Phrase Based Sentiment Classification Using Tree-LSTMs: Does Attention Help?

Anhad Mohananey
Daniel Rivera

What is sentiment analysis?

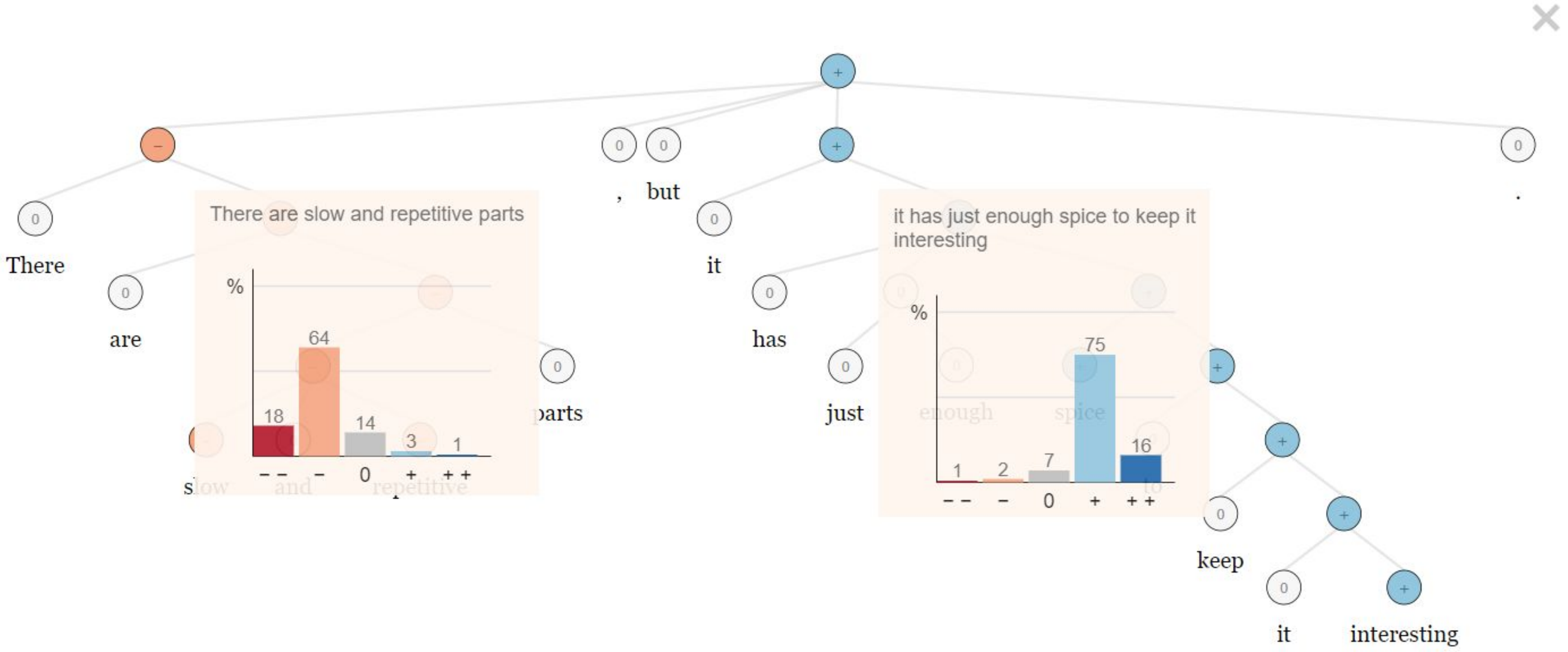
According the Stanford Sentiment Dataset, two types:

1) Binary: Good vs Bad (2 class problem)
2) Fine grained: 5 class problem

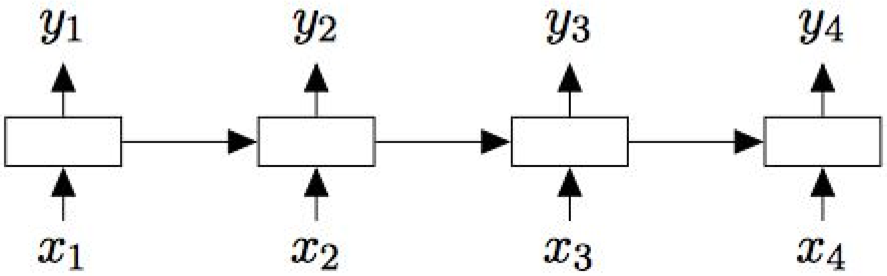
    (Very Negative, Negative, Neutral, Positive, Very Positive)

For the purposes of this project, we worked on **Fine Grained Classification**.

Phrase Sentiment Classification



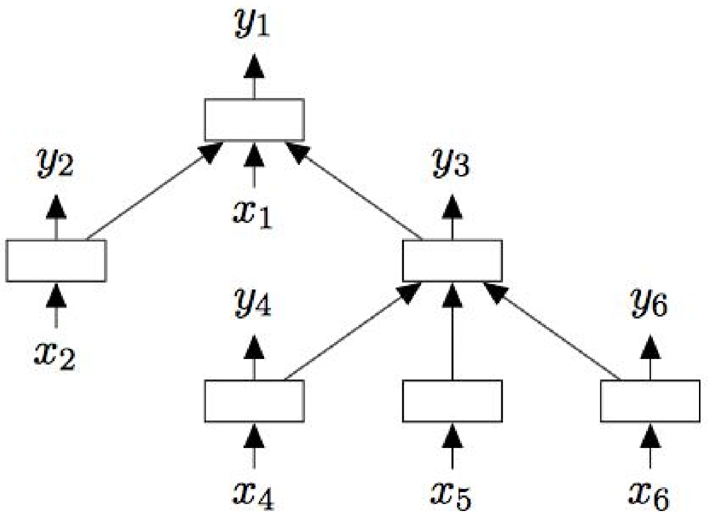
Sequential LSTM



1. There is no notion of phrases (cannot do phrase level predictions).

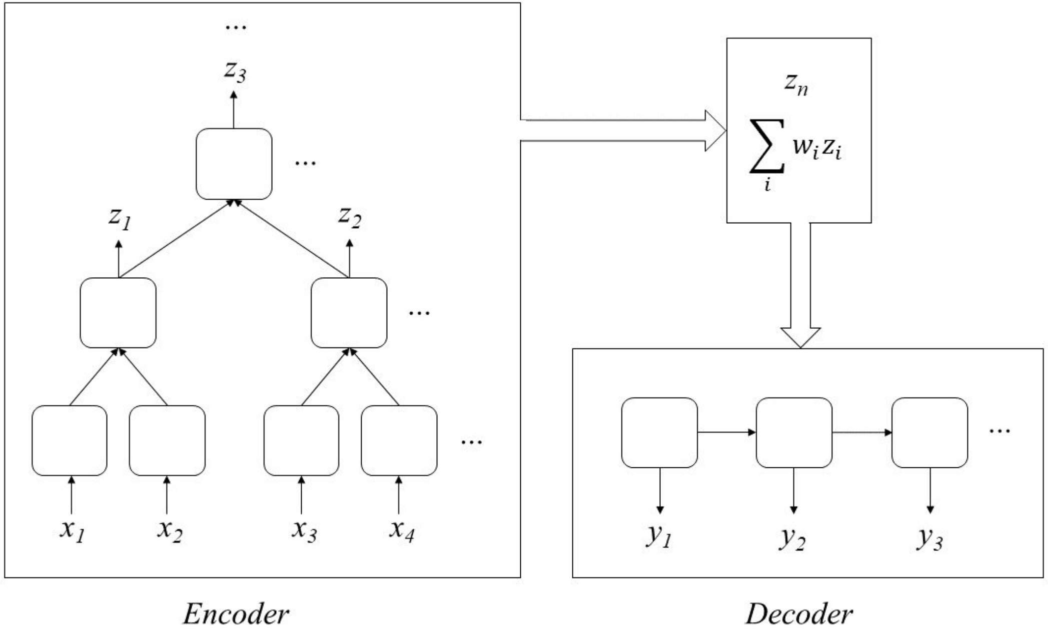
2. Doesn’t capture recursive nature of language (linguistic hypothesis is that trees are better.)

Tree-LSTM (*Tai and Socher 2015*)



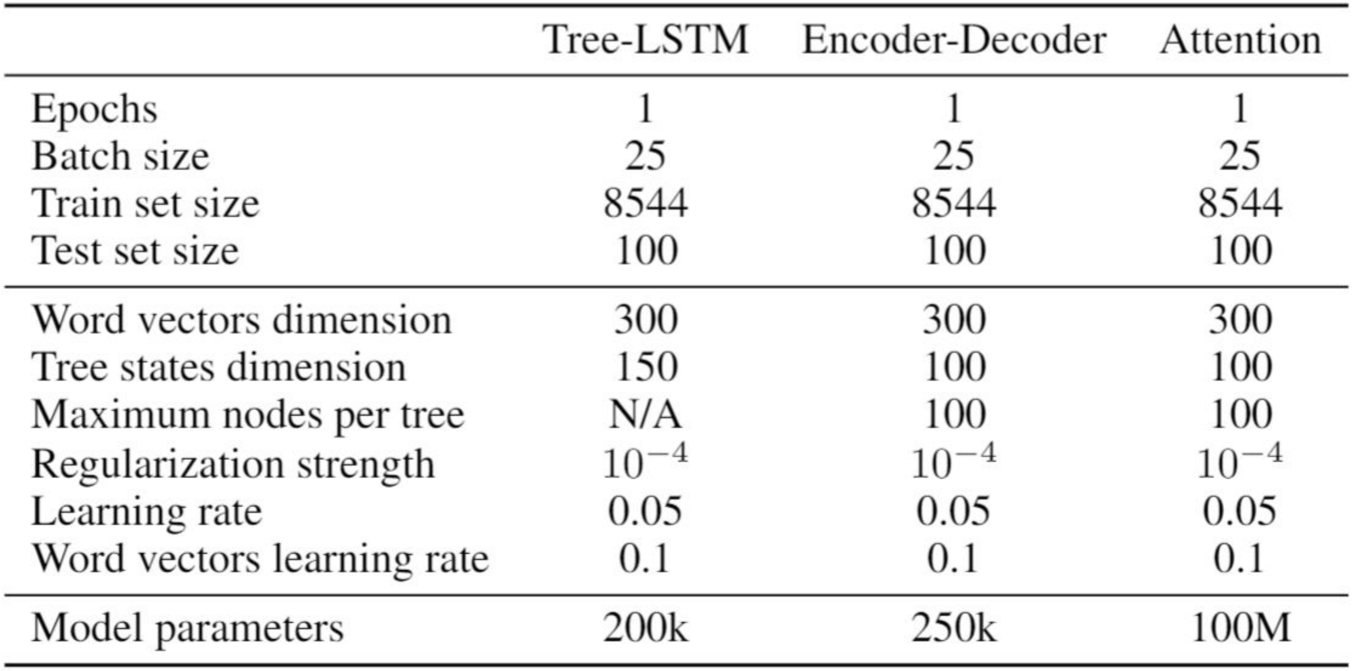
1. Does phrase level classification!
2. Capture nature of language better.

Our hypothesis: Attention?



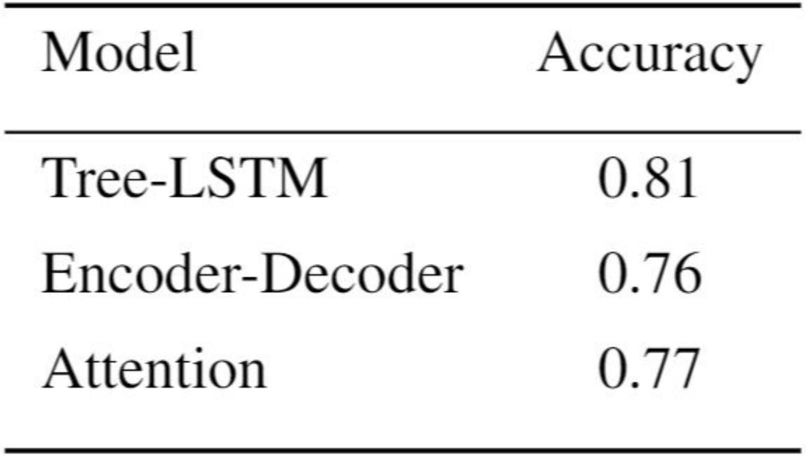
There might be dependencies between phrases that are not in the same root to leaf path!

Experiments: Configuration and Parameters



	Tree-LSTM	Encoder-Decoder	Attention
Epochs	1	1	1
Batch size	25	25	25
Train set size	8544	8544	8544
Test set size	100	100	100
Word vectors dimension	300	300	300
Tree states dimension	150	100	100
Maximum nodes per tree	N/A	100	100
Regularization strength	10^{-4}	10^{-4}	10^{-4}
Learning rate	0.05	0.05	0.05
Word vectors learning rate	0.1	0.1	0.1
Model parameters	200k	250k	100M

Results



Model	Accuracy
Tree-LSTM	0.81
Encoder-Decoder	0.76
Attention	0.77

Conclusions

TreeLSTM > Attention > Encoder Decoder

1. Attention takes more time to train as the number of parameters goes up by a significant margin. With only 1 epoch, the results may not be comprehensive.
2. Attention does better than Encoder Decoder. Obvious reason: sentences of variable length.
3. Attention may actually not work. That is, there may be no dependence between phrases not on same root to leaf path.

Eg: ((Marie had) (a good time))

Future work

1. Improve attention model by fine tuning and training for more epochs. (It takes a lot of time to train!)

2. Instead of using Stanford Tree parses, using modern latent tree learning methods such as SPINN and Chart Parsing (as Prof. Sam Bowman talked about) to learn tree structures on the sentiment supervised objective.

Our contributions

1. Evaluation code for the Tree-LSTM in the original paper for phrase level classification.

2. Lua code written using the Torch framework for implementation of the encoder-decoder and Attention-based models.

3. Evaluation code for the encoder-decoder and Attention-based models.

Thanks!

1. Tai and Socher github codebase (https://github.com/stanfordnlp/treelstm)

2. Torch developers.