# Language Modeling and Text Generation

Samuel Remedios, Madison Karrh, Michael Schmidt, Brent Perez, John Westbrooks

# Does adding grammatical context improve text generation with RNNs?

# What is an RNN?

- Neural network
- Sequences
  - Classification
- Uses:
  - Prediction
  - Text generation
  - Image Captioning

# How has text generation been done in the past?



- Markov Chains
  - Present state → future state

| Corpus | I am a _____ |
| --- | --- |
| I am a cat. | Cat: 100% |
| I am a cat.<br>I am a dog. | Cat: 50%<br>Dog: 50% |

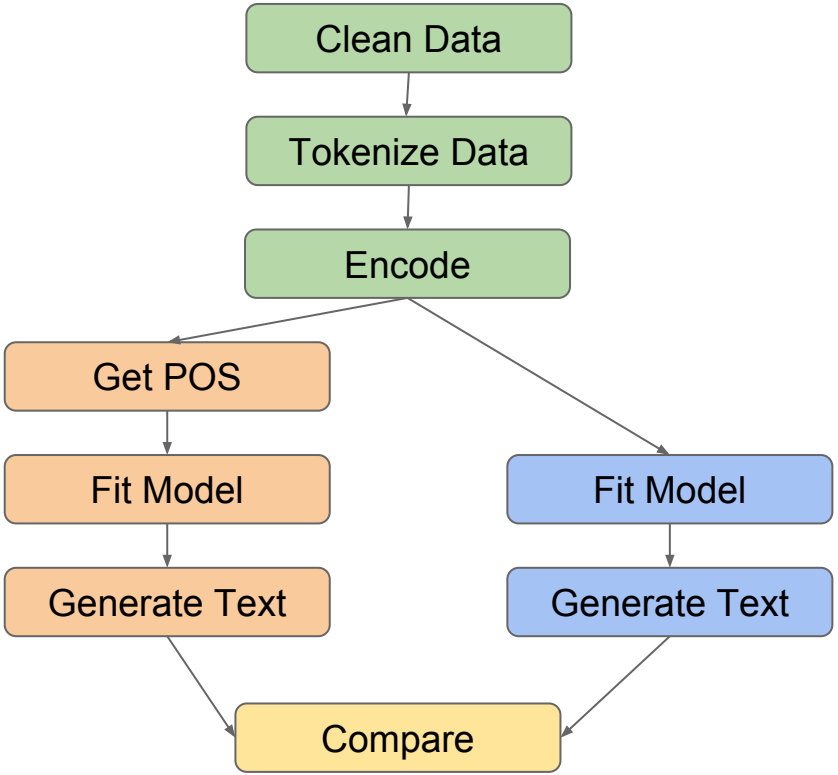
- RNN
  - LSTM

# NLP Basics

- What is NLP?
  - Python's NLTK
- What is "tokenizing"?
  - In short Word + P.O.S
- POS Tagging and grammatical structure.
  - 36 tags in Penn Treebank
    - Verb, present tense, 3rd person singular
  - 9 tags for symbols
  - 2 start/end tokens.

# Experimental Design: Grammar -> Better Results?

Intro
Strategy
Contributions
Demo



# Contributions

# Michael

Intro
Strategy
Contributions
Demo

Data Preparation:

Preprocess data

- Intelligently extract data into sentence
  - Prefix, Suffix, Number, and Abbreviations
  - **Example:** Dr. Schmidt Jr.'s regex must ignore 22.3 percent of all encountered dots (i.e. dots like these) in order to separate sentences correctly.
  - Formatting.
- Tokenizing data
  - <start> and <end> symbols (ST EN)
  - Word w/ Part of Speech

- README.md

[Open to Morty’s room]

Rick: (stumbles in drunkenly, and turns on the lights) Morty! You gotta come on. Jus’... you gotta come with me.

Morty: (rubs his eyes) What, Rick? What’s going on?

RIck: I got a surprise for you, Morty.

Morty: Its the middle of the night. What are you talking about?

ST [open to morty’s room] EN

ST rick: (stumbles in drunkenly, and turns on the lights) morty! EN

ST you gotta come on. EN

ST jus’... EN

ST you gotta come with me. EN

ST morty: (rubs his eyes) what, rick? EN

ST what’s going on? EN

ST rick: i got a surprise for you, morty. EN

ST morty: it’s the middle of the night. EN

ST what are you talking about? EN

# Madison

- Word2vec vs One hot
- Training hyperparameters
- Distance Metrics
- Markov chain

---

**E-RNN:**
I’m this attended put aren’t ST EN

**W-RNN:**
We’re ST ST you’re your principal ST ST ST ST you’re ST a a ST ST ST ST your ST ST ST EN

**Markov Chain:**
ST please, crying at an obituary for grandpa, geez, morty EN

# Brent

Intro
Strategy
Contributions
Demo

Simple RNN vs GRU vs LSTM

- Pros and cons of each
  - RNN
    - Classic RNNs prone to vanishing gradient
  - LSTM
    - Utilizes gates for data control
    - Prevents vanishing and exploding gradient
  - GRU
    - Uses gates similar to LSTM
    - More efficient than LSTM, similar performance
    - No memory unit

# John

- Designed original network:

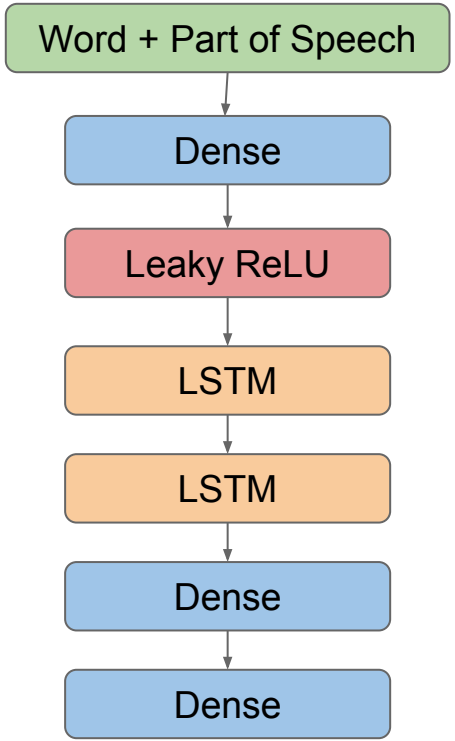
- 2 connected and stateful LSTM layers
- Dropout of 20%
- Dense output layer with softmax
- Categorical Crossentropy loss
- Rmsprop optimization

-How it changed:

- Dropped statefulness and dropout
- Added Dense input layer and Leaky ReLU before LSTM layers
- Added another Dense layer before output layer

# Network Architecture



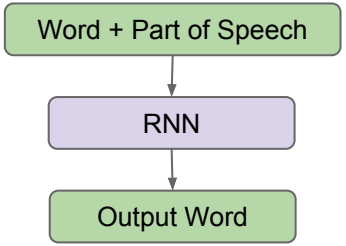


# Sam

## Determinism

RNNs are **DETERMINISTIC**

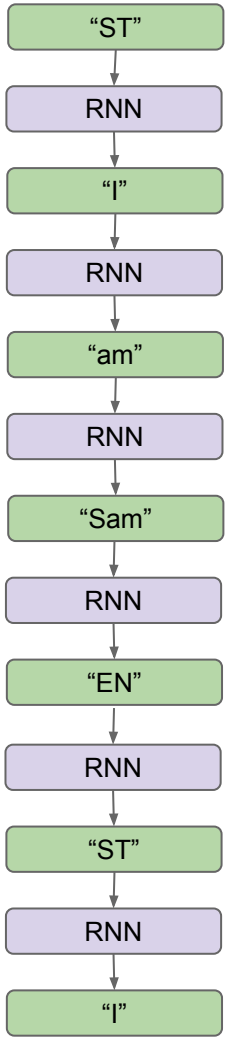
- Like a function, maps x -> y
- ie: Given some input, always the same output



# Sam

## Deterministic Example






CORPUS

ST I am Sam EN
ST Sam I am EN
ST am I Sam EN
ST am I Sam EN

OUTPUT SENTENCE

ST I am Sam EN
ST I am Sam EN
ST I am Sam EN
ST I am Sam EN

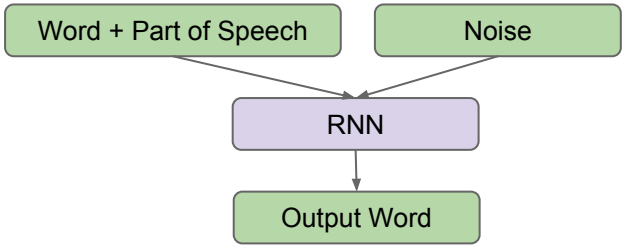
# Sam

## Stochasticity

Intro
Strategy
Contributions
Demo

Goal: Make RNN **STOCHASTIC**

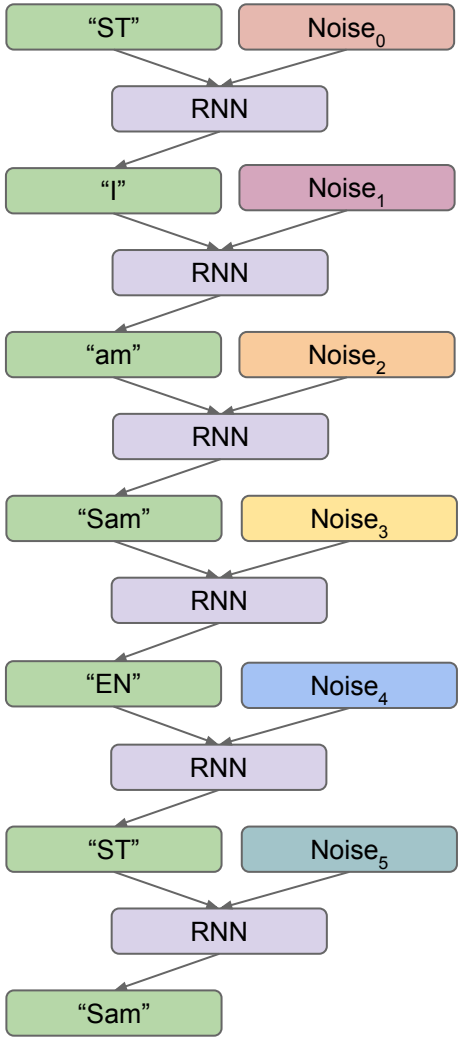
- No longer maps x -> y
- Now maps x + N -> y



# Sam

## Stochastic Example






CORPUS

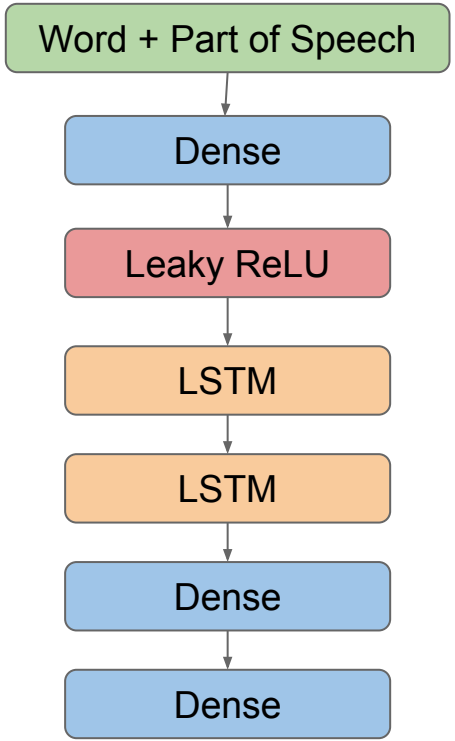
ST I am Sam EN
ST Sam I am EN
ST am I Sam EN
ST am I Sam EN

OUTPUT SENTENCE

ST I am Sam EN
ST Sam Sam Sam EN
ST am am I EN

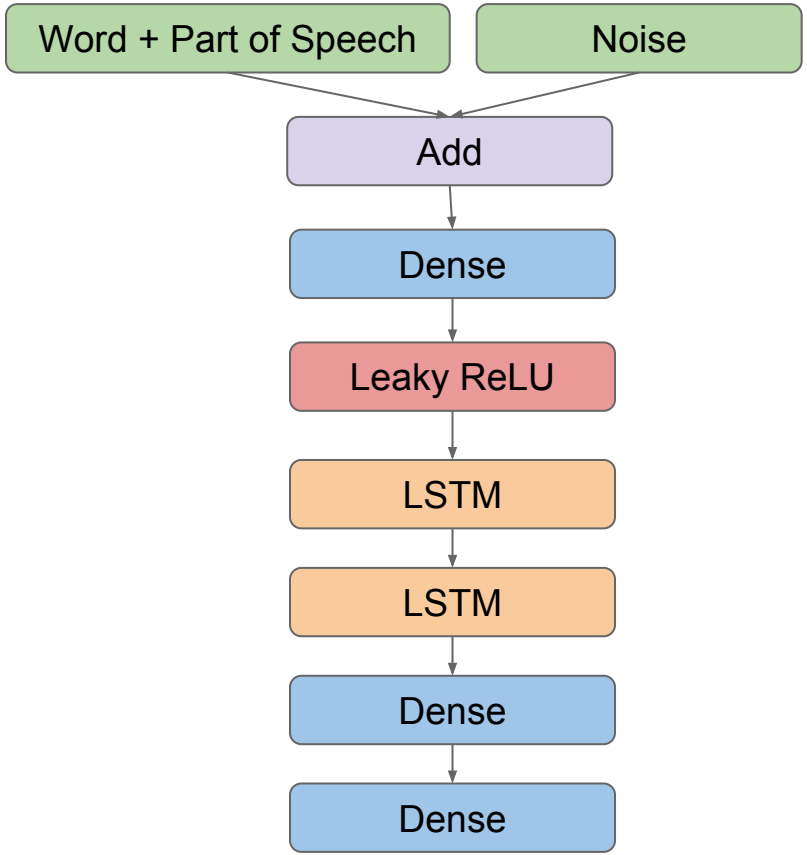
# Network Architecture (previous)





# Network Architecture (revised)





# Demo

# Environment Setup

- Python 3.5
- Install packages
- Optional: Tensorflow GPU + CUDA + cudNN

# Data Preprocessing

- Run preprocess.py
  - Downloads data
  - Runs Mike's cleaner
  - Aggregates (if multiple files)
  - Truncates clean data

```
$ python utils/preprocess.py
***** DOWNLOADING *****
***** CLEANING *****
['data\\train\\rick_and_morty.txt']
***** AGGREGATING *****
***** DONE *****
```

# Model Selection and Training

Intro
Strategy
Contributions
Demo

models/rnn.py

```
def rnn(embedding_size, single_timestep_elements, single_timestep_gt, recurrent_dropout=0, learning_rate
=1e-4, loss='mean_squared_error'):
    inputs = Input(shape=(None, single_timestep_elements))

    # add noise
    noise = Input(shape=(None, single_timestep_elements))
    x = add([inputs, noise])

    x = Dense(embedding_size//4)(x)
    x = LeakyReLU()(x)
    x = LSTM(embedding_size, return_sequences=True, recurrent_dropout=recurrent_dropout, name='a')(x)
    x = LSTM(embedding_size//2, return_sequences=True, recurrent_dropout=recurrent_dropout, name='b')(x)
    x = Dense(embedding_size//2)(x)
    x = Dense(single_timestep_gt, activation='softmax')(x)

    model = Model([inputs, noise], x)
    model.compile(loss=loss,
                  optimizer=RMSprop(lr=learning_rate),
                  metrics = ['accuracy'])

    return model
```

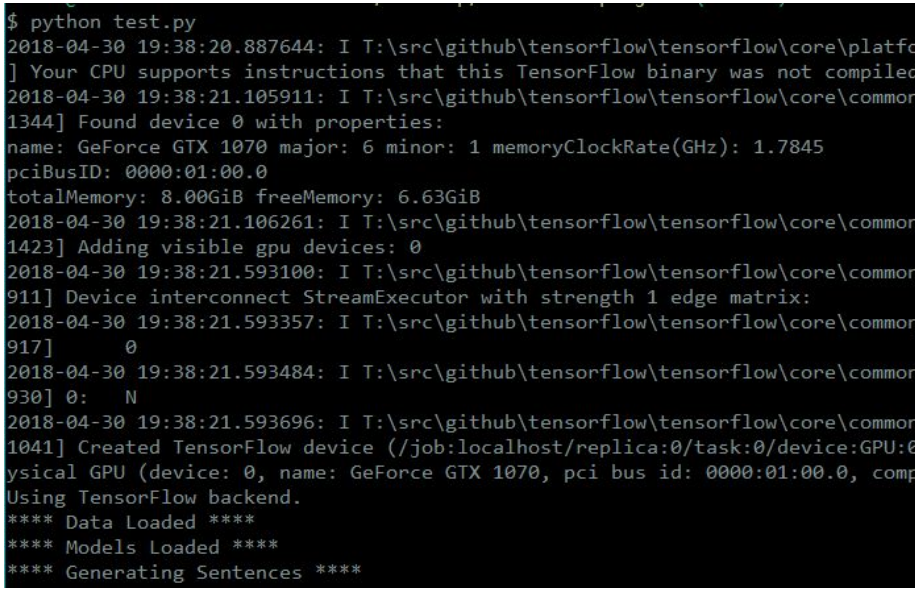
python train.py --include_pos y

```
2018-04-30 19:33:11.155668: I T:\src\github\tensorflow\tensor
1041] Created TensorFlow device (/job:localhost/replica:0/tas
ysical GPU (device: 0, name: GeForce GTX 1070, pci bus id: 00
Training:  47%|####7     | 4723/10000 [03:31<03:57, 22.26it/s
```

# Text Generation

Intro
Strategy
Contributions
Demo

python test.py --include_pos y



# Live Demo

Intro
Strategy
Contributions
Demo

# How to Compare to Other Methods (script)

Intro
Strategy
Contributions
Demo

python metrics.py --file1 myFile1.txt --file2 myFile2.txt

```
$ python metrics.py --file1 markovSentences.txt --file2 ../po
ST . EN --vs-- ST EN
['NNP', '.'] ['NNP', 'NNP']
hamming:
5
cosine:
0.8486684247915055
gotoh:
9.6
levenshtein:
3


ST this-this guy he does n't you 're young . EN --vs-- told S
['NNP', 'JJ', 'NN', 'PRP', 'VBZ', 'RB', 'PRP', 'VBP', 'JJ', '
hamming:
57
cosine:
0.545544725589981
gotoh:
-3.000000000000001
levenshtein:
48
```

# How to Compare to Other Methods (table)

| Metric | Ground Truth | MC | w-RNN | e-RNN |
|---|---|---|---|---|
| Hamming | 0 | 3.901 | 4.439 | **3.427** |
| Cosine | 0.05 | 0.035 | 0.030 | **0.0296** |
| Gotoh | 3.732 | **1.225** | 0.347 | 0.119 |
| Levenshtein | 0 | **2.532** | 3.301 | 2.684 |

# Conclusion

Intro
Strategy
Contributions
Demo

Q: Does adding grammatical context improve text generation with RNNs?

A: Qualitatively yes, but more work is necessary to empirically evaluate how much more realistic the generated text is with grammatical context.

Future work:

- Investigate deeper, larger networks
- Examine other word2vec embedding parameters or techniques
- Search for appropriate metrics to better evaluate the generated text
- Explore different stochastic seeding techniques, using more than 1 prior word in a “seq-to-seq” LSTM model

# Questions?