

# Seq2Seq in Action: Column Segmentation

## Part 1

Bill Watson

S&P Global

January 23, 2020

# What are Sequence to Sequence Models?

- ▶ Generally used to convert one set of tokens into another
  - ▶ Many - to - Many RNN
- ▶ Bleak view: map a sequence of indexes to another independent set of indexes



**Figure:** The pile gets soaked with data and starts to get mushy over time, so it's technically recurrent.

## Use Case: Extracting Tables from Images

## Primer: What are we trying to accomplish

- ▶ Tabular data is locked in PDFs
  - ▶ Either as a typeset document
  - ▶ Or as a scanned image
- ▶ Sometimes copy & paste won't suffice (or impossible with images)
- ▶ Let's take a quick look at the full pipeline to see where Seq2Seq models can fit in

# What is our data?

## CITY OF COFFEYVILLE, KANSAS

### Schedule of Receipts and Expenditures - Actual and Budget

#### Regulatory Basis:

For the Year Ended December 31, 2017

(With Comparative Actual Amounts for the Prior Year Ended December 31, 2016)

Schedule 2  
1 of 41

Receipts	Current Year		
	Year Actual	Budget	Variance Over (Under)
Taxes and Shared Receipts			
In Advance Prepayment Tax	\$ 1,779,884.36	\$ 1,873,543.39	\$ 23,660.03
Delinquent Tax	357,335.16	85,439.02	251,936.14
Motor Vehicle Tax	186,143.00	200,000.00	(13,856.99)
Recreational Vehicle Tax	1,275.20	—	1,275.20
(A) 2014 Motor Vehicle Tax	1,275.20	1,275.20	—
Interest on Delinquent Tax	50.35	51.81	(1.46)
Commercial Vehicle Tax	9,649.34	9,846.33	197.99
Recreational Vehicle Tax	507.49	520.00	(12.51)
Special Assessment	33,217.47	31,620.34	1,597.13
Delinquent Tax	600,413.27	610,000.00	(9,586.73)
Sales Tax	3,127,239.37	4,850,214.72	(1,722,975.35)
Local Alcohol Liquor Tax	14,091.23	28,974.71	14,883.48
Special Highway Tax	200,000.00	16,327.00	183,672.99
Highway User Tax	78,750.00	78,750.00	—
Highway County Aid	65,094.05	65,975.73	983.68
Landfill Tax	100,125.00	100,000.00	1,125.00
Fines, Forfeitures and Penalties	163,440.84	207,414.76	(44,973.92)
Charitable Contribution	1,000.00	1,000.00	—
Use of Money and Property	50,041.28	21,205.49	28,835.79
Interest Income	67,425.00	38,490.00	28,935.00
Sale of Equipment and Scrap	3,259.49	3,294.94	5,000.00
Other Receipts	—	—	—
Donations	1,186.00	—	1,186.00
Reimbursement Expense	44,132.89	44,000.00	132.89
Insurance Proceeds	30,750.00	—	30,750.00
Miscellaneous	8,239.70	24,700.00	6,460.30
Operating Transfers From:			
General Fund	2,327,005.74	2,485,000.76	157,995.02
Water and Sewer Utility Fund	859,324.64	777,650.10	82,674.54
Community Development Fund	1,620,632.00	—	1,620,632.00
Total Receipts	<b>\$ 13,853,737.06</b>	<b>\$ 12,349,772.30</b>	<b>\$ 1,504,965.06</b>
Expenditures			
General Government	838,060.11	782,088.99	55,971.82
Public Safety	208,200.00	213,777.00	(15,577.00)
General Services	19,780.03	12,130.62	14,249.41
Commodities	1,842.98	1,803.93	159.05
Capital Outlay	—	—	—
Total Expenditures	<b>\$ 1,075,090.06</b>	<b>\$ 987,861.50</b>	<b>\$ 98,828.56</b>
Net Expenditures	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Capital Outlays	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Current Liabilities	<b>\$ 313,977.00</b>	<b>\$ 241,910.80</b>	<b>\$ 72,066.20</b>
Less Long-Term Liabilities</td			

# Current Pipeline for Table Extraction

- ▶ Convert to an Image
- ▶ Binary Page Classification: Is there a table on this page?
- ▶ Table Segmentation: U-Net Model
- ▶ Optical Character Recognition via Tesseract
- ▶ **Column Segmentation**
- ▶ Table Alignment
- ▶ Dump to CSV

# Visual: Table Segmentation

**6. CAPITAL LEASE OBLIGATIONS**

The City has entered into a capital lease agreement in order to finance the purchase of a Fire Truck. Payments are made at \$87,750 per year for a term of 10 years at an interest rate of approximately 5.00%. Final maturity for the lease is in 2026. Future minimum lease payments are as follows:

Year Ended December 31	Total
2018	\$ 135,458.44
2019	135,458.44
2020	135,458.44
2021	135,458.44
2022	135,458.44
2023	135,458.44
2024	135,458.44
2025	135,458.44
2026	135,458.44
<b>Less Unearned Interest</b>	<b>(128,142.81)</b>
<b>Less Unearned Interest, Net Present Value of Minimum Lease Payments</b>	<b>(11,315.63)</b>
<b>Less Current Maturing Lease</b>	<b>(11,138.87)</b>
<b>Total Capital Lease Obligation</b>	<b>\$ 99,924.32</b>

The City has entered into a capital lease agreement in order to finance the purchase of a Fire Truck. Payments are made at \$87,750 per year for a term of 10 years at an interest rate of approximately 5.00%. Final maturity for the lease is in 2027. Future minimum lease payments are as follows:

Year Ended December 31	Total
2018	\$ 85,046.36
2019	85,046.36
2020	85,046.36
2021	85,046.36
2022	85,046.36
2023	85,046.36
2024	85,046.36
2025	85,046.36
2026	85,046.36
<b>Less Unearned Interest</b>	<b>(84,534.21)</b>
<b>Less Present Value of Minimum Lease Payments</b>	<b>(3,512.02)</b>
<b>Less Current Maturing Lease</b>	<b>(308,663.29)</b>
<b>Total Capital Lease Obligation</b>	<b>\$ 1,818,301.29</b>

**6. OPERATING LEASES**

As of December 31, 2017 the City has entered into an operating lease for office equipment. Rent expense for the year ended December 31, 2018 and 2019 was \$0. Under the current lease agreements, the future minimum lease payments are as follows:

Year	Total
2018	\$ 4,216.68
2019	3,160.51

Schedule 2  
1 of 41

Schedule 1

**CITY OF COFFEYPOLIS, KANSAS**  
**GENERAL FUND**  
Schedule of Receipts, Disbursements, Actual and Budget  
Regulatory Basis  
For the Year Ended December 31, 2017  
(With Comparative Actual Amounts for the Prior Year Ended December 31, 2016)

Description	Current Year		
	Prior Actual	Actual	Budget
Receipts			
Sales and Use Taxes	\$ 1,779,864.36	\$ 1,873,343.39	\$ 2,000,362.00
Delinquent Property Tax	157,235.00	83,479.02	80,000.00
Motor Vehicle Tax	140,100.00	176,000.00	200,000.00
Residential Vehicle Tax	1,373.30	1,766.98	1,200.00
1/2 CT M. Vehicle Tax	827.20	366.64	841.00
1/2 CT M. Residential Vehicle Tax	97.00	7.64	100.00
Commercial Vehicle Tax	9,600.34	9,861.33	8,157.00
Special Assessments	422,000.00	422,000.00	422,000.00
Dedicated Revenue	63,571.47	73,633.39	30,000.00
Gasoline Tax	60,440.00	60,440.00	60,440.00
Sales Tax	5,137,235.57	5,852,214.72	5,145,316.00
Food and Beverage Tax	4,650.00	4,650.00	4,650.00
Alcohol License Tax	10,893.55	38,876.71	10,351.71
Special Highway Tax	209,235.61	256,036.36	256,330.00
Highway Fuel Use Tax	29,200.00	39,200.00	30,000.00
Highway County Aid	31,879.00	45,973.73	31,143.73
General Fund	30,320.00	30,320.00	30,320.00
Fines, Penalties and Pleadings	183,460.84	207,414.75	208,443.00
Taxes on Income	10,000.00	10,000.00	10,000.00
Use of Money and Property	30,461.00	21,461.49	12,000.00
Interest Income	67,410.00	65,000.00	65,000.00
Sale of Equipment and Seigniorage	2,238.00	3,299.94	5,000.00
Total	\$ 13,061,130.00	\$ 13,871,777.06	\$ 15,342,772.00
Disbursements			
General Fund	\$ 13,061,130.00	\$ 13,871,777.06	\$ 15,342,772.00
Capital Projects	-	-	-
Capital Projects - Building	-	-	-
Capital Projects - Equipment	-	-	-
Capital Projects - Land	-	-	-
Capital Projects - Natural Resources	-	-	-
Capital Projects - Other	-	-	-
General Current Expenditures	-	-	-
General Current Expenditures - Building	-	-	-
General Current Expenditures - Equipment	-	-	-
General Current Expenditures - Land	-	-	-
General Current Expenditures - Natural Resources	-	-	-
General Current Expenditures - Other	-	-	-
General Current Expenditures - Personal Services	-	-	-
General Current Expenditures - Purchases	-	-	-
General Current Expenditures - Utilities	-	-	-
General Current Expenditures - Wages and Salaries	-	-	-
Community Development Fund	-	-	-
State Bond Fund	-	-	-
Reserve Funds	-	-	-
Operating Transfers In:	-	-	-
General Fund	3,275,261.74	8,486,000.00	5,000,000.00
Water and Sewer Utility Fund	698,334.04	777,054.33	722,773.00
Community Development Fund	6,410,432.32	-	55,279.00
Total	\$ 13,061,130.00	\$ 13,871,777.06	\$ 15,342,772.00
Expenditures			
General Government	\$ 828,080.11	\$ 782,088.99	\$ 987,382.00
General Government - Personal Services	393,200.00	349,200.00	393,200.00
General Government - Purchases	157,788.03	123,000.00	142,945.00
Commodities	1,442.98	1,853.92	16,435.00
Contribution (Grant)	-	-	(16,617.00)
Total	\$ 4,216.68	\$ 3,160.51	-

**CITY OF COFFEYPOLIS, KANSAS**  
**GENERAL FUND**  
Summary of Expenditures, Actual and Budget  
Regulatory Basis  
For the Year Ended December 31, 2017

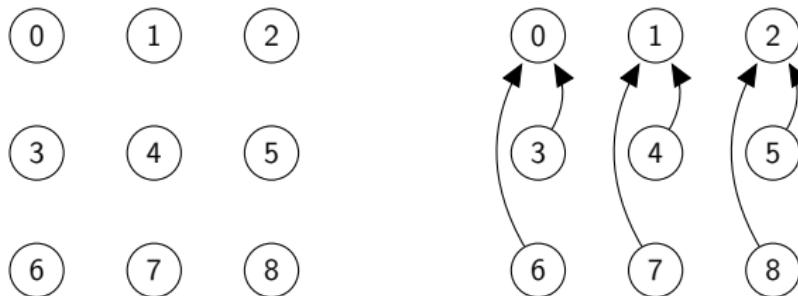
Description	Current Year		
	Prior Actual	Actual	Budget
Expenditures			
General Government	\$ 13,061,130.00	\$ 13,871,777.06	\$ 15,342,772.00
General Government - Personal Services	3,275,261.74	8,486,000.00	5,000,000.00
General Government - Purchases	698,334.04	777,054.33	722,773.00
General Government - Utilities	6,410,432.32	-	55,279.00
General Government - Wages and Salaries	-	-	-
General Current Expenditures	-	-	-
General Current Expenditures - Building	-	-	-
General Current Expenditures - Equipment	-	-	-
General Current Expenditures - Land	-	-	-
General Current Expenditures - Natural Resources	-	-	-
General Current Expenditures - Other	-	-	-
General Current Expenditures - Personal Services	-	-	-
General Current Expenditures - Purchases	-	-	-
General Current Expenditures - Utilities	-	-	-
General Current Expenditures - Wages and Salaries	-	-	-
Community Development Fund	-	-	-
State Bond Fund	-	-	-
Reserve Funds	-	-	-
Operating Transfers Out:	-	-	-
General Fund	3,275,261.74	8,486,000.00	5,000,000.00
Water and Sewer Utility Fund	698,334.04	777,054.33	722,773.00
Community Development Fund	6,410,432.32	-	55,279.00
Total	\$ 4,216.68	\$ 3,160.51	-

# Visual: Tesseract OCR Output

level	page_num	block_num	par_num	line_num	word_num	left	top	width	height	conf	text
5	1	1	1	1	1	1092	100	62	27	95	For
5	1	1	1	1	2	1165	100	55	26	95	the
5	1	1	1	1	3	1239	100	82	25	96	Year
5	1	1	1	1	4	1335	100	114	24	96	Ended
5	1	1	1	1	5	1463	100	82	24	93	June
5	1	1	1	1	7	1630	100	88	22	36	ZU18
5	1	1	1	2	1	1295	146	219	43	96	(Continued)
5	1	1	1	3	1	460	263	128	32	96	NOTE
5	1	1	1	3	2	605	263	16	31	90	1
5	1	1	1	3	3	637	280	13	5	87	-
5	1	1	1	3	4	665	260	245	34	96	SUMMARY
5	1	1	1	3	5	925	260	61	32	95	OF
5	1	1	1	3	6	1002	257	303	34	96	SIGNIFICANT
5	1	1	1	3	7	1319	254	318	34	95	ACCOUNTING
5	1	1	1	3	8	1651	252	220	34	95	POLICIES
5	1	1	1	3	9	1886	251	232	42	96	(Continued)
5	1	1	1	4	1	537	372	39	31	93	H.
5	1	1	1	4	2	612	369	206	34	96	Inventories
5	1	1	1	4	3	833	368	64	33	96	and
5	1	1	1	4	4	911	367	141	43	96	Prepaid
5	1	1	1	4	5	1066	368	98	31	96	Items
5	1	1	1	5	1	611	477	207	34	96	Inventories
5	1	1	1	5	2	831	487	54	22	96	are
5	1	1	1	5	3	896	476	123	32	96	valued
5	1	1	1	5	4	1031	480	32	27	96	at
5	1	1	1	5	5	1073	475	55	32	96	the
5	1	1	1	5	6	1141	474	101	33	96	lower
5	1	1	1	5	7	1253	473	41	33	96	of
5	1	1	1	5	8	1301	477	73	28	93	cost
5	1	1	1	5	9	1385	471	147	42	90	(first-in,
5	1	1	1	5	10	1545	470	163	41	96	first-out)

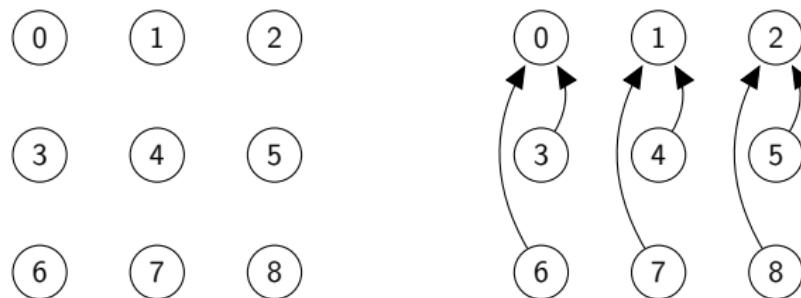
# Column Segmentation

- ▶ Naive approach is to merge words close together
  - ▶ But spacing is **variable** in tables
  - ▶ Text justification (left, center, right)
- ▶ **Union-Find Disjoint Set (UFDS) Algorithm** to group columns by **Intersection Over Union (IOU)**



# Column Segmentation

- ▶ Naive approach is to merge words close together
  - ▶ But spacing is **variable** in tables
  - ▶ Text justification (left, center, right)
- ▶ **Union-Find Disjoint Set (UFDS) Algorithm** to group columns by **Intersection Over Union (IOU)**



- ▶ **Idea:** Use a Seq2Seq to learn correct segmentation

# Visual: Alignment

-- -- -- #SumYear1-(SOSCS-#BS					
	Prior				Variance -
	Year				Over
	Actual		Actual	Budget	(Under)
<b>Receipts</b>					
Taxes and Shared Receipts					
Ad Valorem Property Tax	\$1,779,884.26		\$1,872,543.39	\$8	2,225,362 \$ -352,818.61
Delinquent Tax	107,335.16		83,439.02	60,000	23,439.02
Motor Vehicle Tax	185,104.64		198,703.03	206,631	-7,927.97
Recreational Vehicle Tax	1,372.3		1,786.98	1,521	265.98
16/20 M Vehicle Tax	827.2		566.64	841	-274.36
Vehicle Rental Excise Tax	95.1		51.82	177	-125.18
Commercial Vehicle Tax	9,660.34		9,861.33	8,137	1,724.33
Watercraft Tax	657.49		532.7	517	15.7
Special Assessments	63,572.47		73,635.34	30,000	43,635.34
Franchise Tax	604,481.17		688,423.06	476,972	211,491.06
Sales Tax	5,137,239.57		4,652,214.72	5,145,516	-293,301.28
Federal Grants	8,650.01		2,705.56	-	2,705.56
Local Alcohol Liquor Tax	18,091.55		28,878.71	16,327	12,561.71
Special Highway Tax	259,225.61		256,024.36	256,330	-305.64
Highway Connecting Links	76,750.82		76,645.19	76,700	-54.81
Highway County Aid	45,694.05		45,975.73	40,830	5,145.73
Licenses and Permits	94,108.5		92,606	-	92,606
Fines, Forfeitures and Penalties	183,460.84		207,414.75	308,445	-101,030.25
Charges for Services	99,811.84		90,317.95	96,596	-6,278.05
Use of Money and Property					
Interest Income	20,461.28		21,263.49	12,000	9,263.49
Rents	67,423		36,490.5	67,400	-30,907.5
Sale of Equipment and Scrap	2,228.4		3,294.94	5,000	-1,705.06
Other Receipts					
Donations	-		1,180	-	1,180
Reimbursed Expense	46,151.89		40,804.82	-	40,804.82
Insurance Proceeds	-		38,750	-	38,750
Miscellaneous	8,239.7		4,526.25	6,000	-1,473.75
Operating Transfers from:					
Electric Utility Fund	2,327,045.74		2,408,082.7	2,583,695	-175,612.3
Water and Sewer Utility Fund	694,334.64		777,054.1	723,775	53,279.1
Community Development Fund	218,655.33		-	-	-
Total Receipts	12,060,562.9		11,913,775.08	\$8	12,348,772 \$5 -434,996.92
Expenditures			TT		
General Government					
Personal Services	838,080.11		782,088.99	\$	987,382 \$8 -205,293.01
Contractual Services	259,202.84		269,145.94	313,797	-44,651.06
Commodities	10,788.03		12,130.62	14,340	-2,169.58
Capital Outlay	1,442.98		1,832.92	18,450	-16,617.08

# Problem Statement: Column Segmentation

Year Ended December 31	Totals	Year Ended December 31	Totals
2018	\$ 155,458.46	2018	\$155,458.46
2019	155,458.46	2019	155,458.46
2020	155,458.46	2020	155,458.46
2021	155,458.46	2021	155,458.46
2022	155,458.46	2022	155,458.46
2023-2026	<u>505,870.61</u>	2023-2026	<u>505,870.61</u>
	1,283,162.91		1,283,162.91
Less imputed interest	(174,862.64)		
Net Present Value of Minimum Lease Payments	1,108,300.27	Less imputed interest	-174,862.64
Less: Current Maturities	<u>(118,236.87)</u>	Net Present Value of Minimum Lease Payments	
Long-Term Capital Lease Obligations	<u>\$ 990,063.40</u>	Less: Current Maturities	-118,236.87
		Long-Term Capital Lease Obligations	\$990,063.40

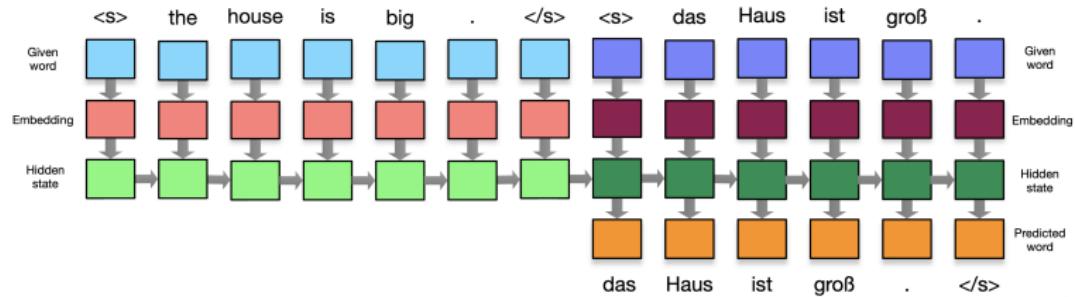
- Given a sequence of text, can we predict where to insert columns?

# Revisiting Column Segmentation as a Seq2Seq Problem

- ▶ Translation Task
- ▶ Given a sequence of tokens
- ▶ Translate into the same set of tokens, but with a special **<SEG>** token

Less imputed interest ( 174,862.64 )  
↓  
Less imputed interest <SEG> ( 174,862.64 )

# Seq2Seq Architecture



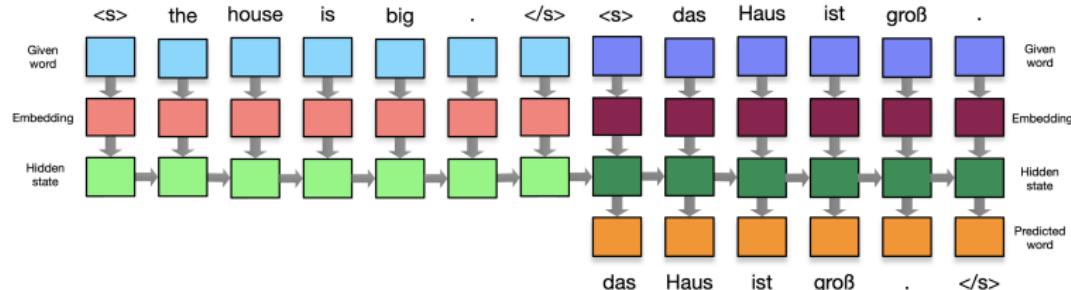
- ▶ We can use a **Encoder-Decoder** style model!

# Encoder-Decoder Architecture

## Overview: Encoders and Decoders

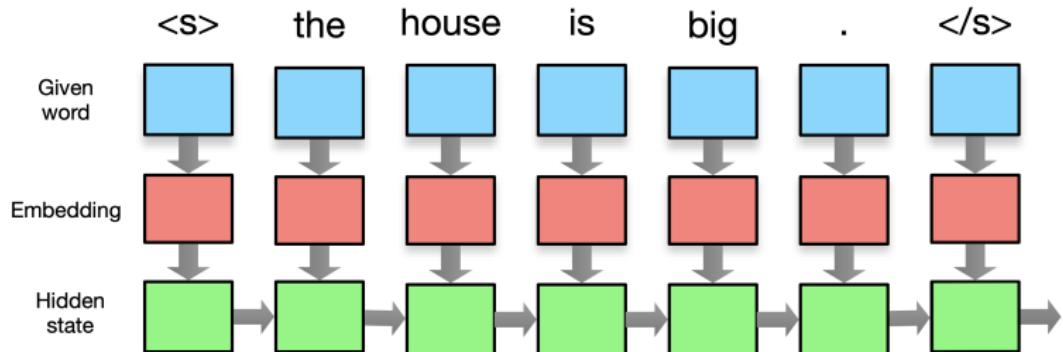
- ▶ We have 2 sub-networks: an **Encoder** and a **Decoder**
- ▶ Encoders
  - ▶ Give the source sentence meaning
- ▶ Decoders
  - ▶ Emit a variable-length sequence
- ▶ We will discuss how to connect the two for joint training

# Overview: Encoders and Decoders



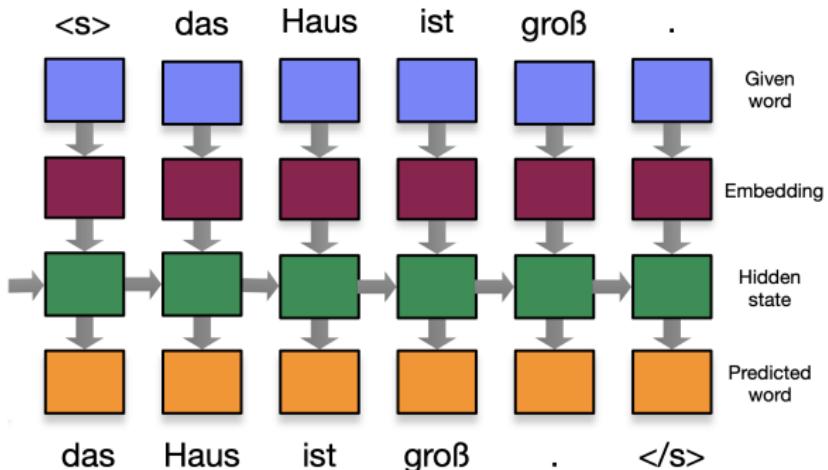
- ▶ Encapsulation allows for flexible design choices
- ▶ **Embeddings**
  - ▶ Pre-trained
  - ▶ DIY
- ▶ **Recurrent Layer**
  - ▶ Type
  - ▶ Depth
  - ▶ Directionality

# What makes an Encoder?



- ▶ Recall: **Encoders** give the source sentence meaning
- ▶ Effectively a language model, without a layer to predict the next word
- ▶ Idea is to pass on the hidden state, and possibly use the encodings directly

# What makes a Decoder?



- ▶ Recall: **Decoders** provide a new sequence conditioned on the Encoder's hidden state
- ▶ Starts with the Encoder's hidden state, and predicts one token at a time
- ▶ Re-feed the predicted token back into the decoder

## Word Embeddings: Practical Considerations

## Column Segmentation Issue: Vocabulary Size

- ▶ Is the text actually important?
- ▶ Over 10K **unique tokens**, mostly numbers

## Column Segmentation Issue: Vocabulary Size

- ▶ Is the text actually important?
- ▶ Over **10K unique tokens**, mostly numbers
- ▶ **Solutions:**
  - ▶ Token → Part-of-Speech
  - ▶ Numeric Value → **(NUM)**
- ▶ Reduces vocabulary size to **21 tokens**
- ▶ But lets take a look at alternative solutions to managing vocab size

## Recap: Word Embeddings

$$\begin{array}{l} \text{the} \rightarrow \begin{bmatrix} 1.23 & -0.58 & 0.22 & -0.80 & 0.61 \end{bmatrix} \\ \text{cat} \rightarrow \begin{bmatrix} -1.10 & 1.23 & 0.17 & -0.21 & 1.43 \end{bmatrix} \\ \text{is} \rightarrow \begin{bmatrix} -0.26 & 0.70 & 0.27 & 0.59 & -1.04 \end{bmatrix} \\ \text{black} \rightarrow \begin{bmatrix} -1.13 & -0.81 & -0.53 & -0.59 & 0.26 \end{bmatrix} \end{array}$$

- ▶ A trick to map tokens to vector representations

# Recap: Pretrained Word Embeddings

- ▶ Co-occurrence Matrix
- ▶ Pointwise Mutual Information
- ▶ SVD Co-occurrence
- ▶ Ngram
- ▶ CBOW, Skip Gram
- ▶ GloVe
- ▶ ELMo
- ▶ BERT

# DIY Embeddings

- ▶ We can always train our own!

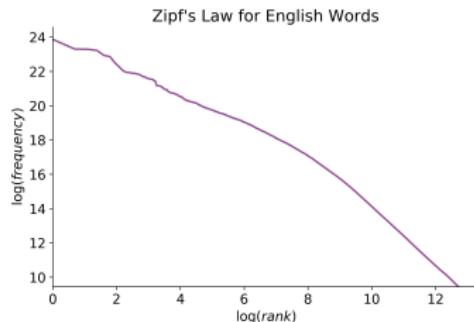
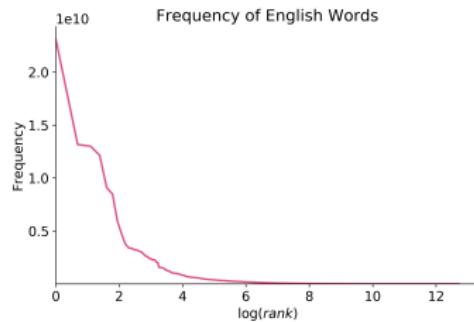
# Special Tokens for Sequence Modeling

- ▶  $\langle \text{PAD} \rangle$  → Padding / Masking
- ▶  $\langle \text{UNK} \rangle$  → Unknown words
- ▶  $\langle \text{SOS} \rangle$  → Start of Sentence
- ▶  $\langle \text{EOS} \rangle$  → End of Sentence

```
<SOS> Data Science is the <UNK> ! <EOS>
<SOS> I <UNK> for S&P . <EOS> <PAD>
<SOS> Hello World ! <EOS> <PAD> <PAD> <PAD>
```

# Managing the Vocab Size

- ▶ Languages are unevenly distributed
- ▶ Many rare words, names → inflates the size of the vocabulary
- ▶ **Problem:**
  - ▶ Large embedding matrices for source, target language
  - ▶ Large output layers for prediction and softmax
- ▶ Naive Solution: Limit the vocab size to most frequent



# Morphology, Compounding, and Transliteration

- ▶ Morphological Analysis

*tweet, tweets, tweeted, tweeting, retweet, ...*

- ▶ Compound Splitting

- ▶ `homework` → `home·work`
- ▶ `website` → `web·site`

- ▶ Names, Places, Proper Nouns

- ▶ `Hoboken, Baltimore, Obama, Michelle`
- ▶ Can do Transliteration

## Handling Numbers

- ▶ Do we really need to encode every number? **NO!**

I pay **950.00** in May **2007** > I pay **2007** in May **950.00**

# Handling Numbers

- ▶ Do we really need to encode every number? **NO!**

I pay **950.00** in May **2007** > I pay **2007** in May **950.00**

- ▶ **Solution 1:** Replace with a **<NUM>** token, but

I pay **<NUM>** in May **<NUM>** = I pay **<NUM>** in May **<NUM>**

# Handling Numbers

- ▶ Do we really need to encode every number? **NO!**

I pay 950.00 in May 2007 > I pay 2007 in May 950.00

- ▶ **Solution 1:** Replace with a  $\langle \text{NUM} \rangle$  token, but

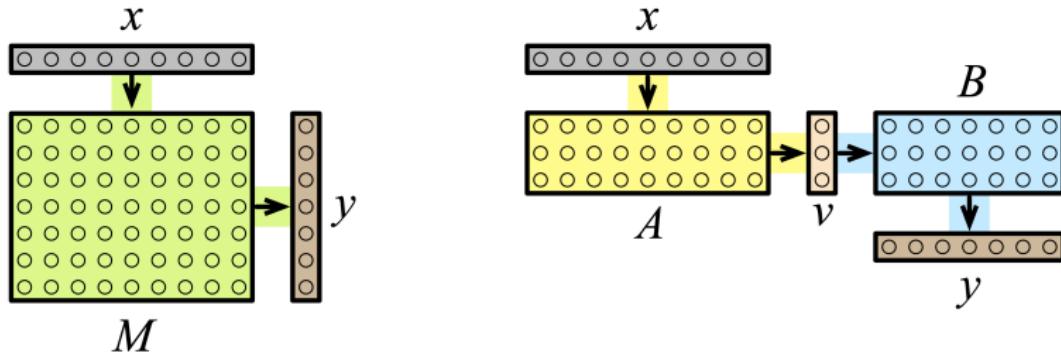
I pay  $\langle \text{NUM} \rangle$  in May  $\langle \text{NUM} \rangle$  = I pay  $\langle \text{NUM} \rangle$  in May  $\langle \text{NUM} \rangle$

- ▶ **Solution 2:** Replace each digit with a unique symbol, e.g. 5

I pay 555.55 in May 5555 > I pay 5555 in May 555.55

- ▶ This reduces the need for embeddings, when we can simply do transliteration

# Factored Decomposition



- ▶ **Problem:** Large input and output vectors
  - ▶  $|x| = 20,000, |y| = 50,000 \rightarrow |M| = 1,000,000,000$
- ▶ **Solution:** Use a bottleneck with smaller matrices  $A, B$ 
  - ▶  $|v| = 100 \rightarrow |A| = 2,000,000, |B| = 5,000,000$
  - ▶ Total Parameters: 7,000,000

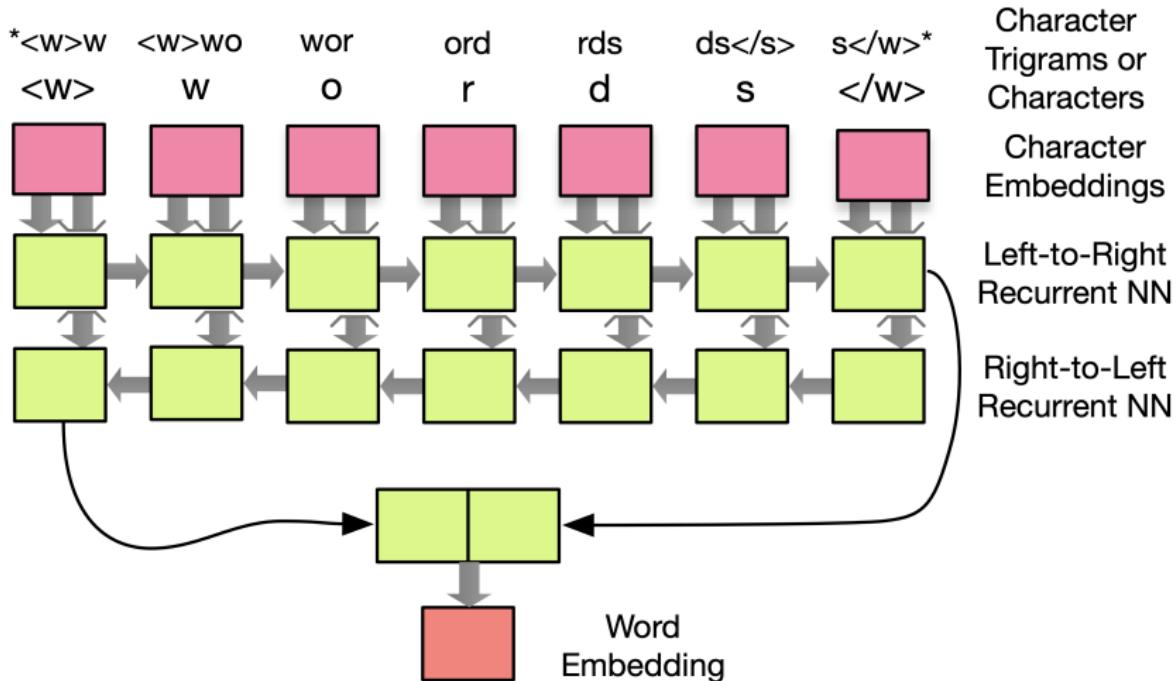
# Character-Based Models

- ▶ Instead use embeddings for character string  
`b e a u t i f u l`
- ▶ Idea is to induce embeddings for unseen morphological variants:  
`beautiful`
- ▶ Tokens are single characters, symbols, whitespace

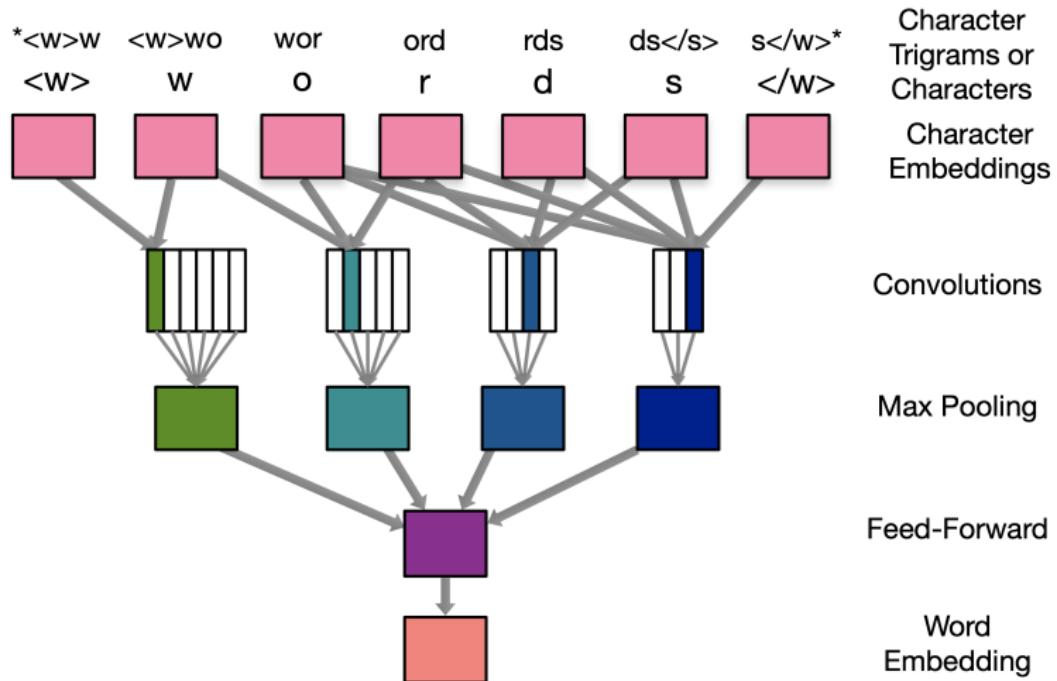
# Character-Based Models

- ▶ Instead use embeddings for character string  
`b e a u t i f u l`
- ▶ Idea is to induce embeddings for unseen morphological variants:  
`beautiful`
- ▶ Tokens are single characters, symbols, whitespace
- ▶ Generally poor performance

# Character-Based Models



# Character-Based Models



# BPE Subwords

```
the · f a t · c a t · i s · i n · t h e · t h i n · b a g  
t h → th     the · f a t · c a t · i s · i n · t h e · t h i n · b a g  
a t → at     the · f a t · c a t · i s · i n · t h e · t h i n · b a g  
i n → in     the · f a t · c a t · i s · i n · t h e · t h i n · b a g  
t h e → the   the · f a t · c a t · i s · i n · t h e · t h i n · b a g
```

- ▶ Breaks words into subwords
  - ▶ Starts with the character set
  - ▶ Merges the most frequent pairs, one per iteration
- ▶ Unsupervised (accidental) morphology (frequency suffixes)

# BPE Tokenization on HHGTTG

- ▶ Unique Characters: 101
- ▶ Unique BPE Tokens (1K iterations): 1,093
- ▶ Unique BPE Tokens (5K iterations): 4,817
- ▶ Unique Tokens: 9,541

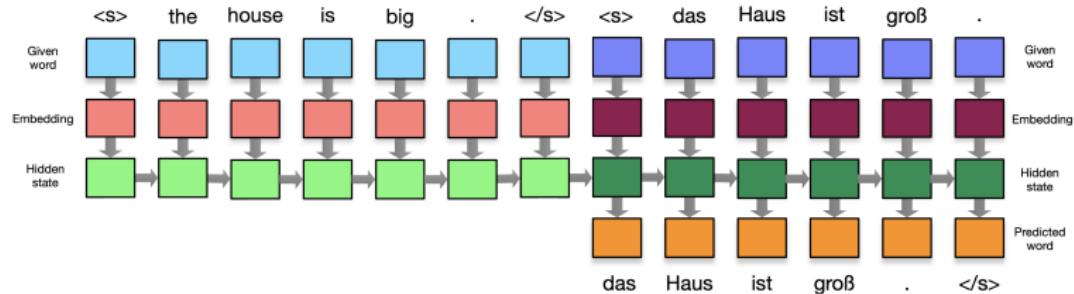
there is a theory which states that if ever anyone  
discovered other universes exist actually what the universe is for  
and why it is here , it will instantly die soon after  
and be replaced by something even more bizarre  
and interesting like conceivable .

there is another theory which states that  
this has already happened .

in the beginning the universe was created .  
this has made a lot of people very angry and  
been widely regarded as a bad move .

# Column Segmentation Architecture

# Seq2Seq Architecture

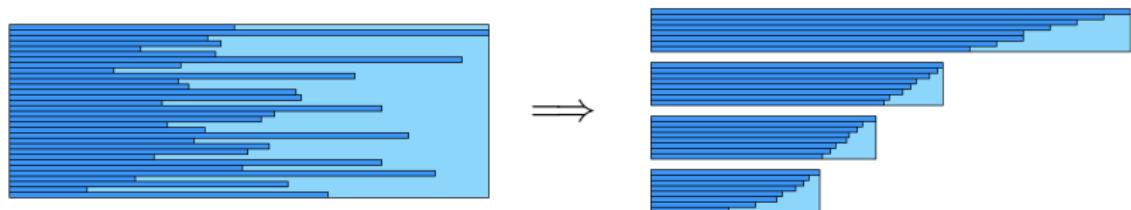


```
Seq2Seq(  
    encoder: Encoder(  
        embedding: Embedding(21, 256)  
        (gru): GRU(256, 128, num_layers=2, batch_first=True)  
    )  
    decoder: Decoder(  
        embedding: Embedding(22, 256)  
        (gru): GRU(256, 128, num_layers=2, batch_first=True)  
        (fc): Linear(in_features=128, out_features=22, bias=True)  
    )  
)
```

The model has 508,438 trainable parameters

## Training Considerations

# Increasing Throughput through Batching



- ▶ We can pad sentences of different lengths to increase batch size
- ▶ While also minimizing the use of padding
- ▶ Matrix Operations are faster
- ▶ **Warning!** Make sure padding is not influencing the hidden state

## Teacher Forcing

- ▶ Instead of refeeding the predicted token, replace it with the true token randomly
- ▶ This is only done during training, not decoding

$$y_{i+1} = \begin{cases} \operatorname{argmax}_j \theta_i & \mathcal{U}(0, 1) > \text{TF} \\ t_{i+1} & \text{else} \end{cases}$$

- ▶  $t_{i+1}$  is the true token
- ▶ TF is the teacher forcing ratio

## Cross Entropy and Label Smoothing

$$\begin{aligned}\ell(\mathbf{x}, y_i) &= -\log \left( \frac{\exp x_{y_i}}{\sum_j \exp x_j} \right) \\ &= -\underbrace{x_{y_i}}_{\max} + \log \underbrace{\sum_j \exp x_j}_{\min}\end{aligned}$$

- ▶ Softmax and Cross-Entropy loss assign all the probability mass to a single word
  - ▶ LogSumExp is minimized on confident predictions
- ▶ Solution: **smooth** the distribution

## Quick Maths: LogSumExp Bounds

$$\begin{aligned}\max \{x_1, \dots, x_n\} &= \log (\exp(\max x_i)) \\ &\leq \log (\exp(x_1) + \dots + \exp(x_n)) \\ &\leq \log (n \cdot \exp(\max x_i)) \\ &= \max \{x_1, \dots, x_n\} + \log(n).\end{aligned}$$

- ▶ The lower bound is *approached* when **all but one** of the arguments approach  $-\infty$
- ▶ The upper bound is met when **all** the arguments are equal (uniform)
- ▶ Hence LogSumExp is minimized on a confident prediction

## Cross Entropy and Label Smoothing

- ▶ Softmax

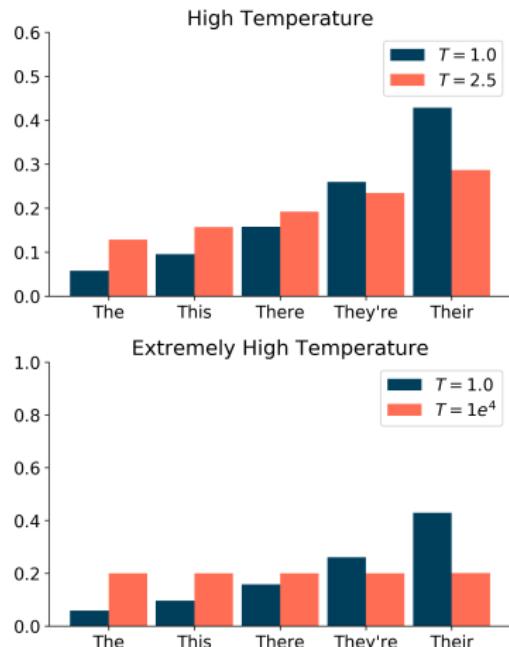
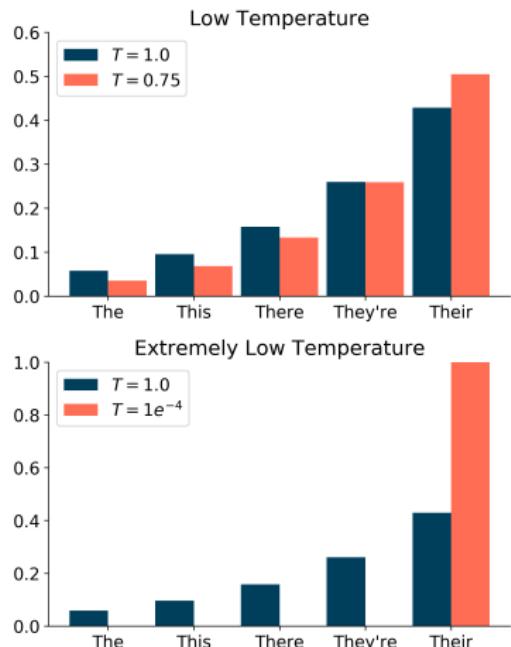
$$p(y_i) = \frac{\exp x_{y_i}}{\sum_j \exp x_j}$$

- ▶ Smoothed Softmax with Temperature  $T$

$$p(y_i) = \frac{\exp (x_{y_i}/T)}{\sum_j \exp (x_j/T)}$$

- ▶ As  $T \rightarrow \infty$ , the distribution is smoother, uniform
- ▶ As  $T \rightarrow 0$ , the distribution approaches a kronecker delta centered on the class with the most mass
- ▶ **Question:** Why do we divide instead of adding/subtracting?

# Visualizing Temperature



## Monte Carlo Decoding

- ▶ Recall how we select the next token:
  - ▶ **Greedy:** Top token weight
  - ▶ **Teacher Forcing:** Randomly select the true token
- ▶ Note that the outputs are a distribution over the target vocabulary
- ▶ **Use these weights in a multinomial to randomly select a continuation**

$$y_{i+1} \sim \text{Multinomial}(\theta_i)$$

# Different Token Decoding Schemes

- ▶ Greedy:

$$y_{i+1} = \operatorname{argmax}_j \theta_i$$

- ▶ Teacher Forcing:

$$y_{i+1} = \begin{cases} \operatorname{argmax}_j \theta_i & \mathcal{U}(0, 1) > \text{TF} \\ t_{i+1} & \text{else} \end{cases}$$

- ▶ Monte Carlo:

$$y_{i+1} \sim \text{Multinomial}(\theta_i)$$

- ▶  $\theta_i$  are the output weights from the Decoder
- ▶  $t_{i+1}$  is the true token at position  $i + 1$
- ▶ TF is the teacher forcing ratio

## Masked Loss

- ▶ Remember, we don't care what gets predicted after seeing a  $\langle \text{EOS} \rangle$
- ▶ Hence, we need to mask out the loss for predicted tokens associated with  $\langle \text{PAD} \rangle$

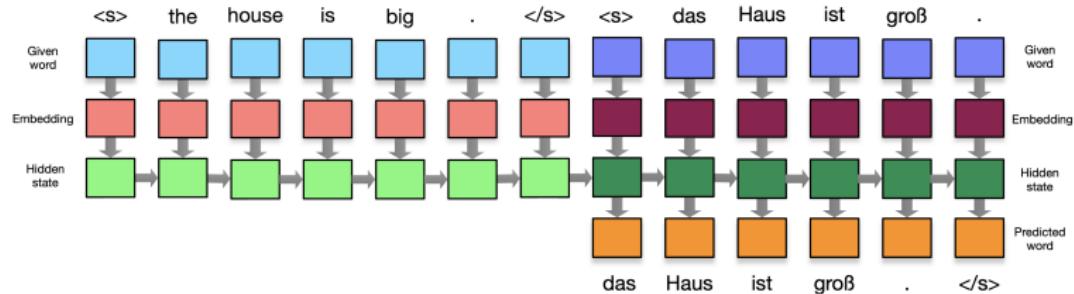
pred	$\rightarrow$	<i>le</i>	<i>le</i>	<i>chat</i>	<i>chat</i>	<i>chat</i>	<i>chat</i>
		$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
		$\ell$	$\ell$	$\ell$	$\ell$	$\ell$	0
		$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
true	$\rightarrow$	<i>le</i>	<i>chat</i>	<i>est</i>	<i>noir</i>	$\langle \text{EOS} \rangle$	$\langle \text{PAD} \rangle$

- ▶ **Solution:** Zero out elements by either:
  - ▶ Multiply pad outputs by 0
  - ▶ Specify the label to ignore in Cross Entropy call

$$\ell(\mathbf{x}, y_i) = \mathbb{1}_{\{y_i \neq \langle \text{PAD} \rangle\}} \cdot \left( -x_{y_i} + \log \sum_j \exp x_j \right)$$

## Initial Training Results

# Recall: Seq2Seq Architecture



```
Seq2Seq(  
    (encoder): Encoder(  
        (embedding): Embedding(21, 256)  
        (gru): GRU(256, 128, num_layers=2, batch_first=True)  
    )  
    (decoder): Decoder(  
        (embedding): Embedding(22, 256)  
        (gru): GRU(256, 128, num_layers=2, batch_first=True)  
        (fc): Linear(in_features=128, out_features=22, bias=True)  
    )  
)  
The model has 508,438 trainable parameters
```

## Seq2Seq: Sample Results

```
> num num num punct
= num <SEG> num <SEG> num <SEG> punct
< num <SEG> num <SEG> num <SEG> punct
```

## Seq2Seq: Sample Results

```
> num num num punct
= num <SEG> num <SEG> num <SEG> punct
< num <SEG> num <SEG> num <SEG> punct

> adj num num num
= adj <SEG> num <SEG> num <SEG> num
< adj <SEG> num <SEG> num <SEG> num num num num num
```

## Seq2Seq: Sample Results

> num num num punct

= num <SEG> num <SEG> num <SEG> punct

< num <SEG> num <SEG> num <SEG> punct

> adj num num num

= adj <SEG> num <SEG> num <SEG> num

< adj <SEG> num <SEG> num <SEG> num num num num num

> verb noun noun noun noun noun verb

= verb <SEG> noun noun <SEG> noun noun noun <SEG> verb

< verb noun <SEG> <SEG> noun noun noun noun <SEG> noun ...

## Translation Results: TL;DR

- ▶ Reasonably for small sequences
- ▶ Struggles to terminate properly
  - ▶ Maybe more training?
- ▶ Long Sequences are a mess

## Translation Results: TL;DR

- ▶ Reasonably for small sequences
- ▶ Struggles to terminate properly
  - ▶ Maybe more training?
- ▶ Long Sequences are a mess
- ▶ **However** this indicates that enough signal can be transferred via the hidden state

# Review

- ▶ We have covered:
  - ▶ What are Seq2Seq Models?
  - ▶ How can I reduce my vocabulary size?
  - ▶ How do I train a Seq2Seq model?
- ▶ Part 2 will explore more advanced concepts such as:
  - ▶ Decoding with Beam Search
  - ▶ Modeling Recurrent Relations
  - ▶ Other applications of Sequence Modeling

Thank You!

# Seq2Seq in Action: Column Segmentation

## Part 2

Bill Watson

S&P Global

January 30, 2020

# Review

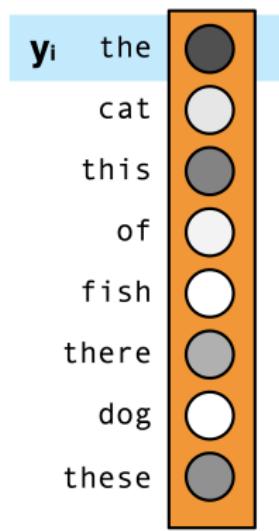
- ▶ We have covered:
  - ▶ What are Seq2Seq Models?
  - ▶ How can I reduce my vocabulary size?
  - ▶ How do I train a Seq2Seq model?
- ▶ Part 2 will explore more advanced concepts such as:
  - ▶ Decoding with Beam Search
  - ▶ Modeling Recurrent Relations
  - ▶ Other applications of Sequence Modeling

## Decoding: Making better Translations

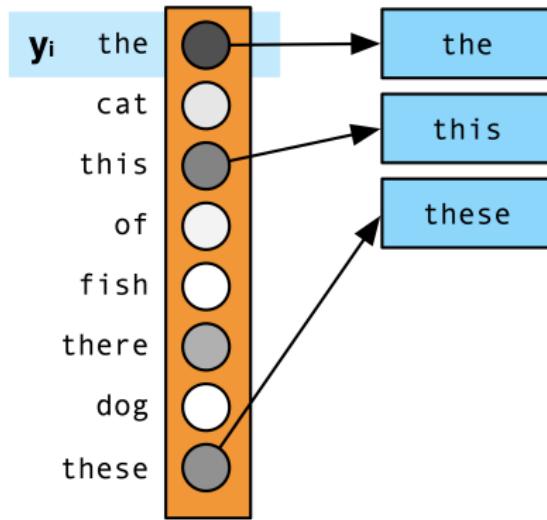
## Beam Search

- ▶ Beam Search is a technique to explore different translation paths
- ▶ Based on the idea that you **shouldn't take the greedy path**
- ▶ And that the best path may be sub-optimal during early iterations

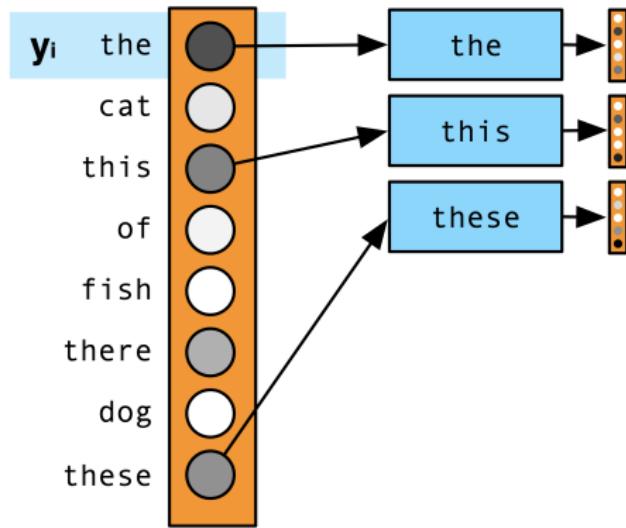
# Beam Search



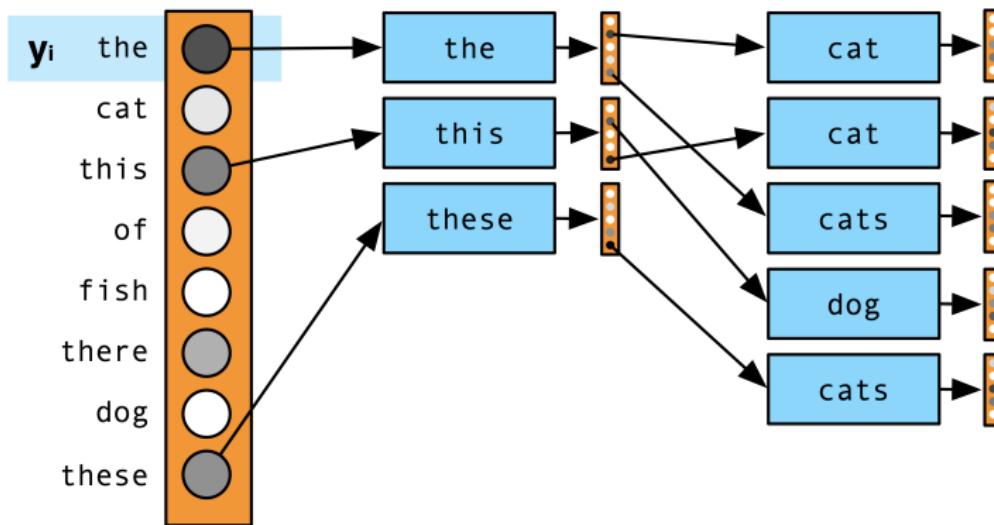
# Beam Search



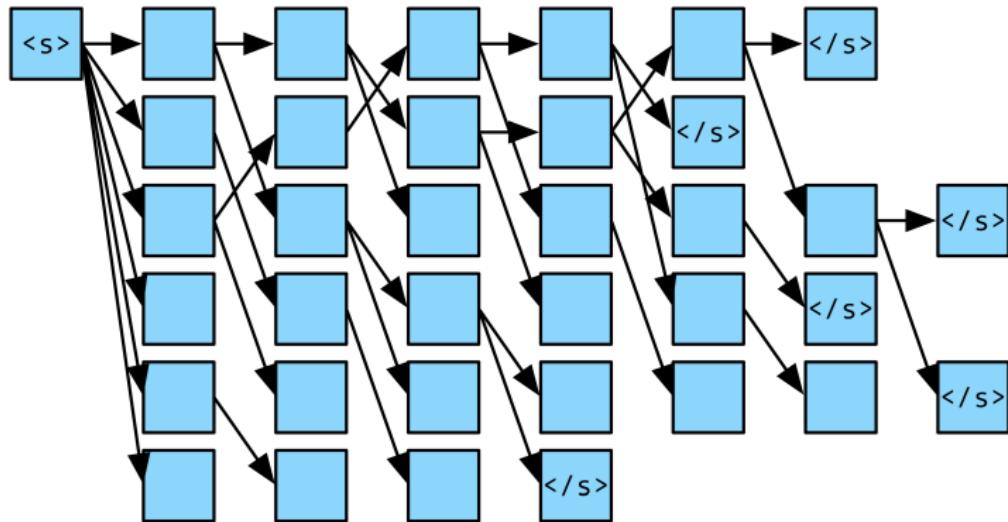
# Beam Search



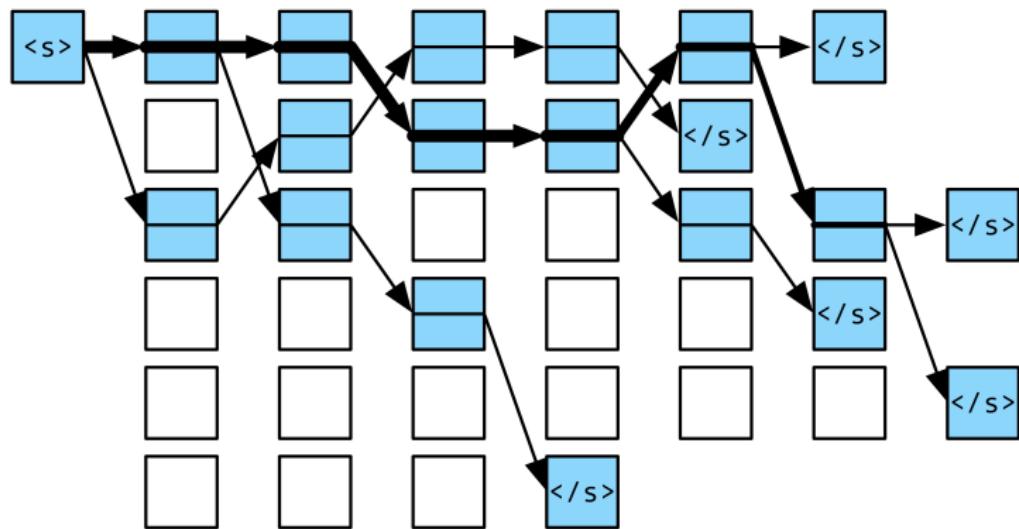
# Beam Search



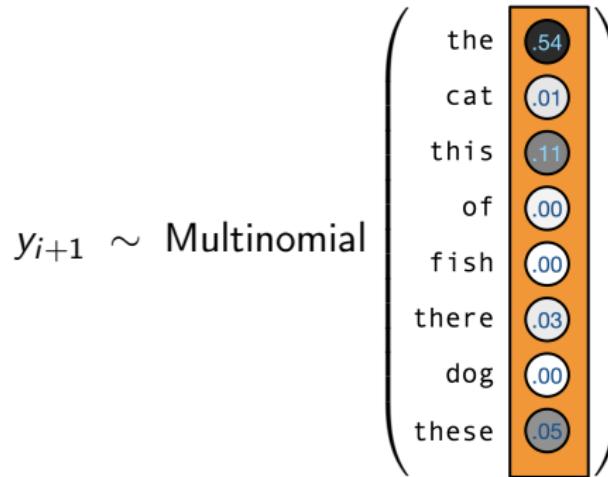
# Beam Search



# Beam Search

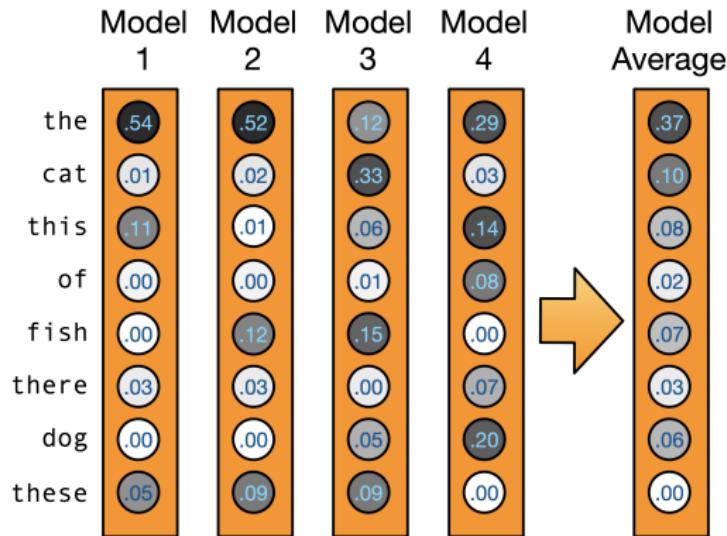


# Monte Carlo Beam Search



- ▶ Why not sample  $n$  words based on their probabilities?
- ▶ Adds more diversity to beam search results

# Ensembling



- ▶ Why not average different models?
- ▶ Random initialization leads to different local solutions
- ▶ Could also use model dumps from different iterations

## Modeling Recurrent Relations

# Vanilla RNNs

$$h_t = \tanh \left( \underbrace{W_{ih}x_t + b_{ih}}_{\text{input}} + \underbrace{W_{hh}h_{t-1} + b_{hh}}_{\text{hidden}} \right)$$

- ▶  $h_t$  is the hidden state at time  $t$
- ▶  $x_t$  is the input at time  $t$
- ▶  $h_{t-1}$  is the previous hidden state
- ▶  $h_0$  is initialized to 0

# Long Short Term Memory (LSTM)

$$\begin{aligned} \text{Gates} \rightarrow & \left\{ \begin{array}{l} i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\ f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\ o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\ g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \end{array} \right. \\ \text{Outputs} \rightarrow & \left\{ \begin{array}{l} c_t = f_t \odot c_{t-1} + i_t \odot g_t \\ h_t = o_t \odot \tanh(c_t) \end{array} \right. \end{aligned}$$

- ▶  $h_t$  is the hidden state at time  $t$
- ▶  $c_t$  is the cell state at time  $t$
- ▶  $x_t$  is the input at time  $t$

# Gated Recurrent Units (GRU)

$$\text{Gates} \rightarrow \begin{cases} r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \\ z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \\ n_t = \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{(t-1)} + b_{hn})) \end{cases}$$

$$\text{Outputs} \rightarrow \begin{cases} h_t = (1 - z_t) \odot n_t + z_t \odot h_{(t-1)} \end{cases}$$

- ▶  $h_t$  is the hidden state at time  $t$
- ▶  $x_t$  is the input at time  $t$

## Aside: Different Perspectives on Deep Recurrent Models

- ▶ So far we've only seen Left to Right Sequencing

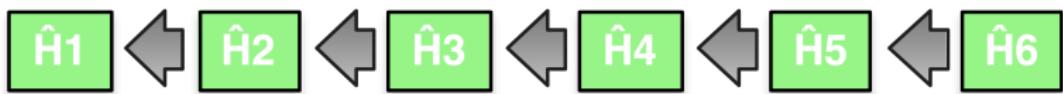


## Aside: Different Perspectives on Deep Recurrent Models

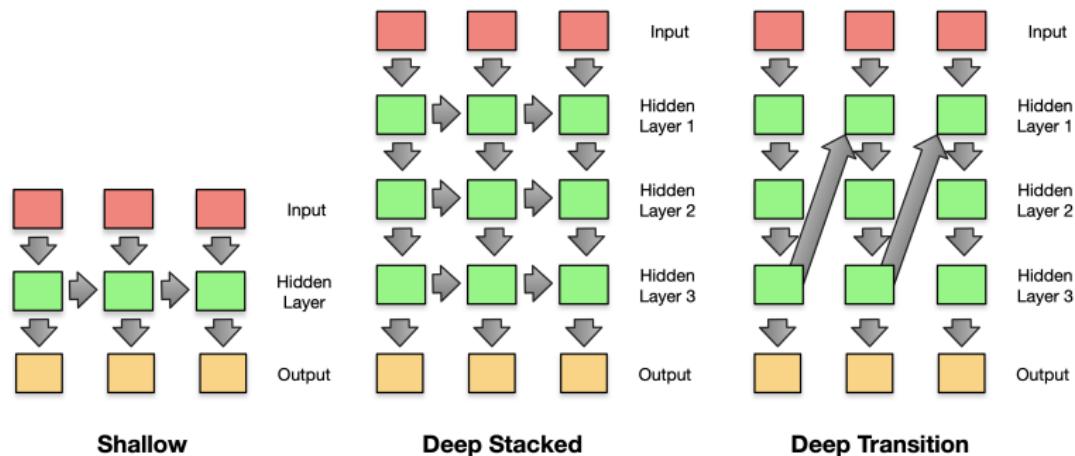
- ▶ So far we've only seen Left to Right Sequencing



- ▶ Why not Right to Left?

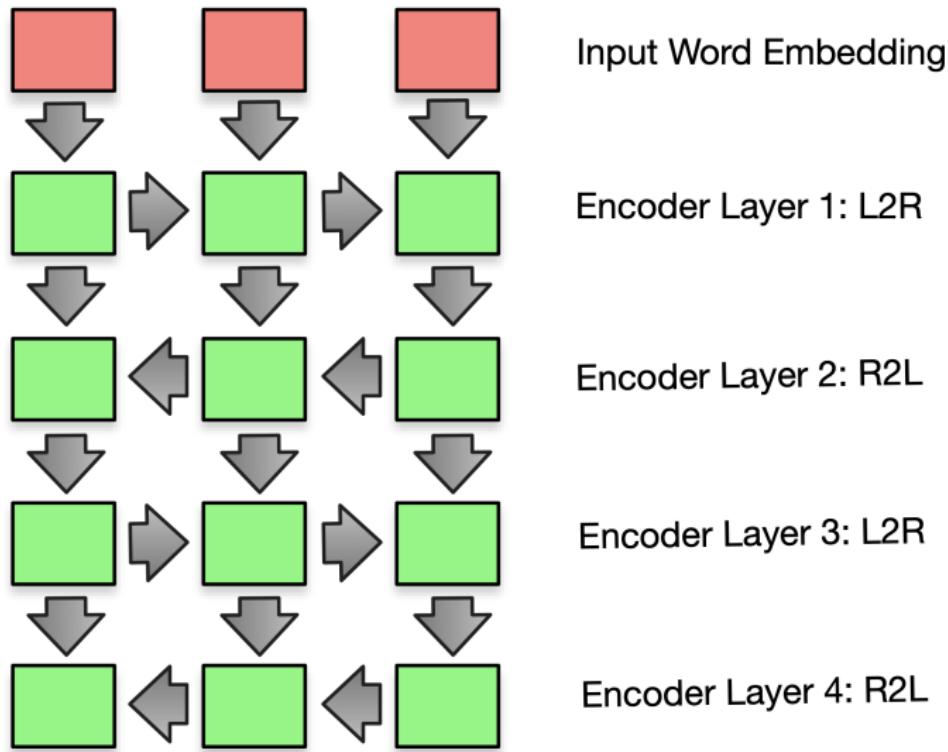


## Aside: Different Perspectives on Deep Recurrent Models

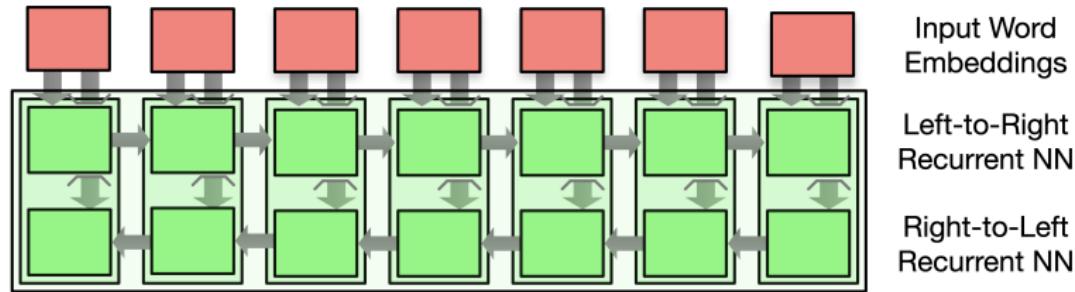


- ▶ Experiment with different stacking techniques

# Alternating Recurrent Directions



# Bidirectional Sequence Modeling



- ▶ Can capture both left and right context
- ▶ Implementation usually concatenates RNN states

## Aside: Dimensionality of Inputs and Outputs (Batch First)

Type	RNN	LSTM	GRU
In	$B, L, H_{in}$	$B, L, H_{in}$	$B, L, H_{in}$
$h_{t-1}$	$N_L \cdot N_D, B, H_{out}$	$N_L \cdot N_D, B, H_{out}$	$N_L \cdot N_D, B, H_{out}$
$c_{t-1}$	-	$N_L \cdot N_D, B, H_{out}$	-
$h_t$	$N_L \cdot N_D, B, H_{out}$	$N_L \cdot N_D, B, H_{out}$	$N_L \cdot N_D, B, H_{out}$
$c_t$	-	$N_L \cdot N_D, B, H_{out}$	-
Out	$B, L, N_D \cdot H_{out}$	$B, L, N_D \cdot H_{out}$	$B, L, N_D \cdot H_{out}$

- ▶  $B$  is the batch size
- ▶  $L$  is the sequence length
- ▶  $N_D$  is the number of directions
- ▶  $N_L$  is the number of layers
- ▶  $H_{in}, H_{out}$  are the input and hidden size

## Aside: The Influence of Padding in RNNs

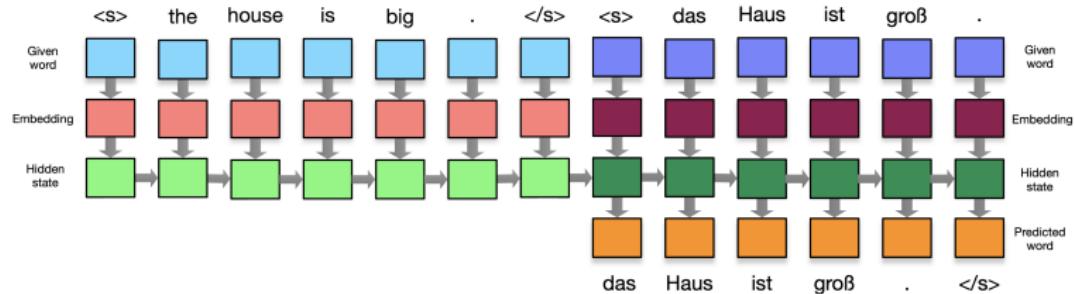
- ▶ Assume the embedding  $E[\langle \text{PAD} \rangle] = \mathbf{0}$
- ▶ Are we safe?

## Aside: The Influence of Padding in RNNs

- ▶ Assume the embedding  $E[\langle \text{PAD} \rangle] = \mathbf{0}$
- ▶ Are we safe? NO!
- ▶ Because of the bias term, zero input does not result in a zero output
  - ▶ This alters the hidden state being passed onto the next iteration
- ▶ Don't even think about the mess bidirectionals become...
- ▶ **Question:** Does this mean we learn the amount of padding for a given sequence?

## Applications to Column Segmentation

# Applications to Column Segmentation



```
Seq2Seq(  
    (encoder): Encoder(  
        (embedding): Embedding(21, 256)  
        (gru): GRU(256, 128, num_layers=2, batch_first=True)  
    )  
    (decoder): Decoder(  
        (embedding): Embedding(22, 256)  
        (gru): GRU(256, 128, num_layers=2, batch_first=True)  
        (fc): Linear(in_features=128, out_features=22, bias=True)  
    )  
)  
The model has 508,438 trainable parameters
```

## Column Segmentation: Throw away the decoder

- ▶ Just use an encoder, with an output layer
- ▶ For every token, predict if we should insert a `\langle SEG \rangle`
- ▶ No longer have to learn the actual sequence

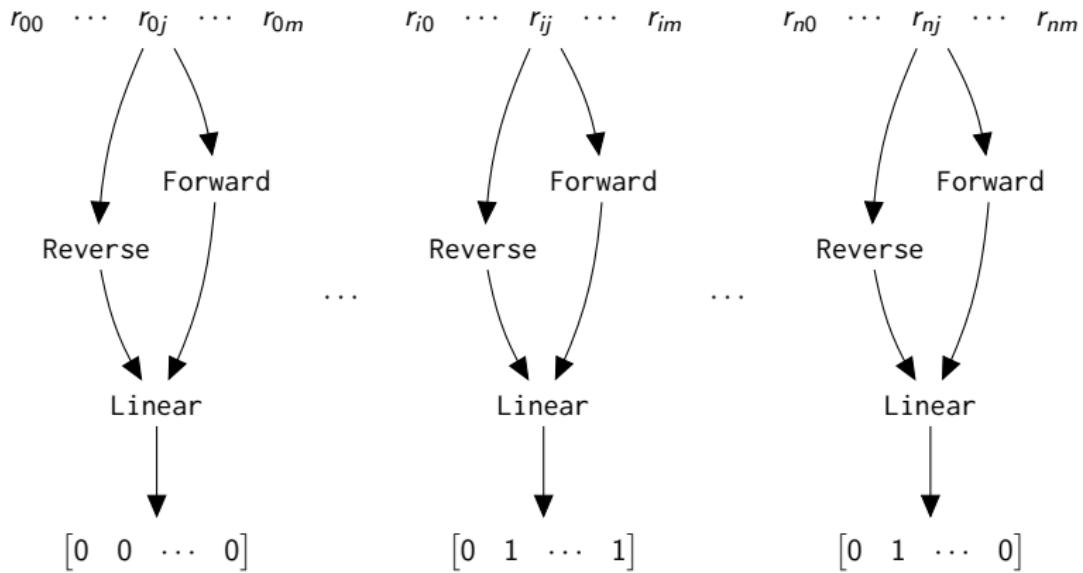
```
Classifier(  
    (encoder): Encoder(  
        (embedding): Embedding(21, 128)  
        (gru): GRU(128, 128, batch_first=True, bidirectional=True)  
    )  
    (linear): Linear(in_features=256, out_features=1, bias=True)  
)  
The model has 201,089 trainable parameters
```

## Independent Row: Training Labels

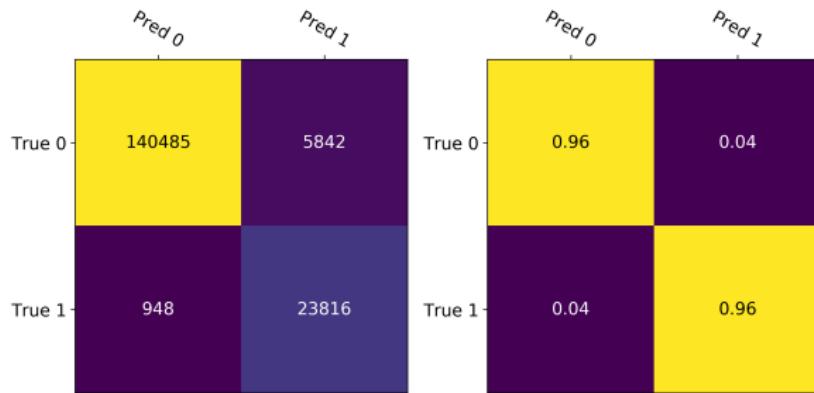
- ▶ Labels are now binary vectors, where  $x_i = 1$  implies that there is a **(SEG)** between  $x_i$  and  $x_{i+1}$

Less imputed interest	(	174,862.64	)		
0	0	1	0	0	0
Less imputed interest	<b>(SEG)</b>	(	174,862.64	)	

# Independent Row Training Visual

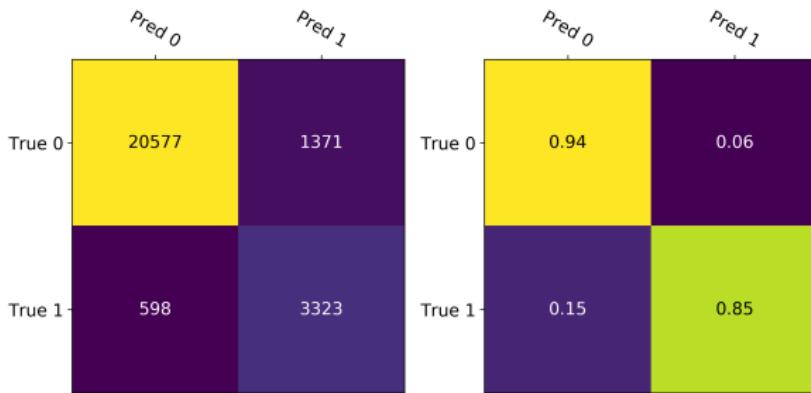


## Independent Row: Sample Results (Training)



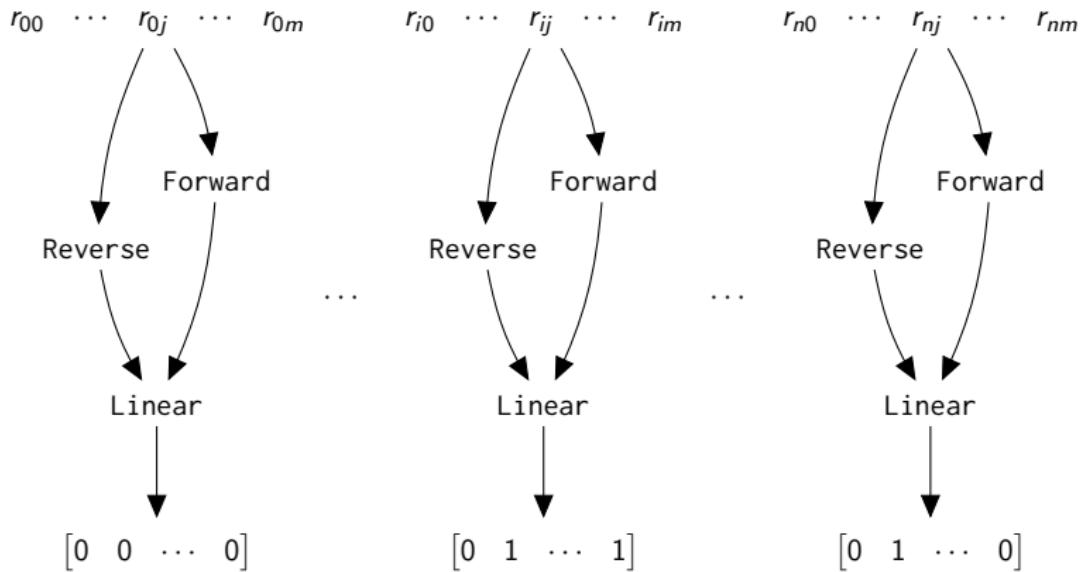
	Precision	Recall	F1-Score	Support
Label: 0	0.99	0.96	0.98	146,327
Label: 1	0.80	0.96	0.88	24,764
accuracy			0.96	171,091
macro avg	0.90	0.96	0.93	171,091
weighted avg	0.97	0.96	0.96	171,091

## Independent Row: Sample Results (Validation)



	Precision	Recall	F1-Score	Support
Label: 0	0.97	0.94	0.95	21,948
Label: 1	0.71	0.85	0.77	3,921
accuracy			0.92	25,869
macro avg	0.84	0.89	0.86	25,869
weighted avg	0.93	0.92	0.93	25,869

## How can we improve on this?

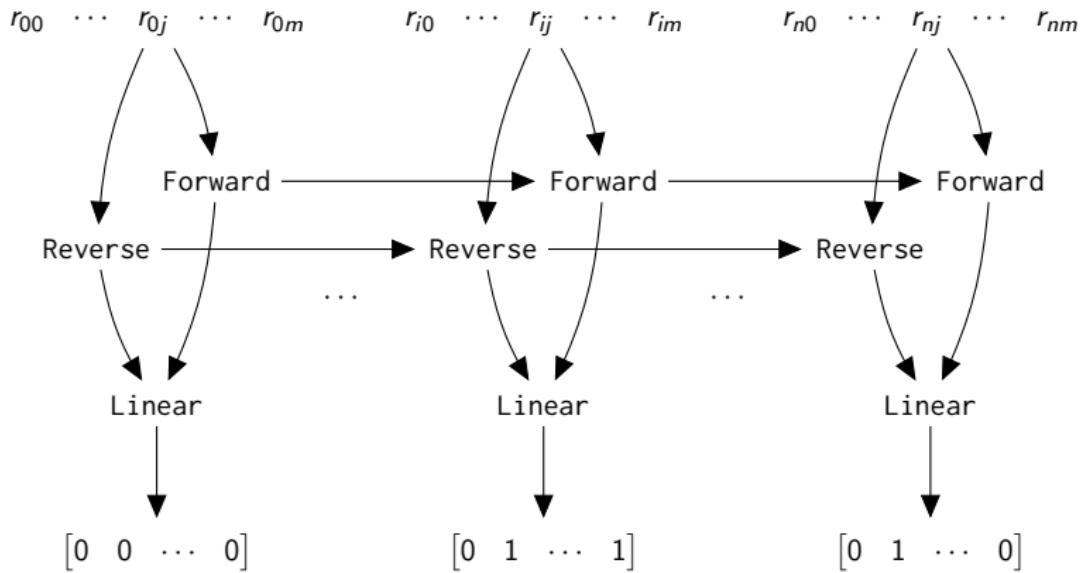


## Joint Training on a Table

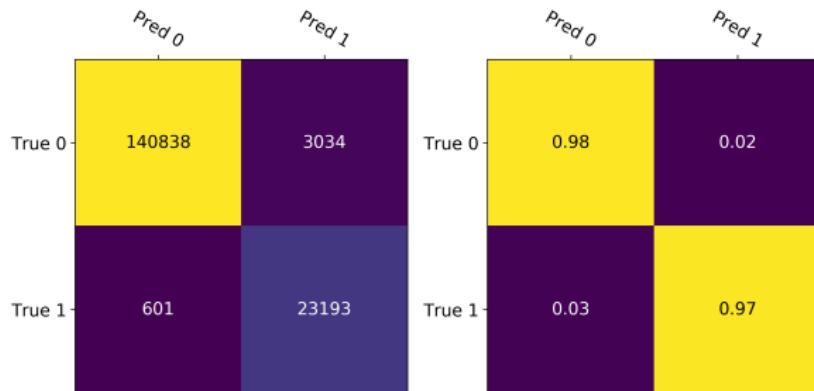
- ▶ We saw that we could regenerate a sequence from its hidden state alone.
- ▶ Why not use this information to link rows within the same table?
  - ▶ Bidirectional for each row
  - ▶ Propagate hidden state to next row
  - ▶ Linear layers for prediction

```
TableClassifier(  
    (embedding): Embedding(21, 128)  
    (rnn): GRU(128, 128, batch_first=True, bidirectional=True)  
    (ff): Linear(in_features=256, out_features=1, bias=True)  
)  
The model has 201,089 trainable parameters
```

# Joint Table Training Visual

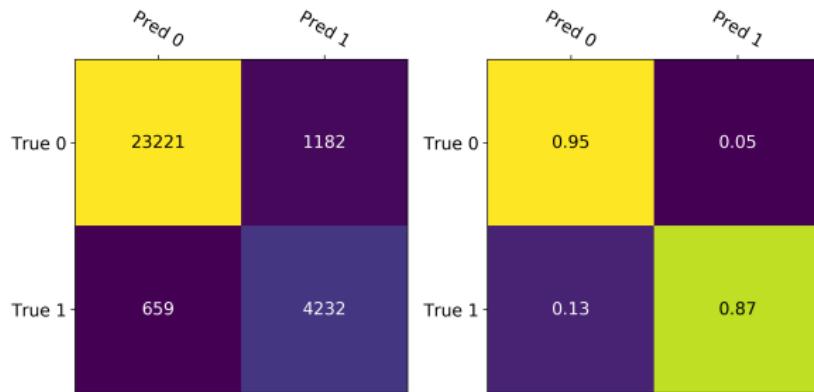


## Joint Table Training: Sample Results (Training)



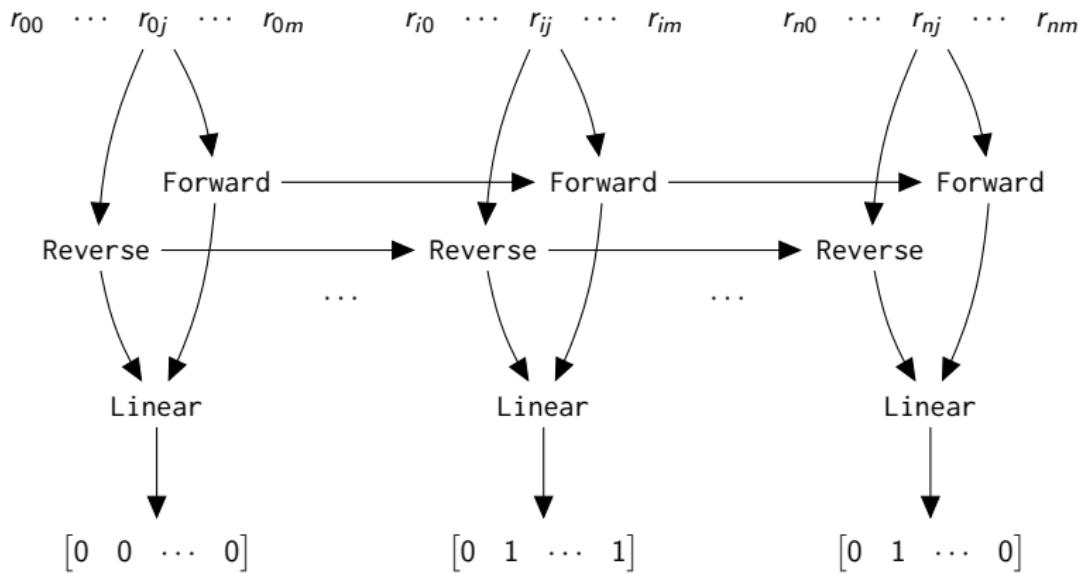
	Precision	Recall	F1-Score	Support
Label: 0	1.00	0.98	0.99	143,872
Label: 1	0.88	0.97	0.93	23,794
accuracy			0.98	167,666
macro avg	0.94	0.98	0.96	167,666
weighted avg	0.98	0.98	0.98	167,666

## Joint Table Training: Sample Results (Validation)

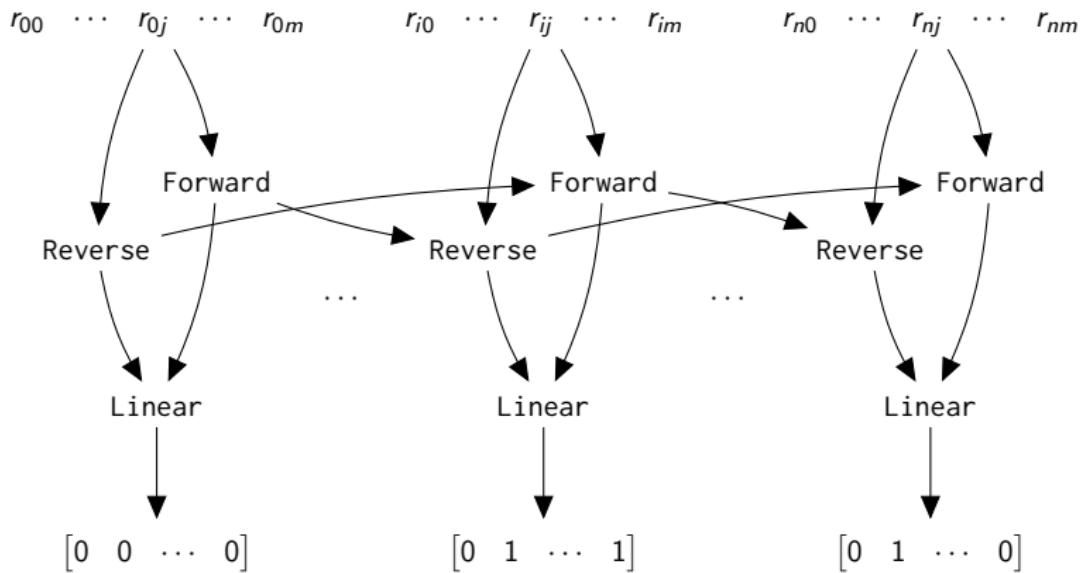


	Precision	Recall	F1-Score	Support
Label: 0	0.97	0.95	0.96	24,403
Label: 1	0.78	0.87	0.82	4,891
accuracy			0.94	29,294
macro avg	0.88	0.91	0.89	29,294
weighted avg	0.94	0.94	0.94	29,294

# Is this a correct way to link rows?



# Joint Table Training with Swapping Visual

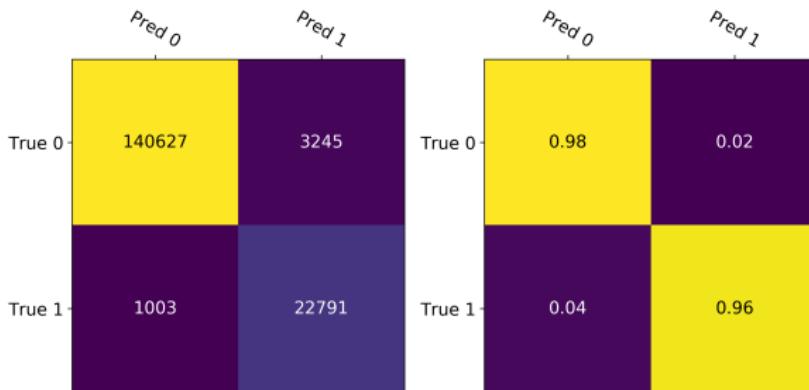


# Trick for Swapping in Joint Table Training

- ▶ Note that the hidden state is biased towards the items closest to it
- ▶ So the beginning of a new row has context from the end of the last row
  - ▶ Is this a good choice?
  - ▶ What if we flip the bidirectional hidden states every iteration
  - ▶ AKA the beginning of a new row has the hidden state info from the beginning of the last row
  - ▶ And vice-versa

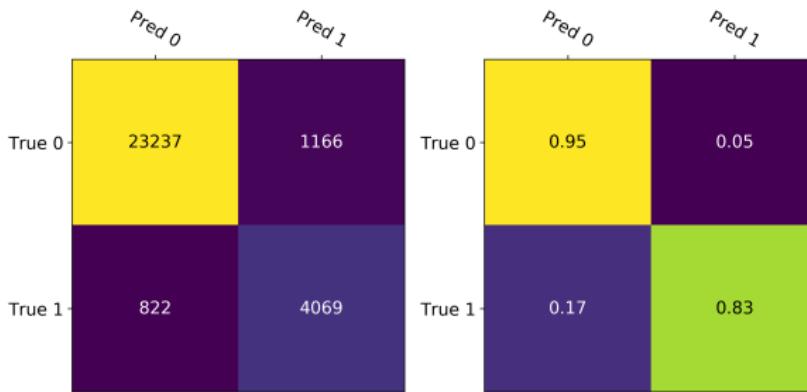
```
TableClassifierSwap(  
    (embedding): Embedding(21, 128)  
    (rnn): GRU(128, 128, batch_first=True, bidirectional=True)  
    (ff): Linear(in_features=256, out_features=1, bias=True)  
)  
The model has 201,089 trainable parameters
```

# Joint Table Training with Swapping: Sample Results (Training)



	Precision	Recall	F1-Score	Support
Label: 0	0.99	0.98	0.99	143,872
Label: 1	0.88	0.96	0.91	23,794
accuracy			0.97	167,666
macro avg	0.93	0.97	0.95	167,666
weighted avg	0.98	0.97	0.98	167,666

# Joint Table Training with Swapping: Sample Results (Validation)



	Precision	Recall	F1-Score	Support
Label: 0	0.97	0.95	0.96	24,403
Label: 1	0.78	0.83	0.80	4,891
accuracy			0.93	29,294
macro avg	0.87	0.89	0.88	29,294
weighted avg	0.93	0.93	0.93	29,294

## Summary of Methods (Validation)

		Independent	Joint	Joint Swapped
Label: 0	Precision	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>
	Recall	0.94	<b>0.95</b>	<b>0.95</b>
	F1-Score	0.95	<b>0.96</b>	<b>0.96</b>
Label: 1	Precision	0.71	<b>0.78</b>	<b>0.78</b>
	Recall	0.85	<b>0.87</b>	0.83
	F1-Score	0.77	<b>0.82</b>	0.80
Accuracy		0.92	<b>0.94</b>	0.93

## What have we learned?

- ▶ We can learn text segments purely from the sequence of Part-of-Speech tags
- ▶ It is useful to jointly train/predict on the table instead of assuming independence
- ▶ Is there any benefit to swapping hidden states during propagation?
- ▶ **Note:**
  - ▶ We use the concept of *deep transition* in Seq2Seq for joint training
  - ▶ Where our training is dependent on the hidden state encoding useful information

# Future Plans

- ▶ Features
  - ▶ Use word location information
  - ▶ Tesseract gives bounding boxes
- ▶ Encoder Structure
  - ▶ Convolution
  - ▶ Transformer

## Other Applications For Sequence Modeling

## Recall: Encoders, Decoders, and Seq2Seq Models

- ▶ **Encoders** given a sequence meaning
- ▶ **Decoders** generate a new sequence
- ▶ **Seq2Seq** generate sequences conditioned on another sequence

# What to use for which application?

- ▶ **Encoders**
  - ▶ POS Tagging
  - ▶ Sentence Embeddings
  - ▶ Anything where you are given the sentence at test time
- ▶ **Decoders**
  - ▶ Text Generation
  - ▶ Language Modeling
  - ▶ Anything where you need to create a sequence at test time
- ▶ **Seq2Seq**
  - ▶ Translation
  - ▶ Speech Recognition
  - ▶ Summarization
  - ▶ Question/Answering
  - ▶ Anything where you convert a sequence into another sequence

## Tools, References, and Further Reading

# Acknowledgment

- ▶ Pieces adapted from my college Machine Translation Course by Philipp Koehn
- ▶ Additional acknowledgment:
  - ▶ Adam Lopez
  - ▶ Matt Post
  - ▶ Chris Callison-Burch
  - ▶ Philipp Koehn

## Papers

- ▶ Sutskever et al., Sequence to Sequence Learning with Neural Networks
- ▶ Cho et al., Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
- ▶ Sennrich et al., Neural Machine Translation of Rare Words with Subword Units
- ▶ Koehn, Neural Machine Translation
- ▶ Koehn, Six Challenges for Neural Machine Translation
- ▶ Curated Machine Translation Reading List

# Tutorials

- ▶ Pytorch
  - ▶ Official PyTorch Seq2Seq Tutorial
  - ▶ PyTorch Seq2Seq with Torchtext
  - ▶ Ben Trevett Seq2Seq Tutorial
- ▶ Tensorflow
  - ▶ NMT with Attention

# Libraries

- ▶ Facebook: fairseq (PyTorch)
- ▶ Open NMT (PyTorch)
- ▶ Open NMT (Tensorflow)

Thank You!