Outperforming Long-form Neural Abstractive Summarizers with Simple Unsupervised Baselines

Shira Eisenberg and Sam Wiseman

University of Chicago and TTIC

Motivation and Overview

Neural generation methods have made great strides when applied to the abstractive summarization of news documents, but their effectiveness at summarizing longer form scientific articles is less well understood.
We show that:

- When applied to the scientific article summarization tasks recently proposed by Cohan et al. (2018), neural methods only minimally outperform simple lede-style baselines
- Neural methods outperformed by LexRank (Erkan and Radev, 2004), an unsupervised, non-neural, extractive baseline
- LexRank with original TF-IDF-based similarity scores outperforms similarity scores based on BERT

Task Background

- The summarization of longer form scientific articles has received less attention than the summarization of news articles.
- Cohan et al. (2018) have recently introduced two datasets for the summarization of longer form scientific articles, one consisting of PubMed articles paired with their abstracts, and the other consisting of arXiv articles paired with their abstracts. Statistics shown in Table 1
- To tackle this summarization task, Cohan et al. (2018) propose an attentional sequence-to-sequence style model with copy attention and a coverage penalty, augmented with a hierarchical document encoder and a discourse-aware decoder.
- We compare the Cohan et al. (2018) model, which we take to embody many aspects of state-of-the-art neural text generation systems, with the LexRank model of Erkan and Radev (2004), which is neither neural nor supervised

Example of a Generated Summary

- **Abstract:** in this paper , the author proposes a series of multilevel double hashing schemes called cascade hash tables . they use several levels of hash tables . in each table , we use the common double hashing scheme . higher level hash tables work as fail - safes of lower level hash tables . by this strategy , it could effectively reduce collisions in hash insertion . thus it gains a constant worst case lookup time with a relatively high load factor (@xmath0) in random experiments . different parameters of cascade hash tables are tested .
- **Cohan et al. (2018):** cascade hash tables are a common data structure used in large set of data storage and retrieval . such a time variation is essentially caused by possibly many collisions during keys hashing . in this paper , we present a set of hash schemes called cascade hash tables which consist of several levels (@xmath2) of hash tables with different size . after constant probes , if an item ca âĂŹnt find a free slot in limited probes in any hash table , it will try to find a cell in the second level , or subsequent lower levels . with this simple strategy , these hash tables will have descendant load factors , therefore lower collision probabilities .
- **this work:** hash table entry in wikipedia. there are various hash functions on strings , such as crc ,we call these methods unlimited. a method is limited , if the number of probes can not exceed some limit . any of these methods may probe indefinite number of locations , even as many as (@xmath1) in the worst case . well known probe sequences include : linear probing , in which the interval between probes is fixed often at 1 ; quadratic probing , in which the interval between probes increases linearly (hence , the indices are described by a quadratic function) ; double hashing , in which the interval between probes is fixed for each record but is computed by another hash function . a hash collision is resolved by probing through alternate locations in the array(the probe sequence) until either the target record is found , or an unused array slot is found , which indicates that there is no such key in the table . open addressing hash tables store the colliding records directly within the array .

LexRank Background

- LexRank (Erkan and Radev, 2004) is an unsupervised extractive approach to text summarization, which attempts to score sentences in terms of eigenvector centrality, using a slight modification of the PageRank algorithm (Page et al., 1998).
- LexRank first generates a graph representation of an input document, where nodes represent sentences and weighted edges represent the pairwise similarity between sentences, ($\mathbf{B} \in \mathbb{R}^{N \times N}$; $\mathbf{B}_{ij} = \mathbf{B}_{ji}$ is the non-negative normalized similarity score between sentences x_i and x_j)
- The eigenvector centrality of each node is calculated by normalizing the rows of $\mathbf{B}$ to form a stochastic matrix,then finding the principal eigenvector typically using power iteration.

Dataset Statistics

Dataset	Num Docs	Avg. Doc Len	Avg. Summ. Len
PubMed	133K	3.0K	203
arXiv	215K	4.9K	220

Table 1: Dataset statistics comparison for PubMed and arXiv (Cohan et al., 2018) corpora; document and summary lengths are measured in words.

Results

Threshold τ	RG-1	RG-2	RG-L
PubMed			
0.03	41.050	15.258	36.854
0.1	41.745	15.604	37.505
arXiv			
0.03	37.681	13.925	33.721
0.03	38.691	14.216	34.804

Table 2: Top-two settings of τ threshold hyperparamter by ROUGE scores on PubMed and arXiv validation datasets, when using standard TF-IDF similarities. In general, performance is fairly sensitive to τ

Model	RG-1	RG-2	RG-L
PubMed			
Lede-6	37.11	12.85	33.78
Attn-Seq2Seq	31.55	8.52	27.38
Pntr-Gen-Seq2Seq	35.86	10.22	29.69
Cohan et al.	38.93	15.37	35.21
LexRank **42.09**	**15.91**	**37.84**	
arXiv			
Lede-5	34.25	8.70	30.44
Attn-Seq2Seq	29.30	6.00	25.56
Pntr-Gen-Seq2Seq	32.06	9.04	25.16
Cohan et al.	35.80	11.05	31.80
LexRank	**37.91**	**14.34**	**33.82**

Table 3: ROUGE performance of the best neural models to date, and of LexRank and lede-style baselines, on PubMed and arXiv test sets. âĂIJAttn-Seq2SeqâĂİ and âĂIJPntr-Gen-Seq2SeqâĂİ numbers are taken from Cohan et al. (2018), as are their main results.

Conclusions and Key Points

- On longer summarization datasets, current abstractive neural generation models underperform unsupervised, non-neural baselines
- We believe these results offer a challenge to the next generation of neural generation models, which must be able to handle longer documents and longer summaries
- Based on the success of LexRank on these larger problems, it would be interesting to see to what extent improving similarity scores, perhaps with improved sentence representations, can improve summarization performance
- Scaling neural abstractive methods up may require explicitly modeling both the graph-structure of the documents (or collections of documents) they are attempting to summarize, as well as the graph strucutre of the summaries they are attempting to generate