POLITECNICO DI TORINO
Repository ISTITUZIONALE

Investigating Large Language Models' Linguistic Abilities for Text Preprocessing

Original
Investigating Large Language Models' Linguistic Abilities for Text Preprocessing / Braga, M., Carlo Milanese, G., Pasi, G.. - (2025), pp. 136-143. (2025 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) London (UK) 15-18 November 2025) [10.1109/WI-IAT67162.2025.00026].

Availability:
This version is available at: 11583/3010273 since: 2026-04-26T15:36:35Z

Publisher:
IEEE

Published
DOI:10.1109/WI-IAT67162.2025.00026

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

28 November 2026

Investigating Large Language Models' Linguistic Abilities for Text Preprocessing

Marco Braga
University of Milano-Bicocca
Milan, Italy
Politecnico di Torino
Turin, Italy
m.braga@campus.unimib.it

Gian Carlo Milanese
University of Milano-Bicocca
Milan, Italy
giancarlo.milanese@unimib.it

Gabriella Pasi
University of Milano-Bicocca
Milan, Italy
gabriella.pasi@unimib.it

***Abstract*—Text preprocessing is a fundamental component of Natural Language Processing, involving techniques such as stopword removal, stemming, and lemmatization to prepare text as input for further processing and analysis. Despite the context-dependent nature of the above techniques, traditional methods usually ignore contextual information. In this paper, we investigate the idea of using Large Language Models (LLMs) to perform various preprocessing tasks, due to their ability to take context into account without requiring extensive language-specific annotated resources. Through a comprehensive evaluation on web-sourced data, we compare LLM-based preprocessing (specifically stopword removal, lemmatization and stemming) to traditional algorithms across multiple text classification tasks in six European languages. Our analysis indicates that LLMs are capable of replicating traditional stopword removal, lemmatization, and stemming methods with accuracies reaching 97%, 82%, and 74%, respectively. Additionally, ML models trained on LLM-preprocessed texts are on par with and occasionally surpass the performance of models trained on traditionally preprocessed data, with gains of up to 6% in the F_1 measure. Our code, prompts, and results are publicly available at https://github.com/GianCarloMilanese/llm_pipeline_wi-iat.**

***Index Terms*—Large Language Models, Web Data, Data Cleaning, Text Classification**

I. Introduction

Text preprocessing is a fundamental step in Natural Language Processing (NLP), involving techniques such as stopword removal, stemming, and lemmatization to standardize text for further processing or downstream tasks, including input preparation for Machine Learning (ML) algorithms. By reducing text to its basic features, text preprocessing decreases the computational cost of the subsequent processing and mitigates noise and irrelevant information [1]. The choice of preprocessing strategy can significantly impact downstream performance, sometimes enabling even simple models to outperform complex transformer-based architectures [2].

Several preprocessing techniques, such as stopword removal and lemmatization, are inherently context-dependent. Indeed, what qualifies as a stopword often varies across tasks and domains, as each is characterized by a distinct word distribution. Additionally, the context of a text is crucial in determining whether a word should be treated as a stopword [1]. Similarly, in lemmatization, the part of speech of a word often determines how it should be processed: for instance, the word "saw" may be reduced to either "see" or "saw" depending on whether it functions as a verb or a noun. Moreover, the broader context of a document is also valuable for accurate lemmatization, as word meanings can shift significantly based on the subject matter. For example, the noun "leaves" could be lemmatized to "leaf" in a document about botany, but it would be lemmatized to "leave" in a text about employee absences. As the above examples show, text preprocessing depends not only on the task at hand or on the part of speech of a word, but also on the broader context of a sentence or document. However, traditional preprocessing techniques rely only marginally on contextual information. Indeed, they often make use of predefined stopwords lists and stemming or lemmatization rules that overlook domain-specific information. These issues highlight the need for techniques that enable a more context-sensitive text preprocessing. To fill this gap, we investigate the ability of pre-trained Large Language Models (LLMs) to preprocess a text. Due to their ability to take the linguistic context into account [3], [4], [5], [6] without requiring extensive language-specific annotated resources, we hypothesize that LLMs can dynamically detect stopwords, lemmas and stems based on the input document, context and task. Although prior work by [7] has examined the role of LLMs in stemming within information retrieval pipelines, the reported study primarily focuses on retrieval effectiveness rather than the quality of preprocessing itself. In this paper, we thoroughly investigate the ability of LLMs to perform text preprocessing, guided by the following research questions: **(RQ1)** How effectively can pre-trained LLMs perform stopword removal, stemming, and lemmatization, and how does their performance vary across different languages? **(RQ2)** Does the use of LLMs for text preprocessing, as opposed to traditional methods, improve the performance on downstream tasks?

To address these questions, we employ recent LLMs, namely Gemma-2 [8] and -3 [9], LLama-3 [10], Phi-4 [11], and Qwen-2.5 [12], and we instruct them to remove stopwords, and to lemmatize or stem a document given a few examples and the task we are tackling. Furthermore, to comparatively evaluate the effectiveness of LLM-based preprocessing, we train three different ML-based classification models by using data preprocessed by the LLMs. In detail, we preprocess data from

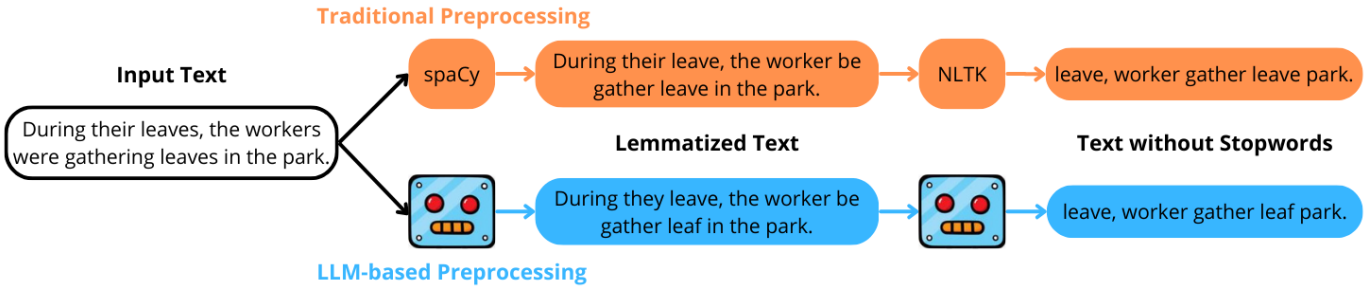


Fig. 1. Example of traditional vs. LLM-based text preprocessing. In this case, the LLM correctly disambiguates the word "leaves", distinguishing between employee absences and foliage in its two occurrences, and applies lemmatization accordingly.

multiple sources, e.g., English Social Networks and News, to study the impact of LLM preprocessing on both dirty and clean data. Furthermore, we analyse the preprocessing of both English-only and multilingual LLMs on six European Languages, i.e. English, French, German, Italian, Portuguese and Spanish, to understand the impact of LLM preprocessing on different languages. Our analysis shows that LLMs can replicate traditional stopword removal, lemmatization, and stemming methods with accuracies of up to 97%, 82%, and 74%, respectively. Furthermore, we note that ML algorithms trained on texts preprocessed by LLMs achieve an improvement of up to 6% with respect to the F_1 measure compared to traditional techniques. The paper is organised as follows: after discussing the related works in Section II, Section III presents in detail the methodology of our analysis, while in Section IV we describe the experimental setup used. Then, Section V discusses the results of our evaluation, and Section VI addresses the limitations of using LLMs for preprocessing. Finally, Section VII concludes the study and outlines future research directions.

II. Related Works

LLMs have achieved state-of-the-art performance across a wide range of tasks and research fields [13], [14]. They are particularly effective in few-shot settings, where they can be applied to unseen tasks or domains without requiring additional supervised fine-tuning [15], [16], [17], [18], which demands a large amount of computational resources [19], [20] and labelled data that are not always available [21]. The relationships between preprocessing operations, such as lemmatization and stopword removal, and the context of input texts has been studied for a long time [22], [23], [1], [24], [25], [26]. For instance, [1] shows how to define context-specific stopwords within an information retrieval pipeline: removing context-specific stopwords achieves higher performance compared to removing them from a predefined list. Recently, LLMs have been applied in a few-shot scenario for stemming queries and documents in an information retrieval pipeline [7]. The authors found that, while LLM-based stemming alone does not improve retrieval performance, using LLMs to identify named entities that should not be stemmed leads to significant improvements. However, no prior work has conducted a comprehensive analysis of LLMs for text preprocessing - including stopword removal, lemmatization, and stemming - by comparing their outputs to those produced by traditional methods, and by assessing their impact on text classification. The study presented in this paper aims to fill that gap.

III. Methodology

Our investigation into the text preprocessing capabilities of LLMs involves defining prompts that guide these models through each preprocessing task. In detail, the LLMs are provided with (i) a formal description of the target preprocessing operation, (ii) a few examples of how it should be performed, (iii) the text to be preprocessed, (iv) the language of the text, and (v) the context of the downstream task that we are addressing. The text is directly fed into the LLMs, which output the corresponding preprocessed version. Note that our methodology relies on in-context learning, as we provide the LLMs with a few examples of stopwords, lemmas and stems inside the prompt. With respect to stopword removal, we additionally instruct the LLMs to retain certain context- and task-specific words that are generally considered stopwords. For example, in the sentiment analysis tasks, the LLMs are instructed to keep the word "not" in the text, due to its key role in determining polarity. Additionally, we evaluate our method across multiple languages – English, French, German, Italian, Portuguese and Spanish – to investigate cross-linguistic performance. For each non-English language, we perform experiments with the same prompts written both in English and in that specific language to assess whether using the native language offers additional contextual benefits. To address RQ1, as shown in Figure 1, we compare the output of each LLM with the one produced from the same text preprocessed by using traditional methods. Specifically, these include removing words from a predefined stopwords list, applying stemming algorithms such as Porter [27], Lancaster [28], and Snowball [29], and utilizing off-the-shelf implementations of rule-based or edit tree lemmatizers [30]. With respect to RQ2, we analyze the impact of preprocessing on downstream classification tasks. We represent the preprocessed texts as bag-of-words with TF-IDF [31]. Then, we train three well-known ML algorithms, i.e. Decision Tree [32], Logistic Regression [33], and Naive Bayes [34]. To assess the overall impact of text preprocessing

across the previously mentioned ML algorithms, we average the single models' performances.

IV. Experimental Setup

In this section, we describe the datasets, the evaluation metrics and the models used to assess the effectiveness of LLM-based preprocessing.

a) Datasets: We select a suite of publicly available datasets encompassing binary and multiclass classification tasks across multiple languages, including English, French, German, Italian, Portuguese and Spanish. Specifically, for the evaluation of texts in English, we use the Twitter datasets from SemEval-18 on emoji prediction [35] and irony detection [36], as well as from SemEval-19 on hate detection [37], offensive language identification [38] and sentiment analysis [39]. In addition, we evaluate LLM-based preprocessing on the task of News classification using the Reuters [40] and AG News [41] datasets. Our main focus is on web-sourced data from platforms like Twitter, which present unique challenges due to their informal language and higher noise levels [42]. Unlike curated news content, which tends to be cleaner and more uniform, Twitter data typically require more extensive preprocessing to handle issues such as misspellings and the presence of hashtags. For non-English languages, we employ five datasets from the Tweet Sentiment Multilingual corpus [43]. Due to high computational costs, we randomly sample up to 3000 documents for training and 3000 documents for evaluation while keeping the original class distributions. Additionally, we create a validation set of 2000 documents, extracted from the original SemEval-19 sentiment analysis training set. These documents are used for tuning the hyperparameters of the ML algorithms.

b) Models: We compare five open source state-of-the-art LLMs, encompassing different sizes and architectures: Gemma-2-9B [8], Gemma-3-4B [9] LLama-3.1-8B [10], Phi-4-mini (3.8B parameters) [11], and Qwen-2.5-7B [12] in their instruction-tuned version. While Gemma-2 and Phi-4 have been primarily trained on English data, Gemma-3, Llama-3.1, Qwen-2.5 are natively multilingual, supporting Italian, Spanish, French, German and Portuguese. We rely on the Hugging Face library to run the models, we set the temperature to 0.7 and, while generating texts, we use Sample Decoding (i.e. `do_sample=True`). Table I provides examples of prompts used for lemmatization, stemming, and stopword removal in English texts. These examples are based on the SemEval-19 sentiment analysis dataset, with prompts for other datasets being either straightforward adaptations or, in the case of multilingual datasets, translations of those shown here.

c) Baselines: traditional preprocessing: We employ the stopword lists and stemmers provided by NLTK, and the lemmatizers provided by spaCy. The word "not" and language-specific negation lexicon are removed from the NLTK's stopwords lists when preprocessing the SemEval and Twitter Sentiment Multilingual datasets.

d) Machine Learning algorithms: We use the scikit-learn implementations of the Multinomial Naive Bayes [34], Decision Tree [32], and Logistic Regression [33] algorithms.

e) Evaluation metrics: With respect to RQ1, for each preprocessing operation, we evaluate the accuracy of LLM-based preprocessing by computing the percentage of words in a text that are processed by the LLM in the same way as the corresponding traditional method. Regarding RQ2, we use the micro F_1 measure for evaluating the performance of the considered ML classification algorithms.

f) Hyperparameters settings: To ensure a fair evaluation, we optimize the TF-IDF hyperparameters, such as the number of features and the n-grams length, on the Semeval-19 sentiment analysis validation set using traditional preprocessing methods. These optimized settings are consistently applied to both traditionally processed and LLM-preprocessed text.

V. Results

In Section V-A, we examine the extent to which LLMs are able to replicate traditional preprocessing techniques. As mentioned in the introduction, LLMs may identify different stopwords, stems, and lemmas compared to traditional techniques, due to their ability to manage contextual information. We investigate whether this leads to improved performance in text classification (RQ2) in Section V-B.

A. LLMs' preprocessing abilities

Tables II and III compare the preprocessing output produced by the LLMs with that produced by traditional methods. Specifically, *SW* refers to the percentage of words removed by the LLM that match NLTK's stopwords list, while *NSW* measures the percentage of words removed by the LLM among those that are *not* considered stopwords by NLTK. Additionally, *L* and *S* represent the percentage of words that are respectively lemmatized and stemmed by the LLM exactly like the corresponding traditional techniques. For stemming in English (Table II), the LLMs are first compared against each of the Porter, Lancaster and Snowball algorithms, then they are compared against the three algorithms collectively (i.e., the LLM's output is valid if it matches the output of *any* of the three algorithms). The reported values are averages over all texts in the same language. These measures assess the similarity between LLM-based preprocessing and traditional techniques, with the best-performing LLM being the one that maximizes *SW*, *L* and *S*, while minimizing *NSW*. The best scores within each dataset and preprocessing type are highlighted in **bold** in Tables II and III. Since Gemma-2 is trained primarily on English data and it is the model with the largest number of parameters, it would be expected to perform best on English texts. Indeed, Gemma-2 consistently outperforms all other models in stopword removal, lemmatization, and stemming on English texts. However, Phi-4 shows the most conservative behaviour in non-stopword removal, achieving the lowest rate of non-stopwords removed compared to the other LLMs. Notably, this pattern is mostly consistent across all the analyzed languages: Gemma-2 outperforms all the other LLMs in stopword removal and stemming (the only exception being Gemma-3 for stopword removal in Portuguese), while Phi-4 removes the fewest percentage of non-stopwords. The

TABLE I
PROMPTS USED TO LEMMATIZE, STEM AND REMOVE STOPWORDS FROM THE TEXTS OF THE SEMEVAL SENTIMENT DATASET.

Stopword removal
You specialize in removing stopwords from text. Stopwords are words that are not relevant for processing a text. Stopwords typically include articles, prepositions, pronouns, and auxiliary verbs. For example, the words 'is', 'are', 'being', 'you', 'me', 'the', 'an', 'and', 'I', 'which', 'that', 'have', 'by', 'for' and their alternative forms are usually considered stopwords. Note that whether a word is a stopword or not depends on the context of the text or of an application. In this case, the relevant task is detecting the sentiment of a tweet (positive, negative or neutral). In this task, the word 'not' is often not considered a stopword, and it should be kept in the text. Please provide a version without stopwords of the following paragraph: '{paragraph}'. Print only the paragraph without stopwords, do not add any explanation, details or notes.
Lemmatization
You specialize in text lemmatization. Text lemmatization is a natural language processing technique that is used to reduce words to their lemma, also known as the dictionary form. The process of lemmatization is used to normalize text and make it easier to process. For example, the verbs 'is', 'are', and 'being' must all be reduced down to the common lemma 'be'. As another example, "he's going" must be lemmatized to "he be go". Lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighbouring sentences or even an entire document. Please provide the lemmatized version of this paragraph: '{paragraph}'. Print only the lemmatized paragraph, do not add any explanation, details or notes.
Stemming
You specialize in text stemming. Text stemming is a natural language processing technique that is used to reduce words to their base form, also known as the root form. The process of stemming is used to normalize text and make it easier to process. For example, the words 'programming,' 'programmer,' and 'programs' can all be reduced down to the common stem 'program'. As another example, the words 'argue', 'argued', 'argument', 'arguing', and 'arguer' all stem to 'argu'. Please provide the stemmed version of this paragraph: '{paragraph}'. Print only the stemmed paragraph, do not add any explanation, details or notes.

TABLE II
ACCURACY OF DIFFERENT LLMS IN PERFORMING TEXT PREPROCESSING IN ENGLISH.

Model	**SW**	**NSW**	**L**	**S**			
				Porter	**Lanc.**	**Snow.**	**Any**
Gemma-2	**84.29**	13.95	**82.61**	**74.93**	**63.15**	**75.65**	**81.14**
Gemma-3	75.17	22.92	77.48	69.75	57.46	71.53	74.51
Llama-3.1	81.61	30.04	79.20	66.54	57.74	68.05	73.14
Phi-4	43.53	**12.74**	78.56	61.61	51.53	63.22	65.81
Qwen-2.5	79.37	22.15	82.37	61.11	53.08	61.95	67.36

lemmatization results, however, are not clear-cut: while Qwen-2.5 achieves the best scores in French, Italian, and Spanish – and performs comparably to Gemma-2 in English – Gemma-3 and Phi-4 outperform the other models in German and Portuguese, respectively. It is also interesting to highlight that Gemma-3 – despite being roughly half the size of Gemma-2, Llama-3.1, and Qwen-2.5 – consistently achieves performance comparable to, and occasionally surpassing, that of the larger models. Additionally, we note that language-specific prompts usually achieve the highest *SW* scores, while the specification of the same prompts in English in most cases produces the best lemmatization (*L*) and stemming (*S*) performance. Overall, there is no consistent evidence that prompting in the target language leads to better results. Specifically, Gemma-3 and Llama-3.1 show improved performance with language-specific prompts in 55% and 60% of cases, respectively, whereas Gemma-2, Qwen-2.5, and Phi-4 perform better with English prompts in 65% of the cases analyzed.

We further observe that LLMs often eliminate words not traditionally considered stopwords (*NSW* column). This behaviour supports our hypothesis that LLMs' contextual understanding influences stopword selection. For instance, "user" is frequently removed, which is reasonable given that the datasets include social media text from Twitter [43]. Regarding stemming, the lower overall scores compared to traditional preprocessing may be due to LLMs generating different stems for the same word across different texts. Although this deviates from traditional stemming rules, it might allow for a more context-specific preprocessing, as also observed by [7].

To concretely answer RQ1, these results show that LLMs are quite effective at identifying stopwords across multiple languages, with Gemma-2 detecting more than 97% of stopwords in French texts, and at least 79% in other languages. Additionally, they show strong lemmatization capabilities in English, with all the evaluated models correctly identifying over 77% of lemmas.

B. Text classification

Tables IV and V report the average performance score of the three ML models that we have trained on English (Table IV) and non-English texts (Table V), by applying the LLM-based preprocessing and traditional preprocessing methods. Each column corresponds to a specific preprocessing task: *SW* denotes stopword removal, *SW+L* applies lemmatization followed by stopword removal, *L* represents lemmatization alone, *SW+S* combines stopword removal and stemming, and *S* applies stemming only. For each dataset and preprocessing task, the best results are highlighted in **bold**, while the second-best

TABLE III

ACCURACY OF DIFFERENT LLMS IN PERFORMING TEXT PREPROCESSING IN FIVE EUROPEAN LANGUAGES. FOR EACH PREPROCESSING OPERATION, THE VALUES ON THE LEFT AND RIGHT REFER TO THE SCORES OBTAINED WITH AN ENGLISH PROMPT AND WITH A LANGUAGE-SPECIFIC PROMPT, RESPECTIVELY.

Language	Model	SW		NSW		L		S (Snowball)	
French	Gemma-2	96.83	**97.94**	28.02	33.12	61.06	53.28	**51.51**	34.47
	Gemma-3	95.01	68.46	34.18	25.30	63.93	62.64	50.91	34.28
	Llama-3.1	65.02	32.01	37.20	23.11	54.86	55.47	45.80	34.15
	Phi-4	38.77	11.37	14.70	**9.43**	62.69	63.99	47.31	36.26
	Qwen-2.5	82.98	86.33	31.72	31.32	**65.70**	61.00	46.86	43.05
German	Gemma-2	74.52	**79.73**	24.35	32.99	59.80	64.45	**68.46**	61.06
	Gemma-3	77.63	53.32	41.28	33.51	64.95	**67.71**	65.46	58.32
	Llama-3.1	58.08	25.65	35.44	**16.54**	58.38	60.07	58.54	57.32
	Phi-4	25.74	30.33	17.46	20.84	57.24	58.90	50.84	45.30
	Qwen-2.5	57.23	47.04	31.29	22.88	64.84	64.29	53.17	48.86
Italian	Gemma-2	86.48	**89.20**	20.82	22.36	59.30	58.31	**56.65**	50.86
	Gemma-3	86.45	86.70	31.20	26.98	63.37	61.66	46.29	48.80
	Llama-3.1	67.27	62.73	31.51	25.78	53.22	51.31	39.83	40.72
	Phi-4	28.28	19.06	15.40	**13.76**	60.21	60.47	38.51	34.13
	Qwen-2.5	69.67	84.88	28.18	33.06	**63.92**	62.27	47.30	46.66
Portuguese	Gemma-2	83.08	86.05	23.38	31.17	65.44	60.01	50.20	**57.41**
	Gemma-3	**89.03**	72.96	41.74	32.54	64.20	61.78	48.24	49.33
	Llama-3.1	65.00	72.52	23.94	36.58	61.80	62.16	44.04	47.80
	Phi-4	34.75	49.13	**15.10**	19.11	**70.42**	68.92	42.02	36.70
	Qwen-2.5	68.30	76.09	25.06	32.79	70.17	69.02	49.50	46.74
Spanish	Gemma-2	85.91	**87.90**	20.46	26.68	57.86	57.68	**63.76**	62.14
	Gemma-3	83.67	69.84	28.54	25.82	58.98	61.14	54.31	55.31
	Llama-3.1	69.99	33.53	29.02	21.44	51.02	52.11	48.69	51.90
	Phi-4	28.42	18.66	19.06	**14.94**	55.26	55.02	45.86	44.24
	Qwen-2.5	67.46	75.55	26.84	25.81	**62.30**	61.11	54.78	52.05

scores are underlined. The best result within each dataset is marked with a †.

a) English language: We first note that LLMs outperform traditional methods in all datasets except for Sentiment, and more specifically in 25 out of the 35 examined combinations of datasets and preprocessing tasks. Moreover, in 80% of these cases, the second-best result is also achieved by an LLM. Traditional preprocessing outperforms LLM-based preprocessing in 10 out of 35 combinations, with a margin greater than 1 point in F_1 in 5 of them (Offensive SW+S and S, Sentiment SW+S and S, and AG News S). Notably, LLMs achieve the highest performance in stopword removal combined with lemmatization in all the proposed datasets, indicating their ability to dynamically identify task-relevant stopwords and lemmas in a more context-sensitive manner than traditional techniques. In particular, Gemma-2 achieves a 6.16% improvement over traditional techniques in the AG News dataset. Additionally, LLM-based preprocessing outperforms traditional stopword removal and lemmatization in 6 out of 7 datasets. The only exception is the Sentiment dataset, where Gemma-2 shows a slight underperformance.

Our results indicate however that stemming with LLMs is not as effective as other preprocessing operations: indeed LLM-based preprocessing outperforms traditional stemming in only 3 out of the 7 datasets. Several factors may contribute to this outcome. First, stemming is a task where context plays a limited role, making it less sensitive to the contextual capabilities of LLMs. This aligns with findings by [7], who show that LLMs' stemming performance is suboptimal in an information retrieval pipeline. Furthermore, we note that LLMs exhibit inconsistencies in stemming across documents. Unlike traditional algorithms, which apply fixed rules, LLMs may stem the same word differently depending on context. For instance, in some cases, an LLM may generate a stem that matches the Porter stemmer, while in others, it may align with the Lancaster stemmer or be completely different. This lack of consistency results in non-standardized text representations, which can negatively impact downstream tasks such as lexical feature extraction, and consequently their classification performance.

b) Non-English languages: Table V presents the performance of text classification across French, German, Italian, Portuguese and Spanish. Overall, LLMs achieve performance on par (within 1 point) with or even better than traditional techniques in half of the evaluated cases. Notably, LLMs achieve the highest performance in 4 out of 5 datasets when lemmatization is applied, showing their ability to understand contextual information even in non-English languages. Moreover, LLMs achieve the highest performance in the Italian and Spanish datasets (marked with †) and perform only marginally lower than the best score in French and German. Interestingly, Gemma-2 and -3 outperform traditional methods even in stemming for Italian and German, respectively. This finding contrasts with the one observed in the English setting. For Spanish, Gemma-2 also shows a significant improvement in stopword removal, underscoring its ability to identify stopwords based on context. Moreover, in Spanish, traditional preprocessing

TABLE IV

COMPARISON OF LLM-BASED AND TRADITIONAL PREPROCESSING ON SEVERAL TEXT CLASSIFICATION TASKS. THE SCORES ARE AVERAGES OF THE RESULTS OBTAINED WITH THREE DIFFERENT ML ALGORITHMS. WHILE APPLYING TRADITIONAL STEMMING, WE REPORT THE VALUES OF THE PORTER | LANCASTER | SNOWBALL STEMMERS, FOLLOWING THIS ORDER.

Dataset	**Model**	**SW**	**SW + L**	**L**	**SW + S**	**S**
					Porter | Lanc. | Snow.	Porter | Lanc. | Snow.
Emoji	Traditional	21.15	21.41	21.42	21.06 | 21.11 | 21.03	21.01 | 20.96 | 20.88
	Gemma-2	21.52	**22.61**†	**21.66**	21.19	21.00
	Gemma-3	**22.00**	22.17	21.09	21.28	21.27
	Llama-3.1	21.71	21.99	21.22	20.51	20.99
	Qwen-2.5	21.63	22.53	21.47	20.97	20.60
	Phi-4	21.73	21.90	21.46	**21.58**	**21.49**
Hate	Traditional	48.73	48.93	49.67	49.47 | 49.40 | 47.87	46.99 | 47.74 | 47.82
	Gemma-2	49.61	47.47	49.52	49.77	49.24
	Gemma-3	**50.87**	**50.68**	50.81	**50.93**	49.49
	Llama-3.1	49.50	49.75	**51.31**†	50.45	50.15
	Qwen-2.5	50.80	49.09	48.93	49.61	50.34
	Phi-4	49.38	49.75	49.91	50.67	**50.58**
Irony	Traditional	61.05	60.11	59.73	61.96 | **63.01** | 61.39	60.96 | 62.15 | 59.73
	Gemma-2	61.64	**62.63**	61.14	59.40	60.63
	Gemma-3	62.20	62.20	61.73	61.01	**62.88**
	Llama-3.1	61.44	62.29	**63.35**	59.31	58.80
	Qwen-2.5	61.99	61.22	63.18	57.95	59.35
	Phi-4	**64.50**†	59.14	61.82	62.50	61.86
Offensive	Traditional	75.62	74.53	73.19	75.73 | **75.93** | 74.61	74.22 | 75.46 | **75.65**
	Gemma-2	74.81	**74.88**	73.02	73.95	72.71
	Gemma-3	73.37	73.44	71.59	72.63	71.98
	Llama-3.1	**76.71**†	73.95	71.47	73.76	71.20
	Qwen-2.5	74.03	73.84	**74.38**	72.40	71.36
	Phi-4	74.77	74.26	72.09	73.57	70.50
Sentiment	Traditional	**48.89**†	48.05	**48.71**	47.81 | **48.54** | 48.18	47.89 | **48.62** | 48.61
	Gemma-2	48.13	47.77	48.35	46.59	47.39
	Gemma-3	47.60	**48.80**	47.27	43.64	46.47
	Llama-3.1	47.96	48.13	48.02	45.80	47.06
	Qwen-2.5	46.98	45.54	46.84	46.04	46.38
	Phi-4	47.24	47.96	46.95	46.52	45.31
AG News	Traditional	61.44	62.21	62.15	61.87 | 61.71 | 61.78	60.24 | 61.64 | **61.78**
	Gemma-2	**63.04**	**66.04**†	62.44	**62.66**	60.67
	Gemma-3	62.89	63.69	61.34	61.98	59.37
	Llama-3.1	60.44	61.38	60.37	56.79	57.55
	Qwen-2.5	62.90	63.44	**62.69**	58.16	55.50
	Phi-4	62.86	63.07	60.45	60.33	57.84
Reuters	Traditional	85.76	85.90	86.40	**85.83** | 85.82 | 85.73	85.37 | **86.19** | 85.53
	Gemma-2	**86.00**	**87.28**†	**87.00**	84.87	85.83
	Gemma-3	83.73	86.60	86.66	82.80	85.78
	Llama-3.1	83.85	85.14	84.86	81.56	83.24
	Qwen-2.5	84.42	86.74	85.85	80.11	82.99
	Phi-4	83.55	86.23	86.41	82.30	85.17

outperforms LLMs by only 1 point in stemming, and similarly in stopword removal when combined with lemmatization or stemming.

Notably, the performance of Llama-3.1 improves when using language-specific prompts in 80% of the preprocessing tasks. For both Gemma-2 and Gemma-3 this is instead true in more than 60% of the analyzed combinations, and for Qwen-2.5 and Phi-4 only in 40%. This finding is unexpected, given that Qwen-2.5 is inherently multilingual, while Gemma-2 is mostly trained on English data.

VI. LIMITATIONS

Regarding RQ1, the ability of LLMs to perform text preprocessing is evaluated by comparing their outputs to those generated by well-known Python libraries, such as NLTK and spaCy. There may however be instances where LLMs outperform these libraries – for example, by splitting a long hashtag like "`#illegalaliens`" and correctly lemmatizing it as "`illegal alien`" – that are not accounted for in the evaluation

TABLE V

COMPARISON OF LLM-BASED AND TRADITIONAL PREPROCESSING ON THE TASK OF SENTIMENT ANALYSIS IN FIVE EUROPEAN LANGUAGES. THE SCORES ARE AVERAGES OF THE RESULTS OBTAINED WITH THREE DIFFERENT ML ALGORITHMS. FOR EACH COMBINATION OF PREPROCESSING OPERATIONS, THE VALUE ON THE LEFT REFERS TO THE SCORE OBTAINED WITH THE ENGLISH PROMPT, AND THE ONE ON THE RIGHT REFERS TO THE SCORE OBTAINED WITH THE LANGUAGE-SPECIFIC ONE. TRADITIONAL STEMMING IS PERFORMED WITH THE SNOWBALL ALGORITHM.

Dataset	Model	SW		SW + L		L		SW + S		S	
French	Traditional	**52.95**		**52.49**		53.48		**53.98**†		**52.87**	
	Gemma-2	52.18	52.53	49.66	49.54	51.88	52.91	51.65	49.54	47.81	50.34
	Gemma-3	50.73	51.30	48.39	49.96	51.49	52.33	49.04	47.74	50.50	48.70
	Llama-3.1	50.57	52.07	47.13	48.20	50.19	51.38	46.05	47.66	49.04	48.00
	Phi-4	51.69	52.26	48.08	50.04	51.88	52.50	51.03	49.62	50.84	50.15
	Qwen-2.5	50.50	49.89	51.23	49.80	53.45	**53.52**	50.08	47.16	47.85	49.08
German	Traditional	**55.13**†		**52.80**		53.18		**53.52**		54.06	
	Gemma-2	48.47	49.77	50.04	46.93	53.56	50.26	50.27	49.31	**54.48**	52.07
	Gemma-3	48.93	49.46	46.93	47.55	53.14	**53.79**	48.51	48.74	53.64	52.26
	Llama-3.1	47.13	50.57	47.24	50.04	51.99	53.37	46.48	50.46	50.96	53.56
	Phi-4	50.92	51.11	51.49	49.20	52.64	53.45	50.46	48.62	52.99	51.07
	Qwen-2.5	46.59	49.54	47.73	49.27	52.30	51.88	45.86	46.05	51.46	48.35
Italian	Traditional	**51.84**		**52.07**		50.61		**51.30**		52.33	
	Gemma-2	48.16	48.73	45.59	47.16	51.34	50.61	45.51	44.80	52.30	52.53
	Gemma-3	46.74	46.90	44.48	46.25	52.03	51.26	46.36	46.48	**53.68**†	52.49
	Llama-3.1	46.44	46.56	44.94	48.47	52.72	51.46	42.34	43.87	48.62	51.92
	Phi-4	51.26	49.00	48.62	49.66	52.49	52.80	51.12	49.54	52.26	53.49
	Qwen-2.5	45.48	44.71	45.94	41.88	**53.37**	52.99	43.44	44.48	48.97	50.50
Portoguese	Traditional	**56.09**		**57.39**		**57.16**		**56.51**		**57.62**†	
	Gemma-2	54.44	**56.09**	52.80	53.53	54.06	55.02	51.34	53.83	54.25	54.94
	Gemma-3	54.14	54.44	53.10	53.71	54.33	54.94	53.64	53.87	54.48	54.48
	Llama-3.1	53.26	52.53	49.77	50.88	53.45	54.41	51.34	51.65	51.84	52.34
	Phi-4	52.34	52.07	51.92	50.19	53.98	53.64	48.54	50.22	50.04	50.61
	Qwen-2.5	52.37	53.49	52.91	51.58	55.17	53.03	52.03	47.70	51.00	49.46
Spanish	Traditional	47.47		**49.43**		48.47		**49.88**		**49.61**	
	Gemma-2	49.85	**49.92**†	48.31	48.70	47.05	48.08	47.47	48.74	48.43	48.74
	Gemma-3	47.36	47.62	47.01	46.82	**49.08**	47.78	46.82	46.09	46.63	47.32
	Llama-3.1	48.40	48.62	45.67	45.10	48.47	46.48	43.80	46.28	45.90	47.55
	Phi-4	47.93	46.51	46.25	44.67	46.44	47.51	47.28	46.32	44.71	45.52
	Qwen-2.5	49.39	47.70	49.08	45.75	48.12	48.74	46.67	45.33	46.86	45.86

metrics. Another limitation of LLM-based preprocessing is the high computational cost of using LLMs. For instance, applying Llama-3.1 on a paragraph requires 6.18 TFLOPs on average, which is significantly greater than the cost of applying traditional methods.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we investigate the capability of LLMs to perform text preprocessing, including stopword removal, lemmatization, and stemming. We conduct a comparative analysis of various LLMs, differing in both size and architecture, to assess their ability to replicate traditional preprocessing techniques across five languages. Additionally, we evaluate the impact of LLM-based preprocessing on multiple downstream Machine Learning classification tasks by training models on text preprocessed using traditional and LLM-based approaches. Our findings indicate that LLMs outperform traditional lemmatization techniques across most of the evaluated languages and datasets, and consistently improve stopword removal in English, both with and without lemmatization. However, LLMs do not appear to perform competitively in stemming. Although preprocessing for European languages has been extensively studied, we note that stemmers and lemmatizers for many other languages have received significantly less attention [44], often resulting in reduced effectiveness. Given the promising results achieved, future work will explore the potential of LLMs as stemming and lemmatization tools for low-resource languages, such as languages that lack the extensive amounts of annotated resources that are needed to develop or train lemmatizers.

ACKNOWLEDGMENT

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support. We also acknowledge Renzo Alva Principe, Rafael Peñaloza and Effrosyni Sokli for their help in writing the prompts in non-English languages.

REFERENCES

[1] S. Hofstätter, A. Lipani, M. Zlabinger, and A. Hanbury, "Learning to re-rank with contextualized stopwords," in *29th ACM International Conference on Information & Knowledge Management*, 2020.

[2] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Information Systems*, vol. 121, 2024.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, 2019.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, 2020.

[5] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *EACL*, 2021.

[6] F. M. Plaza-del Arco, D. Nozza, and D. Hovy, "Leveraging label variation in large language models for zero-shot text classification," *arXiv preprint*, 2023.

[7] S. Wang, S. Zhuang, and G. Zuccon, "Large language models based stemming for information retrieval: Promises, pitfalls and failures," in *SIGIR*, 2024.

[8] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé *et al.*, "Gemma 2: Improving open language models at a practical size," *arXiv preprint arXiv:2408.00118*, 2024.

[9] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin *et al.*, "Gemma 3 technical report," *arXiv preprint arXiv:2503.19786*, 2025.

[10] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[11] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison *et al.*, "Phi-4 technical report," *arXiv preprint arXiv:2412.08905*, 2024.

[12] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, and Z. Fan, "Qwen2 technical report," *CoRR*, 2024.

[13] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Comput. Surv.*, 2023.

[14] M. Braga, P. Kasela, A. Raganato, and G. Pasi, "Synthetic data generation with large language models for personalized community question answering," in *IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2024.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020.

[16] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, "Large language models are few-shot clinical information extractors," in *Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022.

[17] S. Wang, H. Scells, S. Zhuang, M. Potthast, B. Koopman, and G. Zuccon, "Zero-shot generative large language models for systematic review screening automation," in *ECIR*. Springer, 2024.

[18] M. Braga, P. Kasela, A. Raganato, and G. Pasi, "Investigating task arithmetic for zero-shot information retrieval," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 2738–2743. [Online]. Available: https://doi.org/10.1145/3726302.3730216

[19] M. Braga, A. Raganato, and G. Pasi, "AdaKron: An adapter-based parameter efficient model tuning with kronecker product," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds. Torino, Italia: ELRA and ICCL, May 2024, pp. 350–357. [Online]. Available: https://aclanthology.org/2024.lrec-main.32/

[20] M. Braga, "Personalized large language models through parameter efficient fine-tuning techniques," in *Proc. of the 47th SIGIR*, 2024.

[21] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[22] L. Dolamic and J. Savoy, "When stopword lists make the difference," *Journal of the American Society for Information Science and Technology*, 2010.

[23] A. Zaman, P. Matsakis, and C. Brown, "Evaluation of stop word lists in text retrieval using latent semantic indexing," in *2011 Sixth International Conference on Digital Information Management*. IEEE, 2011.

[24] O. Toporkov and R. Agerri, "On the role of morphological information for contextual lemmatization," *Computational Linguistics*, 2024.

[25] D. Kondratyuk, T. Gavenčiak, M. Straka, and J. Hajič, "LemmaTag: Jointly tagging and lemmatizing for morphologically rich languages with BRNNs," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 4921–4928. [Online]. Available: https://aclanthology.org/D18-1532/

[26] T. Bergmanis and S. Goldwater, "Context sensitive neural lemmatization with Lematus," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, M. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1391–1400. [Online]. Available: https://aclanthology.org/N18-1126/

[27] M. F. Porter, "An algorithm for suffix stripping," *Program*, 1980.

[28] C. D. Paice, "Another stemmer," *SIGIR Forum*, 1990.

[29] M. F. Porter, "Snowball: A language for stemming algorithms," 2001.

[30] T. Muller, R. Cotterell, A. Fraser, and H. Schütze, "Joint lemmatization and morphological tagging with lemming," *arXiv preprint arXiv:2405.18308*, 2024.

[31] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, 2003.

[32] B. De Ville, "Decision trees," *Wiley Interdisciplinary Reviews: Computational Statistics*, 2013.

[33] T. G. Nick and K. M. Campbell, "Logistic regression," *Topics in biostatistics*, 2007.

[34] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve bayes." *Encyclopedia of machine learning*, 2010.

[35] F. Barbieri, J. Camacho-Collados, F. Ronzano, L. E. Anke, M. Ballesteros, V. Basile, V. Patti, and H. Saggion, "Semeval 2018 task 2: Multilingual emoji prediction," in *The 12th international workshop on semantic evaluation*, 2018.

[36] C. Van Hee, E. Lefever, and V. Hoste, "Semeval-2018 task 3: Irony detection in english tweets," in *The 12th international workshop on semantic evaluation*, 2018.

[37] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, "Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter," in *13th international workshop on semantic evaluation*, 2019.

[38] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," in *13th International Workshop on Semantic Evaluation*, 2019.

[39] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter," in *The Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

[40] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of machine learning research*, 2004.

[41] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015.

[42] P. Kasela, M. Braga, G. Pasi, and R. Perego, "Se-pqa: Personalized community question answering," in *Proc. of the ACM Web Conference 2024*, 2024.

[43] F. Barbieri, L. Espinosa Anke, and J. Camacho-Collados, "XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond," in *LREC 2022*.

[44] G. Silvello, R. Bucco, G. Busato, G. Fornari, A. Langeli, A. Purpura, G. Rocco, A. Tezza, and M. Agosti, "Statistical stemmers: A reproducibility study," in *40th ECIR, Grenoble, France*. Springer, 2018.