CAROUGE - the new metric for text summarization

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

Summarization of scientific papers is a unique case of text summarization which allows us to use abstracts as human-created labels for our dataset because they are the summaries written by the authors. This allows us to collect great amounts of labeled data for our models to learn from.

In this work, we overview different approaches to text summarization and compare their results with different evaluation metrics. We demonstrate the downsides of ROUGE, the most commonly used summarization metric and introduce our own metric called CAROUGE, which gives more accurate scores for abstractive summaries.

We also present our new dataset of 2000 scientific papers collected from arXiv. All experiments, described in this paper are performed on the data from our dataset, except for the final stage of our project where we involve 5 human judges to do a manual summarization of several papers from the dataset.

Keywords scientific text summarization, ROUGE, seq2seq, NLP, word embeddings

1 Introduction

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

1.1 Abstractive vs extractive summarization

Extractive text summarization identifies and extracts important words, phrases, or sentences from a document and puts them together as a summary. As a result, the produced summary is a subset of the words from original summary[5]. Most work in the field of text summarization has been done around extractive summarization. Generally, it produces better results in terms of the amount of information preserved in a short summary.

Abstractive text summarization allows us to write summaries that are almost as good as those written by humans. Generally speaking, abstractive models perform worse than extractive ones. They are hard to train and often deviate too much from the original text. And since they try to describe the meaning of a text using different words, they preserve less information. Nevertheless, abstractive summarization is closer to what humans do when they write text summaries, and for that reason this is a very promissing AI topics.

1.2 Scientific text summarization

The problem of summarizing scientific texts is very different from general-purpose text summarization. Every scientific paper starts with an abstract, which by its nature is the summary of a document created by its human-author. This allows us to can create a human-labeled dataset of scientific papers just by separating abstracts from the text body.

2 Related work

Das et al. made a survey of the most successful text summarization techniques as of the year 2007[2]. Simmilar survey was made 10 years later by Allahyari et al[1].

ROUGE - the most widely used metric for text summarization was presented by Chin-Yew Lin in July 2004[7]. The metric developed by us is greatly influenced by the the work of this author. Great overview of existing evaluation metrics was also done in [2] and [1].

Fedus, Goodfellow, and Dai[3] wrote an amazing paper about their application of generative adversarial networks to the problem of filling the gaps in text. However, as we show in section 4.3, this approach can not be easily applied to our problem.

3 Data collection and preparation

Thanks to the open policy of arXiv we were able to collect our own dataset of scientific papers from *stat.ML* category[1]. We publish it together with this paper an encourage everyone to use it as they like in their own projects.

Structure of the data Each paper on arXiv has a unique identifier (for example: 1801.01587). It can be used to extract any data that is available and related to that paper, including its metadata and full text as PDF.

Our dataset contains 2000 documents (papers). Each one of them is represented by a row with the following properties:

1. arXiv's unique identifier
2. title
3. abstract
4. body of the paper

Our script[2] 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)

[1] https://arxiv.org/list/stat.ML/recent

[2] Code and instructions for collecting the data is available in our repository: https://github.com/Carouge/TextSummarization

dataset which is among the most commonly used in the feld of text summarization has 30 sets with approximately 10 documents each[3].

Cleaning the text We collected our data as PDF files and parsed them to extract the raw text. This produced a huge amount of uninterpretable UTF-8 characters from mathematical expressions. After removing those extra characters.

We wanted to remove everything that is not a known English word, but that would filter out words like "GAN", "backprop" etc. as most standard corpora of words, such as nltk.corpus, don't include specialized scientific terminology. So we filtered the words using manually created rules based on features like word length and frequency of vowels. This leaves some noise behind, but it will not affect our models.

4 Models for text summarization

Three main classes of models that are used for text summarization task are statistical frequency computation models (TFIDF etc.), graph methods (TextRank, LexRank etc.) and machine learning approach.

4.1 Statistical methods

These methods are based on the assumption that the importance of a word or sentence in a text depends on the total number of times it appears in the document. This means that this classical approach ignores context and lexical features of the text. Furthermore, they are able to perform only extractive summarization.

4.2 Graph models

We can build a graph of each document where words or sentences are nodes and the edges are the connections between each pair of nodes. The weights on these edges represent similarity between words or sentences in the whole text. While proving to have better results than simple frequency-based methods, graph models are still bounded by the absence of lexical understanding and ability to perform extractive summary only.

4.3 Machine learning approach

In the last years, lots of attention was focused on learning how to apply neural networks to NLP tasks, including text summarization. Using encoder-decoder models it is now possible to produce abstract summaries. In [10] the off-the-shelf attentional encoder-decoder RNN that was originally developed for machine translation was applied to summarization and outperformed state-of-the-art systems on two different English corpora. However, there is not much information about neural networks usage in scientific text summarization.

Generative adversarial networks (GAN) are another promising approach for text summarization. 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.

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

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. This is an amazing result that can become an inspiration for others to develop GAN-based solutions for the problems of text sumarization. Nevertheless, summarization is way more complex than the problem of filling the gaps, and therefore, to our knowledge, there are no successful examples of applying adversarial networs to problems similar to ours. We ourselves have tried this new approach, but failed to produce good results.

5 Evaluation metrics

Evaluating a summary is a difficult task because there is no such thing as a single summary that would be ideal for a given document. In most cases even human evaluators can not agree on which of the given summaries is better[2]. Unlike other NLP problems, such as translation or parsing, when it comes to text summarization we can not clearly define what makes a summary good or bad. Therefore we must make assumptions about the space of good summaries.

1. assume that a good summary would be close to some *ideal* summary manually created by humans
2. assume that the goodness of summary can be measured as the amount of important information it contains (this assumption can be inferred from the definition of text summarization).

In the following sections we describe one commonly used summarization metric that is based on the first assumption and propose our own metric that is based on second one (we will show that the proposed metric can be formulated in a different way to work with the first assumption).

5.1 ROUGE

The most widely used score for evaluating text summarizations is ROUGE (Recall-Oriented Understudy for Gisting Evaluation) introduced by Chin-Yew Lin in 2004[7].

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

.

Here $\Phi_n(d)$ is a binary vector representing the n-grams contained in document d, s is the generated summary and r is the human-created summary. In simple terms, ROUGE-N is a fraction q/k, where q is the number of n-grams that are

[3] https://www-nlpir.nist.gov/projects/duc/guidelines/2001.html

present in both s and r, and k is the total number of n-grams in r.

ROUGE works well for extractive text summarization. But if we need to evaluate the generated summary which contains different words from the ones that occurred in paper, the score will always be small because, even though the new words can be close to the expected ones, two summaries don't overlap in terms of word equality.

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

5.2 CAROUGE

We propose a metric that uses word embeddings to evaluate summaries based on their semantic distance to the space of good summaries. Our assumption is that a good summary of a document contains words that are semantically close to the most important words or n-grams in that document.

Let s be the generated summary. If $|s|$ is the number of words in summary s, then $m = |s| - n + 1$ is the number of n-grams in this summary. Let $s_1, s_2, \ldots, s_m$ be all n-grams of summary s and let $c_1, c_2, \ldots, c_k$ be the k most important n-grams in the document. As we will show in section 5.2.1, there are many ways of measuring the importance of n-grams in a document. The definition of n-gram importance is closely related to the two base assumptions that were mentioned at the beginning of section 5.

The metric we propose is in fact a continuous version of ROUGE-N. Instead of testing the equality of n-grams in the compared summaries we use the continuous measure of semantic distance between those n-grams.

For each n-gram in the generated summary we calculate the embedding-based score as its distance to the closest important n-gram in the document[4].

$$\text{CAROUGE-N}(s_i) = \frac{1 - \min_j ||s_i - c_j||}{k}$$

Now we define the score of the whole summary as the average score of its words

$$\text{CAROUGE-N}(s) = \frac{1}{n} \sum_{i=1}^{n} \frac{1 - \min_j ||s_i - c_j||}{k}$$

5.2.1 Measuring word importance

Deciding which words or sentences are important in a piece of text is part of the extractive summarization problem. Therefore, one way to choose k important words would be to use part of a simple extractive model. For example, we could use tf-idf (term frequency-inverse document frequency) which would assign the highest scores to the words (n-grams) which are very frequent in the given document and very unfrequent in other documents.

Another way of choosing important words would be to follow the assumption of ROUGE, according to which a good summary should be as close as possible to the human-created summaries. This means that we will compare generated summaries to the corresponding paper abstract (which are in fact author-created summaries).

Using the same example as in the section 5.1 we can see that CAROUGE score of a decent abstractive summary will be greater than 0.

$$r = \text{"The great paper"}$$

$$s = \text{"A wonderful article"}$$

$$\text{ROUGE}(s) = 0$$

$$\text{CAROUGE}(s) = 0.8956$$

6 Experiments and results

We have tried different methods of text summarization, both extractive (TextRank, RAKE) and abstractive (Seq2Seq, SeqGAN).

6.1 RAKE

Rapid Automatic Keyword Extraction algorithm (RAKE)[11] is a keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurrance with other words in the text. Its main advantages include time efficiency, operating on individual documents. It can be easily applied to new domains and multiple types of documents. RAKE is very sensitive to unclean data, as it relies on word frequency.

Table 1. Example of an abstract generated by RAKE

Title	The Multivariate Generalised von Mises distribution: Inference and applications
Real abstract	...Previously proposed multivariate circular distributions are shown to be special cases of this construction. Second, we introduce a new probabilistic model for circular regression, that is inspired by Gaussian Processes, and a method for probabilistic principal component analysis with circular hidden variables...
Generated abstract	...many data modelling problems since higher order generalised von mises distributions model circular variables using distributional assumptions probabilistic principal component analysis ppca proposed resulting distribution inherits desirable characteristics paper makes three technical contribu tions multivariate generalised...

[4] CAROUGE stands for Continuous Abstractive ROUGE. The French word **carouge** means blackbird

6.2 TextRank

In this research we used a basic model TextRank as a baseline for comparison. This is a graph method, influenced by PageRank algorithm, that represents the documents as a connected graph[9]. We trained TextRank separately for each body of the paper and produced an extractive abstract-length summary. Scores were calculated on the basis of ROUGE score. Unlike RAKE which operates n-grams, TextRank generates summaries using whole sentences. For that reason the generated summaries are more readable.

Table 2. Example of an abstract generated by TextRank

Title	Churn Prediction in Mobile Social Games: Towards a Complete Assessment Using Survival Ensembles
Real abstract	...for each player, we predict the probability of churning as function of time, which permits to distinguish various levels of loyalty profiles ... Our results show that churn prediction by survival ensembles significantly improves the accuracy and robustness of traditional analyses, like Cox regression...
Generated abstract	Conditional inference survival ensembles are constructed based on unbiased trees avoiding this problem the resulting prediction of this model contains for each player a survival function indicating the probability of churn as a function of time since the registration in the game.

6.3 Seq2Seq

We have used deep LSTM seq2seq model with attention for a task of text summarization. Seq2seq have proven to provide state-of-art result in tasks of sequence generation. 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 As input model takes paper abstract converted to the vectorized representation using word embeddings. The input sequence is limited by 600 words. All abstracts that are bigger than limit are omitted. All smaller abstracts are padded with ⟨SOS⟩ word that represents the end of a sequence. Model outputs sequence derived from the probability distribution. Each output word samples from this distribution having input sequence and previously generated samples. Output sequence is limited by 30 words. First ⟨SOS⟩ word represent the end of a generated summary.

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

7 Evaluating results

In the table below you can see the results of our experiments represented by mean scores of several models that were used to generate abstracts for same 2000 papers from our dataset given their text bodies.

Table 3. Evaluating summarization algorithms

Model	ROUGE-1	ROUGE-2	CAROUGE-1
RAKE	0.08	0.02	0.82
TextRank	0.18	0.04	0.89

We have also involved some human judges and asked them to come up with titles for 5 papers from our dataset, given their abstracts.

Table 4. Evaluating human judges

Model	ROUGE-1	ROUGE-2	CAROUGE-1
Human	0.77	0.43	0.98

Conclusions

We have discussed different approaches to scientific text summarization and evaluated them with both ROUGE and CAROUGE (Continuous Abstractive ROUGE) - a new metric proposed by us that uses word embeddings to produce more accurate scores of abstractive summaries.

As we have seen in section 7, both ROUGE and CAROUGE metrics give very high scores to human-created summaries, and much lower scores for the summaries produced by algorithms. However, considering the fact that ROUGE assumes human-judgement to be perfect, the score of 0.77 is way too low. As we explained in section 5.1, this happens because human judgement is abstractive by its nature, and ROUGE score words best on extractive summaries. This problem is fixed by our metric, which gives the score of 0.98 on the same data.

Acknowledgements

We would like to express our gratitude to Anatolii Stehnii, Artem Chernodub, and Maryana Romanyshyn for helping us with their valuable pieces of advice. Also, we are very grateful to Rostyslav Zatserkovnyi, Anton Tarasov, Ivan Ilnytskyi, Taras Panychok, and Bohdan Petryshak for providing their manually created summaries of some papers from our dataset.

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] Yitong Li, Trevor Cohn, and Timothy Baldwin. Bibi system description: Building with cnns and breaking with deep reinforcement learning, 2017.

[7] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. July 2004.

[8] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. Generative adversarial network for abstractive text summarization, 2017.

[9] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[10] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023, 2016.

[11] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. *Automatic Keyword Extraction from Individual Documents*. 03 2010.