Automatic Summarization of Scientific Texts

Maria Dobko
Ukrainian Catholic University
Faculty of Applied Sciences
Lviv, Ukraine
dobko_m@ucu.edu.ua

Oleksandr Zaytsev
Ukrainian Catholic University
Faculty of Applied Sciences
Lviv, Ukraine
oleks@ucu.edu.ua

Yuriy Pryima
Ukrainian Catholic University
Faculty of Applied Sciences
Lviv, Ukraine
y.pryima@ucu.edu.ua

Abstract

In this work we overview recent advancements in the field of abstractive text summarization and discuss the use of deep learning models, especially Generative Adversarial Networks (GAN). We apply different models to our dataset of scientifix papers, acquired from arXiv, and evaluate them in terms of simplicity and performance.

We propose our own metric based on word-embeddings, which we believe will be more suitable for evaluation of abstractive summaries.

Keywords text summarization, NLP

1 Introduction

Abstractive text summarization is ... as opposed to extractive summarization where

1.1 Abstractive vs Extractive summarization

Abstractive text summarization allows us to write summaries that are almost as good as those written by humans.

Abstractive text summarization is more challenging, but it is close to what humans do.

Data-driven approach

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning[1].

1.2 The power of extractive summarization

Though it will not be the focus of our work, extractive text summarization is dominant in this field and serves as a baseline for abstractive models. Before we move any further, we must understand what makes statistical techniques so powerful.

1.3 Structure of the paper

In this study we focus on abstractive text summarization. We do a brief overview of the related work that has been done in this field. Then we proceed to discussing different models for statistical and deep summarization. Then we present our dataset of scientific papers collected from arXiv and compare the performance of different models on this dataset.

2 Related work

Allahyari et al.[1] make a survey of the most successful text summarization techniques as of July 2017.

This September Li et al. [7] described their submission to the sentiment analysis sub-task of ?Build It, Break It: The Language Edition (BIBI)? where they successfully apply generative approach to the problem of sentiment analysis.

In their paper *Generative Adversarial Network for Abstractive Text Summarization* Liu et al.[9] built an adversarial model that achieved competitive ROUGE scores with the state-of-the-art methods on CNN/Daily Mail dataset. They compare the performance of their approach with three methods, including the abstractive model, the pointer-generator coverage networks, and the abstractive deep reinforced model.

In contrast Zhang et al.[4] don't use reinforcement learning but rather introduce TextGAN with an LSTM-based generator and kernelized discrepancy metric.

3 Data collection and preparation

arXiv provides a RESTful API[1] that allows us to search for papers from a specific category and inside a specific time range. The results are returned as an HTML page which can be easily parsed.

Collecting paper IDs We started by making requests to arXiv and parsing the response to acquire unique identifiers of each paper. These are the fixed-length strings that look like this: **1801.01587**. They allow us to access everything related to this paper (including its metadata and full text as PDF). We have only collected 2000 IDs from stat.ML category. We didn't collect all the papers because of network restrictions of arXiv API, huge amount of space required to store them, and the time required to process them. Same approach can be used to collect more data, but for our problem a set of 2000 papers from one category is enough. For example, DUC (Document Understanding Conference) dataset which is among the most commonly used in the feld of text summarization has 30 sets with approximately 10 documents each[2].

Collecting abstracts Having the list of paper IDs it was no trouble to make 2000 requests to the API and collect abstracts of these papers in plain text form. To get the full text of a paper we must parse its PDF, which can introduce noise and loose peces of text. So being able to collect abstracts directly from arXiv greatly simplifies the task.

[1] https://arxiv.org/help/api/index
[2] https://www-nlpir.nist.gov/projects/duc/guidelines/2001.html

Downloading PDF files The same way as we did it with abstracts, we were able to construct URLs using paper IDs, and donload each paper using wget tool.

Extracting text from PDF Extracting text from PDF is a complicated task. We used a python binding to Apache Tika[3] to parse the PDF files. The extracted text contained many special characters (noise) which had to be removed manually.

Cleaning the text After removing special characters produced by Tika, we also had to remove all mathematical expressions because the can not be parsed into plain text and Tika just turns an expression like this $y = f(x)$ into something like $y\,f\,x$ or ytx. More complex expressions (like sums, integrals etc.) produced much more noise, and it was very hard to identify and remove it. We wanted to remove everething that is not a known English word. But that would filter out words like "GAN", "backprop" etc. So we filtered the words based on their length and frequence of vowels. That leaves some noise, but it shouldn't cause too much damage to our model (at least less damage than would be caused by removing the word "GAN"). Hopefully, we will come up with a more ellegant solution by the time of second evaluations.

As a result, we have created a database of 2000 papers. For each of these papers we store its name, a list of authors, full text without an abstract, and abstract stored in a separate column.

3.1 Examples

> Significant attention has been given to minimizing a penalized least squares criterion for estimating sparse solutions to large linear systems of equations. The penalty is responsible for inducing sparsity and the natural choice is the so-called l_0 norm. In this paper we develop a Momentumized Iterative Shrinkage Thresholding (MIST) algorithm for minimizing the resulting non-convex criterion and prove its convergence to a local minimizer. Simulations on large data sets show superior performance of the proposed method to other methods.

The title of this paper is **MIST: L0 Sparse Linear Regression with Momentum**.

4 Models for text summarization

4.1 Statistical models

4.2 Deep models

4.2.1 LSTM

4.2.2 Generative adversarial networks

Generative adversarial networks (GAN) have shown a lot of success in image generation. However until recent years they were considered inapplicable to the discrete problems of natural language processing (NLP). The latest papers introduce novel approaches to overcoming these issues by combining GANs with reinforcement learning models and lay the foundation for the whole new field of research of adversarial language processing.

Recent studies have shown that neural networks can be used for solving NLP problems. However, models that were mostly considered for this task were convolutional neural networks and recurrent neural networks.

Applying generative adversarial networks to the problems of NLP is considered to be a complicated task because GANs are only defined for real-valued data, and all NLP is based on discrete values like words, characters, or bytes.

> For example, if you output an image with a pixel value of 1.0, you can change that pixel value to 1.0001 on the next step. If you output the word "penguin", you can't change that to "penguin + .001" on the next step, because there is no such word as "penguin + .001". You have to go all the way from "penguin" to "ostrich".[4]

However, in their latest paper Fedus, Goodfellow, and Dai[3] overcome this problem by using reinforcement learning to train the generator while the discriminator is still trained via maximum likelihood and stochastic gradient descent, and use it to fill the gaps in the text.

Li et al.[6] take a different approach...

However, this approach has not been used for abstract text summarization of scientific papers, yet.

5 Evaluation metrics

Evaluating a summary is a difficult task because there does not exist an ideal summary for a given document or set of documents. [2]. Manual evaluation is too expensive, so we will only be considering the automatic techniques.

Automatic evaluation metrics compare manually written ideal summaries with summaries generated automatically by summarization systems.

The most widely used score for evaluating text summarizations is Recall-Oriented Understudy for Gisting Evaluation (ROUGE) inroduced by Chin-Yew Lin in 2004[8][2][10].

$$\text{ROUGE-N}(s) = \frac{\sum_{r \in R} \langle \Phi_n(r), \Phi_n(s) \rangle}{\sum_{r \in R} \langle \Phi_n(r), \Phi_n(r) \rangle}$$

$$\text{ROUGE-L}(s) = \frac{(1 + \beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}}$$

The problem with these scores is that all of them measure the number of words that occur in both expected and actual summary. This can be useful for extractive summarization but makes little sense for abstractive, because the summary can be generated with completely new words.

[3] https://github.com/chrismattmann/tika-python

[4] Ian Goodfellow's answer to the related question on Reddit: https://www.reddit.com/r/MachineLearning/comments/40ldq6/generative_adversarial_networks_for_text/

For example, if the expeced summary is *"The great paper"* and our model produces the summary *"A wonderful article"*, ROUGE score will be 0, even though the summary is perfect.

5.1 Embedding-based metric

We propose a metric that uses word embeddings to evaluate summaries based on their semmantic distance to the space of expected summaries.

Our assumption is that a good summary of a document contains words that are semantically close to the most important words in that document.

Let's denote an n-word summary as $S = (s_1, s_2, \ldots, s_n)$ and let $c_1, c_2, \ldots, c_k$ be the k most important words in the document. We define the norm of each word in the summary as its distance to the closest important word in the document

$$||x||_{EMB} = \min_j ||s_i - c_j||_2$$

Now the norm of a whole summary is the average of the norms of it words

$$||S||_{EMB} = \frac{1}{n} \sum_{i=1}^{n} \min_j ||s_i - c_j||_2$$

To choose which words are important we can use an algorithm of extractive text summarization. We will use **term frequency-inverse document frequency** (tf-idf) as a measure of importance because of its simplicity and importance (according to [5], tf-idf is one of the universally used terminologies in extractive summarization).

6 Experiments and results

We have tried different methods of text summarization, both extractive (TextRank, TF-IDF) and abstractive (LSTM, SeqGAN).

6.1 TextRank

6.2 Deep LSTM

6.3 GAN

7 Evaluating results

Model	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0	0	0
LSTM	0	0	0
SeqGAN	0	0	0

8 Baseline models

As a baselin model we selected a seq2seq model with deep LSTM[11] together with beam search and attention. At this point it is too hard to produce an abstract from the text of a paper, so we started with a simpler task of generating a title from the text of an abstract. We represented each absract as a numeric vector using word embedings and fed it to the model together with a corresponding title. After some training our model was able to generate meaningful titles for most abstracts. Take a look at this example:

Abstract this is great popcorn and i too have the whirly pop. the unk packs work wonderfully. i have not found it too salty or the packages leak. i have found the recent price of $35 too expensive and have purchased direct from great american for half the price.

Predicted summary great popcorn!!

Actual summary great unk american popcorn

It is not clear yet if we will be able to produce abstracts based on the whole text of an article. Such problems have very big feature space which might overcomplicate our task. So on the next stage of our project we will continue generating titles from abstracts and try doing that with discrete GANs. If our experiments prove to be successful, we will try scaling up to full texts of papers in our dataset.

9 Strength and weakness of the study

GANs have proved to be the most successful when it comes to generative images, but their application to the problems of the text generation is not well studied. So we expect our research to introduce novel approaches and original ideas that may advance the field of natural language processing. However, there are high risks that this approach may not give good results at all, because there are still lots of issues about usage of neural networks with language data. The other possible weakness is a difficulty to compare the results with other papers, as there are not a lot of researches concerning this or relative subject. We are also currently looking for supervisors who might be interested in the following topic, so we could have a mentorship during the research.

Conclusions

In this paper we demonstrated how Generative Adversarial Networks can be used for abstractive text summarization.

References

[1] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. Text summarization techniques: A brief survey, 2017.

[2] Dipanjan Das and André F. T. Martins. A survey on automatic text summarization. 2007.

[3] William Fedus, Ian Goodfellow, and Andrew M. Dai. Maskgan: Better text generation via filling in the gaps, 2018.

[4] Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation, 2017.

[5] Yogan Jaya Kumar, Ong Sing Goh, Halizah Basiron, Ngo Hea Choon, and Puspalata C Suppiah. A review on automatic text summarization approaches, 2016.

[6] Yang Li, Quan Pan, Suhang Wang, Tao Yang, and Erik Cambria. A generative model for category text generation, 2018.

[7] Yitong Li, Trevor Cohn, and Timothy Baldwin. Bibi system description: Building with cnns and breaking with deep reinforcement learning, 2017.
[8] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. July 2004.
[9] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. Generative adversarial network for abstractive text summarization, 2017.
[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. Bleu: a method for automatic evaluation of machine translation. pages 311–318, 2002.
[11] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.