Restoring Accent and Ligature on French

2022313356 Kim Ji Heon
English / Applied Artificial Intelligence

2022312957 Cho Jun Beom
Applied Artificial Intelligence / French

1. Introduction

1.1. Topic and Problem

French is often called “the language of love” for its graceful and delicate pronunciation as well as its rich literary heritage. This charm inspires many people to learn French. However, with some documents named “Why is French hard to learn?” reflect how complex and nuanced the language can be.

In particular, grammatical gender (masculine, feminine) assigned to nouns, as well as for adjectives and most types of words, and the frequent use of accents (à, é, ç etc.) and special characters (œ, æ etc.) at the character level form as a significant barrier for beginners. These are not mere visual embellishments, but elements that directly affect pronunciation, meaning, and grammatical function, making their correct use crucial.

Grammar, vocabulary, spelling, and prosody are not independent elements but they are the parts of a tightly connected linguistic system. Thus, the beauty of the French language is inseparably tied to its complexity. In today’s digital world, however, keyboard limitations or autocomplete systems often omit accents and special characters, resulting in the loss of the language’s authentic character. Moreover, it interrupts the beginners to learn on a precise way, unfortunately leading to misunderstandings and the reinforcement of incorrect usage patterns.

In this context, the goal of our project is to restore the linguistic integrity and accuracy of French texts by automatically recovering accents and combined characters using a deep learning model. This initiative goes beyond the functionality of a simple editing tool and aims to preserve the aesthetic and cultural value of the French language even in digital environments.

Furthermore, from the perspective of French learners, especially beginners, this system can serve as a valuable educational aid. Accent related errors are common yet critical, as they affect both pronunciation and grammar. A system that automatically corrects those errors can function not only as a feedback tool but also as a learning companion. Therefore, this project can support iterative practice for learners and contribute to improving the overall quality of language education.

1.2. Features of French Letters and Grammar

French includes a variety of accent marks as part of its spelling system. Representative examples include é(accent aigu), è(accent grave), ê(accent circonflexe), ç(cédille), ë(tréma). These are not merely phonetic markers, but essential components that influence a word’s meaning and grammatical role. For instance, ‘a’ and ‘à’ serve completely different purposes as a verb and a preposition, and ‘sur’ and ‘sûr’ each plays as a different meaning as ‘on’ and ‘certain’.

In addition, French uses ligatures, which are special combined characters such as œ and æ. These are not arbitrary letter combinations, but part of the standard French orthography. When omitted or incorrectly replaced, they can lower

the quality of a sentence or even distort its meaning.

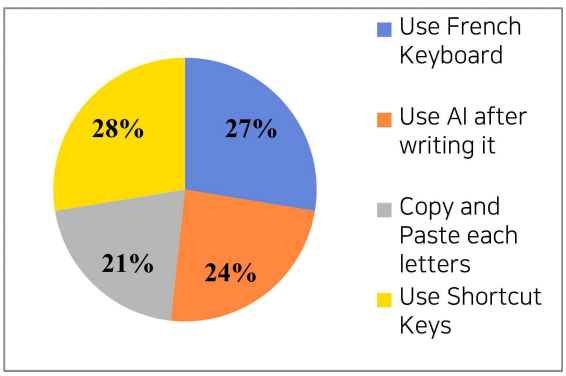
French also categorizes every noun by gender, assigning each to either masculine or feminine. This affects the form of associated words like articles and adjectives. When accents or special characters are missing, the correct form of a word may become ambiguous, leading to grammatical confusion.

Despite these complex features, modern digital environments often ignore or omit accents and special characters, which can damage the integrity of the language and confuse readers. These issues are particularly challenging for learners of French. In the early stages of learning, students often lack the awareness and knowledge to use accents and special characters correctly, making it difficult to input or revise text properly. Frequent mistakes can lead to frustration and reduce motivation in the learning process.

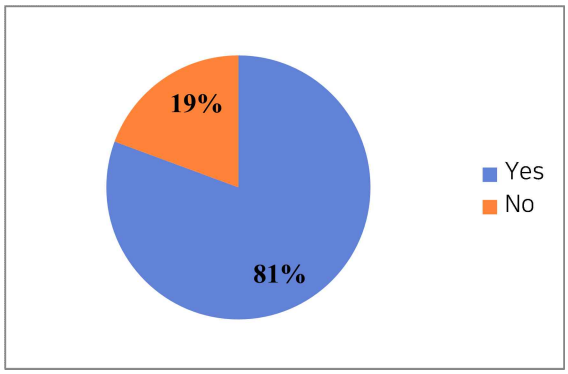
1.3. Survey Results

To investigate whether there are real inconveniences in typing accented and special characters, a survey was conducted among 30 students, who have recently started learning French, from the major of French at Sungkyunkwan University. The questions and their responses are as follows.

1) Ways to write French sentences



2) Experiences of feeling Uncomfortable



The survey results revealed that students use a variety of methods to input French characters, with many relying on shortcuts, dictionaries, or A.I. tools. Except for those who have learned the French keyboard layout, all participants acknowledged discomfort with current input methods. The main reasons cited were the complexity of using shortcuts and the inefficiency of copying and pasting words and characters. For beginners learning French, this can become a psychological barrier to developing their language skills and, at times, lead to the acquisition of incorrect knowledge due to the use of improper accent marks.

1.4. Prior Studies

Accents and special characters are not unique to French: they are prominent features in many European languages, such as Czech and German. Since this is a shared issue across various languages, the development of an automatic French accent restoration model has a long standing research background.

Pioneering work in this field began in the early 1990s with David Yarowsky. He approached the problem of restoring missing accents in Spanish and French texts from the perspective of resolving lexical ambiguity. Yarowsky evaluated and compared the performance of three methods: an N-gram POS

tagger, a Bayesian classifier, and a Decision List Algorithm. The results showed that the Decision List Algorithm, which applies predefined linguistic rules one by one to an input sentence and adopts the first rule that matches, making use of both local syntactic patterns and more distant contextual cues, achieved the highest accuracy at 91.5%.

However, the model based solely on a single highest priority rule lack the flexibility to handle exceptional cases, such as literary expressions that fall outside predefined patterns. Even when multiple contextual clues are present within a sentence, it applies only one rule, making it unsuitable for cases like 'que' clauses that require either a past participle or the subjunctive mood based on some word before. Moreover, it fails to recognize newly coined or borrowed words, and it cannot interpret contracted forms such as "t'es" (short word for "tu es").

Several statistical methods were explored in the early stages, but in 2018, research based on neural networks began in earnest. A notable approach utilized bidirectional RNNs, which were applied to languages such as Czech and Vietnamese. Specifically, two BiLSTM models, LSTM that operates in both forward and backward directions, were used to capture both past and present contextual information. This model also incorporated residual connections, distinguishing it from previous approaches. Compared to traditional statistical methods, this approach achieved performance improvements up to 83%, and attained an accuracy of 99.06% on Czech data.

In 2022, Google's ByT5 transformer-based model was introduced, marking the beginning of byte-level natural language processing research. By using only 256 byte values, the model eliminates the need for a fixed vocabulary and is capable of handling all Unicode characters. This makes it highly flexible for multilingual tasks and unseen data. Additionally, ByT5 is trained to perform both restoration and correction tasks simultaneously, allowing it to effectively account for contextual variations in input.

The ByT5 model achieved an average accuracy of 98% across 13 languages, demonstrating strong performance in languages with accents and special characters like French and German. However, its accuracy drops to around 76% for out-of-vocabulary words, which is notably lower than for known words. Furthermore, due to its byte-level architecture, the model may struggle with maintaining semantic coherence in long or complex sentences. Also it deals as a large transformer model, it also presents challenges in terms of efficiency and deployment.

1.5. Overview of the Model & Service

This study focuses on the task of accent and special character restoration, aiming to correct characters at the letter level by considering the context, grammar, and intrinsic properties of each word in a given sentence. However, to preserve its educational purpose, the system is deliberately limited to function as a supportive tool rather than a fully automated solution. For instance, while French grammar involves adding e for feminine forms or s for plural forms, this project does not account for such gender or number based morphological changes. Automating these grammatical rules could interfere with the learner's ability to internalize core language principles. Therefore, the scope is intentionally restricted to correcting the most prominent difficulties in typing, missing or incorrect accents and special characters.

The model is based on Google's T5-small architecture, a lightweight variant of the

Text-to-Text Transformer model. This T5-small model provides a suitable trade-off between performance and computational efficiency, making an ideal foundation for further training and model compression strategies aimed at real world deployment.

The training data was constructed by pairing input sentences, where accents and combined characters had been transformed into standard alphabetic forms, with their corresponding original output sentences. To ensure broad linguistic coverage, the data was collected from diverse sources, spanning casual spoken style sentences to formal, technical writing. This input-ouput parring was deliberately designed to guide the T5 model, which learns at the sentence level, to not only consider surrounding context and grammatical cues but also focus specifically on restoring accents and combined characters accurately.

After various experiments involving parameter tuning and model optimization, the most stable model was selected and converted into ONNX format for deployment in the service environment. ONNX serves as a standardized format designed to support trained models across diverse platforms such as web and mobile. Due to the nature of the T5 model, encoding embedded input characters and returning newly generated vectors, the model is exported as separate encoder and decoder components. These components, along with the tokenizer, were than integrated into a web environment, allowing users to access and use the service directly through a webpage without the need for additional installation or complex setup.

2. Data

2.1. Data Collection and Description

To train and evaluate the French accent restoration model, text-based data from three major sources were collected. Each source presents unique linguistic characteristics and thematic diversity, which were deliberately chosen to enhance the model's ability to generalize across different domains.

The first, general French corpus, which consists a large scale collection of texts from various genres such as news articles, Wikipedia entries, online forum posts, and official documents. This corpus broadly captures real-life use cases of modern French. In particular, newspaper articles and official reports typically require precise accent usage, helping the model learn standard orthographic conventions. On the other hand forum posts and blog entries often include informal language, abbreviations, and non-standard spelling, which can introduce noise into the data and present a challenge for the model.

The second source is literary novel data, collected from public domain French literature available on digital libraries such as Project Gutenberg. This dataset includes 48 novels, ranging from classic masterpieces to short stories by contemporary authors, reflecting a wide variety of time periods and writing styles. The text in novels is rich in narrative and literary expression, frequently containing complex sentence structures and dialogue. These characteristics encourage the model to develop a deeper understanding of context to accurately restore accents. In particular, areas with character dialogue and frequent tense shifts rely heavily on accents to distinguish meaning, allowing the model to go beyond the word level and learn patterns at the sentence and discourse level.

Lastly, academic papers written in French were extracted from the HAL(Hyper Articles en Ligne) portal. This dataset is distinctly different from

the other two sources due to its frequent use of specialized terminology and proper nouns, as well as its formal and consistent writing style. To cover a wide range of disciplines, material was collected from twelve fields including chemistry, economics, agriculture, life science, law, art, computer science, literature, mathematics, music, and applied science. In this type of data, accents play a crucial role in distinguishing the meaning of word stems and endings. Moreover, the precise spelling of technical terms is essential, enabling the model to capture detailed information at the lexical level.

2.2. Data Preprocessing

To construct input-output pairs suitable for training, three types of collected data were preprocessed according to their specific characteristics.

For the Oscar dataset, which is provided in json1 format, only the relevant parts of each entry were extracted and converted into a format that is easier for a program to interpret. This preprocessing involved filtering out duplicated entries, removing emojis and unnecessary special characters, and applying a minimum length requirement to retain only meaningful sentences. These filtered sentences serve as the original reference texts for the model's output during training.

In the literary text preprocessing step, txt files that had already undergone some degree of manual refinement. Unlike the Oscar data, where a simple and fast approach were used such as splitting lines or using periods to separate sentences, a French-specific sentence tokenizer provided by NLTK was employed for more accurate segmentation. Given the cleaner nature of literary texts, lightweight filtering was also applied to remove structural elements like section dividers or end markers commonly found in print-based content.

For the academic paper dataset, which spans 12 different fields, roughly 20KB of text per document was extracted and stored. This selective sampling was intended to balance time efficiency and content diversity, ensuring that the model encounters a wide range of technical vocabulary without needing to process the full corpus.

For all three types of data, both original (target) and transformed(input) versions were constructed. Within each sentence, accented or combined characters, such as é or œ, were replaced with their ASCII equivalents, like e or oe. To avoid indexing issues during substitution such as shifting caused by insertions or deletions, this process was performed in reverse, from the end of each sentence. The final converted sentences were paired with their originals and stored as input data for training.

3. Method

3.1. Computational Resources

The training process was conducted using a high-performance computing environment. The hardware specifications include an NVIDIA A100 GPU with 40 GiB of VRAM, 90 GiB of system memory, and a 9-core processor.

3.2. Training Configuration

The base model was fine-tuned using a maximum input sequence length of 128 tokens. The training procedure was configured with the Hugging Face Transformers library as below.

To enhance multilingual robustness, additional special tokens that were previously unrecognized by the tokenizer were incorporated: ë, ï, À, È, Ç, œ, æ, Œ, Æ. This augmentation addresses

tokenization challenges in languages that include accented characters and ligatures.

Parameters	
evaluation_strategy	steps
eval_steps	5000
save_strategy	steps
save_steps	5000
learning_rate	1e-4
train_batch_size	8
gradient_steps	8
num_train_epochs	2
weight_decay	0.01
save_total_limit	2
load_best_model_at_end	True
logging_dir	./logs
logging_strategy	steps
bf16	True
report_to	[]

3.3. Model Compression

Given that the primary objective of this study is to enable practical deployment of language models, model compression plays a pivotal role. Large transformer models like T5-small, despite their efficiency relative to larger variants, still pose challenges for deployment on resource-constrained devices. Thus, various compression techniques are explored to reduce the model's size and inference latency while retaining performance.

3.3.1. Knowledge Distillation

Knowledge distillation (KD) is a model compression technique based on a teacher–student paradigm. A large, high-capacity "teacher" model generates soft target outputs—typically probabilistic distributions over classes—which are then used to guide the training of a smaller "student" model.

This process enables the student to capture the generalization behavior of the teacher model beyond what can be learned from the ground-truth labels alone. KD often improves performance compared to training the student from scratch, particularly in low-resource or low-capacity settings.

In this study, the fine-tuned T5-small model(77M Params) served as the teacher, while a smaller 'T5-mini' variant (31M Params) was trained to mimic its outputs. The goal was to retain task-specific performance while minimizing model size and runtime requirements, enabling efficient deployment on resource-constrained hardware.

3.3.2. Pruning

Pruning refers to the systematic removal of non-critical parameters—such as weights, neurons, or attention heads—from a neural network. By identifying and eliminating such redundancies, pruning reduces the number of active computations during inference, thereby decreasing latency and energy consumption.

There are several strategies for pruning, including magnitude-based pruning, structured pruning, and dynamic pruning. In this study, unstructured weight pruning was applied to the fine-tuned T5-small model to minimize performance degradation while reducing model complexity.

This compression technique is particularly beneficial in scenarios where inference speed or deployment memory footprint is constrained, such as on edge devices or embedded systems.

3.3.3. QLoRA (Quantization + LoRA)

Quantization involves reducing the precision of model parameters, typically from 32-bit floating point to 16-bit floating point or 8-bit integers, and even smaller. This process significantly reduces memory usage and accelerates inference,

particularly on hardware optimized for low-precision arithmetic.

LoRA (Low-Rank Adaptation) is a parameter ⁻efficient fine tuning approach that adapts large pre-trained models (such as transformers) to specific downstream tasks by injecting low-rank trainable matrices into selected layers while keeping the original model weights frozen.

Both techniques contribute to more efficient training and inference by reducing computational and memory overhead, thereby facilitating lightweight deployment of large-scale models. Since the T5-small model supports 8-bit quantization via the load_in_8bit option, LoRA layers were integrated and soft fine-tuning was performed on the int8-quantized T5-small model. The subsequent application of QLoRA was expected to further improve training efficiency and compression. However, when trained on an NVIDIA A100 GPU, a increase in training time was observed. This overhead may be attributed to the runtime cost of dynamic quantization and dequantization operations, which are not fully optimized for the A100's tensor core architecture that favors dense matrix operations over irregular low-precision dataflows.

3.4. Deployment of ONXX

The Open Neural Network Exchange (ONNX) is an open standard format designed to facilitate the interoperability of machine learning models across different frameworks and hardware platforms. Originally developed by Microsoft and Facebook, ONNX enables models trained in frameworks such as PyTorch or TensorFlow to be exported and run in a variety of optimized inference runtimes.

The key advantage of ONNX lies in its support for cross-platform deployment and hardware acceleration. ONNX Runtime, for instance, offers highly optimized inference backends for CPUs, GPUs, and even specialized accelerators like FPGAs and NPUs.

In the context of this study, converting the quantized and compressed T5-small model to the ONNX format enabled inference acceleration and deployment in environments where Python-based runtime dependencies are undesirable. This conversion facilitated seamless integration into production pipelines and supported efficient execution in latency-sensitive applications.

4. Experiments Results

As mentioned before, over 670,000 lines of the OPUS dataset were used to evaluate if the model performs well. For scoring, CHRF (character n-gram F-score) metric (Popović, 2015) was integrated. French sentences with removed accents into the model were inputed and evaluated by its generated output against the correct, accent-restored target sentences using the CHRF metric.

4.1. Basic Model

Since Nvidia's A100 GPU fully supports the BF16 (Brain Floating Point 16) data type, the basic model was trained with BF16 for accelerating workloads.

The average score of the basic model was at 98.4357, and the percentage of 100% reproduction rate (meaning how much the generated output is perfectly the same as the target) is 91.16%. Total inference time took 15321.39 seconds. These results demonstrate a high level of generation accuracy and fidelity, suggesting the model's strong capacity for precise reproduction under the given task conditions.

4.2. Knowledge Distilled Model

Knowledge Distillation was performed using

the basic model as the teacher and T5-mini (https://huggingface.co/google/t5-efficient-mini) as the student model. The distilled model achieved an average score of 98.6555, slightly surpassing that of the basic model. The proportion of samples with a 100% reproduction rate was 89.46%. The total inference time required was 14,339.21 seconds. These results indicate that the knowledge distillation process successfully transferred performance to a more compact model while maintaining competitive accuracy and efficiency.

4.3. Pruned Model

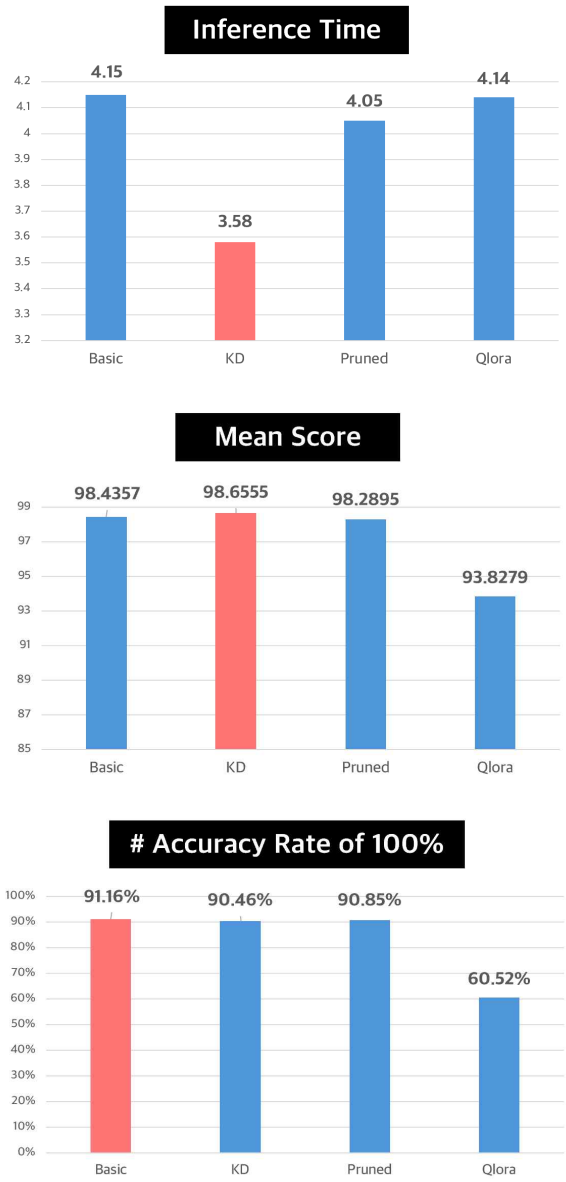
Unstructured L1-norm pruning was applied to all linear layers within the model. Specifically, for each linear layer, 30% of the weights with the smallest absolute values, as determined by the L1-norm criterion, were pruned by setting them to zero. Following pruning, the reparameterizations introduced during pruning were permanently removed using prune.remove, yielding a sparsified model that retains the original parameter dimensions but contains a substantial proportion of zero-valued weights.

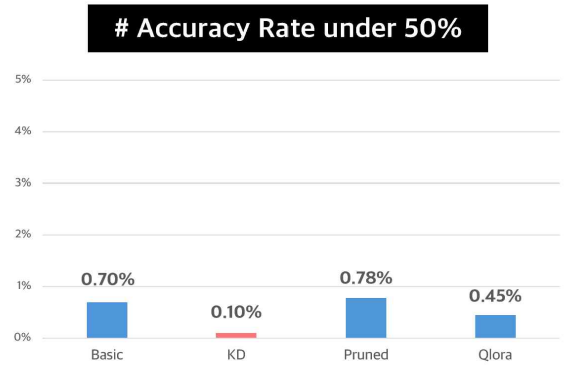
As a result, the model achieved an average score of 98.2895, which is marginally lower than that of the baseline model. The proportion of samples exhibiting a 100% reproduction rate was 90.85%. The total inference time recorded was 14,706.61 seconds.

4.4. QLoRA Model

Finally, the QLoRA model, which incorporates both quantization and Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning, has been evalutaed. The model achieved an average score of 93.8279, which is approximately 5 percentage points lower than that of the baseline model. The proportion of samples achieving a 100% reproduction rate was 60.52%, indicating a notable degradation in generation fidelity. This performance drop is presumed to stem from the compound effect of dual compression techniques, quantization and LoRA, which may have constrained the model's representational capacity while significantly reducing memory and computational requirements. Despite these limitations, QLoRA offers a favorable trade-off in scenarios where resource efficiency is prioritized over exact output replication.

4.5. Comparison Figures





5. Model Serving

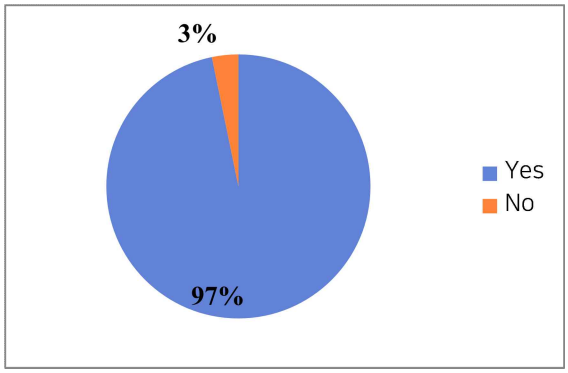
5.1. Web Integration of the Model

Based on the previous experiments, the Knowledge Distillation model, which achieved the highest average performance, the shortest inference time, and the lowest error rate, was ultimately selected. To make this model accessible to potential users who experience inconvenience due to missing accents, the model is proposed to be deployed in a web environment. Rather than using the original PyTorch model directly, it is converted into the ONNX format to ensure consistent performance not only on the web but also across various other platforms in the future.

As a new sentence is entered, it is first embedded using the tokenizer, then passed through the encoder and decoder to generate a new vector representation. This vector is then converted back into a French sentence using the tokenizer and replaces the original input, which is then displayed to the user.

5.2. Survey Results

Question : Was the Service Useful?



A follow-up survey conducted with the original participants revealed that, except for one person, all respondents preferred this program over the traditional method and expressed satisfaction and willingness to use it again.

6. Conclusion

6.1. Significance

This study effectively defined a concrete and practically relevant research problem in the domain of text generation fidelity, specifically focusing on the restoration of diacritical marks in French—a task with both linguistic and pedagogical significance. Leveraging a large-scale dataset and substantial computational resources, we conducted rigorous and extensive experiments under time constraints, enabling robust evaluation across various compression and optimization strategies.

Particular emphasis was placed on model efficiency and deployment feasibility, with lightweight optimization techniques such as pruning and quantization being successfully applied. The current implementation of the low-rank adaptation (LoRA) technique requires further refinement to reach comparable performance. The resulting models not only

maintained high accuracy but also achieved substantial reductions in inference time and memory footprint, making them suitable for real-world applications.

The restored French text has potential utility as an educational tool, particularly in supporting language learners in developing orthographic accuracy and fluency. Beyond education, such models may also contribute to accessibility tools, digital publishing, and linguistic preservation efforts. Future work will involve continued collaboration with our academic advisor to refine the methodology and broaden the application scope.

6.2. Limitations

While the accent restoration model demonstrated generally strong performance across most test cases, several notable limitations were identified during evaluation. These issues, although not pervasive, highlight areas where the model's reliability and linguistic fidelity can be improved.

6.2.1. Overgeneration of Accents

In some instances, the model introduced accents where none were needed, leading to incorrect word forms. This phenomenon, referred to as overgeneration, resulted in output tokens that are not valid French words. For example, "Debout" was incorrectly predicted as "Débout", which is a non-existent or semantically incorrect form. This indicates that the model may be over-reliant on surface-level patterns or may lack sufficient grammatical constraint mechanisms.

6.2.2. Out-of-Vocabulary (OOV) Errors

The model struggled with certain rare or domain-specific words that were likely underrepresented or absent from the training data. In such cases, the model often produced phonetically or structurally similar but incorrect tokens. This underscores the need for a more comprehensive vocabulary or improved generalization techniques, especially for long-tail linguistic phenomena.

6.2.3. Special Token Handling Issues

There were notable difficulties in correctly handling special characters and ligatures such as Œ, œ, Æ, æ, É, À, which are important in French orthography. These characters were occasionally omitted, replaced, or mishandled during prediction. This suggests that the tokenizer or preprocessing pipeline may need refinement to fully support extended Latin characters and diacritics.

6.3. References

Popović, M. (2015). chrF:
Character n-gram F-score for automatic MT evaluation. Proceedings of the Tenth Workshop on Statistical Machine Translation, 392–395. https://aclanthology.org/W15-3049/

Hinton, G., Vinyals, O., & Dean, J. (2015).
Distilling the knowledge in a neural network.
arXiv preprint arXiv:1503.02531.
https://arxiv.org/abs/1503.02531

Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution:
Application to accent restoration in Spanish and French. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL). https://aclanthology.org/P94-1013/

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., & Raffel, C. (2022). ByT5:
Towards a token-free future with pre-trained byte-to-byte models.
arXiv preprint arXiv:2205.05131.
https://arxiv.org/abs/2205.05131

Huang, Z., Xu, W., & Yu, K. (2015).
Bidirectional LSTM-CRF models for sequence tagging.
arXiv preprint arXiv:1508.01991.
https://arxiv.org/abs/1508.01991