# Corrección Eficiente y Precisa de Textos en Español Antiguo OCR con LLMs Locales

**I. Introducción: El Desafío de Digitalizar y Corregir Textos Históricos en Español**

El acceso y análisis de documentos históricos escritos en español durante los siglos XVIII y XIX reviste una gran importancia para la investigación histórica, lingüística y cultural. Estos textos primarios ofrecen perspectivas únicas sobre la evolución de la lengua, las costumbres sociales y los acontecimientos de la época. La tecnología de Reconocimiento Óptico de Caracteres (OCR) desempeña un papel fundamental en la transformación de estos documentos físicos en texto digital, permitiendo que vastas colecciones sean buscables y analizables a gran escala 1. De hecho, la inteligencia artificial está revolucionando la investigación histórica al facilitar el procesamiento de miles de documentos frágiles sin manipulación física, convirtiéndolos en formatos digitales consultables 1.

Sin embargo, la aplicación de OCR a documentos históricos presenta desafíos inherentes 1. La antigüedad y el estado de conservación del papel, que a menudo presenta amarilleamiento y deterioro, así como la calidad desigual de la tinta, que puede estar desvanecida o ser inconsistente, dificultan la correcta identificación de los caracteres 1. Además, el uso de fuentes arcaicas y, en ocasiones, ornamentadas, que difieren significativamente de las tipografías modernas, y la presencia de texto manuscrito en algunos documentos, complican aún más el proceso de reconocimiento 2. Estos factores contribuyen a la aparición de diversos errores en el texto digitalizado, incluyendo el reconocimiento incorrecto de caracteres, la mala interpretación de la puntuación y la introducción de caracteres espurios 2. La calidad del resultado inicial del OCR influye significativamente en la eficacia de cualquier proceso de corrección posterior. Una mayor cantidad de errores en el OCR requerirá una corrección más exhaustiva, lo que podría aumentar el tiempo de procesamiento y reducir la precisión si el método de corrección tiene dificultades con el volumen o el tipo de errores.

A estos desafíos técnicos se suman las considerables dificultades lingüísticas que plantea el español de los siglos XVIII y XIX 1, que difiere del español moderno en aspectos como la ortografía (variaciones en la grafía de las palabras), la gramática (por ejemplo, conjugaciones verbales, uso de pronombres), las reglas de acentuación 3 y las opciones léxicas (uso de palabras que hoy son arcaicas o han cambiado su significado). Esta evolución lingüística exige métodos de corrección que sean sensibles a estas variaciones históricas y que puedan distinguirlas de los errores genuinos del OCR. La necesidad del usuario de "eficiencia y precisión" subraya la importancia de equilibrar los recursos computacionales con la calidad del texto corregido. Diferentes enfoques, como el uso directo de indicaciones (prompts) o el ajuste fino (fine-tuning), tendrán distintos costos computacionales y potencial de precisión. La solución óptima dependerá de las restricciones y requisitos específicos del usuario.

El usuario busca métodos eficientes y precisos para corregir estas variaciones inducidas por el OCR y las lingüísticas históricas utilizando un Modelo de Lenguaje Grande (LLM) local. Además, muestra un interés específico en la viabilidad de alterar el tokenizador del LLM para manejar conversiones directas de formas de palabras entre el español arcaico y el moderno.

**II. Entendiendo la Evolución Lingüística del Español (Siglos XVIII-XIX)**

El español de los siglos XVIII y XIX presenta diferencias gramaticales y ortográficas clave en comparación con el español moderno 1. Por ejemplo, se observan variaciones comunes en la ortografía de ciertas palabras, como "avía" en lugar de "había", "fierro" en lugar de "hierro" o "estoi" en lugar de "estoy". Las reglas de acentuación también experimentaron cambios; en el pasado, era más frecuente el uso de acentos, o estos se colocaban de manera diferente. En cuanto a la gramática, aunque los cambios no fueron tan radicales como en la ortografía, se pueden encontrar diferencias en las conjugaciones verbales y el uso de pronombres, como el uso de "vosotros" en algunas regiones de América Latina durante ciertos periodos.

El proyecto Historical Ink ha identificado cambios específicos en la forma de las palabras 3, Apéndice B] que ilustran estas variaciones ortográficas. Algunos ejemplos incluyen la alternancia entre á y a (hara → hará), é y e (fué → fue), í y i (decia → decía), ó y o (ocasion → ocasión), ú y u (ningun → ningún), i y y (mui → muy), j y g (jente → gente), v y b (gravado → grabado), s y x (espiró → expiró), j y x (méjico → méxico), c y s (faces → fases) y s y z (dies → diez). Estos ejemplos concretos responden directamente a la pregunta del usuario sobre las conversiones de formas de palabras y resaltan la naturaleza sistemática de algunos de estos cambios. Comprender estos patrones puede informar el desarrollo de estrategias de corrección, incluyendo la posibilidad de reglas o modificaciones específicas, o para la creación de datos de entrenamiento en el ajuste fino.

Es fundamental que un sistema de corrección sea lo suficientemente inteligente como para distinguir entre errores genuinos del OCR (por ejemplo, una sustitución de caracteres completamente sin sentido) y estas variaciones lingüísticas históricas 1. Esto evitará una sobrecorrección o la modernización inapropiada del texto cuando la preservación de la forma original es deseable para fines académicos. La metodología del proyecto Historical Ink, que clasifica las correcciones 1, es crucial para esta distinción. Por ejemplo, la clasificación de los cambios de acento como principalmente formas superficiales refleja las reglas de acentuación históricas en lugar de errores del OCR. La capacidad de diferenciar entre errores de OCR y formas históricas es primordial para mantener la integridad del texto histórico. Aplicar indiscriminadamente las reglas gramaticales y ortográficas del español moderno alteraría el carácter histórico del documento, lo que podría ser indeseable para la investigación. El objetivo debería ser corregir los errores introducidos por el proceso de digitalización, no modernizar el lenguaje.

La investigación sobre la detección del cambio semántico 5 subraya aún más la evolución lingüística del español durante este período, aunque el enfoque principal del usuario esté en la sintaxis y la ortografía. Si bien no se trata directamente de la corrección gramatical, la conciencia de posibles cambios semánticos puede proporcionar una comprensión más holística del lenguaje de la época y podría ser relevante en casos matizados donde el significado de una palabra ha cambiado, lo que podría influir en el uso gramatical.

**III. Utilización de Modelos de Lenguaje Grandes Locales para la Corrección Gramatical y Ortográfica: Un Análisis Comparativo de Enfoques**

* **A. Uso Directo de Indicaciones (Prompts) para la Corrección:**  
  Este enfoque consiste en proporcionar el texto obtenido mediante OCR directamente a un LLM local, junto con una indicación que le instruya para que corrija la ortografía y la sintaxis. Es importante especificar que el texto está escrito en español de los siglos XVIII y XIX, y que las formas arcaicas deben manejarse de manera adecuada (ya sea corrigiéndolas a la forma moderna si se desea, o preservándolas si el LLM es capaz de hacerlo).  
  Entre los posibles beneficios de este método se encuentra su relativa simplicidad de implementación, especialmente si ya se dispone de un LLM local configurado 6. Muchos LLMs de código abierto tienen capacidades multilingües que incluyen el español 9. Sin embargo, también existen posibles inconvenientes. Los LLMs de propósito general podrían no estar entrenados específicamente en, o no tener un conocimiento profundo de, los matices del español de los siglos XVIII y XIX. Esto podría llevar a imprecisiones, una modernización excesiva o una falta de comprensión de las estructuras gramaticales históricas 2. El rendimiento podría variar significativamente dependiendo del LLM específico utilizado, sus datos de entrenamiento y la calidad y especificidad de la indicación. Además, el procesamiento de textos muy largos podría verse limitado por el tamaño de la ventana de contexto del LLM 9, lo que requeriría dividir el texto en fragmentos, con la posible pérdida de contexto entre frases.  
  Es crucial una cuidadosa ingeniería de las indicaciones para guiar al LLM de manera efectiva, especificando el idioma, el período histórico y el nivel de modernización deseado (si lo hay). La eficacia del uso directo de indicaciones probablemente dependerá en gran medida del conocimiento base del LLM sobre el español histórico. Los modelos con un entrenamiento multilingüe más amplio y una ventana de contexto mayor podrían ofrecer mejores resultados. Un modelo con cierta exposición a diversos textos en español de diferentes períodos de tiempo podría ser más propenso a reconocer y manejar las formas arcaicas de manera apropiada. Una ventana de contexto más grande permite al modelo considerar más texto circundante para una mejor corrección contextual.
* **B. Ajuste Fino (Fine-tuning) de un LLM Existente con un Corpus de Español Histórico:**  
  Este enfoque implica tomar un LLM multilingüe pre-entrenado (como BLOOM o LLaMA 3 9) y entrenarlo adicionalmente con un conjunto de datos de textos en español de los siglos XVIII y XIX que hayan sido transcritos correctamente y, potencialmente, modernizados (si ese es el objetivo) 14.  
  Entre las posibles ventajas se encuentra la capacidad de adaptar el LLM a las características lingüísticas específicas del período objetivo, lo que podría llevar a una precisión significativamente mayor en el reconocimiento y corrección de formas y errores históricos. También permite al modelo aprender los patrones específicos de ortografía, gramática e incluso errores comunes de OCR asociados con documentos históricos. Sin embargo, existen posibles desventajas. Se requiere acceso a un corpus suficientemente grande y de alta calidad de texto en español histórico con transcripciones y correcciones precisas (un requisito nada trivial y que podría requerir un esfuerzo considerable para crear o seleccionar). El conjunto de datos Latam-XIX 2 podría ser un recurso valioso en este sentido. El ajuste fino de un LLM grande puede ser computacionalmente intensivo y llevar mucho tiempo, requiriendo importantes recursos de hardware (GPU, memoria) y tiempo. Además, podría requerir experiencia en aprendizaje automático, metodologías de entrenamiento de LLMs y preparación de datos.  
  El proyecto de Kaggle 14 es una demostración del ajuste fino para la corrección gramatical utilizando el modelo T5 (aunque en inglés moderno), que destaca el flujo de trabajo general de preparación de datos, entrenamiento del modelo y evaluación. El ajuste fino ofrece el potencial de la mayor precisión al adaptar el LLM a los matices específicos del español histórico, pero exige recursos y datos sustanciales. Al entrenar con ejemplos de español histórico, el LLM puede aprender los patrones y variaciones específicos de ese período, lo que le permite corregir errores de manera más efectiva y manejar las formas arcaicas de manera apropiada.
* **C. Utilización de LLMs Especializados Entrenados con Datos Lingüísticos Históricos:**  
  Se puede explorar la posibilidad de utilizar LLMs que hayan sido entrenados específicamente con corpus de español histórico, como los mencionados en el contexto del proyecto Historical Ink (GPT-3.5, GPT-4o-mini 1) o modelos centrados en el español como MarIA o LINCE 15, si han sido entrenados con datos históricos.  
  Los posibles beneficios incluyen una comprensión significativamente mejor de los matices lingüísticos del español de los siglos XVIII y XIX en comparación con los modelos multilingües de propósito general. Esto podría llevar a correcciones más precisas y contextualmente apropiadas. Además, podría requerir menos ajuste fino o indicaciones muy específicas para lograr buenos resultados. Sin embargo, el acceso a estos modelos especializados podría ser limitado. Los modelos propietarios como GPT-4o-mini podrían requerir acceso a través de una API y no estar disponibles para su implementación local. La disponibilidad local y los datos de entrenamiento específicos de modelos como MarIA y LINCE deberían ser investigados. Su rendimiento y adecuación para el tipo específico de texto del usuario (por ejemplo, periódicos frente a otros tipos de documentos) también deberían ser evaluados.  
  El éxito del proyecto Historical Ink 1 es una fuerte evidencia del potencial de este enfoque, destacando su capacidad para corregir errores de OCR mientras se preservan las grafías históricas. Si están accesibles localmente o a través de una API viable, los LLMs especializados entrenados en español histórico son probablemente la opción más prometedora para lograr una alta precisión con potencialmente menos esfuerzo en la ingeniería de indicaciones o el ajuste fino. Sus datos de entrenamiento deberían haberlos equipado con una comprensión inherente del lenguaje de la época, lo que les permitiría distinguir mejor entre errores y variaciones históricas.
* **Tabla 1: Comparación de Enfoques con LLMs**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Enfoque** | **Eficiencia Potencial** | **Precisión Potencial** | **Requisitos de Recursos** | **Requisitos de Datos** | **Facilidad de Implementación** |
| Uso Directo de Indicaciones | Alta | Baja a Media | Baja | Baja | Alta |
| Ajuste Fino | Media | Alta | Alta | Alta | Baja |
| LLMs Especializados | Media a Baja | Alta | Media | Media | Media |

**IV. Eficiencia vs. Precisión: Equilibrando Recursos Computacionales y Calidad de la Corrección**

Existe una relación de intercambio inherente entre los diferentes enfoques de LLM en términos de recursos computacionales (uso de CPU/GPU, requisitos de memoria para ejecutar el LLM local), tiempo de procesamiento y el nivel esperado de precisión en el texto corregido.

Los enfoques más simples, como el uso directo de indicaciones con LLMs locales más pequeños (por ejemplo, modelos con menos parámetros), podrían ser más eficientes en términos de uso de recursos y velocidad de procesamiento, pero es probable que sean menos precisos al tratar con las complejidades del español histórico. Por otro lado, los enfoques más precisos, como el ajuste fino de LLMs grandes o la utilización de modelos especializados, potencialmente más grandes, probablemente requerirán un hardware más potente y tiempos de procesamiento más largos, pero deberían producir mejores resultados en términos de capturar los matices del lenguaje y distinguir entre errores y formas históricas.

Los pasos de preprocesamiento utilizando herramientas basadas en reglas como LanguageTool 17 podrían desempeñar un papel importante en la mejora de la eficiencia. Al abordar los errores ortográficos comunes y los errores gramaticales básicos antes de alimentar el texto al LLM, se podría reducir la carga computacional del LLM y su enfoque podría dirigirse a las variaciones históricas más complejas y los errores específicos del OCR. La experiencia de un usuario de Reddit 23 al utilizar LanguageTool antes de la indicación de un LLM para inglés moderno podría servir como una analogía potencial. Un enfoque de múltiples etapas que combine diferentes herramientas con distintos niveles de complejidad y requisitos de recursos podría ofrecer el mejor equilibrio general entre eficiencia y precisión. Aprovechar las fortalezas de diferentes herramientas, por ejemplo, LanguageTool para comprobaciones estándar y un LLM especializado para las complejidades del lenguaje histórico, podría ser más eficiente y preciso que depender únicamente de un único modelo muy grande para todas las tareas de corrección.

**V. La Viabilidad e Implicaciones de la Modificación del Tokenizador para el Español Arcaico**

La pregunta del usuario sobre la viabilidad técnica y práctica de alterar el tokenizador del LLM para manejar directamente palabras con conversiones arcaico-modernas comunes (por ejemplo, mapear "vió" a "vio") requiere una consideración detallada del papel fundamental de los tokenizadores en los LLMs 24. Los tokenizadores son responsables de dividir el texto en unidades más pequeñas llamadas tokens, definir el vocabulario (el conjunto de todos los tokens posibles) y asignar identificadores únicos a estos tokens. Algoritmos comunes de tokenización incluyen la codificación por pares de bytes (BPE) 24.

Modificar el tokenizador de un LLM pre-entrenado para manejar específicamente formas de palabras arcaicas del español presenta varios desafíos técnicos. Implicaría identificar las formas arcaicas comunes y sus equivalentes modernos, y potencialmente agregar estas formas arcaicas como nuevos tokens al vocabulario del tokenizador si no están ya presentes. Sin embargo, lo más importante sería establecer un mapeo o asociación directa entre el token arcaico y su contraparte moderna dentro del espacio de incrustación del modelo y las capas posteriores. Esta es una tarea compleja. El proceso podría perturbar la comprensión del lenguaje existente en el modelo, ya que el significado y el contexto asociados con los tokens originales podrían verse afectados por la introducción de estos nuevos tokens, aunque estén relacionados. Además, probablemente sería necesario volver a entrenar o al menos ajustar finamente el LLM después de modificar el tokenizador para garantizar que pueda utilizar eficazmente los nuevos tokens y mantener sus capacidades generales de comprensión del lenguaje 32. La investigación sobre la expansión del vocabulario para LLMs bilingües 32 es una tarea relacionada pero significativamente más compleja, donde el objetivo es agregar tokens para un idioma completo, no solo mapeos específicos de formas de palabras.

Si bien este enfoque podría mejorar la eficiencia para estas conversiones muy específicas y frecuentes al mapear directamente las formas arcaicas a las modernas en la etapa inicial del procesamiento de texto, y podría reducir la necesidad de que el LLM "aprenda" estos mapeos directos durante la indicación o el ajuste fino, existen importantes inconvenientes potenciales. Podría no ser una solución escalable o integral para todos los matices del español histórico, que implica diferencias gramaticales, variaciones ortográficas más complejas y cambios léxicos más allá de simples mapeos directos de palabras. Podría introducir efectos secundarios o inconsistencias no deseadas en el procesamiento de otras partes del texto por parte del modelo. Además, es probable que sea considerablemente más complejo técnicamente y requiera más recursos que simplemente ajustar finamente el LLM o utilizar modelos especializados que ya estén entrenados para manejar tales variaciones. Requeriría una comprensión profunda de la arquitectura del LLM y el proceso de entrenamiento.

Si bien técnicamente concebible, la modificación directa del tokenizador para conversiones específicas de palabras arcaico-modernas probablemente no sea el enfoque más eficiente, práctico o robusto para la corrección integral del español histórico. Las diferencias lingüísticas entre el español histórico y el moderno son multifacéticas y van más allá de simples mapeos palabra por palabra. Ajustar finamente un LLM en un corpus de español histórico o utilizar un modelo ya entrenado en dichos datos tiene más probabilidades de capturar estas complejidades de manera efectiva y consistente.

**VI. Explorando Herramientas y Modelos de Código Abierto Existentes para el Procesamiento de Texto en Español Histórico**

* **A. LanguageTool:**  
  LanguageTool es una herramienta de código abierto que puede ejecutarse localmente 17 y ofrece soporte para español 18. Se sugiere utilizarlo como una valiosa herramienta de primera pasada para corregir la ortografía estándar y los errores gramaticales básicos en el texto obtenido mediante OCR antes de recurrir a un LLM para correcciones más complejas y el manejo de especificidades históricas. Sin embargo, es importante reconocer sus limitaciones con respecto a las formas arcaicas y las estructuras gramaticales específicas prevalentes en el español de los siglos XVIII y XIX, ya que sus reglas están diseñadas principalmente para el lenguaje moderno.
* **B. LLMs de Código Abierto:**  
  Se pueden considerar los LLMs de código abierto con capacidades multilingües que incluyen el español (BLOOM, LLaMA 3, Mistral, GPT-NeoX, etc9.). Se sugiere experimentar con la indicación de estos modelos para corregir el texto en español histórico, proporcionando instrucciones claras sobre el idioma y el período de tiempo. BLOOM 9 destaca por su extenso entrenamiento en un gran corpus multilingüe, que incluye una cantidad significativa de texto en español, potencialmente abarcando diferentes períodos de tiempo. LLaMA 3 10 también es relevante por sus sólidas capacidades generales de comprensión del lenguaje y su gran ventana de contexto, lo que podría ser beneficioso para procesar documentos históricos más largos.
* **C. Proyecto Historical Ink y Recursos Relacionados:**  
  Los resultados significativos del proyecto Historical Ink 1, que utilizó con éxito LLMs (GPT-3.5 y GPT-4o-mini) para la corrección posterior al OCR de periódicos latinoamericanos del siglo XIX, demuestran la viabilidad de este enfoque. Sería útil investigar si los modelos específicos utilizados en este proyecto o versiones ajustadas de otros LLMs basados en su metodología están disponibles públicamente para uso local o con fines de investigación. Además, el conjunto de datos Latam-XIX está disponible en Hugging Face 2, que contiene textos de periódicos latinoamericanos del siglo XIX y podría utilizarse potencialmente para ajustar finamente un LLM local si el usuario dispone de los recursos y la experiencia necesarios.
* **D. Transkribus:**  
  Transkribus 34 es una plataforma especializada que utiliza transcripción impulsada por IA y ofrece modelos específicamente entrenados para el análisis de documentos históricos, incluyendo textos en español antiguo. Se sugiere explorar sus modelos de IA disponibles públicamente para español y la opción de entrenar modelos personalizados para una precisión aún mayor en tipos de documentos o estilos de escritura específicos. También ofrece capacidades de integración a través de una API, lo que podría permitir incorporar su funcionalidad en los flujos de trabajo existentes del usuario. Transkribus proporciona una solución más específica para el dominio en comparación con los LLMs de propósito general o LanguageTool, lo que podría ofrecer una mayor precisión para el procesamiento de texto histórico. Sus modelos están entrenados con una gran cantidad de datos de documentos históricos, lo que los hace más adecuados para manejar los desafíos específicos del OCR en dichos materiales y las características lingüísticas de las lenguas históricas.

**VII. Soluciones y Recomendaciones Propuestas: Un Enfoque Multifacético**

* **Propuesta 1: Enfoque Híbrido con LanguageTool y un LLM Multilingüe:**  
  Utilizar LanguageTool (ejecutado localmente) como un paso inicial para corregir errores ortográficos comunes y gramática básica en el texto en español de los siglos XVIII y XIX obtenido mediante OCR. Posteriormente, emplear un LLM multilingüe local (por ejemplo, LLaMA 3 o BLOOM) e indicarle que refine aún más el texto, instruyéndole específicamente para que comprenda y maneje las convenciones gramaticales y ortográficas de ese período histórico. Experimentar con diferentes estrategias de indicación. Este enfoque aprovecha la eficiencia de una herramienta basada en reglas para correcciones estándar, al tiempo que utiliza la comprensión contextual de un LLM para variaciones históricas más matizadas y posibles errores de OCR que LanguageTool podría pasar por alto.
* **Propuesta 2: Ajuste Fino de un LLM Multilingüe en un Corpus de Español Histórico:**  
  Si el usuario tiene acceso a, o puede seleccionar, un corpus suficientemente grande y de alta calidad de texto en español de los siglos XVIII y XIX transcrito correctamente (y potencialmente modernizado si se desea) (por ejemplo, utilizando el conjunto de datos Latam-XIX como punto de partida), considerar el ajuste fino de un potente LLM local como LLaMA 3 o BLOOM con estos datos. Esto tiene el potencial de producir la mayor precisión al adaptar directamente el LLM a las características lingüísticas específicas del período histórico objetivo. Sin embargo, requiere una cantidad significativa de datos, recursos computacionales y experiencia técnica.
* **Propuesta 3: Exploración y Utilización de LLMs Especializados para Español Histórico:**  
  Investigar a fondo la disponibilidad de versiones locales o APIs accesibles para LLMs entrenados específicamente en español histórico, como los utilizados en el proyecto Historical Ink (si están disponibles públicamente para investigación) u otros modelos como MarIA o LINCE que podrían haber sido entrenados con datos históricos relevantes. Si tales modelos son accesibles, probar su rendimiento en el texto OCR del usuario, comparando su precisión y eficiencia con los LLMs de propósito general. Estos modelos probablemente tengan la mejor comprensión inherente de los matices lingüísticos del español de los siglos XVIII y XIX, lo que podría llevar a las correcciones más precisas con menos necesidad de indicaciones extensas o ajuste fino.
* **Propuesta 4: Evaluación y Potencial Uso de una Plataforma como Transkribus:**  
  Explorar la plataforma Transkribus, sus funciones para transcribir y corregir textos históricos en español utilizando modelos de IA, y su estructura de precios. Considerar el uso de sus modelos disponibles públicamente o la opción de entrenar un modelo personalizado con el tipo específico de documentos del usuario. Investigar su API para una posible integración con otras herramientas o flujos de trabajo que el usuario pueda tener. Transkribus ofrece una solución dedicada y potencialmente muy precisa para el procesamiento de documentos históricos, lo que podría ser más eficiente y eficaz que intentar adaptar LLMs de propósito general sin entrenamiento o indicaciones especializadas.

**VIII. Conclusión: Hacia una Corrección Robusta y Precisa de Textos Históricos en Español con LLMs Locales**

La corrección de textos en español de los siglos XVIII y XIX obtenidos mediante OCR presenta desafíos significativos debido tanto a los errores del OCR como a la evolución lingüística del idioma. Al seleccionar un método de corrección, es crucial considerar cuidadosamente el equilibrio entre la eficiencia y la precisión. Si bien la modificación del tokenizador de un LLM para conversiones específicas de formas de palabras arcaicas es técnicamente compleja y probablemente no sea el enfoque más práctico para una corrección integral, otras estrategias ofrecen vías prometedoras.

Un enfoque híbrido que combine herramientas basadas en reglas como LanguageTool con la comprensión contextual de los LLMs multilingües presenta un gran potencial. Además, el ajuste fino de un LLM en un corpus relevante de español histórico o la utilización de LLMs especializados entrenados específicamente en dichos datos, como lo demuestra el proyecto Historical Ink, podrían lograr beneficios significativos. También se sugiere explorar plataformas como Transkribus como una solución dedicada para el procesamiento de documentos históricos.

Los continuos avances en el procesamiento del lenguaje natural y la creciente disponibilidad de recursos y modelos para el procesamiento de lenguas históricas auguran un futuro prometedor para mejorar la eficiencia y la precisión de estas tareas.

#### Obras citadas

1. Historical Ink: 19th Century Latin American Spanish Newspaper ..., fecha de acceso: marzo 18, 2025, <https://www.promptlayer.com/research-papers/historical-ink-19th-century-latin-american-spanish-newspaper-corpus-with-llm-ocr-correction>
2. Historical Ink: 19th Century Latin American Spanish Newspaper Corpus with LLM OCR Correction - arXiv, fecha de acceso: marzo 18, 2025, <https://arxiv.org/html/2407.12838v1>
3. Historical Ink: 19th Century Latin American Spanish Newspaper Corpus with LLM OCR Correction - arXiv, fecha de acceso: marzo 18, 2025, <https://arxiv.org/html/2407.12838v2>
4. Historical Ink: 19th Century Latin American Spanish Newspaper Corpus with LLM OCR Correction - ResearchGate, fecha de acceso: marzo 18, 2025, <https://www.researchgate.net/publication/386201387_Historical_Ink_19th_Century_Latin_American_Spanish_Newspaper_Corpus_with_LLM_OCR_Correction>
5. Historical Ink: Semantic Shift Detection for 19th Century Spanish - ACL Anthology, fecha de acceso: marzo 18, 2025, <https://aclanthology.org/2024.lchange-1.4.pdf>
6. LLM Writing Tool - Visual Studio Marketplace, fecha de acceso: marzo 18, 2025, <https://marketplace.visualstudio.com/items?itemName=OlePetersen.lm-writing-tool>
7. pluja/llm-language-tool: Use LLMs to enhance, translate and correct your texts - GitHub, fecha de acceso: marzo 18, 2025, <https://github.com/pluja/llm-language-tool>
8. Re: Script to integrate local LLM for proofreading - Adobe Community - 14486527, fecha de acceso: marzo 18, 2025, <https://community.adobe.com/t5/framemaker-discussions/script-to-integrate-local-llm-for-proofreading/m-p/14499275>
9. The Top 10 Open Source LLMs: 2024 Edition - Scribble Data, fecha de acceso: marzo 18, 2025, <https://www.scribbledata.io/blog/the-top-10-open-source-llms-2024-edition/>
10. The 11 best open-source LLMs for 2025 - n8n Blog, fecha de acceso: marzo 18, 2025, <https://blog.n8n.io/open-source-llm/>
11. Top 10 Open-Source LLM Models and Their Uses | by Yugank .Aman | Feb, 2025 - Medium, fecha de acceso: marzo 18, 2025, <https://medium.com/@yugank.aman/top-10-open-source-llm-models-and-their-uses-6f4a9aced6af>
12. 8 Top Open-Source LLMs for 2024 and Their Uses - DataCamp, fecha de acceso: marzo 18, 2025, <https://www.datacamp.com/blog/top-open-source-llms>
13. Evaluating Large Language Models with Tests of Spanish as a Foreign Language: Pass or Fail? - ResearchGate, fecha de acceso: marzo 18, 2025, <https://www.researchgate.net/publication/384295034_Evaluating_Large_Language_Models_with_Tests_of_Spanish_as_a_Foreign_Language_Pass_or_Fail>
14. Grammar Error Correction Using LLM - Kaggle, fecha de acceso: marzo 18, 2025, <https://www.kaggle.com/code/hakim11/grammar-error-correction-using-llm>
15. Large Language Models (LLMs) for Developers in Spain - BytePlus, fecha de acceso: marzo 18, 2025, <https://www.byteplus.com/en/topic/392799>
16. 1. An AI that thinks in Spanish - Cátedra Ciencia y Sociedad - Fundación Rafael del Pino, fecha de acceso: marzo 18, 2025, <https://frdelpino.es/ciencia-y-sociedad/en_gb/1-an-ia-that-thinks-in-spanish/>
17. Top Free Spell Check tools, APIs, and Open Source models - Eden AI, fecha de acceso: marzo 18, 2025, <https://www.edenai.co/post/top-free-spell-check-tools-apis-and-open-source-models>
18. languagetool-org/languagetool: Style and Grammar Checker for 25+ Languages - GitHub, fecha de acceso: marzo 18, 2025, <https://github.com/languagetool-org/languagetool>
19. LanguageTool, your multilingual writing assistant - techforword, fecha de acceso: marzo 18, 2025, <https://www.techforword.com/blog/languagetool-review>
20. Spanish Grammar Checker - LanguageTool, fecha de acceso: marzo 18, 2025, <https://languagetool.org/spellchecking-spanish>
21. LanguageTool: Free Grammar Checker, fecha de acceso: marzo 18, 2025, <https://languagetool.org/>
22. Tool for Proofreading in Spanish - LanguageTool Review - Lingua Linkup, fecha de acceso: marzo 18, 2025, <https://lingualinkup.com/proofreading-spanish/>
23. Tip: Run a local LanguageTool server to correct your spelling/grammar in your prompts, fecha de acceso: marzo 18, 2025, <https://www.reddit.com/r/LocalLLaMA/comments/1alks1j/tip_run_a_local_languagetool_server_to_correct/>
24. LLM Tokenization - Hundred Blocks, fecha de acceso: marzo 18, 2025, <https://hundredblocks.github.io/transcription_demo/>
25. Tokens and Tokenization in Large Language Models in Python.md - GitHub, fecha de acceso: marzo 18, 2025, <https://github.com/xbeat/Machine-Learning/blob/main/Tokens%20and%20Tokenization%20in%20Large%20Language%20Models%20in%20Python.md>
26. Understanding Tokenizers in LLM — Part 1 : Byte Pair Encoding and Beyond - Medium, fecha de acceso: marzo 18, 2025, <https://medium.com/@mukulranjan/understanding-tokenizers-in-llm-part-1-byte-pair-encoding-and-beyond-75e4c4e70e6f>
27. T-FREE: Tokenizer-Free Generative LLMs via Sparse Representations for Memory-Efficient Embeddings - arXiv, fecha de acceso: marzo 18, 2025, <https://arxiv.org/html/2406.19223v1>
28. Introduction to Tokenizers in Large Language Models (LLMs) using Wardley Maps - Medium, fecha de acceso: marzo 18, 2025, <https://medium.com/@mcraddock/introduction-to-tokenizers-in-large-language-models-llms-using-wardley-maps-652ee4dd6227>
29. The Technical User's Introduction to LLM Tokenization - Christopher Samiullah, fecha de acceso: marzo 18, 2025, <https://christophergs.com/blog/understanding-llm-tokenization>
30. Introduction to LLM Tokenization - Airbyte, fecha de acceso: marzo 18, 2025, <https://airbyte.com/data-engineering-resources/llm-tokenization>
31. Understanding tokens - .NET - Microsoft Learn, fecha de acceso: marzo 18, 2025, <https://learn.microsoft.com/en-us/dotnet/ai/conceptual/understanding-tokens>
32. From English-Centric to Effective Bilingual: LLMs with Custom Tokenizers for Underrepresented Languages - arXiv, fecha de acceso: marzo 18, 2025, <https://arxiv.org/html/2410.18836v1>
33. LanguageTool (old version) - Chrome Web Store, fecha de acceso: marzo 18, 2025, <https://chromewebstore.google.com/detail/languagetool-old-version/fbnhlpaknddgpjajidodjdidnaonmhcl>
34. Old Spanish Texts - Transkribus, fecha de acceso: marzo 18, 2025, <https://www.transkribus.org/languages/spanish>