Unsupervised Learning for Automatic Title Generation of Scientific Articles

MINOR PROJECT SYNOPSIS

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

Abdul Waheed
00196407218

Nimisha Mittal
41696402717

Muskan Goyal
35496402717

Guided by
Dr. Deepak Gupta

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)

Index

1 Introduction 2

1.1 Why Unsupervised Learning? 2

1.2 Dataset 3

1.3 Why do we need a domain-specific pipeline? 3

1.4 Language and Tools 4

2 Objective 4

4 Methodology 5

4.1 Pre-processing 5

4.2 Training 5

4.3 Title Generation 6

5 Research Gap Identified 6

5.1 Supervised V/S Unsupervised Learning 6

5.2 Using Pre-trained Language Models with unsupervised learning 7

6 Bibliography 8

1 Introduction

Unsupervised learning could be very crucial when there’s very limited or no labelled data. In this study we aim to attempt towards unsupervised automatic title generation that relies only on unlabeled text corpora. A proper title for a given block of text must express the core meanings of the text in a concise manner. Even a strong title needs to be memorable. Usually, authors will go through several rounds of revisions to get a suitable title. Automatic title generation (ATG) for a given research paper abstract aims at generating a title comparable to a human title [9].

Early title generation methods include Naive Bayesian with limited vocabulary, Naive Bayesian with full vocabulary, Term frequency and inverse document frequency (TF-IDF), K-nearest neighbor, and Iterative Expectation Maximization. These methods, however, only generate an unordered set of keywords as a title without concerning syntax. Using the F1 score metric as a base for comparisons, it was shown through extensive experiments that the TF-IDF title generation method has the best performance over the other five methods [9].

In this study we aim to investigate and build neural network based end-to-end novel title generation model. The advantage of end-to-end model over modular approach is that it requires very little to no feature engineering whatsoever and is easy to train and most of the time it gives way better performance than modular approaches or hand crafted rule based systems. We also aim to study to train our model in an un/self supervised manner which is a very challenging task.

1.1 Why Unsupervised Learning?

Supervised learning is the cornerstone of most recent machine learning progress. However, it can require large, carefully cleaned, and expensive to create datasets to work well. Unsupervised learning is desirable due to its potential for overcoming these disadvantages [13]. Since unsupervised learning removes the bottleneck of explicit human marking, it also correlates well with current trends in-computation and raw data availability. Uncontrolled learning is a very active research field but its practical applications are still still minimal.

A recent move has been made to try and increase language skills through using unsupervised learning and increase systems with vast volumes of unlabeled data; representations of words learned by unsupervised techniques may use huge data sets consisting of information terabytes and, when combined with supervised learning, enhance output on a broad range of NLP tasks. Until recently these unsupervised NLP techniques (e.g., GLoVe and word2vec) used basic models (word vectors) and training signals (local word co-occurrence). Skip-Thought Vectors is a notable early example of the possible changes which could be realized by more complex approaches. But new techniques are now being used which are further boosting performance. These include the use of pre-trained sentence representation models, contextualized word vectors (notably ELMo and CoVE), and approaches which use customized architectures to fuse unsupervised pre-training with supervised fine-tuning, like our own.

1.2 Dataset

We aim to train and evaluate title generation model on recently released arXiv data. It is a machine-readable arXiv dataset: a repository of 1.7 million scientific articles, with relevant features such as article titles, authors, categories, abstracts, full text PDFs, and more [14].

Metadata : This dataset is a mirror of the original ArXiv data. Because the full dataset is rather large (1.1TB and growing), this dataset provides only a metadata file in the json format. This file contains an entry for each paper, containing [14]:

id: ArXiv ID (can be used to access the paper)

submitter: Who submitted the paper

authors: Authors of the paper

title: Title of the paper

comments: Additional info, such as number of pages and figures

journal-ref: Information about the journal the paper was published in

doi: [https://www.doi.org](Digital Object Identifier)

abstract: The abstract of the paper

versions: A version history

categories: 90848 Categories / tags in the ArXiv system

examples : astro-ph: Astrophysics

cond-mat.dis-nn: Disordered Systems and Neural Networks

cond-mat.mes-hall: Mesoscale and Nanoscale Physics

astro-ph.CO: Cosmology and Nongalaic Astrophysics

cs.AI: Artificial Intelligence

cs.CC: Computational Complexity

cs.CL: Computation and Language

cs.CV: Computer Vision and Pattern Recognition

cs.DS: Data Structures and Algorithms

1.3 Why do we need a domain-specific pipeline?

Domain control or often called adaptation is another important research field in natural language processing. The arXiv data that we described earlier has a category label corresponding to each paper. Controlling the domain in an unsupervised manner is also subject to research in this study.

By controlling the domain, title tokens will be sampled from a local distribution of the domain specific vocabulary (which will be a subset of global vocabulary) rather than global vocabulary, in this way the generated title will be catchy and closely related to its associated abstract.
Task agnostic language models are trained on large common language corpus (books, articles, forums, Wikipedia, news articles). As such, they are very suitable for applications in the same domains as they are trained on language and context that can be expected to be understood by people with a regular high school education or the average reader of a newspaper.

Some document types however, contain specialized language and an industry-specific vocabulary, giving rise to the need for training custom word embeddings.

1.4 Language and Tools

Python, Google Colab, PyTorch

2 Objective

1. To generate the title of the articles depicting the central meaning of the text.
2. To generate the title that can be converted to actual title and used by the authors without much effort.
3. To generate the title that can be comparable to titles generated by human writers.
4. The title generated must be domain-specific and have higher similarity to the idea of the article.

3 Feasibility Study

Our approach relies on text corpora with limited labels to generate titles of scientific article abstracts provided. The crux of the (unsupervised) auto-encoding framework is a shared encoder and a pair of attention-based decoders. The encoder attempts to produce semantics preserving representations which can be acted upon by the respective decoders (simple or complex) to generate the appropriate text output (title) they are designed for.

This approach generates a title for the scientific articles by implementing the Seq2Seq (Recurrent and transformer based models) approach. The sequential data will be passed to the encoder producing context vectors which will be fed to the decoder that will act as a language model. We further compute similarities of the generated titles to the existing title if provided to select the title with higher similarities, otherwise are compared to each other to select the appropriate title.

4 Methodology

In this study, we address title generation of the scientific papers and their domain control using abstract and related metadata provided and intend to use unsupervised learning for the same. Our proposed architecture consists of the following modules: pre-processing, training and generating final titles for the abstracts. The following subsections describe each module.

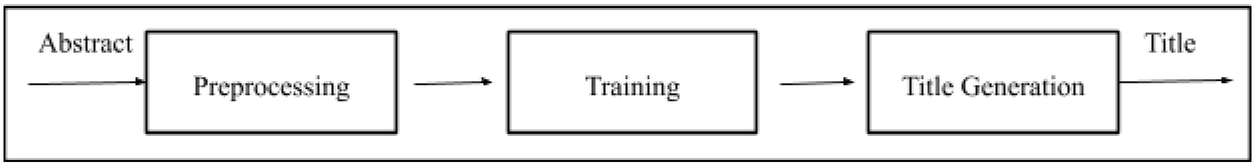

Fig 1: Block diagram of the methodology

4.1 Pre-processing

The abstract is used as input because it is sufficient to represent the research idea in a short manner. Tokenization and stop word removal is used for pre-processing.

4.2 Training

For training, we will be using the sequence-to-sequence (Recurrent based and transformer based models) approach. A sequence-to-sequence model is a typical abstract summarization approach that can map one long sentence (article) to another short sentence (summarization). Figure 2 illustrates the encoder and decoder of the sequence-to-sequence model

Recently, [17] proposed a sequence-to-sequence model [18] with a copying mechanism for keyphrase generation, which is able to produce phrases that are not in the input documents. While the seq2seq model demonstrates good performance on keyphrase generation [17], it heavily relies on massive amounts of labeled data for model training, which is often unavailable for new domains. To overcome this drawback, in this work, we investigate unsupervised learning by leveraging abundant unlabeled corpus.

In our proposed method, the pre-processed data will be passed to the encoder in the form of sequential data producing context vectors which will then be fed to the decoder that will act as a language model generating multiple candidate titles.

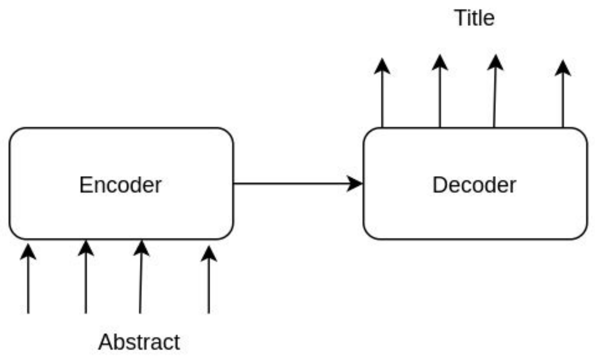

Fig 2: Block diagram of simple encoder-decoder model for sequence-to-sequence mapping. The encoder module takes a text sequence (abstract in this case) and squeezes it into a semantic representation of the input sequence called context vector and this is passed to the decoder which is a language model (next word prediction), which produces a title given context vector.

4.3 Title Generation

The multiple candidates for title generated after training are further compared with the existing title provided in the dataset to select a title with higher similarities. They are compared by calculating cross entropy loss between generated and existing titles.

5 Research Gap Identified

5.1 Supervised V/S Unsupervised Learning

A Supervised learning model is used to make predictions based on information about the targets and the features of data. It infers a function according to a given set of input-output data respectively. Normally, the input data provides a set of observations with which the computer is trained [10]. Each observation consists of an input vector and a desired output value. A supervised learning model trains the data and generates a general rule or function to be used for predicting or classifying new inputs.

In supervised machine learning [15] [16], a batch of text documents are tagged or annotated with examples of what the machine should look for and how it should interpret that aspect. These documents are used to “train” a statistical model, which is then given un-tagged text to analyze. Later, you can use larger or better datasets to retrain the model as it learns more about the documents it analyzes. For example, you can use supervised learning to train a model to analyze movie reviews, and then later train it to factor in the reviewer’s star rating.

The most popular supervised NLP machine learning algorithms are:

1. Support Vector Machines
2. Bayesian Networks
3. Maximum Entropy
4. Conditional Random Field
5. Neural Networks/Deep Learning

Supervised learning is at the core of most of the recent success of machine learning. However, it can require large, carefully cleaned, and expensive to create datasets to work well. Therefore, to address these drawbacks we are going to use unsupervised learning.

Unsupervised machine learning involves training a model without pre-tagging or annotating. Some of these techniques include clustering, latent semantic indexing (LSI), matrix factorization. Clustering means grouping similar documents together into groups or sets. These clusters are then sorted based on importance and relevance (hierarchical clustering). Another type of unsupervised learning is Latent Semantic Indexing (LSI). This technique identifies words and phrases that frequently occur with each other. Data scientists use LSI for faceted searches, or for returning search results that aren't the exact search term. For example, the terms "manifold" and "exhaust" are closely related documents that discuss internal combustion engines. So, when you Google "manifold" you get results that also contain "exhaust". Matrix Factorization is another technique for unsupervised NLP machine learning. This uses "latent factors" to break a large matrix down into the combination of two smaller matrices. Latent factors are similarities between the items. Think about the sentence, "I threw the ball over the mountain." The word "threw" is more likely to be associated with "ball" than with "mountain". In fact, humans have a natural ability to understand the factors that make something throwable. But a machine learning NLP algorithm must be taught this difference. Unsupervised learning is tricky, but far less labor and data-intensive than its supervised counterpart. [12]

5.2 Using Pre-trained Language Models with unsupervised learning

Pre-trained Language Model(LM) is one of the most important research advances in Natural Language Processing(NLP) which focuses on how to make use of language information in large corpus with unsupervised learning. Word2vec [1] and Glove [2] have successfully learned semantic information in word embeddings and have been widely used in NLP tasks as inputs for models. Pre-trained language models explore more by learning syntactic and more abstractive features. These language models enrich embeddings information by adding encoders in pre-trained parts, producing context-aware representations when transfer to downstream tasks. Representative works including ULMFiT [3], which captures general features of the language in different encoder layers to help text classification; ELMo [4], which learns embeddings from Bidirectional LSTM language models; BERT [5], a successful application of training Transformer encoders on large masked corpus and reaching eleven state-of the-art results. After the release of BERT, many super-large-scale Transformer-Based models have been raised including GPT-2 [6], MASS [7] and XLNet [8].

6 Bibliography

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. pages 3111–3119.

[2] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pages 1532–1543.

[3] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 .

[4] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365 .

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[6] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAIBlog 1(8)

[7] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and TieYan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. arXiv preprint arXiv:1905.02450 .

[8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237 .

[9] Shao, Liqun and Jie Wang. “DTATG: An Automatic Title Generator based on Dependency Trees.” KDIR (2016).

[10] Mathivanan, Norsyela Muhammad Noor et al. “Performance Analysis of Supervised Learning Models for Product Title Classification.” IAES International Journal of Artificial Intelligence 8 (2019): 228-236.

[11] Chen, Francine R. and Yan-Ying Chen. “Adversarial Domain Adaptation Using Artificial Titles for Abstractive Title Generation.” ACL (2019).

[12] https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing

[13]https://openai.com/blog/language-unsupervised/#:~:text=Unsupervised%20learning%20is%20a%20very,it%20are%20often%20still%20limited.&text=Until%20recently%2C%20these%20unsupervised%20techniques,co%2Doccurence%20of%20words).

[14] https://www.kaggle.com/Cornell-University/arxiv

[15] Qader, R. et al. “Semi-Supervised Neural Text Generation by Joint Learning of Natural Language Generation and Natural Language Understanding Models.” INLG (2019).

[16] Putra, Jan Wira Gotama and Masayu Leylia Khodra. “Automatic Title Generation in Scientific Articles for Authorship Assistance: A Summarization Approach.” Journal of ICT Research and Applications 11 (2017): 253-267.

[17] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In Proceedings of the 55th Annual Meeting of the Association for Computa- tional Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 582–592.

[18] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural net- works. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural In- formation Processing Systems 2014, December 8- 13 2014, Montreal, Quebec, Canada, pages 3104– 3112.