Gibberfish: Modeling Language by Detecting Nonsense

Bo Peng and Matthew Mistele

bpeng@stanford.edu, mmistele@cs.stanford.edu

CS 230: Deep Learning, Stanford University

Abstract

Data is the limiting reactant of many NLP projects. Data augmentation faces a significant obstacle: perturbations do not always lead to examples that are syntactically valid and that have the same semantic meaning as the original input, and manual filtering of examples doesn't scale.

We built *Gibberfish*: RNNs to predict whether a sequence of words is a valid, sensical English sentence. Our 2-layer word LSTM is 96% accurate at distinguishing valid sentences from sequences of words randomly sampled from the corpus. We analyzed its hidden state activations in search of learned structure representations and have preliminary visualization results.

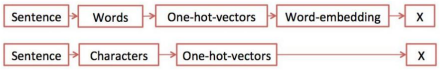
Data

Our dataset consists of 50,000 short English sentences from Tatoeba.org (lowercased) and 50,000 "fake sentences" generated by sampling words from the real sentences at random, labeled with 1 and 0 respectively. Examples:

```
the party ended and everyone went home, 1
do the soul market you please now, 0
```

Features

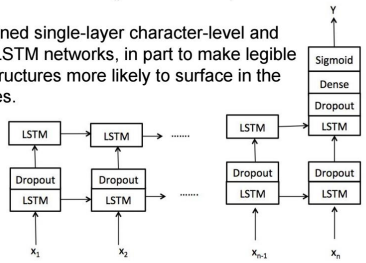
Our character-level model uses one-hot encodings of characters, and our word-level models use pretrained 50-dimensional GloVe word embeddings.



Models

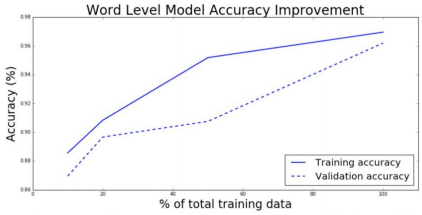
Our primary model is a two-layer, many-to-one word-level LSTM network (pictured below).

We also trained single-layer character-level and word-level LSTM networks, in part to make legible language structures more likely to surface in the hidden states.



Results

Model	Training error (m = 99,080)	Test error (m = 1,841)
Character LSTM	10.3%	8.0%
One-Layer Word LSTM	3.1%	5.54%
Two-Layer Word LSTM	0.65%	3.85%



Discussion

As expected, the word model performed better than the character model, and two layers did better than one. The first was likely thanks to the word embeddings and shorter sequence lengths to "remember" across, and the second two layers flagging or remembering nonsensicality better. Its test error was roughly comparable to human-level performance, and more data would likely help close the gap.

Identifying representations in the hidden states proved to be hard, but we got nice visualizations of how the word LSTM prediction output for a sentence or phrase changes as words are read. (Green = likely valid; red = likely nonsense)

```
you always talk back to me a we your
```

Future Work

1. Do transfer learning on sentence perturbations labeled as syntactically valid or invalid.
2. Use strategically chosen inputs to further train the model and test hypotheses about structure representations in the learned model.

References

Tatoeba.org sentences: http://downloads.tatoeba.org/exports/sentences.tar.bz2 (CC-BY)

GloVe vectors: https://nlp.stanford.edu/projects/glove/

Karpathy, Andrej: The Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/