Stemming Vs. Lemmatization

Stemming and lemmatization are both text normalization techniques used in natural language processing, but they have some key differences:

Stemming:

1. Removes word endings to reduce words to their root form
2. Faster and simpler algorithm
3. Often produces non-words as stems
4. Less accurate but computationally efficient
5. Example: "running" becomes "run", "argument" becomes "argu"

Lemmatization:

1. Reduces words to their base or dictionary form (lemma)
2. More complex, uses linguistic knowledge
3. Always produces valid words
4. More accurate but computationally intensive
5. Example: "running" becomes "run", "argument" becomes "argument"

The main trade-off is between speed and accuracy. Stemming is faster but can sometimes produce incorrect or nonsensical roots. Lemmatization is slower but generally more accurate.

```
import nltk
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

# Download required NLTK data
nltk.download('punkt')
nltk.download('wordnet')

# Sample text
text = "The runner's running shoes are running out of tread."

# Tokenize the text
tokens = word_tokenize(text)

# Stemming
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in tokens]

# Lemmatization
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in tokens]

# Print the results with color and spacing

print("Original: \033[1m\033[94m", tokens, "\033[0m")
print("\n")
print("Stemmed:  \033[1m\033[92m", stemmed_words, "\033[0m")
print("\n")
print("Lemmatized:", "\033[1m\033[91m", lemmatized_words, "\033[0m")
```

Output:

```
Original:  ['The', 'runner', "'s", 'running', 'shoes', 'are', 'running', 'out', 'of', 'tread', '.']

Stemmed:   ['the', 'runner', "'s", 'run', 'shoe', 'are', 'run', 'out', 'of', 'tread', '.']

Lemmatized:  ['The', 'runner', "'s", 'running', 'shoe', 'are', 'running', 'out', 'of', 'tread', '.']
```

1. Stemming:
    - "running" becomes "run"
    - "shoes" becomes "shoe"
2. Lemmatization:
    - "running" remains "running" (as it's used as a verb here)
    - "shoes" becomes "shoe" (singular form)

Note that lemmatization doesn't always change words as it depends on their part of speech in the sentence. For more accurate lemmatization, you can provide the part of speech:

```
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

# Download required NLTK data
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')

# Helper function to convert NLTK POS tags to WordNet POS tags
def get_wordnet_pos(tag):
    tag = tag[0].upper()
    tag_dict = {"J": nltk.corpus.wordnet.ADJ,
                "N": nltk.corpus.wordnet.NOUN,
                "V": nltk.corpus.wordnet.VERB,
```

```
            "R": nltk.corpus.wordnet.ADV}
    return tag_dict.get(tag, nltk.corpus.wordnet.NOUN)

# Sample text
text = "The runner's running shoes are running out of tread."

# Tokenize the text
tokens = word_tokenize(text)

# Lemmatization
lemmatizer = WordNetLemmatizer()

# POS tagging
pos_tags = nltk.pos_tag(tokens)

# More accurate lemmatization with POS tagging
better_lemmatized = [lemmatizer.lemmatize(word, pos=get_wordnet_pos(tag)) for word, tag in pos_tags]

# Print the results with color and spacing
print("Original: \033[1m\033[94m", tokens, "\033[0m")

print("Stemmed:  \033[1m\033[91m", better_lemmatized, "\033[0m")
```

Output:

```
Original:  ['The', 'runner', "'s", 'running', 'shoes', 'are', 'running', 'out', 'of', 'tread', '.']

Stemmed:   ['The', 'runner', "'s", 'running', 'shoe', 'be', 'run', 'out', 'of', 'tread', '.']
```

Stemming and Lemmatization are both techniques used in Natural Language Processing (NLP) to reduce words to their root form, a process often referred to as stemming. However, they differ in their approach and accuracy.

Stemming

1. Process: Removes suffixes from words to obtain the root form, often called a stem. It's a simple rule-based approach.

2. Accuracy: Less accurate than lemmatization, often producing words that don't exist in the dictionary.

3. Speed: Faster than lemmatization due to its simpler algorithm.

4. Use cases: Suitable for tasks where speed and simplicity are prioritized over accuracy, such as search engines, information retrieval systems, and text classification.

Lemmatization

1. Process: Reduces words to their dictionary form, called a lemma, considering the word's part of speech and its context.

2. Accuracy: More accurate than stemming, producing actual words.

3. Speed: Slower than stemming due to the involvement of dictionary lookups and part-of-speech tagging.

4. Use cases: Suitable for tasks where accuracy and meaning preservation are crucial, such as sentiment analysis, topic modeling, and machine translation.

When to Use Which

1. Stemming:

- When speed is critical and some loss of accuracy is acceptable.
- For large datasets where computational efficiency is a concern.
- When dealing with informal or noisy text.

2. Lemmatization:

- When accuracy and meaning preservation are essential.
- For tasks that require in-depth linguistic analysis.
- When dealing with formal or structured text.

In summary, while stemming is a quick and dirty approach, lemmatization is more accurate but computationally expensive. The choice between the two depends on the specific requirements of the NLP task at hand.