# Do we need pre-processing in NLP ?

Gaël Lejeune (gael.lejeune@sorbonne-universite.fr)

January 21st 2026

Sorbonne Université

# Preprocessing in NLP, what is it good for ?

# The dogma

## Understanding Preprocessing

Preprocessing is a critical step in NLP that involves cleaning and preparing text data for analysis. It includes several tasks such as tokenization, removing stop words, stemming, lemmatization, and more. These tasks help in reducing the noise in the data, making it more manageable and meaningful for analysis.

Text preprocessing is an essential step in **natural language processing** (NLP) that involves cleaning anc transforming unstructured text data to prepare it for analysis. It includes **tokenization**, **stemming**, lemmatization, stop-word removal, and part-of-speech tagging. In this article, we will introduce the basics of text preprocessing and provide **Python** code examples to illustrate how to implement these tasks using the **NLTK library.** By the end of the article, readers will better understand how to prepare text data for NLP tasks.

Machine Learning heavily relies on the quality of the data fed into it, and thus, data preprocessing plays a crucial role in ensuring the accuracy and efficiency of the model. In this article, we will discuss the main text preprocessing techniques used in NLP.

**1. Text Cleaning**

In this step, we will perform fundamental actions to clean the text. These actions involve transforming all the text to lowercase, eliminating characters that do not qualify as words or whitespace, as well as removing any numerical digits present.

**I. Converting to lowercase**

Here is a comprehensive list of common techn text preprocessing:

1. Text lowercasing
2. Tokenization
3. Stop-word removal
4. Handling Numerical values
5. Handling Special characters
6. Whitespace stripping
7. Lemmatization/Stemming

# Processing or Preprocessing

**What is the difference ?**

- Preprocessing steps seem harmless (but mandatory ?)
- Are "Processing" steps more noble ?

# Processing or Preprocessing

**What is the difference ?**

- Preprocessing steps seem harmless (but mandatory ?)
- Are "Processing" steps more noble ?
- Which ones are documented and justified ?

# Processing or Preprocessing

**What is the difference ?**

- Preprocessing steps seem harmless (but mandatory ?)
- Are "Processing" steps more noble ?
- Which ones are documented and justified ?

Preprocessing steps are in fact full-fledged processing steps, since they have a non-negligible impact on subsequent operations [Millour, 2020]

# Processing or Preprocessing

**What is the difference ?**

- Preprocessing steps seem harmless (but mandatory ?)
- Are "Processing" steps more noble ?
- Which ones are documented and justified ?

Preprocessing steps are in fact full-fledged processing steps, since they have a non-negligible impact on subsequent operations [Millour, 2020]

From a design perspective :

- They take time
- Do they focus attention on the right problems ?

# Processing or Preprocessing

**What is the difference ?**

- Preprocessing steps seem harmless (but mandatory ?)
- Are "Processing" steps more noble ?
- Which ones are documented and justified ?

Preprocessing steps are in fact full-fledged processing steps, since they have a non-negligible impact on subsequent operations [Millour, 2020]

From a design perspective :

- They take time
- Do they focus attention on the right problems ?
- Do they actually improve results ?

# Processing or Preprocessing

**What is the difference ?**

- Preprocessing steps seem harmless (but mandatory ?)
- Are "Processing" steps more noble ?
- Which ones are documented and justified ?

Preprocessing steps are in fact full-fledged processing steps, since they have a non-negligible impact on subsequent operations [Millour, 2020]

From a design perspective :

- They take time
- Do they focus attention on the right problems ?
- Do they actually improve results ?

# Which preprocessing steps are we talking about ?

*In the literature there is no convention adopted, and each work tests some preprocessing techniques rather than others.*

- Lowercase letters.
- Spelling Correction.

# Which preprocessing steps are we talking about ?

*In the literature there is no convention adopted, and each work tests some preprocessing techniques rather than others.*

- Lowercase letters.
- Spelling Correction.
- Removing HTML tags / URLs.

# Which preprocessing steps are we talking about ?

*In the literature there is no convention adopted, and each work tests some preprocessing techniques rather than others.*

- Lowercase letters.
- Spelling Correction.
- Removing HTML tags / URLs.
- Removing punctuation.
- Removing stop words.

## Which preprocessing steps are we talking about ?

*In the literature there is no convention adopted, and each work tests some preprocessing techniques rather than others.*

- Lowercase letters.
- Spelling Correction.
- Removing HTML tags / URLs.
- Removing punctuation.
- Removing stop words.
- Removing emojis.

# Which preprocessing steps are we talking about ?

*In the literature there is no convention adopted, and each work tests some preprocessing techniques rather than others.*

- Lowercase letters.
- Spelling Correction.
- Removing HTML tags / URLs.
- Removing punctuation.
- Removing stop words.
- Removing emojis.
- Tokenization.
- Stemming.
- Lemmatization.

# Which preprocessing steps are we talking about ?

*In the literature there is no convention adopted, and each work tests some preprocessing techniques rather than others.*

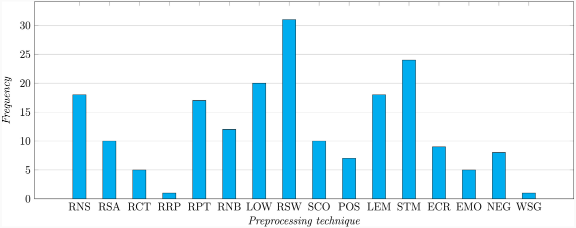
- Lowercase letters.
- Spelling Correction.
- Removing HTML tags / URLs.
- Removing punctuation.
- Removing stop words.
- Removing emojis.
- Tokenization.
- Stemming.
- Lemmatization.

How good is your tokenizer ? on the monolingual performance of multilingual language models [Rust et al., 2020]

Stemming impact on arabic text categorization performance : A survey (Al Anzi 2015)

**Is text preprocessing still worth the time ? A comparative survey on the influence of popular preprocessing methods** . . . [Siino et al., 2024]

# A more detailed overview (Siino et al.)




| | | | |
|---|---|---|---|
| **DON** | Do Nothing | **SCO** | Spelling Correction |
| **RNS** | Replace Noise | **POS** | Part-of-Speech Tagging |
| **RSA** | Replace Slang/Abbreviations | **LEM** | Lemmatization |
| **RCT** | Replace Contraction | **STM** | Stemming |
| **RRP** | Remove Repeated Punctuation | **ECR** | Remove Elongation |
| **RPT** | Removing Punctuation | **EMO** | Emoticon Handling |
| **RNB** | Remove Numbers | **NEG** | Negation Handling |
| **LOW** | Lowercasing | **WSG** | Word Segmentation |
| **RSW** | Remove Stop Words | | (some trending topic) |

# Sentiment on Reviews en [Siino et al., 2024]

| IMDB | | | | | | |
|---|---|---|---|---|---|---|
| **Preprocessing** | **RoBERTa** | **XLNet** | **ELECTRA** | **ANN** | **CNN** | **BiLSTM** |
| **DON (D)** | 0.884 ± 0.00 | 0.885 ± 0.00 | 0.888 ± 0.00 | 0.835 ± 0.01 | 0.856 ± 0.00 | 0.847 ± 0.00 |
| **LOW (L)** | 0.877 ± 0.00 | 0.881 ± 0.01 | **0.895 ± 0.04** | 0.842 ± 0.01 | **0.857 ± 0.00** | 0.843 ± 0.01 |
| **RSW (R)** | **0.885 ± 0.00** | **0.886 ± 0.00** | 0.890 ± 0.07 | 0.840 ± 0.01 | 0.855 ± 0.00 | 0.843 ± 0.01 |
| **STM (S)** | 0.853 ± 0.00 | 0.852 ± 0.03 | 0.857 ± 0.05 | 0.834 ± 0.01 | 0.856 ± 0.00 | 0.837 ± 0.02 |
| **(L)→(R)** | 0.875 ± 0.04 | 0.878 ± 0.01 | 0.888 ± 0.01 | 0.840 ± 0.01 | 0.854 ± 0.00 | 0.844 ± 0.01 |
| **(L)→(S)** | 0.849 ± 0.00 | 0.847 ± 0.01 | 0.860 ± 0.03 | **0.845 ± 0.00** | 0.855 ± 0.00 | 0.845 ± 0.02 |
| **(R)→(L)** | 0.876 ± 0.04 | 0.874 ± 0.00 | 0.890 ± 0.01 | 0.844 ± 0.01 | 0.855 ± 0.00 | 0.847 ± 0.01 |
| **(R)→(S)** | 0.826 ± 0.02 | 0.823 ± 0.32 | 0.832 ± 0.02 | 0.839 ± 0.00 | 0.855 ± 0.00 | 0.844 ± 0.02 |
| **(S)→(L)** | 0.849 ± 0.00 | 0.845 ± 0.03 | 0.864 ± 0.01 | 0.839 ± 0.00 | 0.854 ± 0.00 | 0.840 ± 0.01 |
| **(S)→(R)** | 0.798 ± 0.07 | 0.817 ± 0.01 | 0.832 ± 0.01 | 0.843 ± 0.01 | 0.854 ± 0.00 | 0.843 ± 0.01 |
| **(L)→(S)→(R)** | 0.806 ± 0.04 | 0.782 ± 0.12 | 0.824 ± 0.01 | 0.837 ± 0.01 | 0.855 ± 0.00 | 0.839 ± 0.34 |
| **(L)→(R)→(S)** | 0.838 ± 0.34 | 0.820 ± 0.02 | 0.837 ± 0.04 | 0.842 ± 0.01 | 0.854 ± 0.00 | 0.845 ± 0.00 |
| **(S)→(L)→(R)** | 0.812 ± 0.01 | 0.645 ± 0.18 | 0.818 ± 0.02 | 0.840 ± 0.01 | 0.856 ± 0.00 | 0.845 ± 0.01 |
| **(S)→(R)→(L)** | 0.818 ± 0.02 | 0.820 ± 0.05 | 0.837 ± 0.01 | 0.843 ± 0.01 | 0.853 ± 0.00 | 0.839 ± 0.01 |
| **(R)→(L)→(S)** | 0.829 ± 0.03 | 0.837 ± 0.17 | 0.825 ± 0.05 | 0.838 ± 0.01 | 0.855 ± 0.00 | **0.848 ± 0.01** |
| **(R)→(S)→(L)** | 0.806 ± 0.03 | 0.822 ± 0.07 | 0.848 ± 0.01 | 0.838 ± 0.01 | **0.857 ± 0.00** | 0.838 ± 0.34 |

**Figure 1** – Median accuracy over 5 runs + max difference. For each model, the best result is in bold, the worst in red.

# A more detailed overview (en) (Siino et al.)

IMDB : Review Polarity, PCL : Press Condescending Language

FNS : Fake News, 20N : Forum Categorization

| | IMDB | | | PCL | | | FNS | | | 20N | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Preprocessing** | **NB** | **SVM** | **LR** | **NB** | **SVM** | **LR** | **NB** | **SVM** | **LR** | **NB** | **SVM** | **LR** |
| **DON** | 0.767 | **0.835** | 0.798 | 0.726 | **0.729** | **0.693** | 0.685 | 0.630 | 0.640 | 0.040 | **0.160** | **0.140** |
| **LOW** | 0.771 | 0.831 | 0.801 | **0.736** | 0.696 | 0.668 | 0.695 | 0.665 | 0.650 | 0.040 | 0.140 | 0.100 |
| **RSW** | 0.787 | 0.831 | **0.833** | 0.719 | 0.651 | 0.686 | 0.705 | **0.715** | 0.660 | 0.020 | 0.100 | 0.060 |
| **STM** | 0.741 | 0.794 | 0.773 | 0.683 | 0.678 | 0.691 | 0.675 | 0.645 | 0.640 | 0.040 | **0.160** | 0.080 |
| **LOW → RSW** | 0.787 | 0.828 | **0.833** | 0.706 | 0.671 | 0.683 | 0.720 | 0.690 | 0.680 | 0.040 | 0.140 | 0.040 |
| **LOW → STM** | 0.725 | 0.803 | 0.770 | 0.678 | 0.668 | 0.688 | 0.700 | 0.665 | 0.615 | 0.040 | 0.120 | 0.100 |
| **RSW → LOW** | **0.789** | **0.835** | 0.820 | 0.721 | 0.663 | 0.691 | **0.725** | 0.690 | 0.675 | 0.040 | 0.120 | 0.020 |
| **RSW → STM** | 0.780 | 0.794 | 0.811 | 0.671 | 0.641 | 0.656 | 0.680 | 0.695 | 0.675 | 0.020 | **0.160** | 0.100 |
| **STM → LOW** | 0.725 | 0.803 | 0.800 | 0.678 | 0.668 | 0.673 | 0.700 | 0.665 | 0.635 | 0.040 | 0.120 | 0.060 |
| **STM → RSW** | 0.775 | 0.790 | 0.821 | 0.681 | 0.641 | 0.646 | 0.675 | 0.675 | 0.670 | 0.020 | 0.140 | 0.120 |
| **LOW → STM → RSW** | 0.750 | 0.799 | 0.820 | 0.678 | 0.623 | 0.648 | 0.695 | 0.680 | 0.645 | 0.040 | 0.140 | 0.080 |
| **LOW → RSW → STM** | 0.747 | 0.794 | 0.821 | 0.668 | 0.636 | 0.661 | 0.700 | 0.685 | 0.650 | 0.040 | 0.140 | 0.080 |
| **STM → LOW → RSW** | 0.749 | 0.797 | 0.814 | 0.678 | 0.623 | 0.661 | 0.690 | 0.675 | 0.645 | 0.040 | 0.140 | 0.080 |
| **STM → RSW → LOW** | 0.749 | 0.797 | 0.814 | 0.678 | 0.623 | 0.661 | 0.690 | 0.685 | 0.655 | 0.040 | 0.140 | 0.080 |
| **RSW → LOW → STM** | 0.757 | 0.797 | 0.807 | 0.673 | 0.623 | 0.678 | 0.720 | 0.670 | 0.655 | 0.040 | 0.140 | 0.120 |
| **RSW → STM** | 0.756 | 0.797 | 0.803 | 0.673 | 0.623 | 0.651 | 0.720 | 0.675 | **0.685** | 0.040 | **0.160** | 0.080 |

# Figurative Language in Tweets `fr` [Choi, 2020]

| | Logistic Regression | | Decision Tree | | MNB | | KNN | | Random Forest | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Count | Tfidf | Count | Tfidf | Count | Tfidf | Count | Tfidf | Count | Tfidf |
| **DON** | **50.20** | **52.03** | **50.41** | **42.89** | **51.42** | **52.24** | **38.82** | **45.73** | **53.25** | **51.22** |
| **RPT** | 50.41 | 52.64 | 48.37 | 44.72 | 50.81 | 51.63 | 38.21 | 45.53 | **53.05** | **52.64** |
| **RSW** | **52.24** | 53.86 | 45.93 | 44.11 | 51.22 | 52.24 | 37.40 | 44.31 | 50.00 | 50.20 |
| **ACC** | 49.59 | 52.64 | 49.39 | 43.29 | 51.02 | 52.03 | 35.16 | 45.53 | 52.44 | 52.03 |
| **URL** | **47.56** | **47.36** | **39.43** | **39.43** | **50.20** | **50.61** | **34.35** | **41.46** | **45.53** | **44.51** |
| **LEM** | 50.20 | **54.07** | **49.19** | 44.72 | **52.24** | **53.25** | **39.02** | 45.53 | 50.41 | 51.63 |
| **STM** | 51.63 | 53.86 | 48.37 | **45.93** | 52.03 | 52.44 | 38.41 | **46.75** | 52.44 | 51.42 |

**Table 1** – Average accuracy (in blue : best result, in red : worst result for each classifier)

# Figurative Language in Tweets `fr` (Choi 2020)

| Classifier | Count Vectorizer | Macro F1-score | Tfidf Vectorizer | Macro F1-score |
|---|---|---|---|---|
| **Logistic Regression** | LEM, RSW | 53.53 | LEM, RSW, RAC | 54.35 |
| **Decision Tree** | RPT, accents, RAC, RSW | 49.59 | RAC, RPT | 48.58 |
| **MNB** | LEM, RSW, RAC | 54.59 | LEM, RSW, RAC | **55.89** |
| **KNN** | RAC, RSW, RPT | **38.20** | RAC, RSW | 47.35 |
| **Random Forest** | LEM, RSW, accents, RAC | 51.38 | LEM, RSW, accents, RAC | 53.25 |

**Table 2** – Best macro F1-scores (in blue : best result, in red : worst result. Best result of DEFT2017 : 65%)

# Combinations of preprocessing steps

A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis [Symeonidis et al., 2018]

# Combinations of preprocessing steps

A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis [Symeonidis et al., 2018]

Influence of preprocessing on text classification – Application to tweet polarity classification [Choi, 2020]

# Combinations of preprocessing steps

A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis [Symeonidis et al., 2018]

Influence of preprocessing on text classification – Application to tweet polarity classification [Choi, 2020]

**What do we learn ?**

- Two preprocessing steps can interact negatively
- Performance gains tend to be asymptotic

# Combinations of preprocessing steps

A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis [Symeonidis et al., 2018]

Influence of preprocessing on text classification – Application to tweet polarity classification [Choi, 2020]

**What do we learn ?**

- Two preprocessing steps can interact negatively
- Performance gains tend to be asymptotic
- There is no universal "cocktail" that works regardless of :
    - the task
    - the type of texts
    - the classifier
    - the language model

## Let's put that in practice

https://github.com/rundimeco/Preprocessing_NLP :

- These Slides in PDF
- A simple notebook to compare quickly different pre-processing techniques :
- 01_run_experiments_simple_single_task.ipynb (Kaggle dataset)
- Another example with a multilingual task (using corpus_muli.zip)
- 
- Now it's up to you to find the best pre-processing configuration with another multilingual dataset :
- https://www.kaggle.com/datasets/suraj520/multi-task-learning
- https://www.kaggle.com/datasets/azimulh/tweets-data-for-authorship-attribution-modelling
- The objective is to try different classification models (ML, language models) with differ preprocessing configurations in order to find :
- What preprocessing combinations work best
- How much this depends on the learning method

Choi, H.-S. (2020).
**Influence des pré-traitements sur la classification de textes - application à la classification de tweets selon leur polarité.**
Master's thesis, Sorbonne Université, France.

Millour, A. (2020).
**Myriadisation de ressources linguistiques pour le TA de langues non standardisées.**
PhD thesis, Sorbonne Université, France.

Rust, P., Pfeiffer, J., Vulic, I., Ruder, S., and Gurevych, I. (2020).
**How good is your tokenizer ? on the monolingual performance of multilingual language models.**
*CoRR*.

Siino, M., Tinnirello, I., and La Cascia, M. (2024).
**Is text preprocessing still worth the time ? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers.**
*Information Systems*, 121 :102342.

Symeonidis, S., Effrosynidis, D., and Arampatzis, A. (2018).
**A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis.**
*Expert Systems with Applications*, 110 :298–310.