Text Cleaning and Normalization

Dr. Abhishek Grover
Assistant Professor
Communication and Computer Engineering, LNMIIT

Table of Contents

Why clean text

- Standardizes Input (e.g., "Running" to "run")
- Removes Noise: Stripping away irrelevant data like HTML tags, emojis, or metadata
- Reduces Dimensionality
- Enhances Feature Extraction

Dimensionality in NLP

- Classical NLP (BoW, TF-IDF): Dimensionality is vocabulary size
- Word Embedding: Dimensionality is embedding dimension (e.g. 300)
- Sequence models (RNNs, Transformers): Input dimension is embedding size

Noise in Text Data

- Inconsistent capitalization (Eg. Orange, orange)
- Punctuation and special characters
- Spelling errors and typos
- Abbreviations and Contractions (Eg. don’t)
- Slang, emojis, and informal expressions
- Numbers, dates, units in multiple formats
- HTML tags and URLs
- Multi-lingual or code-mixed text

Text Cleaning v/s Text Normalization

Text Cleaning	Text Normalization
Removing Noise	Reducing Variability
Removing HTML, punctuation, extra spaces, non-textual symbols	Includes operations such as lowercasing, stemming, lemmatization and canonical forms
Makes text processable	Makes text consistent

- Usually cleaning is done before normalization.

Tokens

- Tokenization splits text into meaningful units (tokens)
- Meaningful → Computationally useful
- They need not correspond to linguistic words(Eg. happiness=happ-in-ess)
- What is “meaningful” depends on: Model Architecture, Tokenization strategy and the application
- **Vocabulary of an NLP model is list of tokens.**

Text-Cleaning Operations

- Removing HTML/markup
- Removing URLs (sentence.like_url)
- Removing email addresses (sentence.like_email)
- Removing non-linguistic symbols (token.is_symbol)
- Filtering punctuation-only tokens (token.is_punct)
- Removing metadata/special tokens (rule-based token filtering)
- Dropping empty tokens (token.is_space)

Case Normalization

- Converting text to consistent letter case
- Improves statistical robustness of features
- May remove useful information (Eg. proper noun)
- Converting to lowercase (token.lower_)
- Task dependent decision; It is not always optimal.

Stopword Removal

- Stopwords are frequent function words (Eg. the, is, and)
- Using spaCy: token.is_stop
- Stopword lists are language specific and task-specific.
- Stopwords are usually removed for classification application.
- They are never removed for language generation tasks.

Stemming

- Reduces words to their root form
- Eg. running → run, happily → happi
- Simple, rule based algorithms
- May produce non-dictionary forms
- Common in search engines and classical NLP

Lemmatization

- Maps words to their base (dictionary) form.
- Uses linguistic knowledge (POS, morphology)
- Eg. running (verb) → run, running (noun) → running, better (adjective) → good, better (verb) → better
- More accurate but slower than stemming
- spaCy: token.lemma_

Impact on NLP models

- Directly affects vocabulary size and sparsity
- Influences feature distributions (BoW, TF-IDF)
- Over-cleaning can remove useful information
- Under-cleaning increases noise and variance
- Cleaning decisions should align with the application