Economics 2355: NLP with Noisy Text

Harvard University

March 2021

Outline

The Canonical Deep NLP Training Corpus

A Definition of Noise

The Problem with Noise

Approaches to Denoising for Deep NLP

Final Words

What are SOTA Deep NLP Models Trained On?

- As discussed in Lecture 16, BERT was trained on 16 GB of BookCorpus and Wikipedia data
  - BookCorpus is 11,038 books from the web; English Wikipedia is, well, English wikipedia!
  - Devlin et al. (2018): “For Wikipedia we extract only the text passages and ignore lists, tables, and headers”

What are SOTA Deep NLP Models Trained On?

- Also as discussed in Lecture 16, RoBERTa adds 144GB of additional data, which comes from the Common Crawl News Dataset (63 million articles; 76 GB), the OpenWebText corpus (38 GB), and STORIES from Common Crawl (31 GB)
  - Common Crawl News contains news articles published online as identified by Google News sitemaps
  - OpenWebText is based on a “web scrape which emphasizes document quality. To do this we only scraped web pages which have been curated/filtered by humans”
  - STORIES is a subset of common crawl where “the constituent documents take the form of a story with long chain of coherent events”

What are SOTA Deep NLP Models Trained On?

- The OpenWebText corpus is actually an open source recreation of the training corpus for GPT-2 (“WebText”)
- ALBERT and DistilBERT both use the same training corpus as BERT
- T5 is pre-trained on the “Colossal Clean Crawled Corpus” containing about 750 GB of “reasonably clean and natural English text”

What are SOTA Deep NLP Models Trained On?

- The takeaway is clear: all SOTA deep NLP models are trained on some individual or set of cleaned, high quality text corpuses! These are texts comprised mostly of full sentences, with few misspellings, and with standard english grammar
- In the social sciences, however, the texts we are interested in working with are usually not very clean–they are “noisy,” and this requires some extra care and attention when building our data curation pipelines

Outline

The Canonical Deep NLP Training Corpus

A Definition of Noise

The Problem with Noise

Approaches to Denoising for Deep NLP

Final Words

What Is Noisy Text, Exactly?

- Knoblock et al. (2007) broadly define noise in text as “any kind of difference between the surface form of a coded representation of the text and the intended, correct, or original text”
- For example, typos generated by user input, shorthand and slang used in social media, or filler words and phrases transcribed from audio data

Noisy Texts in the Social Sciences

- When it comes to the data sources we tend to analyze in the social sciences, e.g., historical documents, there are two very common types of noise encountered:
  - 1. OCR noise: noise that originates from OCR errors in translating image data to text data, resulting in insertions and deletions of characters that cause spelling and punctuation errors
  - 2. Layout noise: noise that results from improper layout analysis, e.g., too narrow bounding boxes resulting in large swathes of characters being ignored or mis-translated by OCR on the margins of a layout object, layout object mis-classification resulting in complex tabular-type layouts being OCRed as single-column text
- Even with the state of the art in OCR and layout analysis, much of which we have already discussed in this class, these types of noise remain a problem

Outline

The Canonical Deep NLP Training Corpus

A Definition of Noise

The Problem with Noise

Approaches to Denoising for Deep NLP

Final Words

The Problem with Noise

- From the perspective of a downstream NLP analysis, we will see that text noise is a problem for the same reason any sort of noise is a problem in a quantitative analysis: noisy inputs lead to unexplained variability in a model's output
- For deep learning tasks, where the goal is almost always prediction, this means that predictions may deterministically or stochastically depart from the ground truth
- Understanding how and why noisy text inputs generate noisy outputs in deep NLP models will ultimately help us design strategies to mitigate the downstream effects of text noise
- Because they now form the bedrock of modern NLP, dissecting how and why text noise is a problem for neural language models based on transformer architectures will be the subject of this lecture

Noisy Text and Neural Language Models

- At their core, many transformer-based NLP models are neural language models, e.g., BERT, and they create informative, context-dependent numerical representations of inputs (i.e., in the "encoding" process) that are then consumed for downstream analyses, such as text classification
- Models like BERT are trained at first ("pre-trained") to generate representations of texts that help them succeed at tasks like masked language modeling (MLM) and next sentence prediction (NSP); to be a good language modeler in some sense means to have natural language understanding, i.e., to have learned a great deal about the syntax, semantics, tone, etc., of a corpus of texts, so the representations of text necessary to do good language modeling are very "information dense"–as you've heard before, there's lots in a BERT embedding

Noisy Text and Neural Language Models

- However, a neural language model can only learn from what it has seen before, and, as we've just discussed, almost every SOTA pre-trained transformer has seen very little noise–especially social science-relevant OCR and layout noise
- Consequently, a transformer-based model pre-trained and fine-tuned exclusively on clean corpuses may not possess a true "understanding" of a noisy text, leading to representations of that text in embedding space that deviate from an ideal, ground truth representation, resulting in worse performance on downstream tasks

An Autopsy of Noisy Text NLP, in Theory

- Because of how we know transformer-based neural language models to work, we can also think more about just how a transformer-based neural language model might fail to “understand” a noisy text...
- 1. Tokenization troubles: misspellings in noisy texts can cause the WordPiece tokenizer of models like BERT to end up “breaking the [misspelled] words into subwords whose meaning can be very different from the meaning of the original word”; the looked-up embeddings of these tokens can have meanings that don’t well appoximate the meaning of their ground truth word (Kumar et al., 2020); in the limiting case, when the OCR is completely garbled, the input embeddings will have nearly no relation to the ground truth text they are meant to represent

An Autopsy of Noisy Text NLP, in Theory

- 2. Attention misapprehension: from Clark et al. (2019), we know that, among other things, some of BERT's attention heads pay close attention to punctuation, and when punctuation errors flood a piece of text, a pre-trained BERT will attend to erroneous periods and commas as if they were real periods and commas; similarly, given their clean training corpuses, we can assume that BERT's attention heads and FFNNs will have had little reason to learn to detect patterns in text that should be ignored or re-interpreted in a way befitting of noise, and in a paradigm where there is global self-attention, every contextual embedding in a transformer layer would be affected by earnestly attending to even a few erroneous input tokens induced by a misspelled word

An Autopsy of Noisy Text NLP, in Theory

- 3. Sentence segmentation struggles: in addition to token-level tasks like MLM, BERT and many other transformer-based models are pre-trained on sentence-level tasks like NSP to build natural language understanding; when punctuation and misspellings are frequent, sentences are difficult to reliably segment as inputs to the model, and the processing of sentence fragments will contribute to a further distribtional shift in the noisy data a pre-trained model sees at inference time

An Autopsy of Noisy Text NLP, in Practice

- Kumar et al. (2020) empirically assess BERT's performance across a sampling of downstream tasks when artificially injecting noise into input texts; their most striking plot is depicted below, illustrating a sharply linear decline in performance w.r.t. noise

Results of experiments on both clean and noisy data

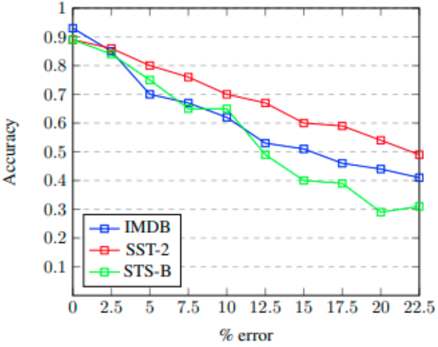


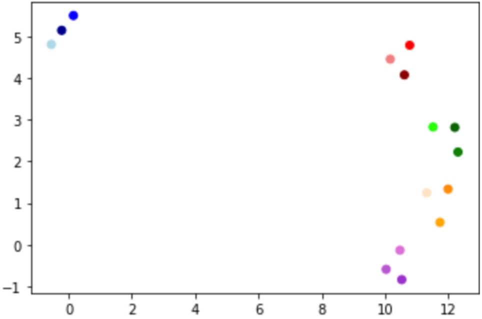
Figure 1: Accuracy vs Error

An Autopsy of Noisy Text NLP, in Practice

- Intuitively, another way of empirically assessing the impact of noise in deep NLP is to embed clean and noisy versions of the same underlying text and measure the distance between the respective respresentations in embedding space relative to, say, a different clean embedded text, or a “denoised” version the same text again; this is something we’ve experimented with in the context of OCR noise for a particular use case in our research, with the **very tentative** finding that, for certain models, like RoBERTa, mild OCR noise from high quality document scans may not disturb a ground truth embedding by very much, which bodes well for downstream analyses–especially those where one may be able to use a classifier/prediction head fine-tuned on noisy data

An Autopsy of Noisy Text NLP, in Practice

- To give a visual sense of this finding, see below a UMAP dimensionality reduced plot of embeddings for a handful of clean, noisy, and “denoised” newspaper article OCR texts, where the same color in varying shades corresponds to the same underlying article (N.B. UMAP is especially nice for this application, as it preserves global structure better than t-SNE)



Outline

The Canonical Deep NLP Training Corpus

A Definition of Noise

The Problem with Noise

Approaches to Denoising for Deep NLP

Final Words

Next Stop: Denoising

- The word “denoised” has come up a few times now, and hints at the obvious question before us now: if we know noise is a problem, and why noise is a problem, how do we use this knowledge to mitigate it?
- For the remainder of this lecture, we will cover three plausible ways of going about this: preclusion, post-processing, and pre-training

Preclusion

- In the context of working with documents and documents scans, one way to reduce noise in your NLP pipeline is to simply cut it off at the source
- By paying careful attention to the configuration of your OCR engine, or by fine-tuning or designing your own use-case-specific OCR engine, you may be able to effectively “preclude” OCR noise from entering your downstream NLP analyses (and I encourage you to revisit Lectures 9, 10, and 11 if this seems like a fruitful strategy for your use case)
- Similarly, layout noise can be reduced by carefully designing a data-rich, active learning training regime for an object detection and recognition model like Faster R-CNN or Mask R-CNN

Post-processing

- Post-processing approaches attempt to clean up your text inputs before passing them to a deep NLP model
- When OCR noise manifests as punctuation errors, off-the-shelf or use-case-tailored "punctuators" may prove quite helpful, e.g., `https://github.com/ottokart/punctuator2`, `https://github.com/nkrnrnk/BertPunc`, `https://github.com/xashru/punctuation-restoration`

Post-processing

- When OCR or layout noise manifests as spelling errors, you could apply a commerical spellchecker or build a custom spellchecker with a spellchecking dictionary, or employ or adapt a more sophisticated model like a lexical normalizer or deep spell corrector, e.g.,
  - Enhancing BERT for Lexical Normalization (Muller et al., 2019)
  - Contextual Text Denoising with Masked Language Model (Sun and Jiang, 2019)
  - Using NLP (BERT) to Improve OCR Accuracy (`https://www.statestitle.com/resource/using-nlp-bert-to-improve-ocr-accuracy/`)
- Depending on how OCR and layout noise manifest in your use case, you may also want to design your own targeted desnoising model or strategy, which could reasonably be in part either learned or rule-based

Post-processing

- For some use cases, we have also experimented with replacing misspelled words with `[MASK]` tokens at *inference* time; in theory, models like BERT should be trained to impute the embeddings of these tokens well based on context alone, given their MLM pre-training objectives

Pre-training

- At this point, you may have asked yourself: “well, if deep NLP models can learn to understand clean text from a clean corpus, why can’t they learn to understand noisy text from a noisy corpus?” We think this is a super interesting question, and there are precedents for doing so
- Relative to Wikipedia, tweets are considered noisy texts, and, just by pre-training on a corpus of 80GB of tweets, a recently developed model called BERTweet was able to surpass vanilla BERT’s performance on a variety of downstream NLP tasks involving Twitter data

Pre-training

- Gururangan et al. (2020) wrote a paper called “Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks” to further echo the importance of pre-training on text domains as close as possible to your downstream corpus of interest, and, in some sense, an OCR noised corpus may just be considered another domain of text
- Karpukhin et al. (2019) found that training a NMT model on the right diet of synthetic noise actually increased the model’s robustness vis-a-vis downstream performance in the presence of naturally noised texts

Which Approach Should You Pursue?

- While pre-training a transformer-based model to better understand noisy texts in your use case can sound compelling, it is a major computational undertaking; BERTweet alone required 4 weeks of distributed training on 8 high performance GPUs
- Practically, therefore, denoising approaches designed around punctuators and spellcheckers will prove most feasible to work into your pipeline
- Thinking carefully about the patterns of noise in your use case and implementing post-processing strategies to counter them will often prove more fruitful than casually applying a one-size-fits-all model or method

Outline

The Canonical Deep NLP Training Corpus

A Definition of Noise

The Problem with Noise

Approaches to Denoising for Deep NLP

Final Words

Final Words: Fully Architectural Approaches to Denoising

- Sergio and Lee (2020) devise a model architecture called Stacked DeBERT which stacks a set of “denoising transformers” on top of vanilla BERT encoders, which can be trained to “reconstruct” the ground truth embeddings of noisy texts
- Off-the-shelf Stacked DeBERT may yield positive results in use cases very similar to the model’s training corpus and evaluation tasks; otherwise, Stacked DeBERT may provide inspiration for doing your own architectural denoising

Final Words: W-NUT

- In the context of deep NLP, there is a nascent but quickly growing literature on approaches to denoising that is mostly focused on cleaning up and managing noisy user-generated text (i.e., the use case for noisy NLP with the most commercial application)
- Every year, the conference on Empirical Methods in Natural Language Processing (EMNLP) holds the Workshop on Noisy User-generated Text (W-NUT), which is a treasure trove of newly published denoising papers that might help spark thoughts and strategies for your own denoising efforts!

Final Words: The Viral Texts Project

- The focus of this lecture was managing and understanding noise in the context of transformer-based neural language models, but, for some use cases and analyses, there may be promising alternatives to NLP with transformers when a corpus is very noisy
- In particular, the Viral Texts Project from Northeastern University has published great softwares and papers on computationally feasibly and accurately detecting, tracking, and clustering on passage similarity in a large corpus of noisily OCRed texts
- See `https://viraltexts.org/` and `https://github.com/dasmiq/passim` for much more on this!