Investigating Large Language Models' Linguistic Abilities for Text Preprocessing

Anonymous ACL submission

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, such as the Porter stemmer, usually ignore contextual information. Moreover, low-resource languages frequently lack the comprehensive linguistic resources needed for defining traditional preprocessing techniques. In this paper, we investigate the idea of using Large Language Models (LLMs) to perform various preprocessing tasks, due to their ability to understand context without requiring extensive language-specific annotated resources. Through a comprehensive evaluation, we compare LLM-based preprocessing – specifically stopword removal, lemmatization and stemming – to traditional algorithms across multiple text classification tasks in five European languages. Our analysis shows that LLMs can correctly replicate traditional lemmatization and stemming methods with up to 83% accuracy. Additionally, we show that ML algorithms trained on texts preprocessed by LLMs achieve an improvement of up to 8.6% with respect to the F_1 measure compared to traditional techniques. Our code and results are publicly available at https://anonymous.4open.science/r/llm_pipeline-7B0D/.

1 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 (Hofstätter et al., 2020).

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 different word distribution. Additionally, the context of a text is crucial in determining whether a word should be treated as a stopword (Hofstätter et al., 2020). 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. Furthermore, these techniques depend on extensive linguistic resources, such as annotated datasets, which makes preprocessing challenging for low-resource languages.

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 understand linguistic context (Radford et al., 2019; Brown et al., 2020b; Schick and Schütze, 2021; Plaza-del Arco et al., 2023) without requiring extensive language-specific annotated resources, we hypothesize that LLMs can dynamically detect stopwords, lemmas

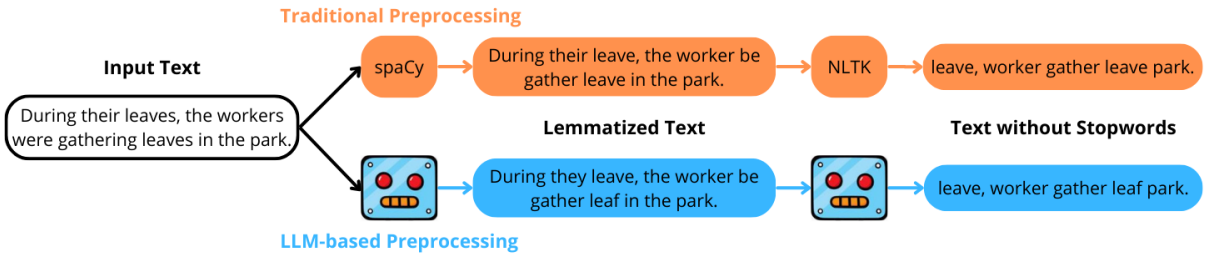

Figure 1: Pipeline of the proposed approach: we compare the output of text preprocessing using traditional techniques and LLMs. In this example, the LLM correctly disambiguates the word "leaves", distinguishing between employee absences and foliage in its two occurrences, and applies lemmatization accordingly.

and stems based on the input document, context and task. Although prior work by Wang et al. (2024b) 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 EuroLLM (Martins et al., 2024), Gemma-2 (Team et al., 2024), LLama-3 (Dubey et al., 2024), and Qwen-2.5 (Yang et al., 2024), 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 our approach, we train three different ML-based classification models by using data preprocessed by the LLMs. Our analysis shows that LLMs replicate the performance of traditional preprocessing with an accuracy up to 83% in English and to 97% in French. Furthermore, we note that ML algorithms trained on texts preprocessed by LLMs achieve an improvement of up to 8.6% with respect to the F_1 measure compared to traditional techniques. The paper is organised as follows: after discussing the related works in Section 2, Section 3 introduces in detail our approach, while in Section 4 we describe the experimental setup used. Then, Section 5 discusses the results of our evaluation. Finally, Section 6 concludes the study and outlines future research directions.

2 Related Works

LLMs have achieved state-of-the-art performance across a wide range of tasks and research fields (Min et al., 2023). They are particularly effective in few-shot settings, where they can be applied to unseen tasks or domains without requiring additional supervised fine-tuning (Brown et al., 2020a; Agrawal et al., 2022; Wang et al., 2024a), which demands a large amount of labelled data that are not always available (Thakur et al., 2021).

The relation between preprocessing operations, such as lemmatization and stopword removal, and the context of the input texts has been studied for a long time (Dolamic and Savoy, 2010; Zaman et al., 2011; Hofstätter et al., 2020; Toporkov and Agerri, 2024). For instance, Hofstätter et al. (2020) show 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. Additionally, in the context of the SIGMORPHON 2019 (McCarthy et al., 2019) shared task, Toporkov and Agerri (2024) propose a BERT (Devlin, 2018) model for context-specific lemmatization. Recently, LLMs have been applied in a few-shots scenario to stemming queries and documents in an information retrieval pipeline (Wang et al., 2024b). The authors found that, although LLM-based stemming alone does not improve retrieval performance, using LLMs to identify named entities that should not be stemmed – while applying the Porter algorithm (Porter, 1980) to the remaining words – significantly enhances the retrieval performance.

No prior work has conducted a comprehensive analysis of LLMs for text preprocessing – including

Language	Task	# Voc.	Avg. Words	# Labels	# Train	# Test
English	Emoji prediction	17405	15	20	3000	3000
	Hate detection	18403	26	2	2999	2970
	Irony detection	11824	18	2	2862	784
	Offensive lang. detection	13977	29	2	3000	860
	Sentiment analysis	19631	21	3	2997	2997
French	Sentiment analysis	8943	18	3	1839	870
German	Sentiment analysis	10902	15	3	1839	870
Italian	Sentiment analysis	9934	18	3	1839	870
Spanish	Sentiment analysis	9184	18	3	1839	870

Table 1: Statistics of the adopted datasets: size of the vocabulary, average number of words in each text, number of labels, and number of examples in the train and test splits.

stopword removal, lemmatization, and stemming – by comparing their outputs to those produced by traditional methods, and by assessing their impact on text classification.

3 Methodology

The proposed approach involves prompting LLMs for text preprocessing by defining prompts that guide them 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 approach is few-shot, as we provide the LLMs with a few examples of stopwords, lemmas and stems. 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 task, 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, 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, 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 (Porter, 1980), Lancaster (Paice, 1990), and Snowball (Porter, 2001), and utilizing off-the-shelf implementations of rule-based or edit tree lemmatizers (Muller et al., 2024). 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 (Aizawa, 2003). Then, we train three well-known ML algorithms, i.e. Decision Tree (De Ville, 2013), Logistic Regression (Nick and Campbell, 2007), and Naive Bayes (Webb et al., 2010). To assess the overall impact of text preprocessing across the previously mentioned ML algorithms, we average the single models' performances.

4 Experimental Setup

In this section, we describe the datasets, the evaluation metrics and the models used to assess the effectiveness of the proposed approach.

Datasets We select a suite of publicly available datasets encompassing binary and multiclass classification tasks across multiple languages, including English, French, German, Italian and Spanish. Specifically, for the evaluation of texts in English, we use the Twitter datasets from SemEval-18 on emoji prediction (Barbieri et al., 2018) and irony detection (Van Hee et al., 2018), as well as from SemEval-19 on hate detection (Basile et al., 2019), offensive language identification (Zampieri et al., 2019) and sentiment analysis (Nakov et al., 2013). For non-English languages, we employ four datasets from the Tweet Sentiment Multilingual corpus (Barbieri et al., 2022). 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. Table 1 shows a few statistics of the adopted datasets. 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

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 2: Prompts used to lemmatize, stem and remove stopwords from the texts of the SemEval Sentiment dataset.

of the ML algorithms.

Models We compare four open source state-of-the-art LLMs, encompassing different sizes and architectures: EuroLLM-9B (Martins et al., 2024), Gemma-2-9B (Team et al., 2024), LLama-3.1-8B (Dubey et al., 2024), and Qwen-2.5-7B (Yang et al., 2024) in their instruction-tuned version. While Gemma has been primarily trained on English data, Llama, Qwen and EuroLLM are natively multilingual, supporting Italian, Spanish, French and German. 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`). The experiments are conducted on an NVIDIA Ampere A100 GPU.

Prompts Table 2 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 straightforward adaptations or, in the case of multilingual datasets, translations of the ones shown here. All prompts used in this study are publicly available in our repository[1].

Baselines: traditional preprocessing We employ the stopword lists and stemmers provided by NLTK[2], and the rule-based (for English, French and Spanish) and edit tree (for German and Italian) lemmatizers provided by spaCy[3]. The word "not" and language-specific negation lexicon are removed from the NLTK's stopwords lists.

Machine Learning algorithms We use the scikit-learn[4] implementations of the Multinomial Naive Bayes (Webb et al., 2010), Decision Tree (De Ville, 2013), and Logistic Regression (Nick and Campbell, 2007) algorithms. Further details on specific ML hyperparameters can be found in Appendix A.

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.

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. The detailed hyperparameters

[1] https://anonymous.4open.science/r/llm_pipeline-7B0D/

[2] https://www.nltk.org/

[3] https://spacy.io/

[4] https://scikit-learn.org/

Model	SW	NSW	L	S			
				Porter	Lanc.	Snow.	Any
EuroLLM	39.46	37.52	54.41	40.52	33.92	41.33	43.58
Gemma	**83.60**	**16.36**	81.96	**73.51**	**61.60**	**74.24**	**79.98**
Llama	82.69	32.45	79.32	66.11	57.07	67.40	73.09
Qwen	77.94	22.54	**83.43**	60.93	52.50	61.96	67.41

Table 3: Accuracy of different LLMs in performing text preprocessing in English.

Language	Model	SW		NSW		L		S (Snowball)	
French	EuroLLM	35.62	43.78	34.98	37.32	49.22	50.39	37.30	28.34
	Gemma	96.83	**97.94**	28.02	33.12	61.06	53.28	**51.51**	34.47
	Llama	65.02	32.01	37.20	**23.11**	54.86	55.47	45.80	34.15
	Qwen	82.98	86.33	31.72	31.32	**65.70**	61.00	46.86	43.05
German	EuroLLM	39.32	26.13	39.80	25.92	46.03	51.15	32.04	40.40
	Gemma	74.52	**79.73**	24.35	32.99	59.80	64.45	**68.46**	61.06
	Llama	58.08	25.65	35.44	**16.54**	58.38	60.07	58.54	57.32
	Qwen	57.23	47.04	31.29	22.88	**64.84**	64.29	53.17	48.86
Italian	EuroLLM	36.32	25.26	35.66	25.24	46.26	52.87	29.04	34.86
	Gemma	86.48	**89.20**	**20.82**	22.36	59.30	58.31	**56.65**	50.86
	Llama	67.27	62.73	31.51	25.78	53.22	51.31	39.83	40.72
	Qwen	69.67	84.88	28.18	33.06	**63.92**	62.27	47.30	46.66
Spanish	EuroLLM	39.83	43.94	38.22	35.12	42.70	48.20	29.91	43.94
	Gemma	85.91	**87.90**	**20.46**	26.68	57.86	57.68	**63.76**	62.14
	Llama	69.99	33.53	29.02	21.44	51.02	52.11	48.69	51.90
	Qwen	67.46	75.55	26.84	25.81	**62.30**	61.11	54.78	52.05

Table 4: Accuracy of different LLMs in performing text preprocessing in four 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.

settings for each model and preprocessing technique can be found in Appendix A.

5 Results

In Section 5.1, we examine how closely LLMs are able to reproduce traditional preprocessing techniques (RQ1). As mentioned in the introduction, LLMs may however 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 5.2.

5.1 LLMs' preprocessing abilities

Tables 3 and 4 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 3), 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 3 and 4.

Since Gemma 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 outperforms all the other models in stopword removal and stemming, although Qwen achieves the highest accuracy in lemmatization. Notably, this pattern is consistent across all the analyzed languages. Interestingly, despite being a natively multilingual model with parameter size comparable to Gemma, EuroLLM does not perform as well as the other models with respect to any preprocessing operation. Additionally, we note that language-specific

Dataset	Model	SW	SW + L	L	SW + S Porter | Lanc. | Snow.	S Porter | Lanc. | Snow.
Emoji	Classic	21.15	21.41	21.42	21.06 | 21.11 | 21.03	**21.01** | 20.96 | 20.88
	EuroLLM	20.82	20.44	20.79	19.82	20.12
	Gemma	21.52	**22.61**†	**21.66**	**21.19**	21.00
	Llama	**21.71**	21.99	21.22	20.51	20.99
	Qwen	21.63	22.53	21.47	20.97	20.60
Hate	Classic	48.73	48.93	49.67	49.47 | 49.40 | 47.87	46.99 | 47.74 | 47.82
	EuroLLM	**51.19**	**53.14**†	**51.54**	**51.14**	**50.40**
	Gemma	49.61	47.47	49.52	49.77	49.24
	Llama	49.50	49.75	51.31	50.45	50.15
	Qwen	50.80	49.09	48.93	49.61	50.34
Irony	Classic	61.05	60.11	59.73	61.96 | **63.01** | 61.39	60.96 | **62.15** | 59.73
	EuroLLM	53.70	54.63	60.24	50.72	59.01
	Gemma	61.64	**62.63**	61.14	59.40	60.63
	Llama	61.44	62.29	**63.35**†	59.31	58.80
	Qwen	**61.99**	61.22	63.18	57.95	59.35
Offensive	Classic	75.62	74.53	73.19	75.73 | **75.93** | 74.61	74.22 | 75.46 | **75.65**
	EuroLLM	71.71	70.93	70.81	71.09	70.58
	Gemma	74.81	**74.88**	73.02	73.95	72.71
	Llama	**76.71**†	73.95	71.47	73.76	71.20
	Qwen	74.03	73.84	**74.38**	72.40	71.36
Sentiment	Classic	**48.89**†	48.05	**48.71**	47.81 | **48.54** | 48.18	47.89 | **48.62** | 48.61
	EuroLLM	45.64	41.30	43.90	45.00	45.01
	Gemma	48.13	47.77	48.35	46.59	47.39
	Llama	47.96	**48.13**	48.02	45.80	47.06
	Qwen	46.98	45.54	46.84	46.04	46.38

Table 5: Comparison of LLM-based and traditional (classic) 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.

prompts achieve the highest *SW* scores, while the specification of the same prompts in English produces the best lemmatization (*L*) and stemming (*S*) performance. More specifically, EuroLLM's performance improves with language-specific prompts in 60% of preprocessing tasks. In contrast, Gemma and Qwen perform better with an English prompt in at least 70% of cases, while Llama benefits from it in 55% of combinations.

We further observe that LLMs often eliminate words not traditionally considered stopwords (*NSW* column). Among them, Gemma aligns most closely with NLTK for English, Italian, and Spanish, while Llama performs analogously for French and German. 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 consist of social media text from Twitter (Barbieri et al., 2022). Regarding stemming, the lower overall scores compared to traditional preprocessing might be due to LLMs producing different stems of the same word appearing in different texts. Although this deviates from traditional stemming rules, this might allow for a more context-specific preprocessing, as also observed by Wang et al. (2024b).

Overall, these results show that LLMs are quite effective at identifying stopwords across multiple languages, with Gemma detecting 97% of stopwords in French texts, and at least 79% in other languages. Additionally, they show a good lemmatization capability in English, with Gemma, Llama and Qwen correctly identifying over 79% of lemmas.

5.2 Text classification

Tables 5 and 6 report the averaged performance score of the three ML models we have trained on English (Table 5) and non-English text (Table 6), 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

Dataset	Model	SW		SW + L		L		SW + S		S	
French	Classic	**52.95**		**52.49**		53.48		**53.98**†		**52.87**	
	EuroLLM	45.44	46.70	41.00	44.21	49.34	48.74	43.76	39.38	45.59	43.30
	Gemma	52.18	52.53	49.66	49.54	51.88	52.91	51.65	49.54	47.81	50.34
	Llama	50.57	52.07	47.13	48.20	50.19	51.38	46.05	47.66	49.04	48.00
	Qwen	50.50	49.89	51.23	49.80	53.45	**53.52**	50.08	47.16	47.85	49.08
German	Classic	**55.13**†		**52.80**		53.18		**53.52**		54.06	
	EuroLLM	46.70	48.92	41.65	43.41	47.28	48.55	39.58	44.41	46.70	48.08
	Gemma	48.47	49.77	50.04	46.93	**53.56**	50.26	50.27	49.31	**54.48**	52.07
	Llama	47.13	50.57	47.24	50.04	51.99	53.37	46.48	50.46	50.96	53.56
	Qwen	46.59	49.54	47.73	49.27	52.30	51.88	45.86	46.05	51.46	48.35
Italian	Classic	**51.84**		**52.07**		50.61		**51.30**		52.33	
	EuroLLM	47.13	49.16	43.14	47.13	50.92	50.38	45.48	46.70	48.08	48.89
	Gemma	48.16	48.73	45.59	47.16	51.34	50.61	45.51	44.80	52.30	**52.53**
	Llama	46.44	46.56	44.94	48.47	52.72	51.46	42.34	43.87	48.62	51.92
	Qwen	45.48	44.71	45.94	41.88	**53.37**†	52.99	43.44	44.48	48.97	50.50
Spanish	Classic	47.47		**49.43**		48.47		**49.88**		**49.61**	
	EuroLLM	43.14	44.02	39.58	43.45	43.52	47.36	40.80	42.30	41.88	45.86
	Gemma	49.85	**49.92**†	48.31	48.70	47.05	48.08	47.47	48.74	48.43	48.74
	Llama	48.40	48.62	45.67	45.10	48.47	46.48	43.80	46.28	45.90	47.55
	Qwen	49.39	47.70	49.08	45.75	48.12	**48.74**	46.67	45.33	46.86	45.86

Table 6: Comparison of LLM-based and traditional (classic) preprocessing on several sentiment analysis tasks in multiple 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.

results are highlighted in **bold**, while the second-best scores are underlined. The best result within each dataset is marked with a †.

English language We first note that LLMs outperform traditional methods in all datasets except for Irony, and more specifically in 16 out of the 25 examined combinations of datasets and preprocessing tasks. Moreover, in over 65% of these cases, the second-best result is also achieved by an LLM. Traditional preprocessing outperforms LLM-based preprocessing in 9 out of 25 combinations, with a margin greater than 1 point in F_1 in 6 of them (Irony SW+S and S, Offensive SW+S and S, and Sentiment SW+S + S). Notably, LLMs achieve the highest performance in stopword removal and lemmatization across 4 out of 5 datasets, indicating their ability to dynamically identify task-relevant stopwords and lemmas in a more context-sensitive manner than traditional techniques. The only exception is the Sentiment dataset, where Gemma underperforms by less than 1 point compared to traditional preprocessing. Additionally, our approach outperforms traditional preprocessing across all datasets when applying lemmatization combined with stopword removal: in particular, in the Hate dataset EuroLLM achieves an 8.6% improvement over traditional techniques.

Our results indicate however that stemming with LLMs is not as effective as other preprocessing operations. 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 Wang et al. (2024b), 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.

Non-English languages Table 6 presents the performance of text classification across French, German, Italian, and Spanish. Overall, LLMs achieve performance on par with or even better than traditional techniques in half of the evaluated cases. Notably, LLMs achieve the highest performance across all datasets when lemmatization is applied, showing their ability to understand contextual in-

formation even in non-English languages. Moreover, LLMs achieve the highest performance in the Italian and Spanish datasets (marked with †) and perform marginally lower than the best score in French and German.

Interestingly, for Italian and German, Gemma outperforms traditional methods even in stemming, in contrast to the findings in the English setting. For Spanish, Gemma also shows a significant improvement in stopword removal, underscoring its ability to identify stopwords based on context. Moreover, in Spanish, traditional preprocessing outperforms LLMs by only 1 point in stemming, and similarly in stopword removal when combined with lemmatization or stemming.

The performance differences among languages may stem from their varying morphological complexity. For instance, Markus Sadeniemi and Honkela (2008) found that English, French, Italian, and Spanish exhibit relatively low morphological complexity, whereas German demonstrates significantly higher complexity, possibly due to its use of compound words.

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

6 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 all evaluated languages 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 (Silvello et al., 2018), 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.

7 Ethics Statement and Limitations

Ethics statement Our approach shares the same possibilities as most of previous works based on LLMs, such as misusage, containing data bias, and suffering from adversarial attacks. Since we are however using LLMs to preprocess text, we conclude that our work will not likely have a negative ethical impact.

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 metrics.

We do not perform extensive prompt engineering in this work, as we are interested in investigating the abilities and raw behaviour of Large Language Models rather than obtaining the best results. This is also due to computational constraints and costs. However, some results may differ if other prompts are considered.

Another limitation of the proposed approach is the high computational cost of using LLMs for text preprocessing, which is significantly greater than that of traditional methods. Therefore, LLM-based preprocessing is best justified for low-resource languages. Our results, demonstrating that LLMs can consistently match or even surpass traditional preprocessing techniques across multiple languages, further support their use in such cases.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. Semeval 2018 task 2: Multilingual emoji prediction. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 24–33.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Barry De Ville. 2013. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):448–455.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ljiljana Dolamic and Jacques Savoy. 2010. When stopword lists make the difference. *Journal of the American Society for Information Science and Technology*, 61(1):200–203.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sebastian Hofstätter, Aldo Lipani, Markus Zlabinger, and Allan Hanbury. 2020. Learning to re-rank with contextualized stopwords. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2057–2060.

Eduardo Laber and Lucas Murtinho. 2019. Minimization of gini impurity: Np-completeness and approximation algorithm via connections with the k-means problem. *Electronic Notes in Theoretical Computer Science*, 346:567–576. The proceedings of Lagos 2019, the tenth Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS 2019).

Tiina Lindh-Knuutila Markus Sadeniemi, Kimmo Kettunen and Timo Honkela. 2008. Complexity of european union languages: A comparative approach. *Journal of Quantitative Linguistics*, 15(2):185–211.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, et al. 2024. Eurollm: Multilingual language models for europe. *arXiv preprint arXiv:2409.16235*.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2).

Thomas Muller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2024. Joint lemmatization and morphological tagging with lemming. *arXiv preprint arXiv:2405.18308*.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320.

Todd G Nick and Kathleen M Campbell. 2007. Logistic regression. *Topics in biostatistics*, pages 273–301.

Chris D. Paice. 1990. Another stemmer. *SIGIR Forum*, 24(3):56–61.

Flor Miriam Plaza-del Arco, Debora Nozza, and Dirk Hovy. 2023. Leveraging label variation in large language models for zero-shot text classification. *arXiv preprint arXiv:2307.12973*.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Gianmaria Silvello, Riccardo Bucco, Giulio Busato, Giacomo Fornari, Andrea Langeli, Alberto Purpura, Giacomo Rocco, Alessandro Tezza, and Maristella Agosti. 2018. Statistical stemmers: A reproducibility study. In *Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings 40*, pages 385–397. Springer.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. 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 (Round 2)*.

Olia Toporkov and Rodrigo Agerri. 2024. On the role of morphological information for contextual lemmatization. *Computational Linguistics*, 50(1):157–191.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 39–50.

Shuai Wang, Harrisen Scells, Shengyao Zhuang, Martin Potthast, Bevan Koopman, and Guido Zuccon. 2024a. Zero-shot generative large language models for systematic review screening automation. In *European Conference on Information Retrieval*, pages 403–420. Springer.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2024b. Large language models based stemming for information retrieval: Promises, pitfalls and failures. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2492–2496, New York, NY, USA. Association for Computing Machinery.

Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. 2010. Naïve bayes. *Encyclopedia of machine learning*, 15(1):713–714.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

ANK Zaman, Pascal Matsakis, and Charles Brown. 2011. Evaluation of stop word lists in text retrieval using latent semantic indexing. In *2011 Sixth International Conference on Digital Information Management*, pages 133–136. IEEE.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

A Hyperparameters Setting

Table 7 shows the two best TF-IDF configurations for each ML model and traditional preprocessing algorithm. The length of the n-grams ranges from 1 to 3, while the number of features can be 1000, 3000, 5000 or 7000. Regarding Naive Bayes, we put the smoothing parameter $alpha = 1$. Decision Tree relies on Gini impurity (Laber and Murtinho, 2019) to measure the quality of each split. Regarding Logistic Regression, we add a L_2 penalty term. For both Decision Tree and Logistic Regression, we set the random seed to 42.

ML algorithm	Preproc.	N-grams	Features dim.	F_1
Decision Tree	SW	1	3000	**57.55**
		2	3000	56.70
	SW + L	1	3000	**57.00**
		2	5000	56.40
	L	2	5000	**57.55**
		1	3000	56.65
	S	2	5000	**56.75**
		3	7000	56.65
	SW + S	1	3000	**58.75**
		1	7000	57.05
Multinomial Naive Bayes	SW	2	5000	**71.00**
		3	5000	70.95
	SW + L	2	7000	**70.60**
		2	5000	70.45
	L	2	7000	**71.15**
		3	7000	70.75
	S	3	7000	**71.25**
		2	7000	71.15
	SW + S	3	7000	**70.40**
		2	5000	70.30
Logistic Regression	SW	2	7000	**68.75**
		1	5000	68.70
	SW + L	3	5000	**68.85**
		2	7000	68.80
	L	2	5000	**69.20**
		2	7000	68.80
	S	3	7000	**70.15**
		3	5000	69.95
	SW + S	2	5000	**68.75**
		1	7000	68.70

Table 7: Study on the impact of hyperparameters using classical preprocessing techniques. The applied stemming algorithm is Porter.