Cadernos BAD

ETD 2019 Fruits of knowledge

Summarizing ETDs with deep learning

William A. Ingram

Virginia Tech, USA

waingram@vt.edu

Bipasha Banerjee

Virginia Tech, USA

bipashabanerjee@vt.edu

Edward A. Fox

Virginia Tech, USA

fox@vt.edu

Abstract

Inspired by the millions of Electronic Theses and Dissertations (ETDs) openly available online, we describe a novel use of ETDs as data for text summarization. We use a large corpus of ETDs to evaluate techniques for generating abstractive summaries with deep learning. Using an extensive ETD collection of over 30,000 doctoral dissertations and master's theses, we examine the quality of state-of-the-art deep learning summarization technologies when applied to an ETD corpus. Deep learning requires a large set of training data to produce satisfactory results. Finding suitable training data is especially difficult due to the widespread use of domain-specific jargon in ETDs, coupled with the wide-ranging breadth of subject matter contained in an ETD corpus. To overcome this significant limitation, we demonstrate the potential of transfer learning on automatic summarization of ETD chapters. We apply several combinations of deep learning models and training data to the ETD chapter summarization task and compare the outputs of the top performers.

Keywords: Text summarization, Text segmentation, Abstractive summarization, Deep learning

Introduction

Millions of Electronic Theses and Dissertations (ETDs) are openly available online, constituting a rich and freely available corpus of graduate research and scholarship. This important corpus of scholarly content can be leveraged in novel ways to advance further education and research. In this paper, we address the research problem *How can we best automatically construct English language summaries of the important information in a large document collection?* Our work focuses on using state-of-the-art deep learning methods for generating abstractive summaries of ETD chapters. ETD chapter summarization poses specific challenges, such as the unstructured format of PDF files, the lack of sufficient labeled training data needed for building supervised deep learning models, and the large size and multidisciplinary scope of the document corpus itself. Using the university's ETD collection, made up of over 30,000 doctoral dissertations and master theses, we explore the difficulty of information extraction from ETD documents, the potential of transfer learning on automatic summarization of ETD chapters, and the quality of state-of-the-art deep learning summarization technologies when applied to the ETD corpus.

Literature Review

Summarization can broadly fit into extractive and abstractive categories. Extractive summarization directly copies the important information from the original text to the summary. Abstractive summarization, on the other hand, rephrases the original text and produces a final summary. We compare three state-of-the-art deep learning techniques for creating abstractive summaries. Sutskever, et al. (2014) introduced Sequence to Sequence Networks, which uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. See, et al. (2017) augment the neural sequence-to-sequence attentional model with their Pointer-Generator Network, which can copy words from the source text via pointing while retaining the ability to produce novel words through the generator. Reinforce-Selected Sentence Rewriting (Chen & Bansal, 2018) tries to mimic the patterns of humans summarizing long documents by first compressing the text then paraphrasing it. This hybrid model combines the advantages of extractive and abstractive paradigms with policy-based reinforcement learning to bridge together two networks.

For extracting chapter text from PDF documents, we use two scholarly PDF data extraction tools, Grobid (Lopez, 2009) and Science Parse (Science Parse, 2019). Following the approach of Peng, et al. (2004), these tools operate by applying conditional random fields to document segmentation. They attempt to convert unstructured PDF documents into structured XML or JSON. Grobid marks up its output using the TEI (Text Encoding Initiative) (Sperberg-McQueen & Bernard, 1990) extensible XML schema, and Science Parse structures its output as simple JSON.

The standard for evaluating a machine-generated summary is to compare it with a "gold standard" summary, usually created by hand. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores (Lin, 2004) are calculated to evaluate generated summaries against the gold standard.

Methodology

Our primary aim is to automatically construct summaries of the critical information in an extensive document collection. In doing do, we devise and tailor a workflow for ETD chapter summarization with a focus on state-of-the-art deep learning techniques. Our corpus of ETDs consists of 13,071 doctoral dissertations and 17,890 master's theses downloaded from the University Libraries' VTechWorks institutional repository system. Before we begin the main task of creating chapter summaries from ETDs, we first need to devise a process for segmenting the documents into chapters. We use Grobid (Lopez, 2009) and Science Parse (Science-Parse, 2019) to create structured data from the unstructured PDFs. From the structured output of these tools, we are able to extract individual chapter text and strip away citations, notes, tables, figures, captions, and all other extraneous content. Finally, we use various Python modules from NLTK (Bird, Klein, & Loper, 2009) to extract sentences and correct punctuation errors.

Deep learning requires a large set of labeled training data to produce satisfactory results, but to our knowledge no sufficient training set exists for ETD chapters. Finding suitable training data is especially difficult due to the widespread use of domain-specific jargon in ETDs, coupled with the wide-ranging breadth of subject matter contained in the ETD corpus. To overcome this significant limitation, we implement various transfer-learning techniques, in which a base language model trained on a different large data set is used to generate summaries for ETD chapters. We compare results from models trained on a CNN/Daily Mail (Hermann et al., 2015) news article data set, scientific papers downloaded from the arXiv.org e-Print archive (arXiv.org, 2019), and a recent English Wikipedia database dump (Meta, 2019). The CNN/Daily Mail data set contains articles and summaries. But for the arXiv and Wikipedia data, we create article-summary pairs using an article's abstract section as the training label and the remaining article content as training input. Additionally, we train models with a combination of data sets (*e.g.*, CNN/Daily Mail + Wikipedia) to see if these models would perform better than those trained on a single source.

We examine three state-of-the-art deep learning techniques for creating abstractive summaries: Sequence to Sequence Networks (S2S), Pointer-Generator Networks (PGN), and Reinforce-Selected Sentence Rewriting (RSSR). To validate and test our deep-learning models, we manually create gold standard summaries at the chapter level for several ETDs. For each thesis or dissertation chosen, we 1) manually extract the text for each chapter; and 2) carefully read each chapter and write a coherent overview, including interesting details from the chapter. Each summary stands on its own and provides enough context so that researchers would understand the overall topic of the thesis if they only read the chapter summary. The result is 150 gold standard chapter summaries from about 30 ETDs on a range of topics from several distinct academic disciplines.

We train our models using various data sets listed above and report the best results from the various combinations of training data sets and deep learning techniques employed.

Model + Dataset	ROUGE-1	ROUGE-2	ROUGE-L
PGN + CNN/Daily Mail	0.238	0.097	0.213
PGN + arXiv	0.222	0.038	0.198
S2S + CNN/Daily Mail	0.205	0.069	0.162
PGN + Wikipedia	0.172	0.057	0.155
PGN + CNN/Daily Mail + Wikipedia	0.137	0.040	0.122
RSSR + CNN/Daily Mail + arXiv	0.139	0.122	-

TABLE 1

ROUGE scores for generated summaries

Results and Discussion

We calculate ROUGE scores to evaluate the generated summaries against our gold standard summaries. The top-performing models are listed in Table 1. The summaries generated by the CNN/Daily Mail models achieve the best ROUGE scores, and the summaries generally consist of easy to read sentences. We attribute that to the simple content and style of the CNN/Daily Mail training data. However, we found it difficult to generate substantive summaries, which we suspect is due to the widespread use of domain-specific jargon in ETDs and the wide-ranging breadth of subject matter contained in the ETD corpus. The vocabulary size for the model trained on CNN/Daily Mail data seems to be too small. The arXiv-trained model also did not have a sufficient vocabulary and was generally unable to generate coherent summaries. Alternatively, the Wikipedia model has better vocabulary coverage, but nonetheless fails to create summaries that are as natural and easy to read as the CNN/Daily Mail model. Finally, we found that models trained on a combination of data sets yield worse results than those trained on a single source. Table 2 displays examples of our machine-generate summaries.

Conclusions

The ETD corpus is an under-explored resource as a data set. Our research shows one way in which this rich corpus of graduate research and scholarship can be used to advance further research and education. By applying various abstractive summarization techniques to the task of automatically summarizing ETD chapters, we demonstrate the power of state-of-the-art deep learning technology. We address the difficulty of obtaining satisfactory training data and explore the potential of transfer learning for ETD summarization.

Gold Standard
An analysis tool was developed to make multi-threaded programming easier for students. Although the tool gave successful results in the user study, there is some room for improvement. When detecting a deadlock, the tool only identifies the line number where the threads deadlock. Willgrind could track which locks are held to indicate the exact acquisition that caused the deadlock. WillgrindPlus could be enhanced to detect atomic variables for different memory consistencies.
PGN + CNN/Daily Mail
Conclusion 7.1 Future This research has demonstrated successful results in the user study. Nonetheless particular, the deadlock detection only shows the line number where the threads deadlock. A research is required to deter- mine how to detect atomic variables at the machines code level, especially for different memory consistencies. None
PGN + Wikipedia
[UNK] is a tool used by computer programmers to measure the detection of atomic variables at the machines code level. It is based on the premise that deadlock detection can be applied to deadlock detection. It is based on the premise that deadlock detection is the problem of finding atomic variables at the machines code level.
PGN + CNN/Daily Mail + Wikipedia
Conclusion 7.1 Future Work research has demonstrated successful results in the user study. In research is required to determine how to detect atomic variables at the machines code level, especially for different memory consistencies. In research is required to determine how to detect atomic variables at the machines code level, especially for different memory consistencies.

TABLE 2

Comparing summaries of an ETD chapter, generated with a PGN trained on various data sets. Summaries based on Chapter 7 of Naciri, W. M. (2017). Bug finding methods for multithreaded student programming projects. (Master's thesis, Virginia Tech, Blacksburg, Virginia). Retrieved from http://hdl.handle.net/10919/78675

One line of future work has already begun. Baghudana (2019) has developed a research toolkit for extracting, segmenting, and summarizing text from PDF files. The toolkit provides a platform API for allowing researchers to easily plug in and swap out various segmentation and summarization modules to generate and compare summaries. Other options for future work include building more comprehensive vocabulary models for ETDs than those presented here. The use of computer vision for document layout analysis is an active area of research (*e.g.,* Grana, Serra, Manfredi, Coppi, & Cucchiara, 2016; Tran, Na, & Kim, 2015). These techniques might provide better results for extracting chapter text, figures and tables from ETD documents, but we leave that for future work.

This year, we were awarded a research grant from the U.S. Institute of Museum and

Library Services for future work aimed at bringing computational access to ETDs. Building on the work described in this paper, we will investigate how to effectively identify and extract key parts from ETDs, further develop automatic classification and summarization techniques, and pilot a more-effective digital library to better serve the information seeking needs and behaviors of our users.

Acknowledgements

This project was made possible in part by the Institute of Museum and Library Services (LG-37-19-0078-19).

References

arXiv.org. (2019). *arXiv.org e-Print archive*. Retrieved May 14, 2019, from https://arxiv.org/

Baghudana, A. (2019). *A Web Framework for Text Extraction, Segmentation, and Summarization of Long Documents*. (Unpublished master's project report). Virginia Tech, Blacksburg, Virginia.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Chen, Y.-C., & Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. *Proceedings of the 56th annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*, 675-686. Melbourne, Australia: Association for Computational Linguistics. Retrieved May 14, 2019, from https://www.aclweb.org/anthology/P18-1063

Grana, C., Serra, G., Manfredi, M., Coppi, D., & Cucchiara, R. (2016). Layout analysis and content enrichment of digitized books. *Multimedia Tools and Applications*, *75*(7), 3879-3900. https://doi.org/10.1007/s11042-014-2360-0

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in neural information processing systems (NIPS)*. Retrieved May 14, 2019, from http://arxiv.org/abs/1506.03340

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens (Ed.), *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74-81). Barcelona, Spain: Association for Computational Linguistics. Retrieved May 14, 2019, from http://www.aclweb.org/anthology/W04-1013

Lopez, P. (2009). GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. *International conference on theory and practice of digital libraries*, 473-474. Springer.

Meta. (2019). *Data dumps — meta, discussion about Wikimedia projects*. Retrieved October 2, 2018, from https://meta.wikimedia.org/wiki/Data_dumps

Peng, F., & McCallum, A. (2004). Accurate information extraction from research papers using conditional random fields. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Science Parse. (2019). *Science Parse parses scientific papers (in PDF form) and returns them in structured form*. Retrieved May 14, 2019, from https://github.com/allenai/science-parse

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *Proceedings of the 55th annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*, 1073-1083. https://doi.org/10.18653/v1/P17-1099

Sperberg-McQueen, C. M., & Bernard, L. (Eds.) (1990). *Guidelines for the encoding and interchange of machine-readable texts* (1.0 ed.). Chicago: Text Encoding Initiative.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 3104-3112). Curran Associates, Inc. Retrieved May 14, 2019, from http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

Tran, T.-A., Na, I.-S., & Kim, S.-H. (2015). Separation of Text and Non-text in Document Layout Analysis using a Recursive Filter. *KSII Transactions on Internet and Information Systems*, *9*(10), 4072-4091.