CS 230

Duplicate Question Detection

{dougc, shgu} @stanford.edu

Applications: Quora/Slack/Twitter/email

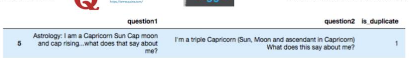
Substantial amount of communication contains Questions.

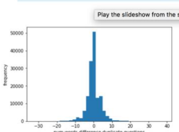
How to automate answering of questions?

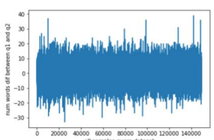
https://youtu.be/fSpvfPnMXuQ

Data Set

Unbalanced dataset

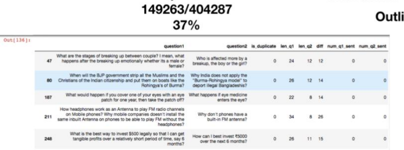






duplicates
149263/404287
37%

Outliers waste memory



nonduplicates
255024/404287
63%

biLSTM Architectures

Start with basic biLSTM + Linear

Good: no overfitting
Bad: poor test accuracy

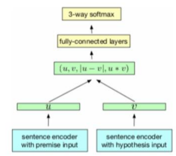


Figure 1: Generic NLI training scheme.

SNLI RTE 2017
What has changed?
Bigger GPUs, V100 16GB

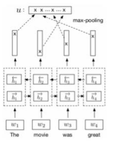
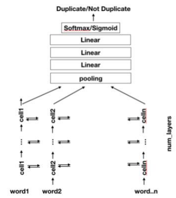


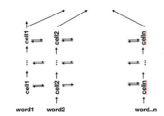
Figure 2: Bi-LSTM max-pooling network.



Modified to 2 classes, num_layers > 1

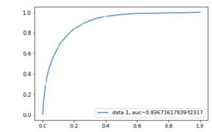
Regularization to Reduce Overfitting

Best performance is with Bi-LSTM @ 83%



Best model used q1,q2, IA + dropout + BN+Tanh 16GB limit

Model	train	dev	test	diff
Bi-LSTM linear	62.8	62.5	62.7	.3
Bi-LSTM(I) 3L	89.0	79.4	78.7	10.4
Bi-LSTM(I) 3L Dropout n_h=2048 n_l=2	83.3	78.7	78.8	4.5
Bi-LSTM(I) 3L Relu n_h=4096	90.2	79.2	79.0	11.2
Bi-LSTM(big) 3L Tanh n_h = 2048	95.06	83.3	83.4	11.65
Bi-LSTM(clean) 3L Tanh n_h = 2048	94.53	79.48	79.31	15.2
hierarchial attention	76.9	77.5	77.8	.59



biLSTM Architectures

$$S = w_1, w_2, w_3, \ldots w_K$$

$$\sum \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

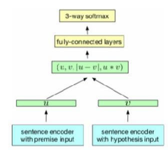


Figure 1: Generic NLI training scheme.

Modified to 2 classes, num_layers > 1

Increase accuracy 16%

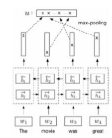


Figure 2: Bi-LSTM max-pooling network.

Increase performance 1-3%

Hierarchial Attention

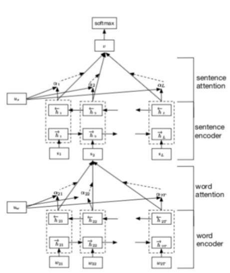
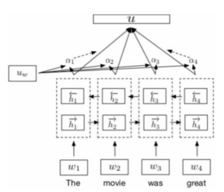


Figure 2: Hierarchical Attention Network.

Inner Attention



num sentences	question 1	question 2
0	376717	380851
1	18615	16396
2	7040	5279
3	1615	1117
4	550	400
5	162	95
6	92	61
7	36	15
8	17	8
9	21	6

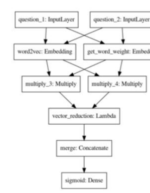
Approach II: Siamese Networks

Siamese Networks using pre-trained word2vec (vocab_size=400k)

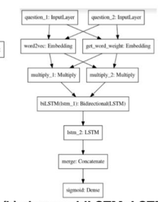
- Network (a): a generalization of tf-idf weighted word vector.
- Network (b) : replace the vector_reduction(summation) layer with biLSTM+ LSTM
  - Regularization: early stopping with patience=10 and split_ratio=0.1
  - Overfitting is mitigated (training perf. is close to testing perf.)

Model	auc_train	auc_test	accuracy_train	accuracy_test
word2vec_v1	0.7143	0.7139	0.6668	0.6656
word2vec_v2	0.7236	0.7229	0.6718	0.6714
word2vec_v3	0.7287	0.7263	0.6790	0.6762
siamese_v1	0.8286	0.7982	0.7632	0.7470
siamese LSTM	0.8578	0.8235	0.7906	0.7669
siamese LSTM(2)	0.8642	0.8252	0.7957	0.7687
siamese biLSTM+LSTM	0.8630	0.8261	0.7969	0.7693

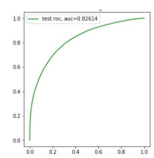
Table 1: Experiment Results



(a) siamese_v1



(b) siamese biLSTM+LSTM



Siamese biLSTM+LSTM

Further Work

LSTMs are hard to regularize, try Merity's AWD-LSTM/QRNN, IF LSTMs are production worthy for an api call

Different dataset, directed intent from Twitter thread postings

LSTMs don't scale, Transformer architecture scales to multiple GPUs but slow inference times, Bert