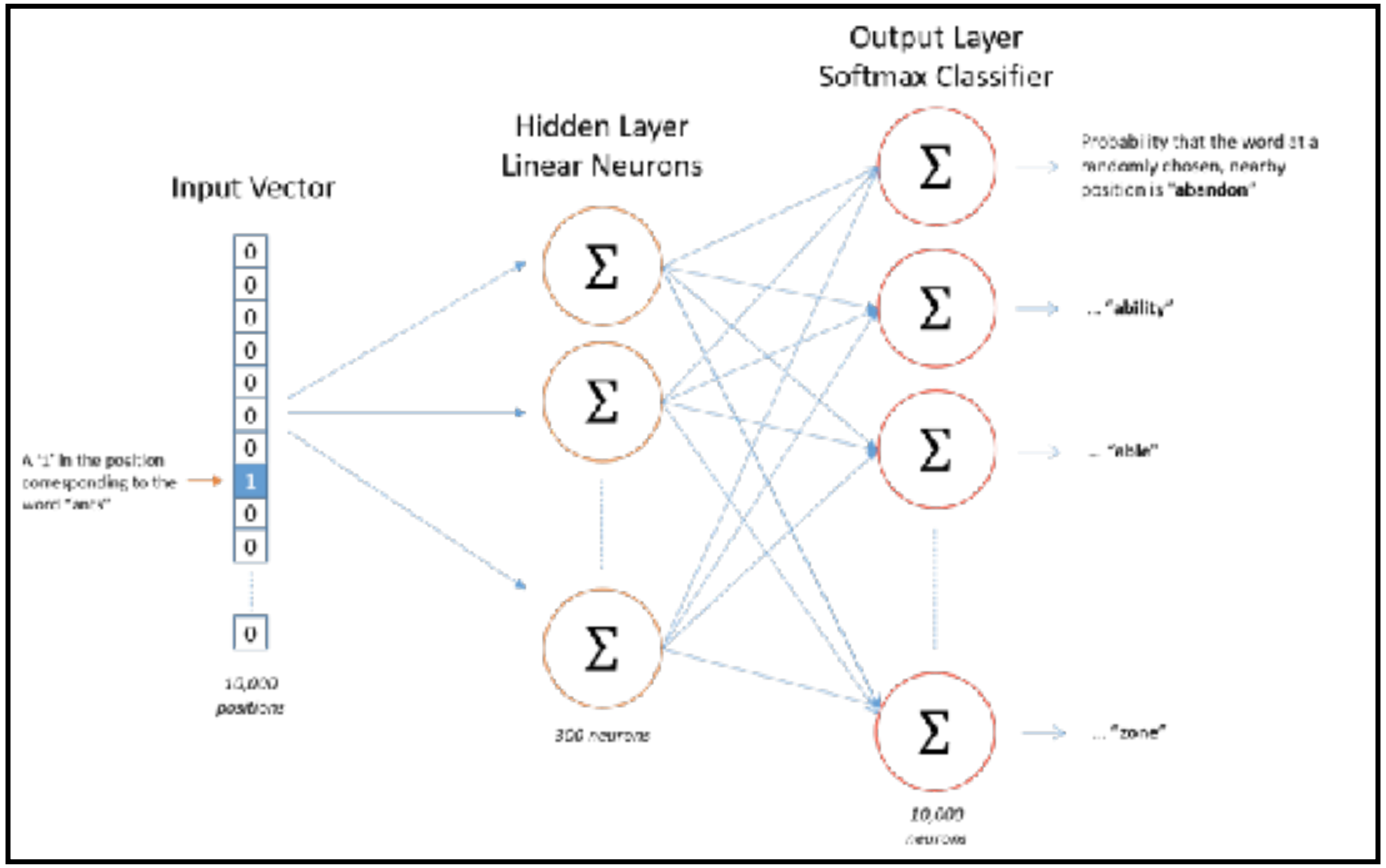
Word2Vec

GloVe
Global Vectors
for Word Representation

Continuous
Skip-gram
architecture

Continuos
Bag-of-Words
Model
(CBOW)

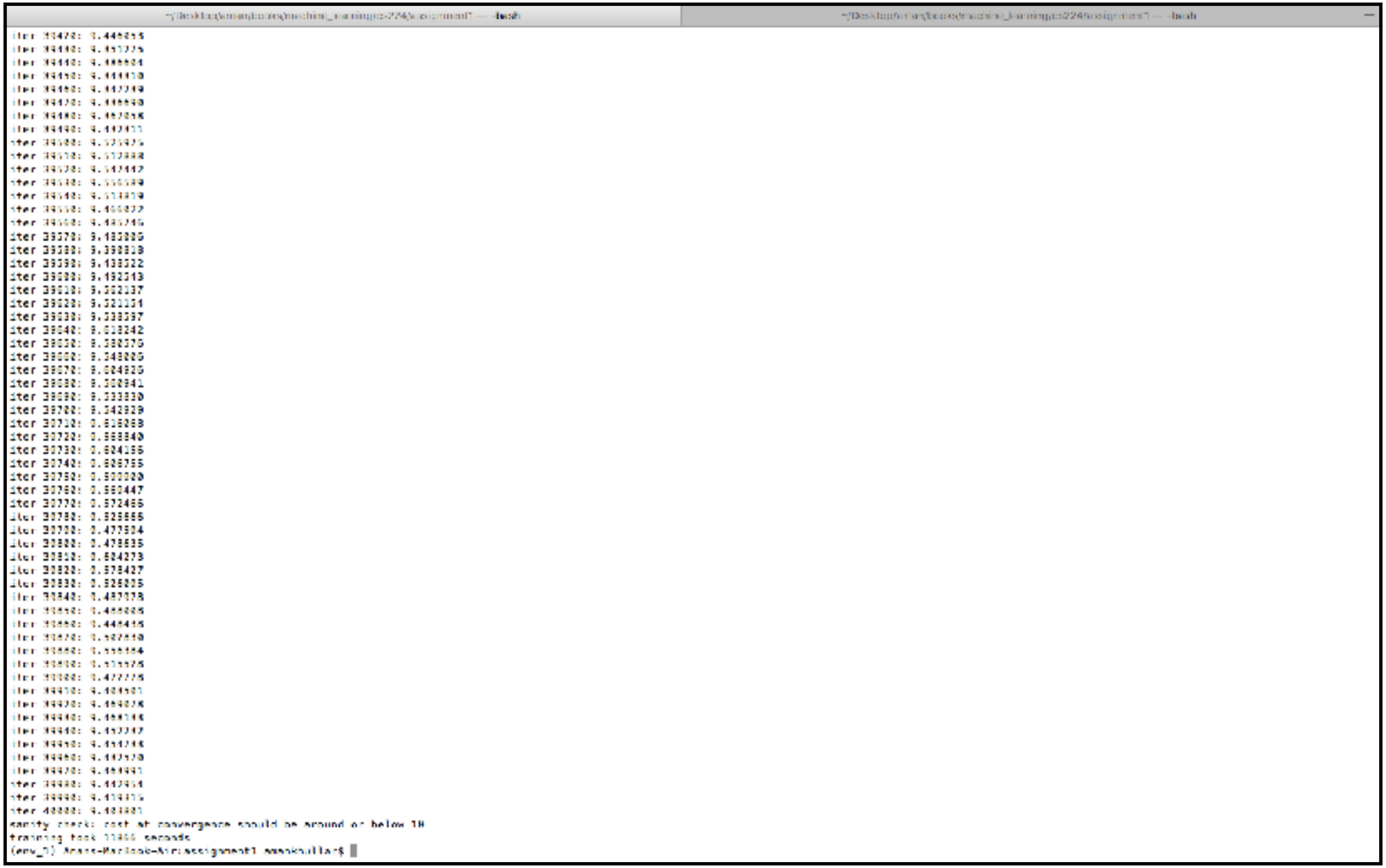
# Skip-Gram Model



$$J(\theta) = -\frac{1}{T}\Sigma_{t=1}^{T}\Sigma_{-m\leq j\leq m} log(p(w_{t+j}|w_t))$$

$$p(o|c) = \frac{exp(u_o^T v_c)}{\Sigma_{w=1}^{v} exp(u_w^T v_c)}$$

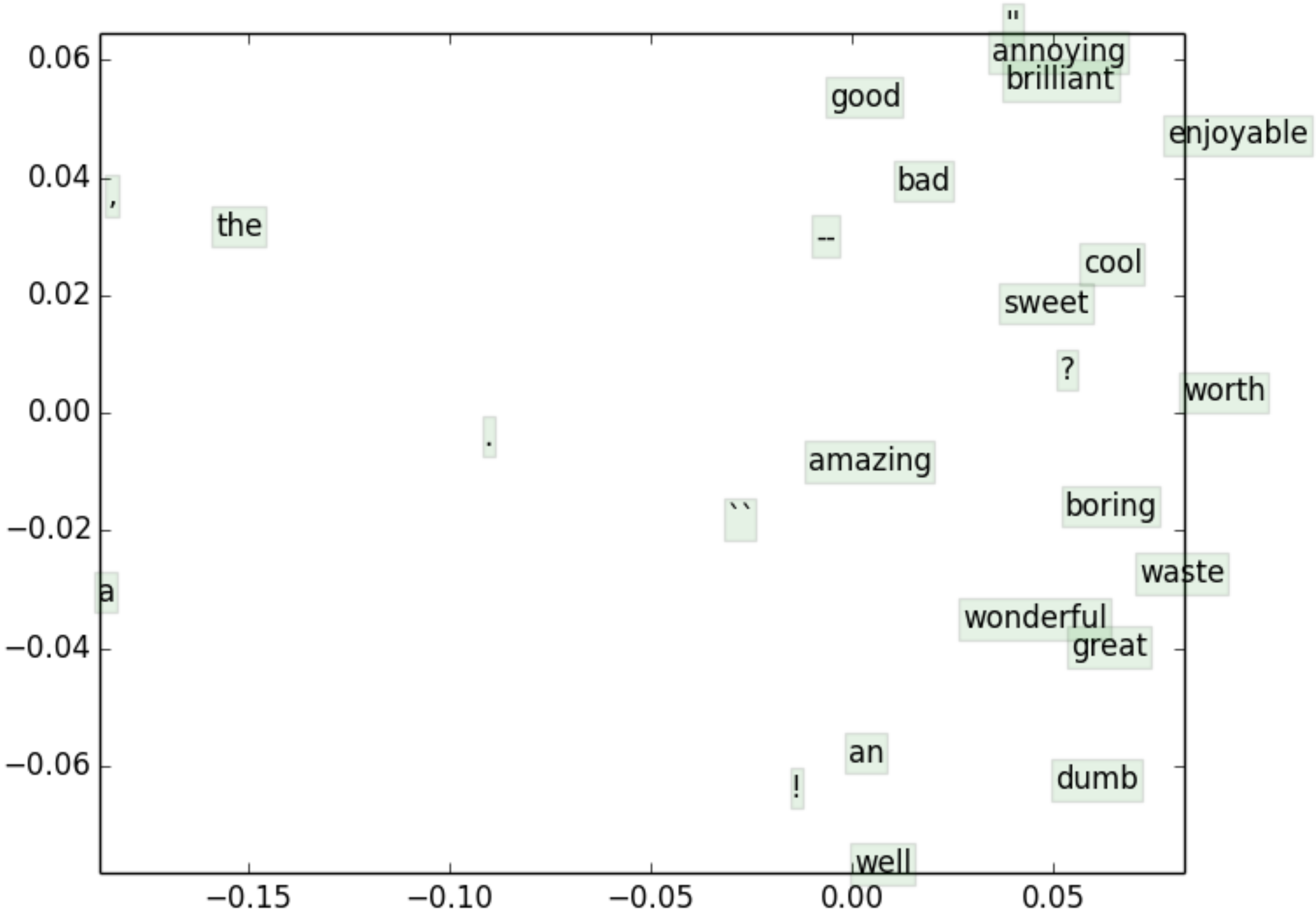
# Skip-Gram Model



~/Desktop/aman/books/machine_learning/cs224/assignment1 — -bash
~/Desktop/aman/books/machine_learning/cs224d/assignment1 — -bash

```
iter 39420: 9.446255
iter 39430: 9.451225
iter 39440: 9.488884
iter 39450: 9.444410
iter 39460: 9.447749
iter 39470: 9.446640
iter 39480: 9.462856
iter 39490: 9.442411
iter 39500: 9.525475
iter 39510: 9.512888
iter 39520: 9.542442
iter 39530: 9.556589
iter 39540: 9.513819
iter 39550: 9.466822
iter 39560: 9.485746
iter 39570: 9.483885
iter 39580: 9.398818
iter 39590: 9.438522
iter 39600: 9.492543
iter 39610: 9.502137
iter 39620: 9.521151
iter 39630: 9.538597
iter 39640: 9.618242
iter 39650: 9.388576
iter 39660: 9.348865
iter 39670: 9.624926
iter 39680: 9.508941
iter 39690: 9.533830
iter 39700: 9.542929
iter 39710: 9.616268
iter 39720: 9.568840
iter 39730: 9.624156
iter 39740: 9.626755
iter 39750: 9.500020
iter 39760: 9.560447
iter 39770: 9.572466
iter 39780: 9.525565
iter 39790: 9.477524
iter 39800: 9.478535
iter 39810: 9.524273
iter 39820: 9.578427
iter 39830: 9.526225
iter 39840: 9.487778
iter 39850: 9.488885
iter 39860: 9.448445
iter 39870: 9.527810
iter 39880: 9.556484
iter 39890: 9.515575
iter 39900: 9.477775
iter 39910: 9.484581
iter 39920: 9.464276
iter 39930: 9.468146
iter 39940: 9.452247
iter 39950: 9.454746
iter 39960: 9.442570
iter 39970: 9.464441
iter 39980: 9.442454
iter 39990: 9.419415
iter 40000: 9.424481
sanity check: cost at convergence should be around or below 10
training took 11866 seconds
(env_1) Amans-MacBook-Air:assignment1 amankhullar$
```

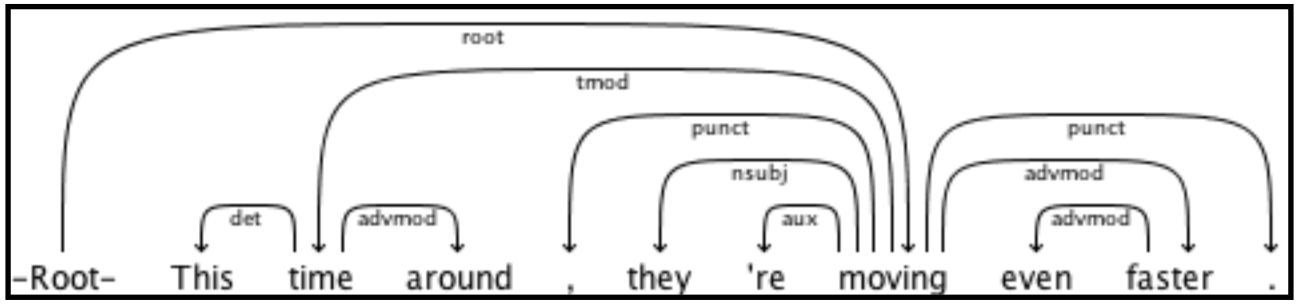
# Skip-Gram Model



# Linguistic Structure

- Two views of linguistic structure : Constituency structure and Dependency structure.
- Constituency structure is the Context Free Grammar form which organizes words into constituents
- Dependency structure shows which words depend on which other words

# Neural Transition Based Dependency Parser



- A dependency parser analyzes the grammatical structure of a sentence, establishing relationship between “head” words and the words which modify those words
- A sentence is parsed by choosing for each word what other word (including ROOT) it is a dependent of.
- This makes dependencies a tree

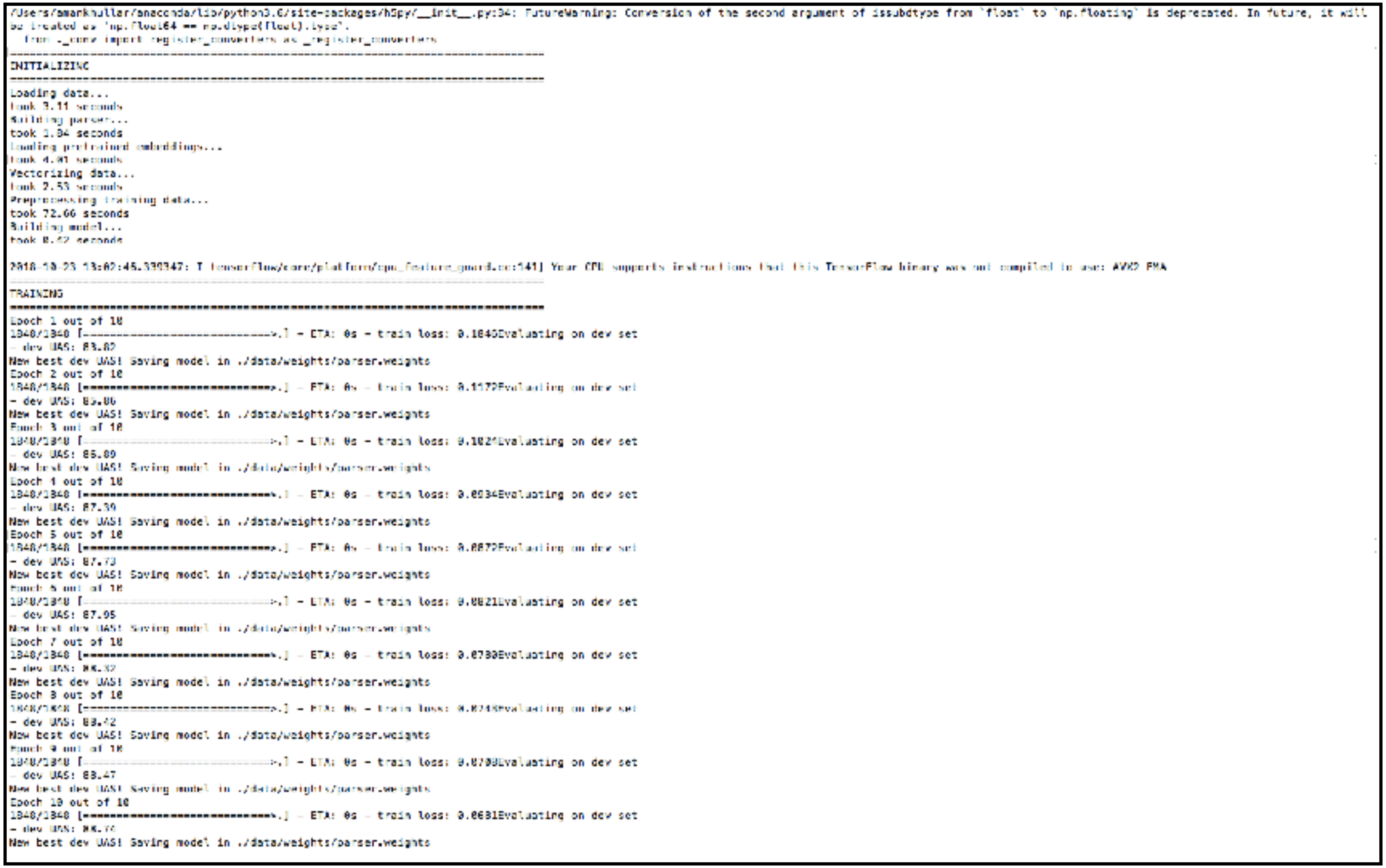
# Neural Transition Based Dependency Parser

- The transition-based dependency parser incrementally builds up a parse one step at a time.
- At each step, it maintains a partial parse with:
    1. A stack of words that are currently being processed.
    2. A buffer if words yet to be processed.
    3. A list of dependencies predicted by the parser.
- Initially the stack contains ROOT, dependencies list is empty and buffer contains all words of sentence in order.
- The list of transitions that can be applied:

  1. Shift        2. Left - Arc        3. Right - Arc

- Neural Network classifier is used at each partial state to decide among the transitions

# Neural Transition Based Dependency Parser



```
/Users/amankhullar/anaconda/lib/python3.6/site-packages/h5py/__init__.py:34: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will
be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
================================================================================
INITIALIZING
================================================================================
Loading data...
took 3.11 seconds
Building parser...
took 1.84 seconds
Loading pretrained embeddings...
took 4.01 seconds
Vectorizing data...
took 2.53 seconds
Preprocessing training data...
took 72.66 seconds
Building model...
took 0.42 seconds

2018-10-23 13:02:45.339347: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
================================================================================
TRAINING
================================================================================
Epoch 1 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.1845Evaluating on dev set
- dev UAS: 83.82
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 2 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.1172Evaluating on dev set
- dev UAS: 85.86
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 3 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.1024Evaluating on dev set
- dev UAS: 86.89
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 4 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0934Evaluating on dev set
- dev UAS: 87.39
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 5 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0872Evaluating on dev set
- dev UAS: 87.73
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 6 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0821Evaluating on dev set
- dev UAS: 87.95
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 7 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0780Evaluating on dev set
- dev UAS: 88.32
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 8 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0743Evaluating on dev set
- dev UAS: 88.42
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 9 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0708Evaluating on dev set
- dev UAS: 88.47
New best dev UAS! Saving model in ./data/weights/parser.weights
Epoch 10 out of 10
1848/1848 [==============================>.] - ETA: 0s - train loss: 0.0681Evaluating on dev set
- dev UAS: 88.74
New best dev UAS! Saving model in ./data/weights/parser.weights
```

# Recurrent Neural Networks

- Language modeling is a central task in NLP
- Given a sequence of words represented as one-hot vectors, a language model predicts the next word $x^{(t+1)}$ by modeling:

$$P(x^{(t+1)} = w_j \,|\, x^{(t)}, x^{(t-1)}, \ldots, x^{(1)})$$

where $w_j$ is a word in vocabulary

**RNN Model**

$$e^{(t)} = Ex^{(t)}$$

$$h^{(t)} = sigmoid(W_h h^{(t-1)} + W_e e^{(t)} + b_1)$$

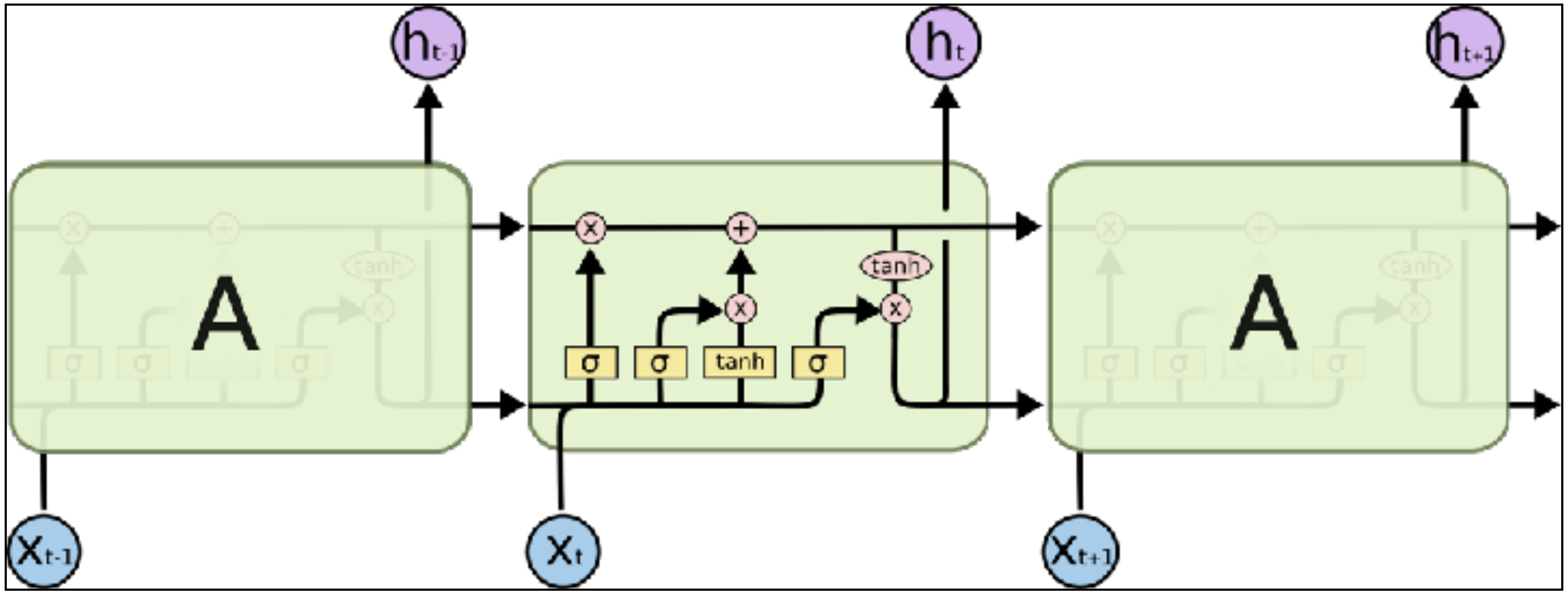
$$\hat{y} = softmax(Uh^{(t)} + b_2)$$

$$P(x^{(t+1)} = w_j \,|\, x^{(t)}, x^{(t-1)}, \ldots, x^{(1)}) = \hat{y}_j^{(t)}$$

The model is trained by minimizing the (un-regularized) cross-entropy loss:

$$J^{(t)}(\theta) = CE(y^{(t)}, \hat{y}^{(t)}) = -\Sigma_{j=1}^{|V|} y_j^{(t)} \log \hat{y}_j^{(t)}$$

# Fancy RNNs - LSTMs and GRUs



$$Forget\ Gate\ :\ f_t = \sigma(W^{(f)}x_{(t)} + U^{(f)}h_{(t-1)} + b_{(f)})$$

$$Input\ Gate\ :\ i_t = \sigma(W^{(i)}x_{(t)} + U^{(i)}h_{(t-1)} + b_{(i)})$$

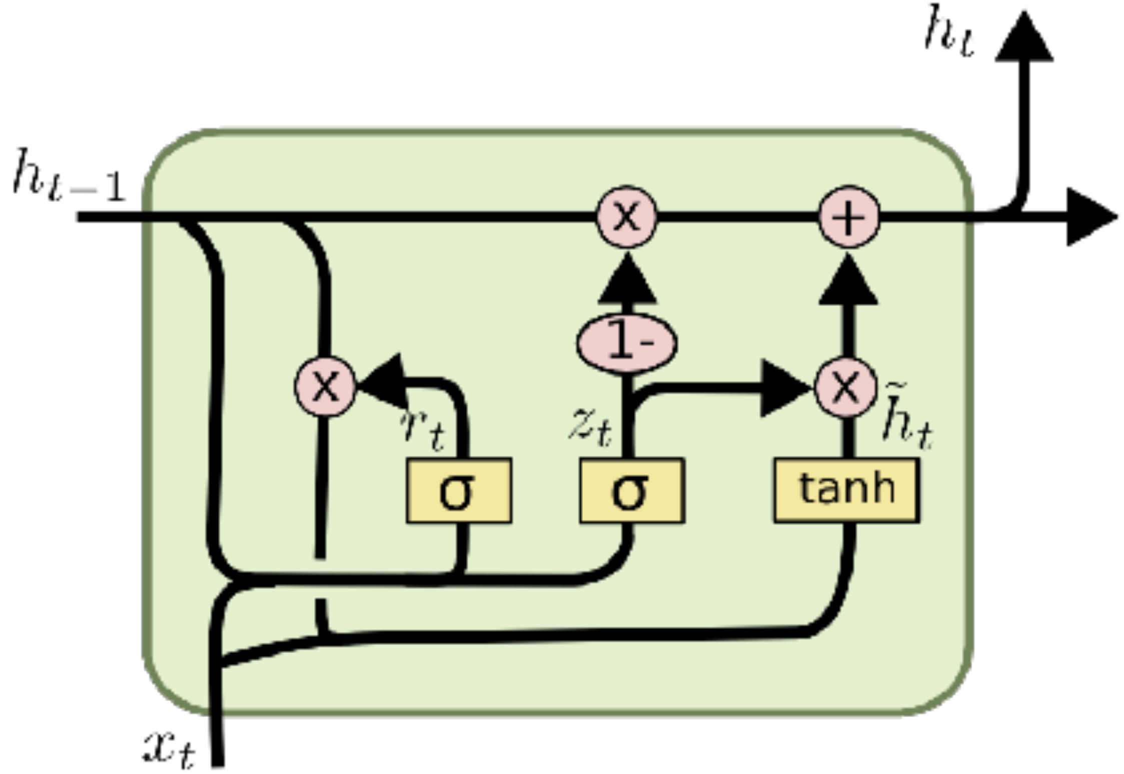
$$Candidate\ Gate\ :\ \tilde{C}_t = \tanh(W^{(c)}x_{(t)} + U^{(c)}h_{(t-1)} + b_{(c)})$$

$$Output\ Gate\ :\ o_t = \sigma(W^{(o)}x_{(t)} + U^{(o)}h_{(t-1)} + b_{(o)})$$

$Cell\ State\ :\ C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t$ **(The addition sign does the magic)**

$$Hidden\ State\ :\ h_t = o_t \circ \tanh(C_t)$$

# Fancy RNNs - LSTMs and GRUs



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

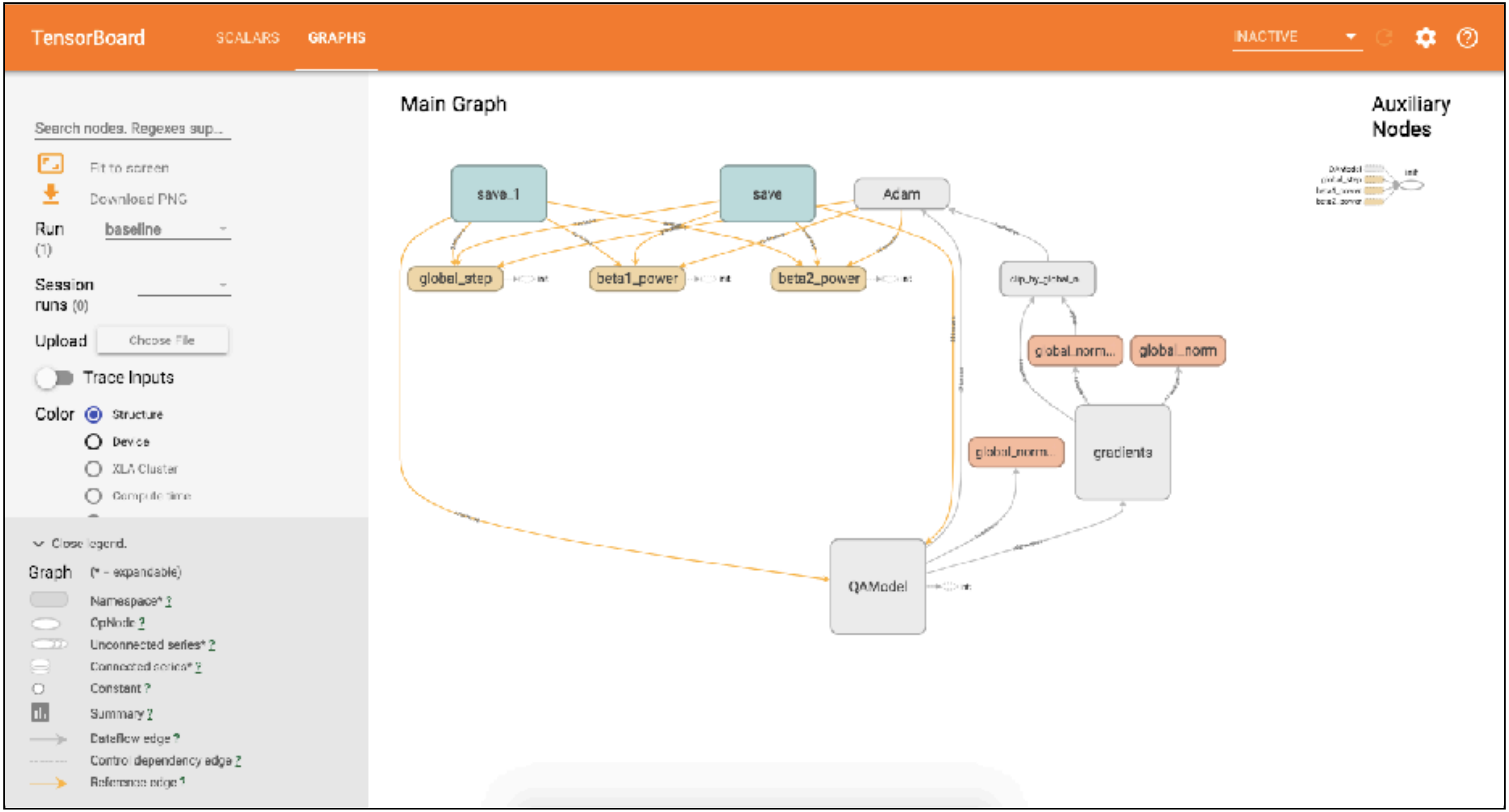
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

# Baseline Model

- The baseline model has three components:
- The RNN encoder layer that encodes both the context and the question into hidden states
- The attention layer that combines the context and the question representations
- Output layer which applies a fully connected layer and then two separate Softmax layers to get the start and the end location of the answer span

# Baseline Model

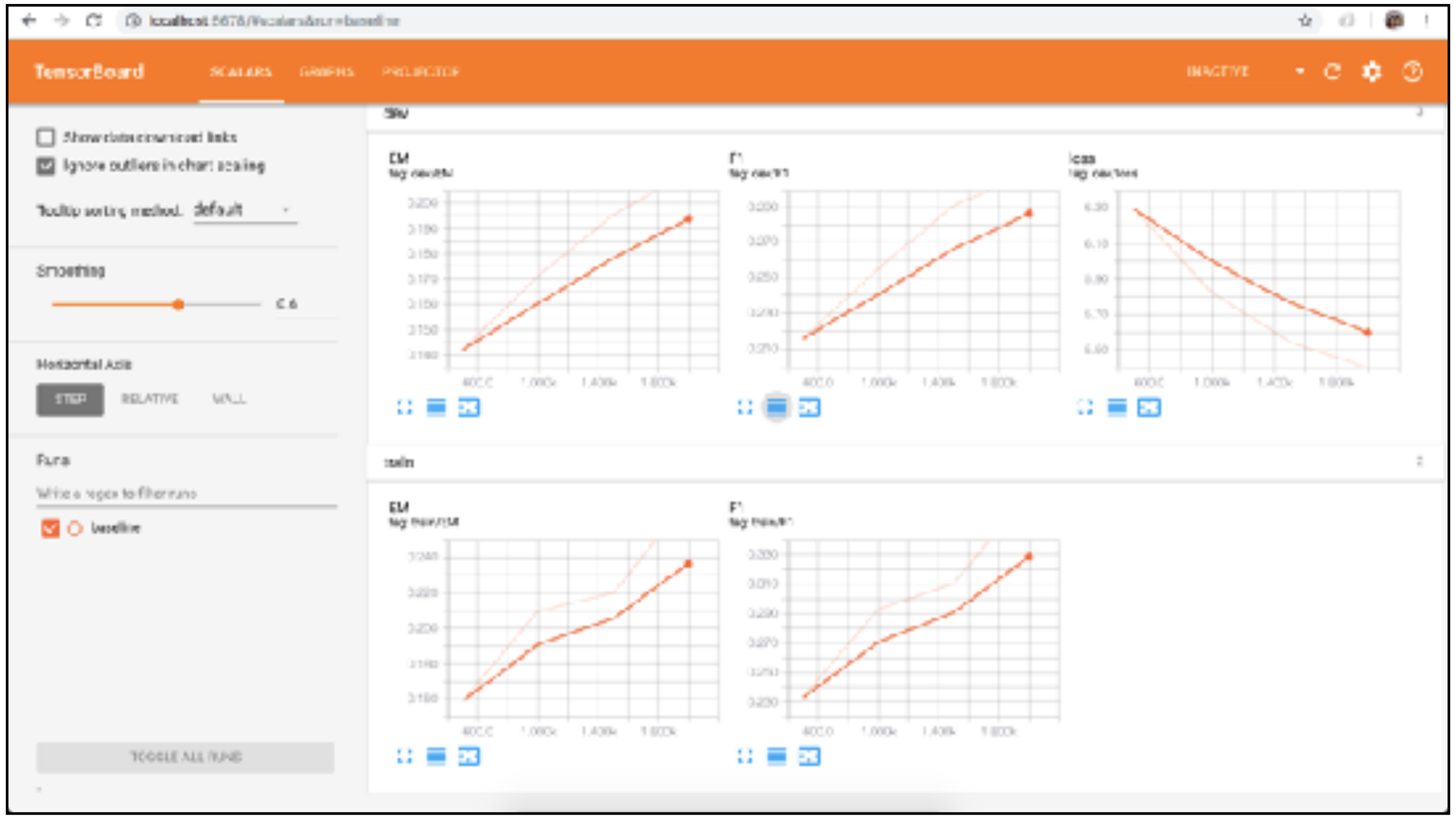


Run (1) baseline

Session runs (0)

Color ● Structure ○ Device ○ XLA Cluster ○ Compute time

# Baseline Model



TensorBoard SCALARS GRAPHS PROJECTOR INACTIVE

☐ Show data download links

☑ Ignore outliers in chart scaling

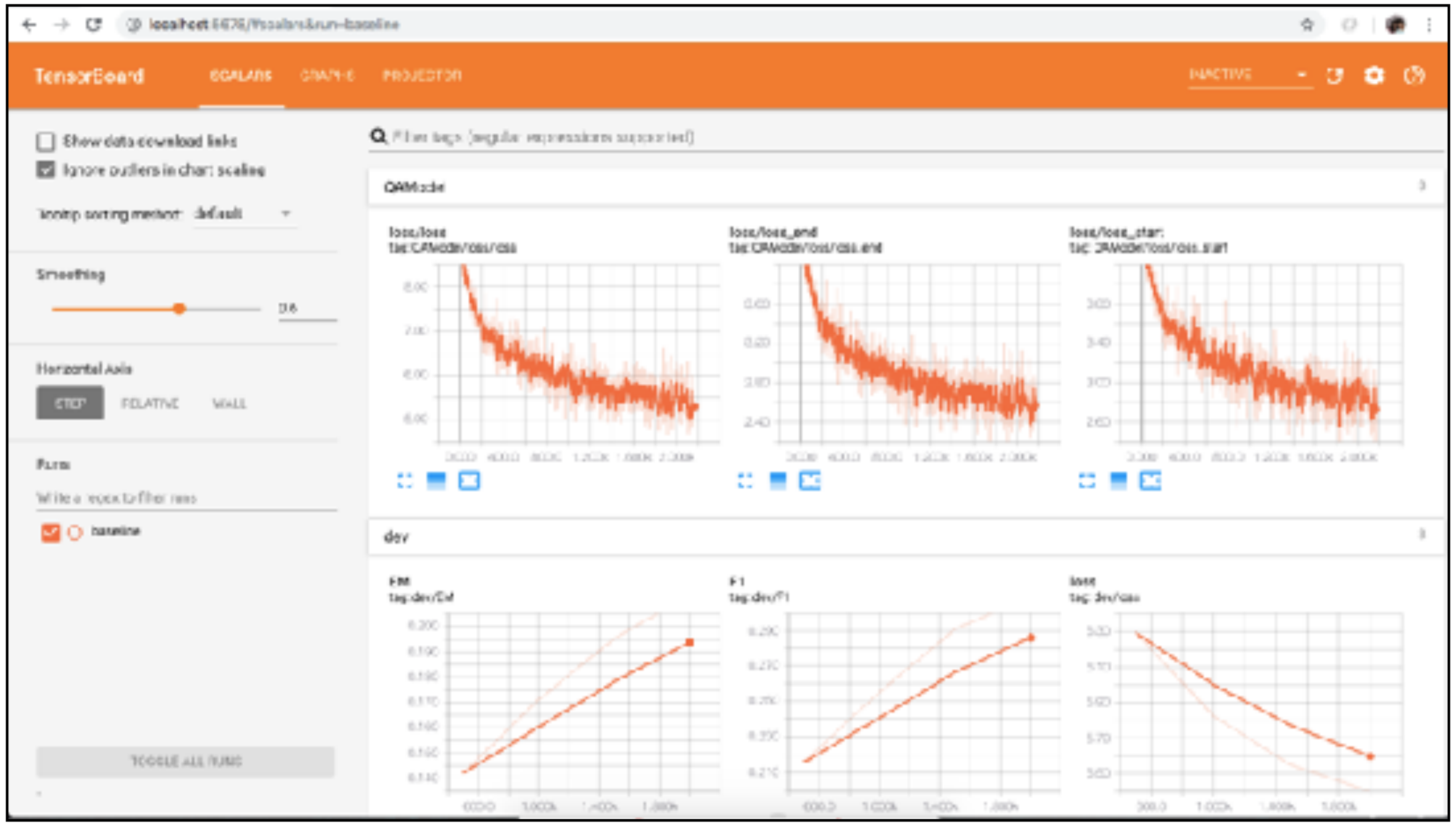
Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis STEP RELATIVE WALL

Runs

☑ ○ baseline



TensorBoard SCALARS GRAPHS PROJECTOR INACTIVE

☐ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

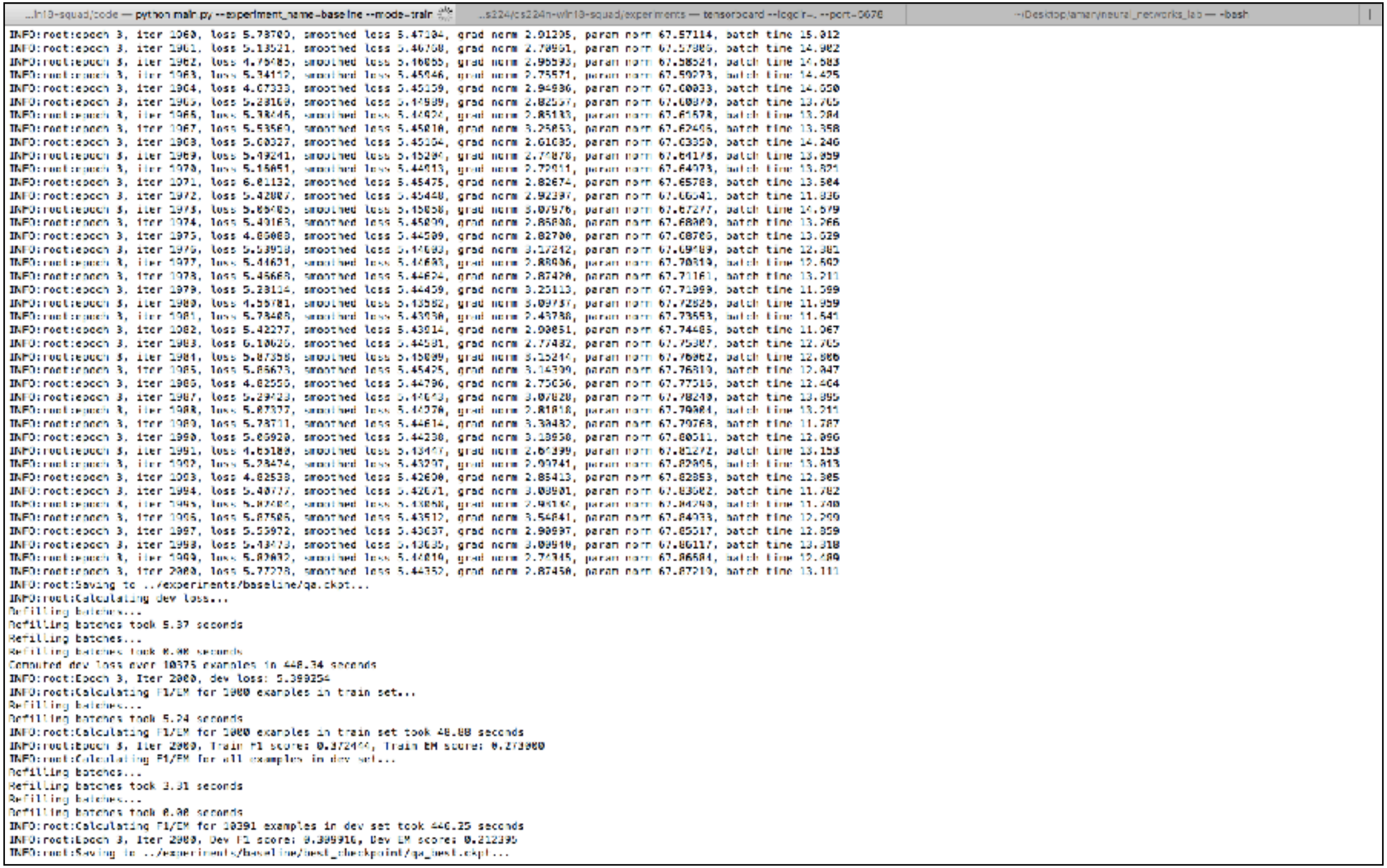
Horizontal Axis STEP RELATIVE WALL

Runs

☑ ○ baseline

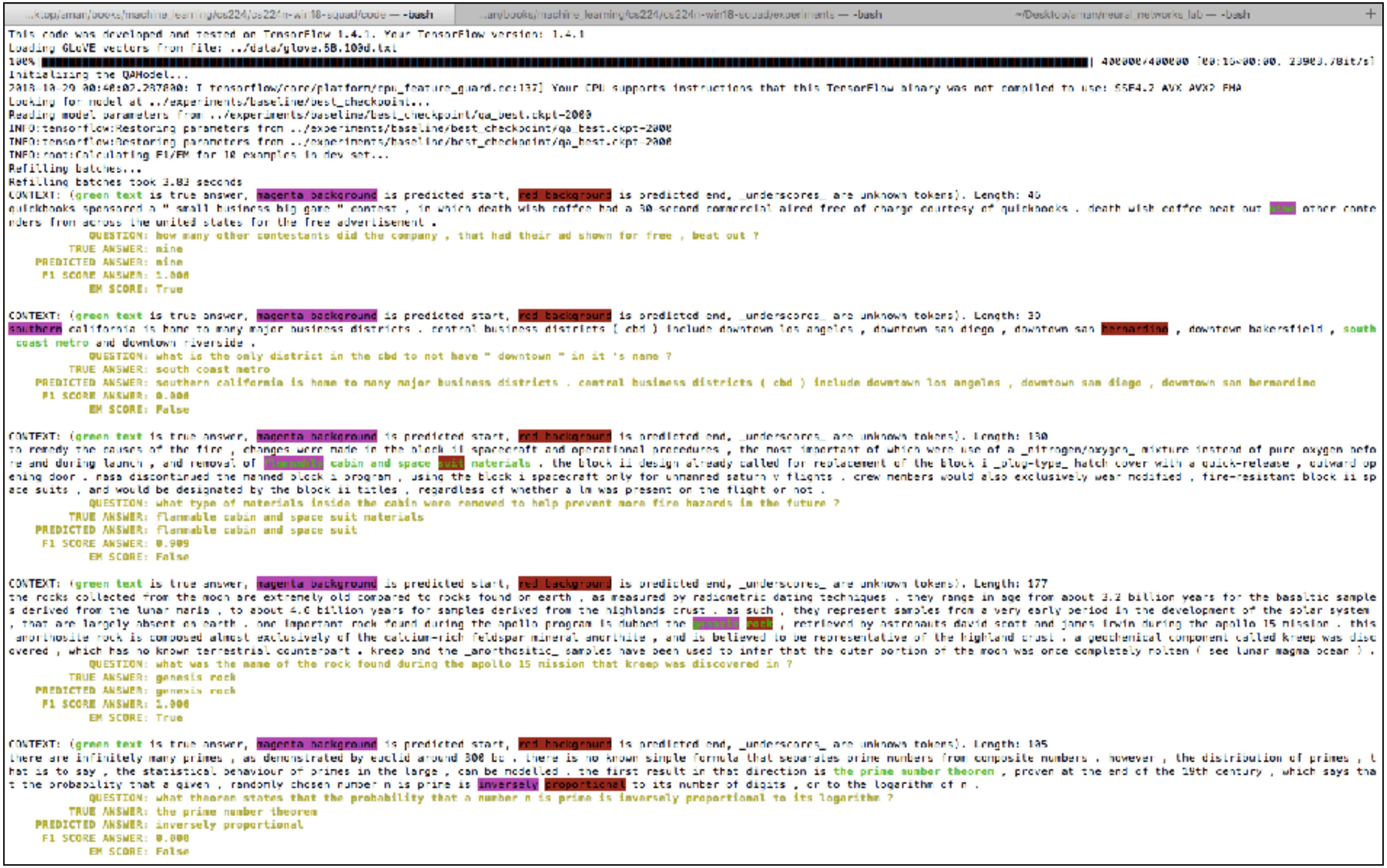
localhost:6676/#scalars&run=baseline
Filter tags (regular expressions supported)
QAModel
loss/loss
tag: QAModel/loss/loss
loss/loss_end
tag: QAModel/loss/loss_end
loss/loss_start
tag: QAModel/loss/loss_start
dev
EM
tag: dev/EM
F1
tag: dev/F1
loss
tag: dev/loss
TOGGLE ALL RUNS

# Baseline Model


...in18-squad/code — python main.py --experiment_name=baseline --mode=train
...s224/cs224n-win18-squad/experiments — tensorboard --logdir=. --port=5678
~/Desktop/amar/neural_networks_lab — -bash

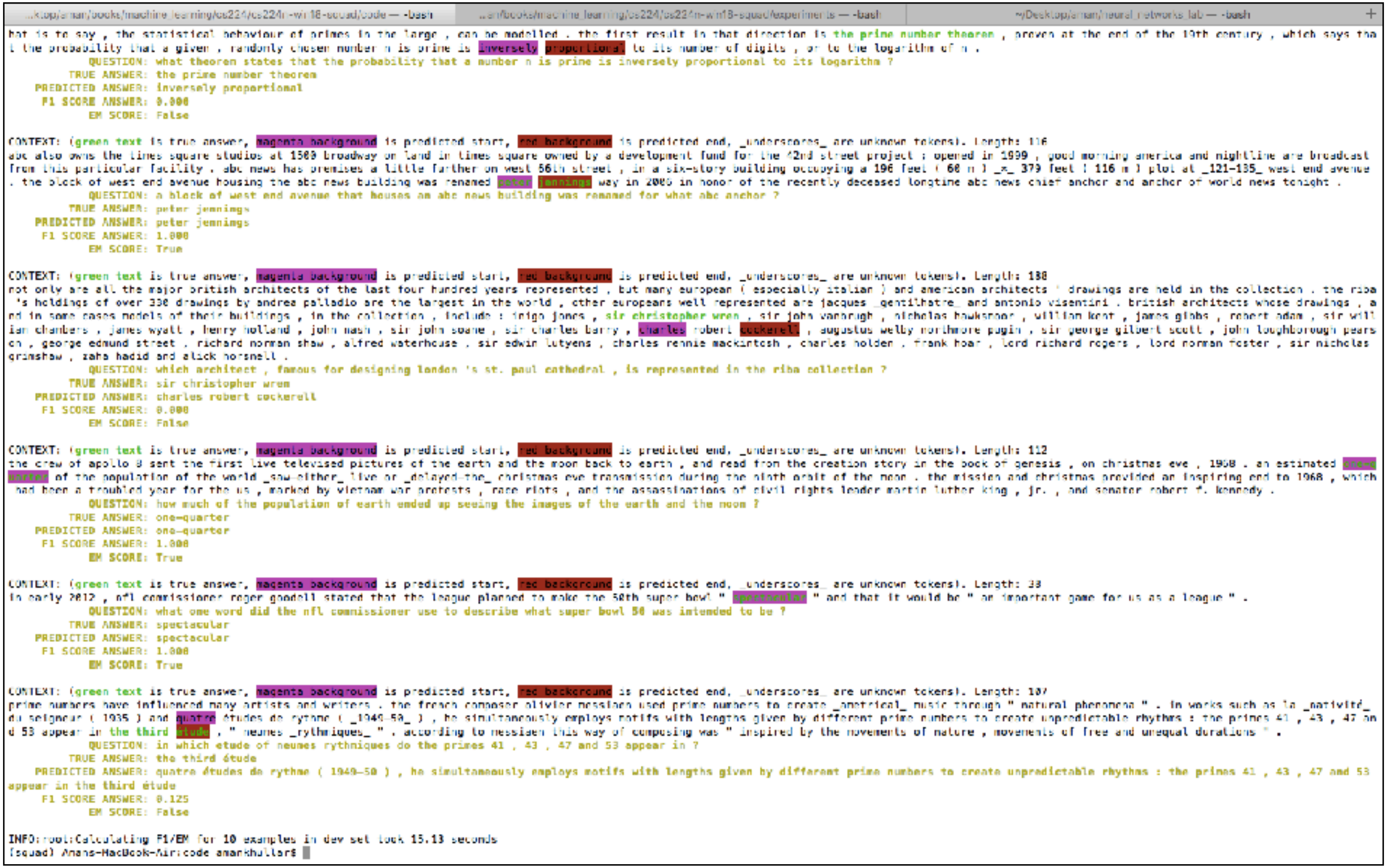
```
INFO:root:epoch 3, iter 1960, loss 5.73700, smoothed loss 5.47104, grad norm 2.91205, param norm 67.57114, batch time 15.012
INFO:root:epoch 3, iter 1961, loss 5.13521, smoothed loss 5.46758, grad norm 2.70951, param norm 67.57806, batch time 14.902
INFO:root:epoch 3, iter 1962, loss 4.75485, smoothed loss 5.46055, grad norm 2.95593, param norm 67.58521, batch time 14.583
INFO:root:epoch 3, iter 1963, loss 5.34112, smoothed loss 5.45946, grad norm 2.75571, param norm 67.59273, batch time 14.425
INFO:root:epoch 3, iter 1964, loss 4.07333, smoothed loss 5.45159, grad norm 2.94936, param norm 67.60033, batch time 14.050
INFO:root:epoch 3, iter 1965, loss 5.28160, smoothed loss 5.44989, grad norm 2.82557, param norm 67.60870, batch time 13.765
INFO:root:epoch 3, iter 1966, loss 5.38445, smoothed loss 5.44924, grad norm 2.85133, param norm 67.61578, batch time 13.284
INFO:root:epoch 3, iter 1967, loss 5.53569, smoothed loss 5.45010, grad norm 3.25053, param norm 67.62495, batch time 13.358
INFO:root:epoch 3, iter 1968, loss 5.60327, smoothed loss 5.45164, grad norm 2.61035, param norm 67.63350, batch time 14.246
INFO:root:epoch 3, iter 1969, loss 5.49241, smoothed loss 5.45204, grad norm 2.74878, param norm 67.64175, batch time 13.859
INFO:root:epoch 3, iter 1970, loss 5.15051, smoothed loss 5.44913, grad norm 2.72911, param norm 67.64973, batch time 13.821
INFO:root:epoch 3, iter 1971, loss 6.01132, smoothed loss 5.45475, grad norm 2.82674, param norm 67.65783, batch time 13.504
INFO:root:epoch 3, iter 1972, loss 5.42007, smoothed loss 5.45448, grad norm 2.92397, param norm 67.66541, batch time 11.836
INFO:root:epoch 3, iter 1973, loss 5.85485, smoothed loss 5.45858, grad norm 3.07976, param norm 67.67277, batch time 14.579
INFO:root:epoch 3, iter 1974, loss 5.40163, smoothed loss 5.45809, grad norm 2.85808, param norm 67.68009, batch time 13.266
INFO:root:epoch 3, iter 1975, loss 4.85083, smoothed loss 5.44509, grad norm 2.82700, param norm 67.68705, batch time 13.529
INFO:root:epoch 3, iter 1976, loss 5.53918, smoothed loss 5.44693, grad norm 3.17242, param norm 67.69489, batch time 12.381
INFO:root:epoch 3, iter 1977, loss 5.44621, smoothed loss 5.44693, grad norm 2.88906, param norm 67.70319, batch time 12.592
INFO:root:epoch 3, iter 1978, loss 5.45668, smoothed loss 5.44624, grad norm 2.87420, param norm 67.71161, batch time 13.211
INFO:root:epoch 3, iter 1979, loss 5.23114, smoothed loss 5.44459, grad norm 3.25113, param norm 67.71999, batch time 11.399
INFO:root:epoch 3, iter 1980, loss 4.55781, smoothed loss 5.43582, grad norm 3.09737, param norm 67.72826, batch time 11.959
INFO:root:epoch 3, iter 1981, loss 5.78408, smoothed loss 5.43930, grad norm 2.43788, param norm 67.73653, batch time 11.541
INFO:root:epoch 3, iter 1982, loss 5.42277, smoothed loss 5.43914, grad norm 2.90051, param norm 67.74485, batch time 11.067
INFO:root:epoch 3, iter 1983, loss 6.10626, smoothed loss 5.44581, grad norm 2.77482, param norm 67.75307, batch time 12.765
INFO:root:epoch 3, iter 1984, loss 5.87358, smoothed loss 5.45009, grad norm 3.15214, param norm 67.76062, batch time 12.806
INFO:root:epoch 3, iter 1985, loss 5.86673, smoothed loss 5.45425, grad norm 3.14309, param norm 67.76810, batch time 12.047
INFO:root:epoch 3, iter 1986, loss 4.82555, smoothed loss 5.44796, grad norm 2.75656, param norm 67.77516, batch time 12.404
INFO:root:epoch 3, iter 1987, loss 5.29423, smoothed loss 5.44643, grad norm 3.07828, param norm 67.78240, batch time 13.895
INFO:root:epoch 3, iter 1988, loss 5.07377, smoothed loss 5.44270, grad norm 2.81818, param norm 67.79004, batch time 13.211
INFO:root:epoch 3, iter 1989, loss 5.78711, smoothed loss 5.44614, grad norm 3.30482, param norm 67.79768, batch time 11.787
INFO:root:epoch 3, iter 1990, loss 5.05920, smoothed loss 5.44238, grad norm 3.18958, param norm 67.80511, batch time 12.096
INFO:root:epoch 3, iter 1991, loss 4.65180, smoothed loss 5.43447, grad norm 2.64399, param norm 67.81272, batch time 13.153
INFO:root:epoch 3, iter 1992, loss 5.28474, smoothed loss 5.43297, grad norm 2.99741, param norm 67.82095, batch time 13.013
INFO:root:epoch 3, iter 1993, loss 4.82538, smoothed loss 5.42690, grad norm 2.85413, param norm 67.82853, batch time 12.305
INFO:root:epoch 3, iter 1994, loss 5.40777, smoothed loss 5.42671, grad norm 3.08901, param norm 67.83582, batch time 11.782
INFO:root:epoch 3, iter 1995, loss 5.82484, smoothed loss 5.43068, grad norm 2.98134, param norm 67.84290, batch time 11.740
INFO:root:epoch 3, iter 1996, loss 5.87505, smoothed loss 5.43512, grad norm 3.54841, param norm 67.84933, batch time 12.299
INFO:root:epoch 3, iter 1997, loss 5.55972, smoothed loss 5.43637, grad norm 2.90997, param norm 67.85517, batch time 12.859
INFO:root:epoch 3, iter 1998, loss 5.43473, smoothed loss 5.43635, grad norm 3.00910, param norm 67.86117, batch time 13.318
INFO:root:epoch 3, iter 1999, loss 5.82032, smoothed loss 5.44019, grad norm 2.74345, param norm 67.86584, batch time 12.489
INFO:root:epoch 3, iter 2000, loss 5.77278, smoothed loss 5.44352, grad norm 2.87450, param norm 67.87210, batch time 13.111
INFO:root:Saving to ../experiments/baseline/qa.ckpt...
INFO:root:Calculating dev loss...
Refilling batches...
Refilling batches took 5.37 seconds
Refilling batches...
Refilling batches took 0.00 seconds
Computed dev loss over 10375 examples in 448.34 seconds
INFO:root:Epoch 3, Iter 2000, dev loss: 5.399254
INFO:root:Calculating F1/EM for 1000 examples in train set...
Refilling batches...
Refilling batches took 5.24 seconds
INFO:root:Calculating F1/EM for 1000 examples in train set took 48.88 seconds
INFO:root:Epoch 3, Iter 2000, Train F1 score: 0.372444, Train EM score: 0.273000
INFO:root:Calculating F1/EM for all examples in dev set...
Refilling batches...
Refilling batches took 3.31 seconds
Refilling batches...
Refilling batches took 0.00 seconds
INFO:root:Calculating F1/EM for 10391 examples in dev set took 446.25 seconds
INFO:root:Epoch 3, Iter 2000, Dev F1 score: 0.309916, Dev EM score: 0.212395
INFO:root:Saving to ../experiments/baseline/best_checkpoint/qa_best.ckpt...
```

# Baseline Model


...ktop/aman/books/machine_learning/cs224/cs224n-win18-squad/code — -bash
...anybooks/machine_learning/cs224/cs224n-win18-squad/experiments — -bash
~/Desktop/aman/neural_networks_lab — -bash
+

```
This code was developed and tested on TensorFlow 1.4.1. Your TensorFlow version: 1.4.1
Loading GLoVE vectors from file: ../data/glove.6B.100d.txt
100%|████████████████████████████████████████████████████████████████████████████████| 400000/400000 [00:15<00:00, 23903.78it/s]
Initializing the QAModel...
2018-10-29 00:40:02.287800: I tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary was not compiled to use: SSE4.2 AVX AVX2 FMA
Looking for model at ../experiments/baseline/best_checkpoint...
Reading model parameters from ../experiments/baseline/best_checkpoint/qa_best.ckpt-2000
INFO:tensorflow:Restoring parameters from ../experiments/baseline/best_checkpoint/qa_best.ckpt-2000
INFO:tensorflow:Restoring parameters from ../experiments/baseline/best_checkpoint/qa_best.ckpt-2000
INFO:root:Calculating F1/EM for 10 examples in dev set...
Refilling batches...
Refilling batches took 3.83 seconds
CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 46
quickbooks sponsored a " small business big game " contest , in which death wish coffee had a 30-second commercial aired free of charge courtesy of quickbooks . death wish coffee beat out nine other conte
nders from across the united states for the free advertisement .
            QUESTION: how many other contestants did the company , that had their ad shown for free , beat out ?
          TRUE ANSWER: nine
     PREDICTED ANSWER: nine
      F1 SCORE ANSWER: 1.000
            EM SCORE: True

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 30
southern california is home to many major business districts . central business districts ( cbd ) include downtown los angeles , downtown san diego , downtown san bernardino , downtown bakersfield , south
 coast metro and downtown riverside .
            QUESTION: what is the only district in the cbd to not have " downtown " in it 's name ?
          TRUE ANSWER: south coast metro
     PREDICTED ANSWER: southern california is home to many major business districts . central business districts ( cbd ) include downtown los angeles , downtown san diego , downtown san bernardino
      F1 SCORE ANSWER: 0.000
            EM SCORE: False

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 130
to remedy the causes of the fire , changes were made in the block ii spacecraft and operational procedures , the most important of which were use of a _nitrogen/oxygen_ mixture instead of pure oxygen befo
re and during launch , and removal of flammable cabin and space suit materials . the block ii design already called for replacement of the block i _plug-type_ hatch cover with a quick-release , outward op
ening door . nasa discontinued the manned block i program , using the block i spacecraft only for unmanned saturn v flights . crew members would also exclusively wear modified , fire-resistant block ii sp
ace suits , and would be designated by the block ii titles , regardless of whether a lm was present on the flight or not .
            QUESTION: what type of materials inside the cabin were removed to help prevent more fire hazards in the future ?
          TRUE ANSWER: flammable cabin and space suit materials
     PREDICTED ANSWER: flammable cabin and space suit
      F1 SCORE ANSWER: 0.909
            EM SCORE: False

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 177
the rocks collected from the moon are extremely old compared to rocks found on earth , as measured by radiometric dating techniques . they range in age from about 3.2 billion years for the basaltic sample
s derived from the lunar maria , to about 4.6 billion years for samples derived from the highlands crust . as such , they represent samples from a very early period in the development of the solar system
, that are largely absent on earth . one important rock found during the apollo program is dubbed the genesis rock , retrieved by astronauts david scott and james irwin during the apollo 15 mission . this
 anorthosite rock is composed almost exclusively of the calcium-rich feldspar mineral anorthite , and is believed to be representative of the highland crust . a geochemical component called kreep was disc
overed , which has no known terrestrial counterpart . kreep and the _anorthositic_ samples have been used to infer that the outer portion of the moon was once completely molten ( see lunar magma ocean ) .
            QUESTION: what was the name of the rock found during the apollo 15 mission that kreep was discovered in ?
          TRUE ANSWER: genesis rock
     PREDICTED ANSWER: genesis rock
      F1 SCORE ANSWER: 1.000
            EM SCORE: True

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 105
there are infinitely many primes , as demonstrated by euclid around 300 bc . there is no known simple formula that separates prime numbers from composite numbers . however , the distribution of primes , t
hat is to say , the statistical behaviour of primes in the large , can be modelled . the first result in that direction is the prime number theorem , proven at the end of the 19th century , which says tha
t the probability that a given , randomly chosen number n is prime is inversely proportional to its number of digits , or to the logarithm of n .
            QUESTION: what theorem states that the probability that a number n is prime is inversely proportional to its logarithm ?
          TRUE ANSWER: the prime number theorem
     PREDICTED ANSWER: inversely proportional
      F1 SCORE ANSWER: 0.000
            EM SCORE: False
```

# Baseline Model


...ktop/aman/books/machine_learning/cs224/cs224n-win18-squad/code — -bash
...an/books/machine_learning/cs224/cs224n-win18-squad/experiments — -bash
~/Desktop/aman/neural_networks_lab — -bash
+

```
hat is to say , the statistical behaviour of primes in the large , can be modelled . the first result in that direction is the prime number theorem , proven at the end of the 19th century , which says tha
t the probability that a given , randomly chosen number n is prime is inversely proportional to its number of digits , or to the logarithm of n .
            QUESTION: what theorem states that the probability that a number n is prime is inversely proportional to its logarithm ?
         TRUE ANSWER: the prime number theorem
    PREDICTED ANSWER: inversely proportional
     F1 SCORE ANSWER: 0.000
            EM SCORE: False

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 116
abc also owns the times square studios at 1500 broadway on land in times square owned by a development fund for the 42nd street project ; opened in 1999 , good morning america and nightline are broadcast
from this particular facility . abc news has premises a little farther on west 56th street , in a six-story building occupying a 196 feet ( 60 m ) _×_ 379 feet ( 116 m ) plot at _121-135_ west end avenue
. the block of west end avenue housing the abc news building was renamed peter jennings way in 2006 in honor of the recently deceased longtime abc news chief anchor and anchor of world news tonight .
            QUESTION: a block of west end avenue that houses an abc news building was renamed for what abc anchor ?
         TRUE ANSWER: peter jennings
    PREDICTED ANSWER: peter jennings
     F1 SCORE ANSWER: 1.000
            EM SCORE: True

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 188
not only are all the major british architects of the last four hundred years represented , but many european ( especially italian ) and american architects ' drawings are held in the collection . the riba
 's holdings of over 330 drawings by andrea palladio are the largest in the world , other europeans well represented are jacques _gentilhatre_ and antonio visentini . british architects whose drawings , a
nd in some cases models of their buildings , in the collection , include : inigo jones , sir christopher wren , sir john vanbrugh , nicholas hawksmoor , william kent , james gibbs , robert adam , sir will
iam chambers , james wyatt , henry holland , john nash , sir john soane , sir charles barry , charles robert cockerell , augustus welby northmore pugin , sir george gilbert scott , john loughborough pears
on , george edmund street , richard norman shaw , alfred waterhouse , sir edwin lutyens , charles rennie mackintosh , charles holden , frank hoar , lord richard rogers , lord norman foster , sir nicholas
grimshaw , zaha hadid and alick horsnell .
            QUESTION: which architect , famous for designing london 's st. paul cathedral , is represented in the riba collection ?
         TRUE ANSWER: sir christopher wren
    PREDICTED ANSWER: charles robert cockerell
     F1 SCORE ANSWER: 0.000
            EM SCORE: False

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 112
the crew of apollo 8 sent the first live televised pictures of the earth and the moon back to earth , and read from the creation story in the book of genesis , on christmas eve , 1968 . an estimated one-q
uarter of the population of the world _saw-either_ live or _delayed-the_ christmas eve transmission during the ninth orbit of the moon . the mission and christmas provided an inspiring end to 1968 , which
 had been a troubled year for the us , marked by vietnam war protests , race riots , and the assassinations of civil rights leader martin luther king , jr. , and senator robert f. kennedy .
            QUESTION: how much of the population of earth ended up seeing the images of the earth and the moon ?
         TRUE ANSWER: one-quarter
    PREDICTED ANSWER: one-quarter
     F1 SCORE ANSWER: 1.000
            EM SCORE: True

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 38
in early 2012 , nfl commissioner roger goodell stated that the league planned to make the 50th super bowl " spectacular " and that it would be " an important game for us as a league " .
            QUESTION: what one word did the nfl commissioner use to describe what super bowl 50 was intended to be ?
         TRUE ANSWER: spectacular
    PREDICTED ANSWER: spectacular
     F1 SCORE ANSWER: 1.000
            EM SCORE: True

CONTEXT: (green text is true answer, magenta background is predicted start, red background is predicted end, _underscores_ are unknown tokens). Length: 107
prime numbers have influenced many artists and writers . the french composer olivier messiaen used prime numbers to create _ametrical_ music through " natural phenomena " . in works such as la _nativité_
du seigneur ( 1935 ) and quatre études de rythme ( _1949-50_ ) , he simultaneously employs motifs with lengths given by different prime numbers to create unpredictable rhythms : the primes 41 , 43 , 47 an
d 53 appear in the third étude , " neumes _rythmiques_ " . according to messiaen this way of composing was " inspired by the movements of nature , movements of free and unequal durations " .
            QUESTION: in which etude of neumes rythmiques do the primes 41 , 43 , 47 and 53 appear in ?
         TRUE ANSWER: the third étude
    PREDICTED ANSWER: quatre études de rythme ( 1949-50 ) , he simultaneously employs motifs with lengths given by different prime numbers to create unpredictable rhythms : the primes 41 , 43 , 47 and 53
appear in the third étude
     F1 SCORE ANSWER: 0.125
            EM SCORE: False

INFO:root:Calculating F1/EM for 10 examples in dev set took 15.13 seconds
(squad) Amans-MacBook-Air:code amankhullar$ 
```

# Improvements and References

- The baseline model can be improved by replacing the bidirectional GRUs by other models :

1. Bidirectional attention flow
2. Coattention
3. Self-attention
4. Character level CNN

- References:

1. CS224n Stanford course on Natural Language processing and Deep Learning

2. McCormick, C. (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model.

3. T. Mikolov et al, Efficient Estimation of Word Representations in Vector Space

4. Danqi Chen and Christopher Manning. 2014.
A Fast and Accurate Dependency Parser Using Neural Networks

5. Christopher Olah's, 'colah's blog' on understanding LSTM Networks