Text as Data

Meeting 2: Text Preprocessing

Petro Tolochko

Text Preprocessing

- Texts are highly dimensional
- When possible, it is nice to reduce this dimensionality
- Ideally, without losing too much information

Text Preprocessing For Unsupervised Learning (Danny & Spirling, 2018)

- Punctuation
- Numbers
- Lowercasing
- Stemming/Lemmatization
- Stop-words
- N-grams
- Removal of words by frequency

Punctuation / Numbers / Lowercasing

- Fairly straighforward
- Often we don't care about punctuation and/or numbers – so, might be better to remove them
- We probably do care about the letter case
  - To what extent?
  - Reduction in dimensions might be worth the reduction in accuracy
  - When would letter case be (un)important?

Stemming / Lemmatization

- A stem is the part of the word responsible for lexical meaning
- A stem is invariable part of the word under inflection
- “wait” is a stem of:
  - Waiting
  - Waits
  - Waited
- A lemma is the base / “original” (dictionary) part of the word
  - “Went”, “gone” $\rightarrow$ “go”
- Both are useful for dimension reduction and often produce similar results

Stop Words

- Words that are filtered out before the analysis begins
- Could be any type of words that you do not want in the analysis
- Usually, function words are used as stop words
  - the
  - is
  - that
  - etc.
- Domain-specific words are also often excluded from the analysis
- E.g., "Global Warming" in the corpus of texts about Global Warming

N-grams

- So far, we've only looked at "unigrams" – individual words
- Texts can be broken down into any n-gram sequences
- "I love ice-cream and bananas"
  - "I" "love" "ice-cream" "and" "bananas"
  - "I love" "love ice-cream" "ice-cream and" "and bananas"
  - 3-grams?

Removal of terms by frequency

- Further removal of dimensionality can be achieved by removing either very frequent or very infrequent terms
- If they are very frequent, they probably don't carry much discriminating information for our analysis (think stopwords)
- If they are very infrequent, they probably carry a lot of discriminating information, but very low statistical power

Text Preprocessing For Unsupervised Learning

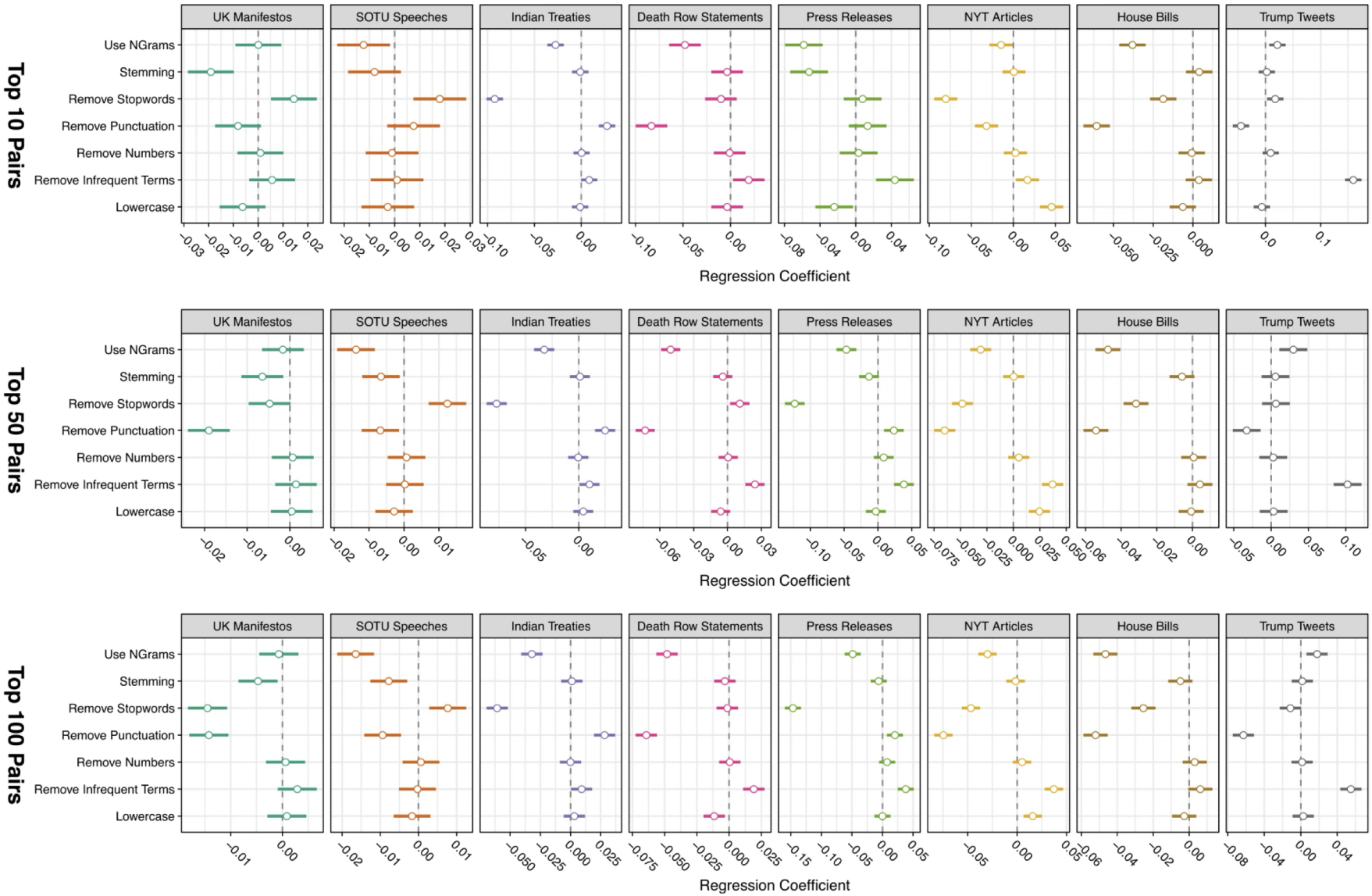
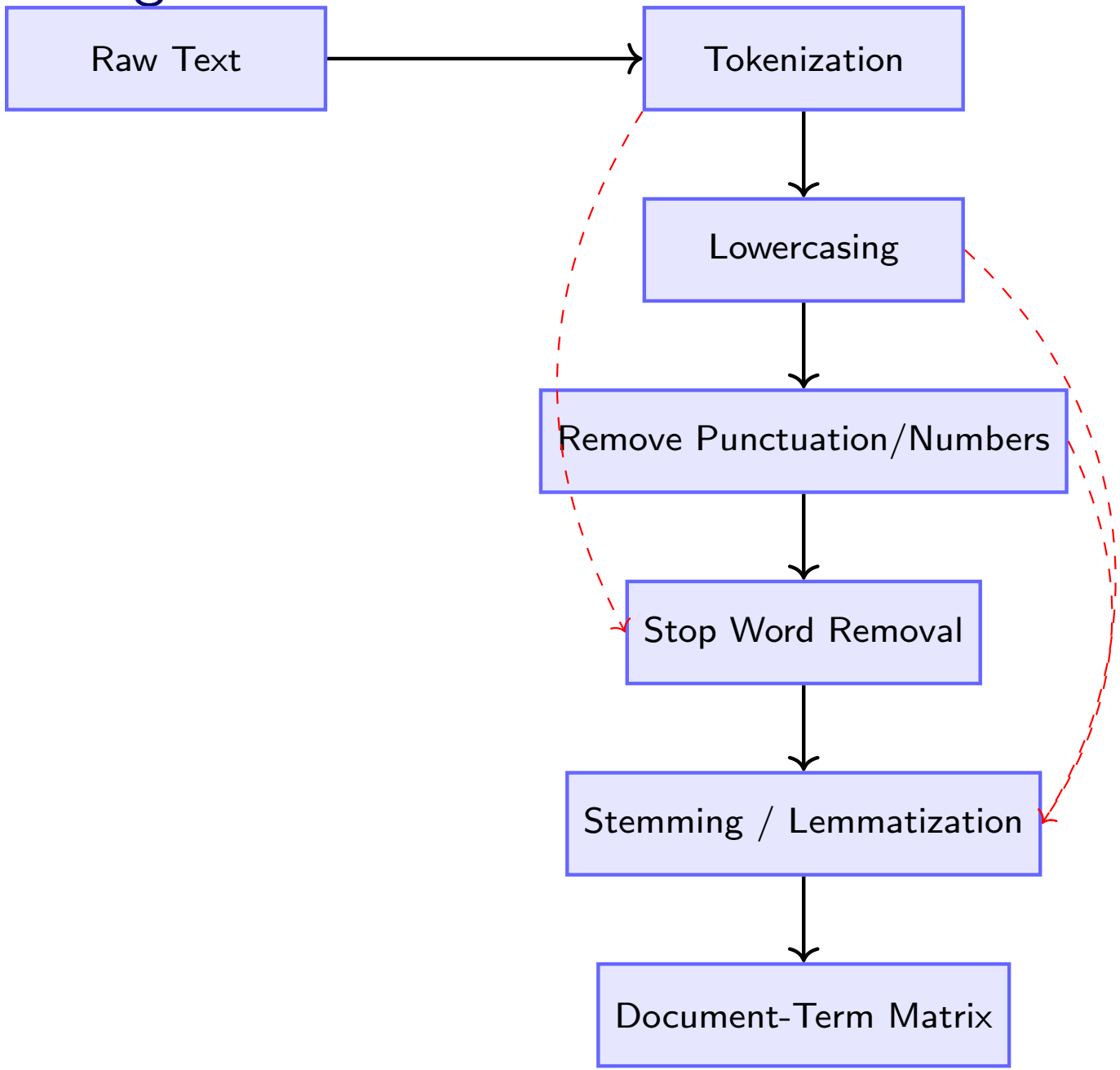


Figure 5. Regression results depicting the effects of each of the seven preprocessing steps on the `preText` score for that preprocessing combination.

Preprocessing Order Matters



TF-IDF

- We can do more than just count words
- We can transform these counts
- Use some sort of a weight in order to transform
- Term frequency inverse document frequency is one form of weighting

TF-IDF: Term Frequency – Inverse Document Frequency

- **Motivation:** Raw counts (bag-of-words) overemphasize frequent words (e.g., "the", "and").
- **Idea:**
  - **TF (term frequency):** How often a term appears in a document.
  - **IDF (inverse document frequency):** Downweight terms that appear in many documents.
- **Formula:**

$$\text{TF–IDF}(t,d) = \underbrace{\frac{\text{count}(t,d)}{\sum_{t'} \text{count}(t',d)}}_{\text{TF: term frequency in document } d} \times \underbrace{\log \frac{N}{\text{df}(t)}}_{\text{IDF: inverse document frequency}}$$

- $N =$ total number of documents.
- $\text{df}(t) =$ number of documents containing term t.

What the hell?

What the hell?

- Exactly

What the hell?

- Exactly
- Introduced in 1972 by a computer scientist (Spärck Jones, 1972)

What the hell?

- Exactly
- Introduced in 1972 by a computer scientist (Spärck Jones, 1972)
- No theoretical justification

What the hell?

- Exactly
- Introduced in 1972 by a computer scientist (Spärck Jones, 1972)
- No theoretical justification
- Apart from "it seems to work. . ."

What the hell?

- Exactly
- Introduced in 1972 by a computer scientist (Spärck Jones, 1972)
- No theoretical justification
- Apart from "it seems to work. . ."
- And sometimes it does

Odds

- **Probability:** $p = \frac{\#\text{ successes}}{\#\text{ trials}}$
- **Odds:** ratio of probability of success to probability of failure

$$\text{Odds} = \frac{p}{1-p}$$

If $p = 0.8$ (80% chance), then

$$\text{Odds} = \frac{0.8}{0.2} = 4 \quad \text{(4-to-1 in favor)}$$

Log Odds

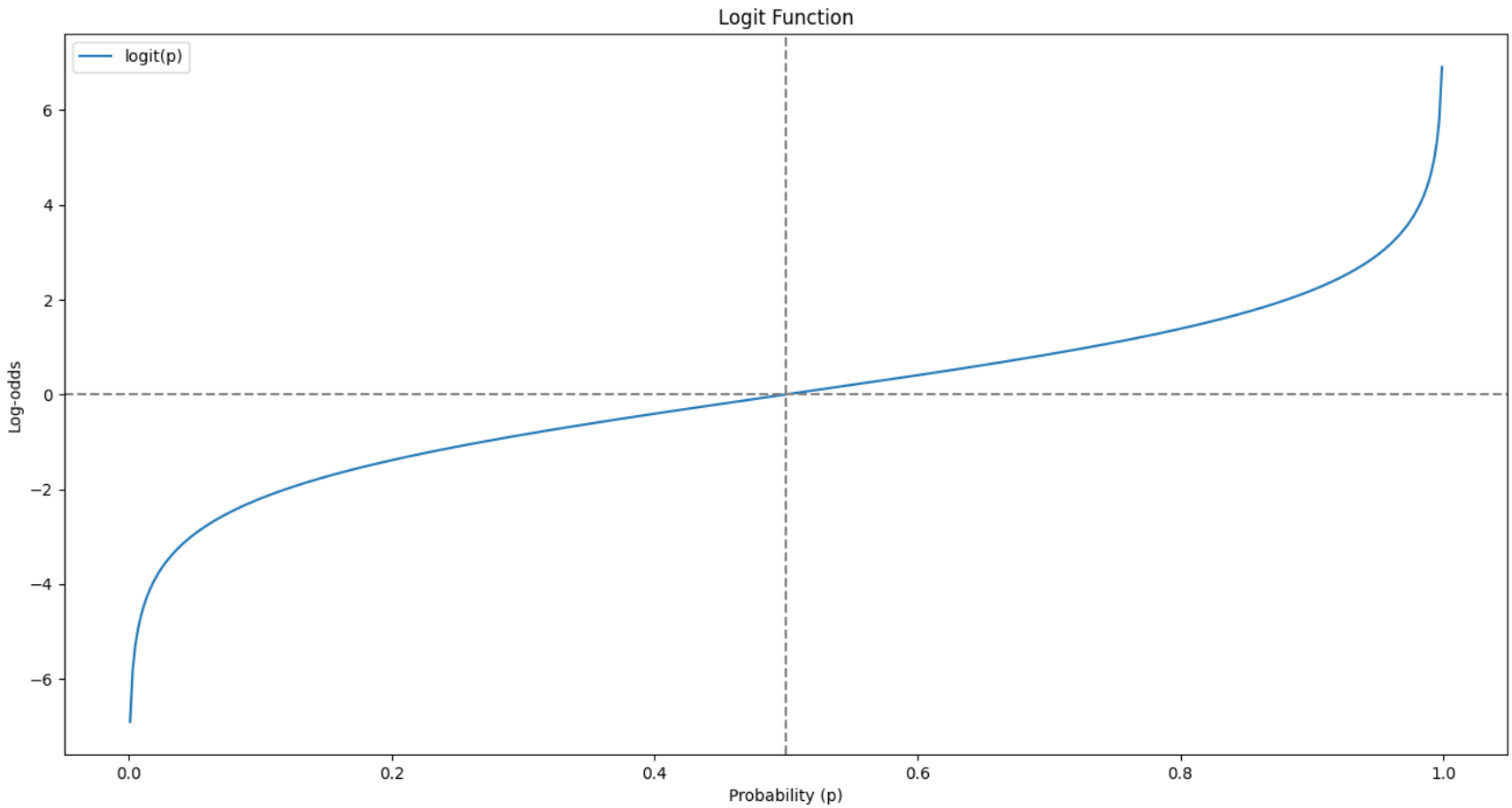
- Log odds are the natural logarithm of odds.
- Maps probabilities $(0, 1)$ to the full real line $(-\infty, +\infty)$.

$$\text{Logit}(p) = \log\left(\frac{p}{1-p}\right)$$

If $p = 0.8$, then

$$\log\left(\frac{0.8}{0.2}\right) = \log(4) \approx 1.39$$

Logit Function



Comparing Two Groups

- Suppose we have two groups (e.g., John vs. Mary).
- Each group has its own probability of using a word.
- We compare their log odds.

$$\text{Log Odds Ratio} = \log\left(\frac{p_1}{1-p_1}\right) - \log\left(\frac{p_2}{1-p_2}\right)$$

Worked Example: "banana"

- John says "banana" in 7 out of 12 words: $p_1 = 7/12 \approx 0.58$.
- Mary says "banana" in 1 out of 16 words: $p_2 = 1/16 = 0.0625$.

$$\text{Log Odds Ratio} = \log\left(\frac{0.58}{0.42}\right) - \log\left(\frac{0.0625}{0.9375}\right)$$

$$= \log(1.38) - \log(0.0667) \approx 0.32 - (-2.71) \approx 3.03$$

Positive log-odds ratio $\Rightarrow$ "banana" is more associated with John.

Why Use Log Odds?

- Probabilities are bounded between 0 and 1, hard to compare directly.
- Odds ratios can be very skewed.
- Log odds:
  - Symmetric around 0.
  - Interpretable: sign = direction, magnitude = strength.
  - Used in logistic regression and text analysis (e.g., Monroe et al. 2008).