Noise Removal From Webpage Data Using Encoder-Decoder Network

Yash Agarwal, Jonathan Song, Andy Jiang

1 Introduction

Clean webpage data plays an increasingly vital role in various domains, including web scraping, sentiment analysis, and academic research. In web scraping, where data extraction from websites is used for applications like e-commerce and market research, the quality of data is very important. Noise and inconsistencies in the data can lead to inaccurate insights and important business decisions based on unreliable information. Content recommendation systems, used in e-commerce, entertainment, and news websites, also rely on clean data to understand user behavior more effectively and provide personalized recommendations. In the domain of sentiment analysis, the accuracy of determining sentiment expressed in textual data depends on clean data free from irrelevant text, advertisements, or other forms of noise. Clean data ensures more reliable insights into public opinion, customer feedback, and market trends, serving applications such as brand reputation management and market research. Lastly, in academic and research settings, reliable and clean data from web pages is crucial for conducting research that is both trustworthy and influential, ensuring results aren't skewed. Across various fields, the integrity of clean webpage data is essential for the success and efficacy of applications and research initiatives that impact hundreds of millions daily.

Al Sharou et. al. (2021) provide a comprehensive commentary on various forms of noise that could arise within data sourced from webpage-like contexts, such as syntactical errors (grammar mistakes, unusual spellings), language variability (slang, multiple languages), and HTML tags or markup. Other causes of noisy webpage data include user-generated content (comments, reviews, forum posts, etc), advertisements, and content redundancy.

We aim to build a noise filtering model that extracts relevant features from text while being robust to these various forms of noise. Our goal is to show that features created using denoising process perform better in downstream tasks (e.g. classification, sentiment analysis etc.) compared to the noisy webpage data.

2 Related Work

2.1 Webpage data related work

The specific problem of denoising webpage data is sparsely studied due to the complexity of web page data, which is often cluttered with advertisements, structural components such as navigation tabs and buttons, and other irrelevant information within the document markup. Much of the existing literature on denoising webpage data falls under the scope of extracting textual data from webpages for the purposes of data mining. These approaches for the most part take a rule-based form: exploiting the fact that HTML pages are structured as a Document Object Model (DOM) tree of parent and child elements and developing rules that determine which elements are and are not relevant.

Yi et. al. (2003) posit that analyzing the DOM tree of a single webpages is often not sufficient for extracting relevant elements since, without context, it is difficult to judge what text forms the main content of the page. They observe that similar webpages e.g. from the same domain are likely to have similar or identical stylistic/structural elements while the content elements are more likely to be dissimilar. Thus, they develop an algorithm that uses this theory to mark HTML elements containing content.

Garg and Kaur (2014) compare using a DOM-based segmentation approach and an Least Recently Used (LRU) based paging approach. This consists of removing least recently used pages within the webpage to process web page data. However, the LRU approach is prone to removing good

content while keeping noisy content since it is not certain that less used webpages contain less relevant content.

The primarily shortcoming of DOM-tree based approaches are their computational cost. Webpages may consist of large nested tree structures: using tree-traversal algorithms such as those aforementioned quickly becomes expensive when we consider they must be applied to every page in the dataset individually.

Aside from tree based algorithms, Shen et al. (2007) use computational webpage summarization techniques to remove noise from data for use in downstream tasks. Unfortunately, such techniques, again, are computationally expensive and fail to utilize the capabilities of powerful large language models (LLMs), such as GPT, to capture relevant document features.

2.2 General denoising and error correction methods

Denoising autoencoders have been explored extensively in fields such as computer vision and natural language processing (NLP). Vincent et. al. (2008) explore denoising autoencoders and their potential in learning robust features from potentially corrupted inputs. However, their proposed model does not target any specific task; in particular, it would underperform with webpage data coming in an HTML/XML format due to the abundance of tags and meta data.

Progress has also been made in the field of grammatical error correction (GEC), especially given the popularity of the CoNLL-2014 shared metric. Chollampatt and Ng (2017) formulate the task of GEC as one of machine translation from erroneous to correct text and demonstrate promising results using state-of-the-art statistical machine translation techniques. Chollampatt and Ng (2018) then best those results using a convolutional encoder-decoder model incorporating an attention mechanism.

Junczys-Dowmunt et. al. (2018) propose that the more accurate formulation of GEC is that of *low-resource* machine translation from incorrect to correct text given the sparsity of grammatical error data. They adapt techniques utilized in low-resource neural machine translation such as domain adaptation, transfer learning, ensembling of multiple models, and experiment with various deep neural model architectures, demonstrating better performance over statistical methods as well as previous neural GEC models. More recently, transformer based sequence-to-sequence solutions have achieved remarkable results on the GEC task.

We build on this idea of utilizing sequence-to-sequence models to denoise erroneous text to clean text, with the added goal of generating rich representations during the encoding-decoding process. Additionally, our model seeks to be more robust, not only handling grammatical errors but structural noise as well (which we define in Section 3.1). Finally, our model improves upon classical webpage denoising techniques in its computational efficiency — being scalable without the bottleneck of manually searching through each individual webpage.

3 Methodology

We construct a **noise filtering** model that takes in noisy input data and extracts **denoised features** using an **encoder-decoder** architecture. We then use these denoised features in a variety of downstream tasks to gauge whether features produced using our extraction mechanism perform better than features produced using conventional frameworks without explicit denoising.

3.1 Data

We use both publicly available Wikipedia and Common Crawl datasets to train our models on. The strength of these datasets lies in the comprehensive nature of their material: there exists data from a wide variety of authors, in different writing styles, and on diverse topics.

The Wikipedia dataset [1] consists of raw dumps from Wikipedia, containing markdown, metatextual data, and another irrelevant material. This dataset will be paired with cleaned Wikipedia data [2] that has all of these materials removed. Similarly, the Common Crawl dataset [3] contains scraped data from billions of webpages. The corresponding clean data will be sourced from C4 [4], a cleaned, filtered version of Common Crawl. These noisy datasets contain various forms of **structural noise** which we define to be noise outside of the content itself, such as HTML markup, advertisements, metadata etc. These datasets will be represented in the form of large strings of text.

[1] https://dumps.wikimedia.org
[2] https://www.tensorflow.org/datasets/catalog/wikipedia
[3] https://commoncrawl.org/
[4] https://www.tensorflow.org/datasets/catalog/c4

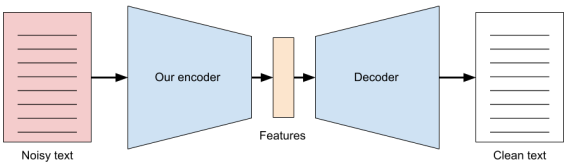

Figure 1: Encoder-Decoder Network Architecture

Additionally, to account for other forms of syntactic noise, such as grammatical errors and typos, and semantic noise, such as irrelevant words and sentences, which may or may not be present in the noisy datasets, we develop a **synthetic noise** algorithm to add noise of this manner to our datasets.

We utilize a 70/15/15 split between our training, testing, and validation data, using 7000 webpages in the training data, 1500 in the testing data, and 1500 in the validation data.

3.2 Synthetic Noise Algorithm

As seen in Raffel et al. (2020), some of the various forms of synthetic noise that exist in modern webpages are syntactical errors (grammar mistakes, unusual spellings), language variability (slang, multiple languages, profanity), citation markers, and "lorem ipsum" placeholder tags. In addition, some other types of noise are HTML tags, user-generated content (comments, reviews, forum posts, etc), citation advertisements, and content redundancy. We insert:

- syntactical errors from the GitHub Typo Corpus from Hagiwara et al.
- slang and profanity from the "List of Dirty, Naughty, Obscene, or Otherwise Bad Words" [5]
- Spanish, French, and German words from their corresponding online dictionaries

3.3 Encoder-Decoder Models

We build a transformer-based encoder-decoder model for the purpose of producing denoised features, as shown in Figure 1.

Training such models from scratch is prohibitively expensive given the large number of parameters (110M parameters in standard BERT[6]). Therefore, to make training more efficient, we warm-start from existing pre-trained models, and fine-tune for our specific task at hand. Work by Rothe et. al. (2020) has demonstrated the effectiveness of constructing encoder-decoders for sequence-to-sequence tasks by initializing encoder parameters from models such as BERT and decoder parameters from models such as GPT. These encoder-decoders achieve state-of-the-art performance on sequence-to-sequence tasks such as summarization and machine translation despite their component models being pre-trained in an unsupervised manner.

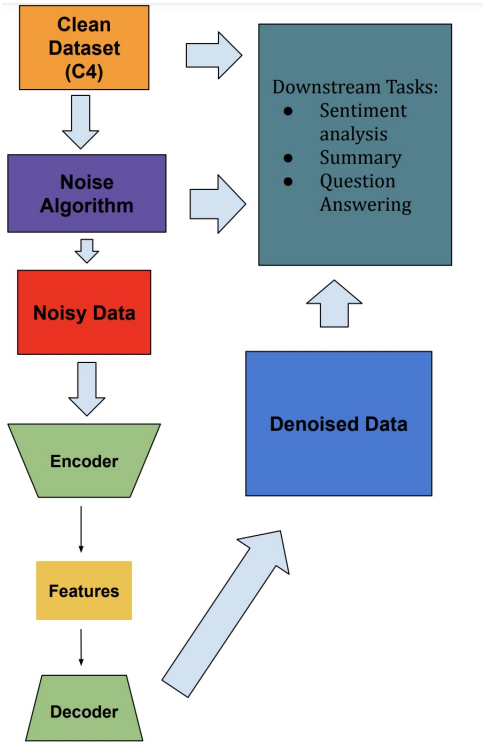

Figure 2: Project Pipeline

We source the architecture for our encoder-decoder model from HuggingFace's EncoderDecoderModel [7]. The model used in Rothe et. al. HuggingFace also provides standard pre-trained models such as BERT, GPT etc. Experimentation suggests combinations of encoders and decoders that perform the best on the seq2seq tasks of translation and summarization, which we seek to evaluate our webpage features on, involve BERT- and RoBERTa-compatible encoders and decoders. We also consider BART, an architecture from Lewis et. al. (2019) specifically pre-trained to produce

[5] https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

[6] https://huggingface.co/transformers/v2.9.1/pretrained_models.html

[7] https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/encoder-decoder

	GLUE	CNNDM	SQuAD
Common Crawl			
C4			
Encoder Decoder Denoised			

Table 1: Performance of our denoised data and raw data on downstream tasks

original text from corrupted text [8].

Our encoder-decoder models are trained using teacher forcing, with pairs of noisy text as input and their corresponding cleaned text as expected output. Our encoder extracts features from the noisy text; these features are used as input to the decoder, whose output is evaluated on similarity to the actual clean text. Aside from forcing the decoder to produce clean text, our paradigm encourages the features to also go through a denoising process in the encoder itself. We use features from the models that attain the best similarity metrics as then evaluate their performance on downstream tasks compared to baselines.

3.4 Performance on Downstream Tasks

The last part of the pipeline consists of our method for evaluating our model's decoded text. We test the performance of our denoised webpage data against Common Crawl webpages on downstream tasks such as classification and sentiment analysis. We measure the performance using the General Language Understanding Evaluation (GLUE) score, ROUGE and METEOR evalation on the CNN / Daily Mail (CNNDM) dataset, and accuracy score on the Stanford Question Answering (SQuAD) dataset, which measure the quality of sentiment analysis, summarization skills, and question-answering ability, respectively. We hope that the variety of downstream tasks produce a more comprehensive model that can encapsulate the meaning of the text, or its key features, in its decoded text.

The GLUE score, introduced by Wang et al. (2019) is a composite score consisting of 9 different NLP tasks, many of which are related to sentiment analysis, inference, and question answering. The CNNDM dataset is useful for evaluating summaries, which are evaluated based on coherency and content. The SQuAD dataset consists of question-answer pairings from Wikipedia articles, and the SQuAD score consists of two primary metrics: exact match with a human annotated answer and F1 score, which considers both precision and recall.

The overall pipeline is shown in Figure 2.

[8] https://huggingface.co/docs/transformers/model_doc/bart

4 Results

We measure the quality of our denoised features by their performance on downstream tasks versus features produced from existing embedding models. Our baselines for this evaluation are established encoding frameworks such as Word2Vec and BERT. These frameworks will be used to extract feature embeddings from the noisy data without seeing any of the cleaned data. From there, we can compare the usefulness of these features with our denoised features on other tasks. Our novel encoder produces more utilitarian and robust features when given noisy data.

5 References

Khetam Al Sharou, Zhenhao Li, and Lucia Specia. 2021. Towards a Better Understanding of Noise in Natural Language Processing. In *Proceedings of Recent Advances in Natural Language Processing*.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Anchal Garg and Bikrampal Kaur. 2014. Web Page Performance Enhancement by Removing Noise. *International Journal of Computer Applications*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha and Kenneth Heafield. 2018. Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task. In *Proceedings of North American Chapter of the Association for Computational Linguistics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer

Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Shahan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Sascha Rothe, Shashi Narayan and Aliaksei Severyn. 2020. Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. In *Transactions of the Association for Computational Linguistics*.

Dou Shen, Qiang Yang, and Zheng Chen. 2007. Noise reduction through summarization for Web-page classification. *Information Processing and Management*.

Yifu Sun and Haoming Jiang. 2019. Contextual Text Denoising with Masked Language Models. *arXiv preprint, arXiv:1910.14080*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*

Ziang Xie, Guillaume Genthal, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*.

Lan Yi, Bing Liu, and Xiaoli Li. Eliminating Noisy Information in Web Pages for Data Mining. 2003. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.