Computational Linguistics-1

Text Pre-Processing

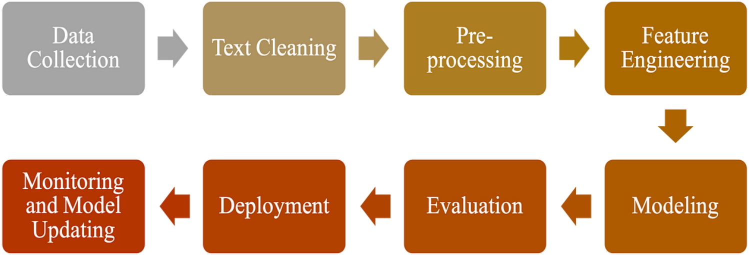
Parameswari Krishnamurthy

Language Technologies Research Centre
IIIT-Hyderabad

param.krishna@iiit.ac.in

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

NLP Pipeline



Data Collection

Data Collection

Gather text data from various sources such as websites, books, articles, and social media.

Data Collection

Gather text data from various sources such as websites, books, articles, and social media.

Challenges:

- **Data Quality**

Data Collection

Gather text data from various sources such as websites, books, articles, and social media.

Challenges:

- **Data Quality**
  - Incomplete or missing data
  - Inconsistent data formats
  - Presence of noise and outliers

Data Collection

Gather text data from various sources such as websites, books, articles, and social media.

Challenges:

- **Data Quality**
  - Incomplete or missing data
  - Inconsistent data formats
  - Presence of noise and outliers
- **Data Privacy and Security**

Data Collection

Gather text data from various sources such as websites, books, articles, and social media.

Challenges:

- **Data Quality**
  - Incomplete or missing data
  - Inconsistent data formats
  - Presence of noise and outliers
- **Data Privacy and Security**
  - Ensuring data anonymization
  - Compliance with regulations (e.g., GDPR)
  - Securing data storage and transfer

Data Collection

Challenges:

- **Data Accessibility**

Data Collection

Challenges:

- **Data Accessibility**
    - Limited access to proprietary or sensitive data
    - High costs of acquiring certain datasets
    - Technical barriers to accessing data from various sources

Data Collection

Challenges:

- **Data Accessibility**
  - Limited access to proprietary or sensitive data
  - High costs of acquiring certain datasets
  - Technical barriers to accessing data from various sources
- **Data Volume and Variety**

Data Collection

Challenges:

- **Data Accessibility**
  - Limited access to proprietary or sensitive data
  - High costs of acquiring certain datasets
  - Technical barriers to accessing data from various sources
- **Data Volume and Variety**
  - Managing large volumes of data (Big Data)
  - Integrating data from multiple sources and formats
  - Handling unstructured data (e.g., text, images, videos)

Data Collection

Challenges:

- **Data Accessibility**
  - Limited access to proprietary or sensitive data
  - High costs of acquiring certain datasets
  - Technical barriers to accessing data from various sources
- **Data Volume and Variety**
  - Managing large volumes of data (Big Data)
  - Integrating data from multiple sources and formats
  - Handling unstructured data (e.g., text, images, videos)
- **Bias and Representativeness**

Data Collection

Challenges:

- **Data Accessibility**
  - Limited access to proprietary or sensitive data
  - High costs of acquiring certain datasets
  - Technical barriers to accessing data from various sources
- **Data Volume and Variety**
  - Managing large volumes of data (Big Data)
  - Integrating data from multiple sources and formats
  - Handling unstructured data (e.g., text, images, videos)
- **Bias and Representativeness**
  - Ensuring the data is representative of the population
  - Avoiding sampling bias
  - Addressing any inherent biases in the data collection process

Text Cleaning

Text Cleaning

- **Remove Noise:**
    - **Punctuation, Numbers, and Special Characters:**

Text Cleaning

- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** "Hello! This is an example text with numbers 12345 and symbols $%&."
    - **Cleaned Text:** "Hello This is an example text with numbers and symbols"

Text Cleaning

- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** “Hello! This is an example text with numbers 12345 and symbols $%&.”
    - **Cleaned Text:** “Hello This is an example text with numbers and symbols”
  - Removing noise helps focus on the meaningful parts of the text.
- **Correct Spelling Errors and Normalize Text:**

Text Cleaning

- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** "Hello! This is an example text with numbers 12345 and symbols $%&."
    - **Cleaned Text:** "Hello This is an example text with numbers and symbols"
  - Removing noise helps focus on the meaningful parts of the text.
- **Correct Spelling Errors and Normalize Text:**
  - **Original Text:** "This sentnce contains a speling error."
  - **Corrected Text:** "This sentence contains a spelling error."

Text Cleaning

- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** "Hello! This is an example text with numbers 12345 and symbols $%&."
    - **Cleaned Text:** "Hello This is an example text with numbers and symbols"
  - Removing noise helps focus on the meaningful parts of the text.
- **Correct Spelling Errors and Normalize Text:**
  - **Original Text:** "This sentnce contains a speling error."
  - **Corrected Text:** "This sentence contains a spelling error."
  - Normalization involves converting text to a standard form, such as converting different forms of a word to a single form (e.g., "color" and "colour" to "color").

Text Cleaning

- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** "Hello! This is an example text with numbers 12345 and symbols $%&."
    - **Cleaned Text:** "Hello This is an example text with numbers and symbols"
  - Removing noise helps focus on the meaningful parts of the text.
- **Correct Spelling Errors and Normalize Text:**
  - **Original Text:** "This sentnce contains a speling error."
  - **Corrected Text:** "This sentence contains a spelling error."
  - Normalization involves converting text to a standard form, such as converting different forms of a word to a single form (e.g., "color" and "colour" to "color").
- **Handle Misspellings, Slang, and Abbreviations:**

Text Cleaning

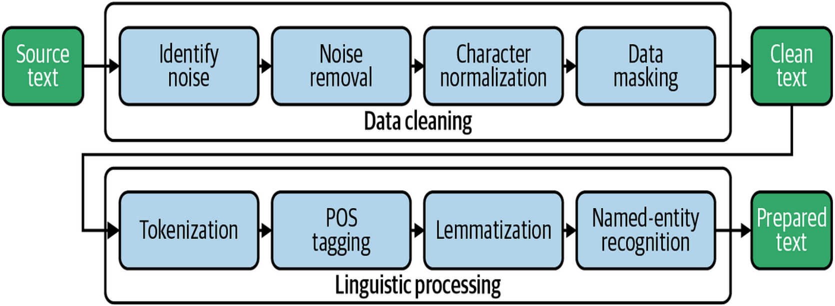
- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** “Hello! This is an example text with numbers 12345 and symbols $%&.”
    - **Cleaned Text:** “Hello This is an example text with numbers and symbols”
  - Removing noise helps focus on the meaningful parts of the text.
- **Correct Spelling Errors and Normalize Text:**
  - **Original Text:** “This sentnce contains a speling error.”
  - **Corrected Text:** “This sentence contains a spelling error.”
  - Normalization involves converting text to a standard form, such as converting different forms of a word to a single form (e.g., “color” and “colour” to “color”).
- **Handle Misspellings, Slang, and Abbreviations:**
  - **Original Text:** “OMG, this txt is gr8!”
  - **Normalized Text:** “Oh my god, this text is great!”

Text Cleaning

- **Remove Noise:**
  - **Punctuation, Numbers, and Special Characters:**
    - **Original Text:** “Hello! This is an example text with numbers 12345 and symbols $%&.”
    - **Cleaned Text:** “Hello This is an example text with numbers and symbols”
  - Removing noise helps focus on the meaningful parts of the text.
- **Correct Spelling Errors and Normalize Text:**
  - **Original Text:** “This sentnce contains a speling error.”
  - **Corrected Text:** “This sentence contains a spelling error.”
  - Normalization involves converting text to a standard form, such as converting different forms of a word to a single form (e.g., “color” and “colour” to “color”).
- **Handle Misspellings, Slang, and Abbreviations:**
  - **Original Text:** “OMG, this txt is gr8!”
  - **Normalized Text:** “Oh my god, this text is great!”
  - Converting slang and abbreviations to their full forms ensures clarity and consistency.

Text Pre-processing

Text Pre-processing



Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.
- **Enhances Accuracy:** Clean and well-processed text improves the performance of NLP tasks like parsing and named entity recognition.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.
- **Enhances Accuracy:** Clean and well-processed text improves the performance of NLP tasks like parsing and named entity recognition.
- **Reduces Noise:** Removing irrelevant information (e.g., stop words) helps focus on meaningful content.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.
- **Enhances Accuracy:** Clean and well-processed text improves the performance of NLP tasks like parsing and named entity recognition.
- **Reduces Noise:** Removing irrelevant information (e.g., stop words) helps focus on meaningful content.
- **Facilitates Consistency:** Normalization techniques ensure uniformity in text data, aiding better understanding and analysis.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.
- **Enhances Accuracy:** Clean and well-processed text improves the performance of NLP tasks like parsing and named entity recognition.
- **Reduces Noise:** Removing irrelevant information (e.g., stop words) helps focus on meaningful content.
- **Facilitates Consistency:** Normalization techniques ensure uniformity in text data, aiding better understanding and analysis.
- **Improves Training Efficiency:** Preprocessed text speeds up training by reducing complexity and dimensionality.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.
- **Enhances Accuracy:** Clean and well-processed text improves the performance of NLP tasks like parsing and named entity recognition.
- **Reduces Noise:** Removing irrelevant information (e.g., stop words) helps focus on meaningful content.
- **Facilitates Consistency:** Normalization techniques ensure uniformity in text data, aiding better understanding and analysis.
- **Improves Training Efficiency:** Preprocessed text speeds up training by reducing complexity and dimensionality.
- **Boosts Model Quality:** Clean and standardized data helps in learning more accurate language patterns.

Text Preprocessing

- Text preprocessing is crucial for improving the quality of text data before applying NLP techniques.
- It improves the quality of text data before applying NLP techniques.
- **Enhances Accuracy:** Clean and well-processed text improves the performance of NLP tasks like parsing and named entity recognition.
- **Reduces Noise:** Removing irrelevant information (e.g., stop words) helps focus on meaningful content.
- **Facilitates Consistency:** Normalization techniques ensure uniformity in text data, aiding better understanding and analysis.
- **Improves Training Efficiency:** Preprocessed text speeds up training by reducing complexity and dimensionality.
- **Boosts Model Quality:** Clean and standardized data helps in learning more accurate language patterns.
- **Mitigates Bias:** Proper preprocessing can help in reducing biases present in the raw text.

Text Preprocessing Steps

- **Tokenization:** Split text into individual words or sentences.
- **Lowercasing:** Convert all text to lowercase to ensure consistency.
- **Stop Words Removal:** Eliminate common words (e.g., “and”, “the”) that add little value.
- **Normalization:** Convert text into a standardized format by addressing various inconsistencies and variations.
- **Stemming/Lemmatization:** Reduce words to their base or root form.

Text Preprocessing Steps

- **Tokenization:** Split text into individual words or sentences.
- **Lowercasing:** Convert all text to lowercase to ensure consistency.
- **Stop Words Removal:** Eliminate common words (e.g., “and”, “the”) that add little value.
- **Normalization:** Convert text into a standardized format by addressing various inconsistencies and variations.
- **Stemming/Lemmatization:** Reduce words to their base or root form.

Text Preprocessing: Tokenization

Tokenization is the process of splitting text into smaller units called tokens (sentences and words).

Text Preprocessing: Tokenization

Tokenization is the process of splitting text into smaller units called tokens (sentences and words).

- **Sentence tokenization** is the process of splitting text into individual sentences.

Text Preprocessing: Tokenization

Tokenization is the process of splitting text into smaller units called tokens (sentences and words).

- **Sentence tokenization** is the process of splitting text into individual sentences.

Challenges:

- Handling punctuation marks that do not indicate the end of a sentence (Dr., e.g., Ph.D. etc.)
- Differentiating between periods in abbreviations and sentence boundaries
- Dealing with sentences that include quotes or parentheses

Text Preprocessing: Tokenization

Tokenization is the process of splitting text into smaller units called tokens (sentences and words).

- **Sentence tokenization** is the process of splitting text into individual sentences.

Challenges:

- Handling punctuation marks that do not indicate the end of a sentence (Dr., e.g., Ph.D. etc.)
- Differentiating between periods in abbreviations and sentence boundaries
- Dealing with sentences that include quotes or parentheses

Sentence tokenization

- **Original Text:** “Dr.Indhu, an expert in AI, visited Chennai. She gave a talk on Ph.D. research at IIT Madras. Her presentation was insightful, e.g., she discussed various algorithms. After the event, we went to 'Marina Beach' for a relaxing evening.”
- **Sentence Tokenized Text:**
  - “Dr. Indhu, an expert in AI, visited Chennai.”
  - “She gave a talk on Ph.D. research at IIT Madras.”
  - “Her presentation was insightful, e.g., she discussed various algorithms.”
  - “After the event, we went to 'Marina Beach' for a relaxing evening.”

Text Preprocessing: Word Tokenization

Word tokenization is the process of splitting text into individual words.

Text Preprocessing: Word Tokenization

Word tokenization is the process of splitting text into individual words.

Challenges:

- Can’t just blindly remove punctuation. Full stops (”.”) are ambiguous; Dr., m.p.h., Ph.D.
- Email addresses, URLs, etc. contain alphabets, numbers, as well as special characters (”@”, “/”, “-”, “_”)
- Languages like English use contractions (”we’re”, “I’m”) which, when tokenized by this approach, creates tokens ”re”, “m”, which are not meaningful.

Lowercasing

Lowercasing is the process of converting all characters in a text to lowercase. This step standardizes text data by eliminating case differences, which helps in uniform analysis.

Lowercasing

Lowercasing is the process of converting all characters in a text to lowercase. This step standardizes text data by eliminating case differences, which helps in uniform analysis.

Why is Lowercasing Important?

- **Uniform Representation:** Treats words with different cases as identical, which is crucial for accurate text analysis and processing.

Lowercasing

Lowercasing is the process of converting all characters in a text to lowercase. This step standardizes text data by eliminating case differences, which helps in uniform analysis.
Why is Lowercasing Important?

- **Uniform Representation:** Treats words with different cases as identical, which is crucial for accurate text analysis and processing.
- **Simplifies Matching:** Helps in text matching and retrieval tasks by reducing case sensitivity.

Lowercasing

Lowercasing is the process of converting all characters in a text to lowercase. This step standardizes text data by eliminating case differences, which helps in uniform analysis.

Why is Lowercasing Important?

- **Uniform Representation:** Treats words with different cases as identical, which is crucial for accurate text analysis and processing.
- **Simplifies Matching:** Helps in text matching and retrieval tasks by reducing case sensitivity.
- **Improves Model Efficiency:** Ensures that text data is consistent, enhancing the performance of machine learning models.

Lowercasing

Example

Original Text:

```
"The quick brown Fox jumps over the lazy DOG."
```

Lowercasing

Example

Original Text:

`"The quick brown Fox jumps over the lazy DOG."`

After Lowercasing:

`"the quick brown fox jumps over the lazy dog."`

- Consider a search engine querying for `"quick Brown fox"` in a database of documents.
- Lowercasing ensures that the search results match regardless of the case used in the query or the documents.

Text Preprocessing: Stopword Removal

Stopword removal involves eliminating common words that add little value (e.g., “and”, “the”).

Text Preprocessing: Stopword Removal

Stopword removal involves eliminating common words that add little value (e.g., “and”, “the”).

Challenges:

- Determining the appropriate stopword list for the specific context: tasks such as information retrieval, sentiment analysis, and topic modeling.
- Ensuring important words are not mistakenly removed (e.g., “no” in “no pain no gain”)

Text Preprocessing: Stopword Removal

Stopword removal involves eliminating common words that add little value (e.g., “and”, “the”).

Challenges:

- Determining the appropriate stopword list for the specific context: tasks such as information retrieval, sentiment analysis, and topic modeling.
- Ensuring important words are not mistakenly removed (e.g., “no” in “no pain no gain”)

Text Preprocessing: Stopword Removal

Stopword removal involves eliminating common words that add little value (e.g., “and”, “the”).

Challenges:

- Determining the appropriate stopword list for the specific context: tasks such as information retrieval, sentiment analysis, and topic modeling.
- Ensuring important words are not mistakenly removed (e.g., “no” in “no pain no gain”)

When to NOT remove stopwords:

- If the task involves understanding the context or sentiment; for example, in sentiment analysis, words like “not” in “not happy” are crucial for understanding the sentiment.

Text Preprocessing: Stopword Removal

Stopword removal involves eliminating common words that add little value (e.g., “and”, “the”).

Challenges:

- Determining the appropriate stopword list for the specific context: tasks such as information retrieval, sentiment analysis, and topic modeling.
- Ensuring important words are not mistakenly removed (e.g., “no” in “no pain no gain”)

When to NOT remove stopwords:

- If the task involves understanding the context or sentiment; for example, in sentiment analysis, words like “not” in “not happy” are crucial for understanding the sentiment.
- For tasks like machine translation or text generation; retaining stopwords is important to preserve the grammatical structure and meaning of sentences.

Text Preprocessing: Stopword Removal

Stopword removal involves eliminating common words that add little value (e.g., “and”, “the”).

Challenges:

- Determining the appropriate stopword list for the specific context: tasks such as information retrieval, sentiment analysis, and topic modeling.
- Ensuring important words are not mistakenly removed (e.g., “no” in “no pain no gain”)

When to NOT remove stopwords:

- If the task involves understanding the context or sentiment; for example, in sentiment analysis, words like “not” in “not happy” are crucial for understanding the sentiment.
- For tasks like machine translation or text generation; retaining stopwords is important to preserve the grammatical structure and meaning of sentences.
- Multi-Word Expressions (MWEs); phrases like “fish and chips”, “kick the bucket” lose their meaning when stopwords (”and”/”the”) are removed.

Text Preprocessing: Normalization

Normalization involves converting text to a standard format, such as lowercasing, expanding abbreviations, and correcting spelling errors.

Text Preprocessing: Normalization

Normalization involves converting text to a standard format, such as lowercasing, expanding abbreviations, and correcting spelling errors.

Challenges:

- Handling variations in spelling (e.g., “favourite” vs “favorite”)
- Dealing with domain-specific abbreviations and slang
- Correcting spelling errors without introducing new errors

Text Preprocessing: Normalization

Normalization involves converting text to a standard format, such as lowercasing, expanding abbreviations, and correcting spelling errors.

Challenges:

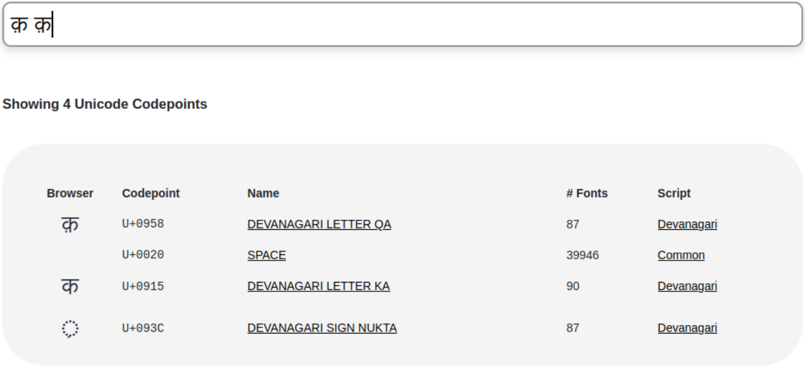
- Handling variations in spelling (e.g., “favourite” vs “favorite”)
- Dealing with domain-specific abbreviations and slang
- Correcting spelling errors without introducing new errors

Example:

- **Original Text:** “LOL, that was the funniest joke ever!!!”
- **Normalized Text:** “Laugh out loud, that was the funniest joke ever”

Unicode Normalization

Normalizaiton in Hindi an Example:



क़ क़

Showing 4 Unicode Codepoints

Browser	Codepoint	Name	# Fonts	Script
क़	U+0958	DEVANAGARI LETTER QA	87	Devanagari
	U+0020	SPACE	39946	Common
क	U+0915	DEVANAGARI LETTER KA	90	Devanagari
◌़	U+093C	DEVANAGARI SIGN NUKTA	87	Devanagari

Figure: Devanagari Example for Normalization

Spelling Normalization

- A Telugu word can be written in different forms:
  taruvatā
  tarvatā
  taravatā
- Spellings of these kinds which might be valid and most frequent in corpus need to be normalized.

Stemming and Lemmatization

Stemming

Stemming is a process that removes suffixes from words to reduce them to a base form. It uses heuristic rules and does not always produce valid dictionary words.

Lemmatization

Lemmatization reduces words to their base or dictionary form (lemma) by considering the context and ensuring the root form is a valid word. It involves more complex analysis compared to stemming.

Stemming and Lemmatization

Stemming Example:

- **Original Words:** “flies”, “flying”, “flied”
- **Stemmed Form:** “fli/fly”

Lemmatization Example:

- **Original Words:** “flies”, “flying”, “flied”
- **Lemmatized Form:** “fly”

Stemming and Lemmatization

Key Differences

- **Approach:** Stemming uses heuristic rules to strip suffixes, while lemmatization uses a dictionary and context.
- **Output:** Stemming can produce non-words, while lemmatization produces valid words.
- **Complexity:** Lemmatization involves more sophisticated analysis and is more accurate but computationally more expensive than stemming.

Conclusion

- Importance of Text Preprocessing: Proper preprocessing is essential for effective NLP applications. It ensures that the data is clean, consistent, and ready for analysis.

Conclusion

- Importance of Text Preprocessing: Proper preprocessing is essential for effective NLP applications. It ensures that the data is clean, consistent, and ready for analysis.
- Key Steps: The main steps include data collection, text cleaning, and preprocessing techniques like tokenization, lowercasing, stopword removal, and normalization.

Conclusion

- Importance of Text Preprocessing: Proper preprocessing is essential for effective NLP applications. It ensures that the data is clean, consistent, and ready for analysis.
- Key Steps: The main steps include data collection, text cleaning, and preprocessing techniques like tokenization, lowercasing, stopword removal, and normalization.
- Challenges: Each step comes with its own set of challenges, including handling noise, ensuring data privacy, managing different text formats, and addressing biases.

Conclusion

- Importance of Text Preprocessing: Proper preprocessing is essential for effective NLP applications. It ensures that the data is clean, consistent, and ready for analysis.
- Key Steps: The main steps include data collection, text cleaning, and preprocessing techniques like tokenization, lowercasing, stopword removal, and normalization.
- Challenges: Each step comes with its own set of challenges, including handling noise, ensuring data privacy, managing different text formats, and addressing biases.
- Best Practices: Always adapt preprocessing steps to the specific requirements of your NLP task and ensure that the processed text maintains its integrity and meaning.

Conclusion

- Importance of Text Preprocessing: Proper preprocessing is essential for effective NLP applications. It ensures that the data is clean, consistent, and ready for analysis.
- Key Steps: The main steps include data collection, text cleaning, and preprocessing techniques like tokenization, lowercasing, stopword removal, and normalization.
- Challenges: Each step comes with its own set of challenges, including handling noise, ensuring data privacy, managing different text formats, and addressing biases.
- Best Practices: Always adapt preprocessing steps to the specific requirements of your NLP task and ensure that the processed text maintains its integrity and meaning.
- Future Directions: As NLP continues to evolve, keeping up with advancements in preprocessing techniques and tools will be crucial for improving the accuracy and efficiency of text analysis.