Headline Generator

Dane Hankamer dhank@stanford.edu

David Liedtka dliedtka@stanford.edu

Key Question: Can we combine knowledge of article structure with RNNs to predict headlines?

Objective

- Based on previous study by Lopyrev
- Use encoder to process article word-by-word, decoder to generate headline word-by-word
- Lopyrev used the first 50 words of each article, can we accomplish better summarization using other parts of the article?

Data

- Used "All the News" dataset by Andrew Thompson
- Processed 96,543 articles from 12 sources
- Example article:
  HEADLINE: `Hillary Clinton: Her Notable Moments`
  ARTICLE: `The name Hillary Rodham first appeared in the pages of The New York Times 47 years ago in a 1969 article about her commencement address at Wellesley the women's college in Massachusetts...`

Model

- **RNN ARCHITECTURE:**

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 50, 100)	4000000
lstm_1 (LSTM)	(None, 50, 512)	1255424
dropout_1 (Dropout)	(None, 50, 512)	0
lstm_2 (LSTM)	(None, 50, 512)	2099200
dropout_2 (Dropout)	(None, 50, 512)	0
lstm_3 (LSTM)	(None, 50, 512)	2099200
dropout_3 (Dropout)	(None, 50, 512)	0
simplecontext_1 (Lambda)	(None, 25, 944)	0
time_distributed_1 (TimeDist	(None, 25, 40000)	37800000
activation_1 (Activation)	(None, 25, 40000)	0

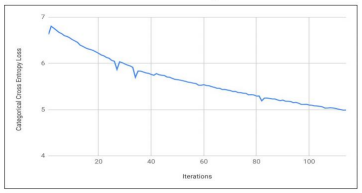
Total params: 47,253,824
Trainable params: 47,253,824
Non-trainable params: 0

Model (continued)

- **Hyperparameters:** Adam Optimizer, first moment 0.9, second moment 0.999, epsilon 1e-8, learning rate 0.0001
- **Loss function:** Categorical cross entropy loss,

$$L(\hat{y}, y) = \frac{-1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \in C_c} \log p_{model}[y_i \in C_c]$$

minimized over 114 iterations of training:



Results

BLEU Score	Training	Validation	Test
50	0.01365756221	0.01451416069	0.01226555188
25 + 25	0.01334964053	0.01323200369	0.01297593721
50 + 25	0.01382171917	0.01177343727	0.01119147869

Levenshtein	Training	Validation	Test
50	54.77231648	54.52419509	54.98365409
25 + 25	54.55219124	55.39320007	54.40124380
50 + 25	54.92491911	54.19084872	54.42568945

- Where **50** corresponds to inputting the first 50 words of an article, **25 + 25** corresponds to the first 25 and last 25 words, and **50 + 25** corresponds to the first 50 and last 25 words
- Objective: maximize BLEU score, minimize Levenshtein score

Discussion

- Selected (cherry-picked) examples:
  - TRUE: `Federal Judge Blocks Obama Administration Protections For Transgender People^`
    PREDICTED: `Federal Judge Blocks Obama Administration Protections For Gay People Under A order:']^ order:']^`
  - TRUE: `Neil deGrasse Tyson and Al Gore on the future of our planet — and everything else`
    PREDICTED: `Neil deGrasse Tyson and Al Gore on the future of our planet — and everything else and everything else and everything else and else`
  - TRUE: `Brothers share what it was like quitting their corporate jobs to sell ties on the beach and cofound^ Vineyard Vines,^ a company worth nearly $1 billion`
    PREDICTED: `Brothers share what it was like quitting their corporate jobs to sell ties on the beach and worked\n^ Vineyard worked\n^ a company worth a theory`
- BLEU score results match Lopyrev's early results, but with further training he is able to attain an average BLEU score of about 0.09
- No evidence of overfitting
- Did not effectively leverage different parts of article
- Difficulty predicting <EOS> token

Future

- Train until convergence, prohibitive cost led to non-optimal results
- Further experiment with hyperparameters, architecture
- Clean data, prevalent meta/non-article text caused problems
- How to better leverage key parts of article for summarization (include parts other than exclusively introduction and conclusion)?

References

- Levenshtein, V. Levenshtein distance, 1965.
- Lopyrev, K. Generating news headlines with recurrent neural networks. *arXiv preprint aXiv:1512.0712* (2015).
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (2002), Association for Computational Linguistics, pp. 311-318.
- Thompson, A. ALl the news. https://components.one/datasets/#all-the-news, 2019
- See paper for complete list of references