DeepDowNN: Automatic Crossword Clue Generation

Angad Rekhi, Ting Chia Chang, Charlotte Kirk
{arekhi, tcchang3, ckirk}@stanford.edu https://github.com/arekhi/DeepDowNN2

Motivation

Crosswords are a very popular type of puzzle. Programs now exist that can create the word grids for crosswords and choose clues to match the words in the grids from a database of previously used clues. However, generating novel clues for words takes time and skill and is still typically done by humans.

We propose the use of *Recurrent Neural Networks* to generate novel clues for crossword words

Data: Word/Definition/Clue

Unfiltered word/clue data: 742,149 pairs from the New York Times crossword (01/1994–03/2018)

Unfiltered word/definition data: 98,855 pairs from the GCIDE open-source dictionary (first non-obsolete definition)

Filtering methods: (1) remove punctuation, (2) convert words to lowercase, (3) only keep words that are present in GloVe vocabulary, (4) limit the number of clues per word to 5*.

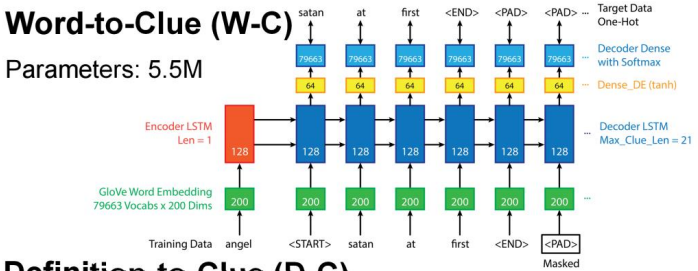
156,007 word/clue pairs (*576,344)
47,801 definition/clue pairs (*211,038)
Reduced GloVe vocab size from 400k to **~80k**

Word	Definition	Clue(s)
overripe	matured to excess	somewhat spoiled, like a mushy banana say
spate	a river of flood an overflow or inundation	sudden flood

Encoder-Decoder RNN Models

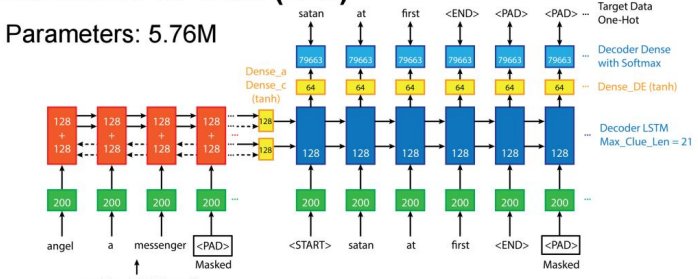
Word-to-Clue (W-C)

Parameters: 5.5M



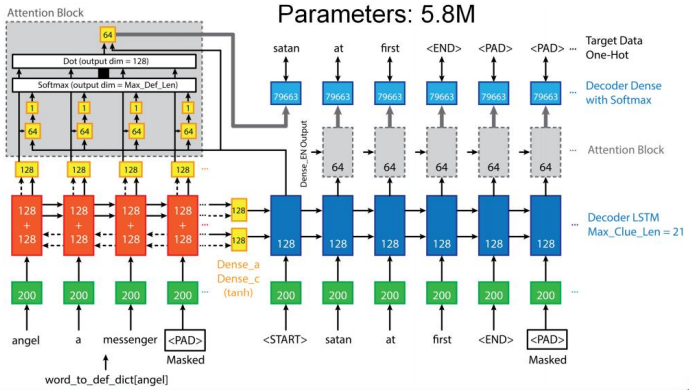
Definition-to-Clue (D-C)

Parameters: 5.76M



Definition-to-Clue with Attention (D-CA)

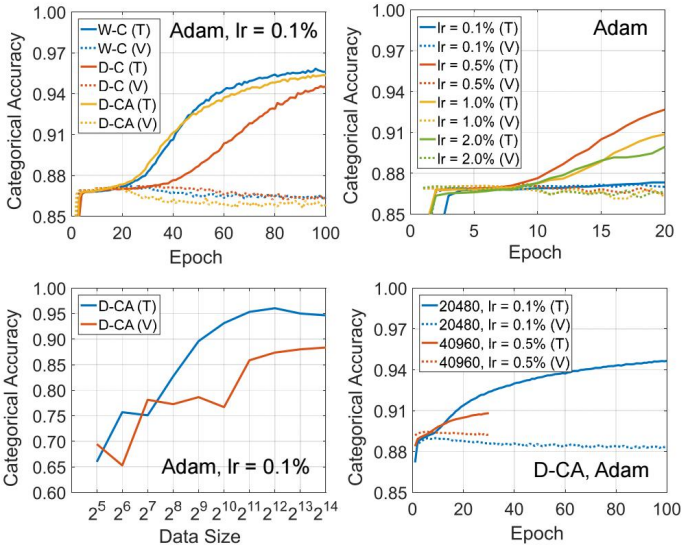
Parameters: 5.8M



Results

To quickly get a simple working model from which to proceed, we first trained the three models shown on a small amount of data (2560 W/D/C triples with an 80/20 split).

All models achieve *low bias* when trained long enough, but have *poor variance*. Our models already included some degree of regularization (e.g. dropout layers), so we looked to scale up the amount of data. We first tuned the learning rate at small scale to ensure efficient use of resources when scaling up. We also investigated how the accuracies scaled with data size.



We then ran two larger-scale experiments with our D-CA network: using 20480 triples (100 epochs) and 40960 triples (30 epochs), both with an 80/20 split. The network trained on more data achieves a smaller variance, at the expense of bias. We used this network to generate clues by sampling from the softmax-generated distribution at each time step.

Word	Generated Clues	Top $\hat{y}_1$ Values
overripe*	turning like a mushy banana, like spoiled, not bright	45%, 21%
spate	giving indian, napoleons method, head on an egg, serious offering	2.8%, 2.7%
reagent	dirty physics, kind of food, base acting, kitchen gizmo, jock	8.3%, 4.7%

*in training set

Discussion & Future Work

Our networks can combine passed clues for seen words, but generate poor clues for unseen words.

- Definitions and clues often do not refer to the same meaning of the word; word embedding may not have enough (or the right) information to overcome this challenge
- Multiple different clues for the same word/definition pair might be confusing the networks
- Accuracy calculation probably includes <PAD> tokens (reduced dynamic range for accuracy metric)
- Using a newer dictionary with definitions that correspond to clues might help
- Implement <UNK> with pointing [3] to increase data size

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," In Intl. Conf. on Learning Representations, 2015.
[2] J. Pennington, R. Socher, and C. D. Manning," GloVe: Global vectors for word representation," Proc. EMNLP 2014.
[3] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," In Proc. 55th Annual Meeting ACL, pp. 1073–1083, July 2017.