Memoria de prácticas

Traducción automática

Iñaki Diez Lambies

Curso 2023 2024

Tabla de contenido

[1. Introducción 2](#_Toc158465523)

[2. Metodología 3](#_Toc158465524)

[3. Experimentación 4](#_Toc158465525)

[3.1. MOSES 4](#_Toc158465526)

[3.1.1. Sin ajuste de pesos 4](#_Toc158465527)

[3.1.2. Valores más altos de iteraciones del MERT 5](#_Toc158465528)

[3.1.3. Variación en los valores de n-gramas 5](#_Toc158465529)

[3.1.4. Otras técnicas de suavizado 6](#_Toc158465530)

[3.1.5. Moses monótono 6](#_Toc158465531)

[3.2. OpenNMT 7](#_Toc158465532)

[3.2.1. Variación del tamaño de representación de palabras 7](#_Toc158465533)

[3.2.2. Variación del número de capas 8](#_Toc158465534)

[4. Comparativa de resultados 9](#_Toc158465535)

[5. Conclusiones 11](#_Toc158465536)

# Introducción

El presente trabajo tiene como objetivo explorar y comparar dos de las principales metodologías en el ámbito de la TA: la traducción estadística basada en frases, implementada mediante la herramienta MOSES, y la traducción basada en redes neuronales dinámicas, realizada a través de OpenNMT.

A través de los ejercicios propuestos, se busca no solo determinar cuál de estas herramientas proporciona los mejores resultados bajo condiciones específicas, sino también comprender cómo las variaciones en los parámetros afectan la calidad de la traducción automática.

Para llevar a cabo este trabajo, se ha seleccionado un conjunto de datos bilingües que permiten entrenar y evaluar los modelos de TA en un escenario controlado, proporcionando así una base sólida para la comparación objetiva de los resultados. La métrica principal utilizada para evaluar el rendimiento de los modelos es el score BLEU, un estándar en la industria que mide la calidad de las traducciones automáticas comparándolas con traducciones humanas de referencia.

# Metodología

En la metodología aplicada a este trabajo, se ha dado prioridad a la automatización y reproductibilidad de los experimentos mediante la implementación de scripts especializados para ambas prácticas, MOSES y OpenNMT. Los scripts diseñados se encargan de cada una de las etapas críticas del proceso de traducción automática: desde la limpieza de corpus y tokenización hasta el entrenamiento, traducción y evaluación de los modelos. Esta estructuración en scripts facilita no solo la ejecución secuencial de tareas complejas sino también asegura que se puedan replicar los experimentos con exactitud en diferentes entornos o en futuras investigaciones.

La utilización de scripts como *p1\_clean\_corpus.sh*, *p1\_evaluate.sh* o *p1\_train\_model.sh* en MOSES y *p2\_train.sh*, *p2\_translate.sh* o *p2\_evaluation.sh* en OpenNMT ofrece una capa de abstracción que permite centrarse en los resultados experimentales en lugar de en la gestión manual de los procesos, reduciendo significativamente el margen de error humano y aumentando la eficiencia del tiempo. Además, la estandarización de los procesos a través de scripts garantiza que los pasos seguidos en la preparación de los datos, las configuraciones de entrenamiento y las metodologías de evaluación sean transparentes y puedan ser examinados y verificados.

La decisión de ejecutar las herramientas de traducción en contenedores Docker ha sido fundamental para reforzar aún más la reproductibilidad de los experimentos. Docker proporciona entornos aislados llamados contenedores, donde el software se ejecuta con sus dependencias en un ecosistema controlado. Esto significa que los modelos de TA se entrenan y se evalúan en un ambiente uniforme y predecible, independientemente de las diferencias en las máquinas locales o en la infraestructura de cloud donde puedan ser desplegados. La dockerización de herramientas, ilustrada por scripts como *build\_moses.sh* y *run\_moses.sh* para MOSES, y *build\_opennmt.sh* y *run\_opennmt.sh* para OpenNMT, asegura que las variaciones en el entorno de desarrollo no afecten la comparabilidad de los resultados, lo cual es un desafío común en el campo del PLN.

La automatización mediante scripts y la dockerización tienen un impacto significativo en la eficacia y la escala de proyectos. Permite replicar y ampliar estudios previos con facilidad, garantizan que las pruebas se realicen de manera consistente y ofrecen una solución robusta para la gestión de la diversidad y complejidad de las configuraciones experimentales. En suma, estas decisiones metodológicas optimizan los recursos y esfuerzos en el presente estudio.

# Experimentación

En este trabajo, se ha realizado una serie de experimentos estructurados y metódicos, cada uno asociado con un script específico que facilita su ejecución. Esta sección detalla los experimentos realizados, los scripts que los automatizan y los resultados obtenidos, incluyendo la métrica BLEU para evaluar la calidad de las traducciones.

La selección de los experimentos y la creación de los scripts correspondientes han sido diseñadas para abordar aspectos específicos de cada práctica. Cada experimento se ha ejecutado utilizando su script correspondiente, lo que garantiza que el proceso se realice de manera sistemática y que los resultados sean reproducibles. Los resultados, recopilados y analizados tras cada ejecución, proporcionan una comprensión profunda del comportamiento del modelo bajo diferentes configuraciones, y el score BLEU obtenido sirve como referencia cuantitativa del rendimiento de la traducción.

A continuación, se presentará cada experimento junto con su script de ejecución y los resultados del BLEU, proporcionando una visión integral y detallada de la experimentación realizada en este estudio.

## MOSES

### Sin ajuste de pesos

Para el primer ejercicio utilizando MOSES, se ha desarrollado un script *run.sh* que describe paso a paso el proceso seguido para entrenar y evaluar el modelo de traducción estadística. Este proceso se inicia con la preparación de los datos, donde se crea un directorio de entrenamiento y se limpia el corpus mediante *p1\_clean\_corpus.sh*, asegurando que los datos de entrada estén en el formato adecuado y libres de incongruencias. Posteriormente, se genera el modelo de lenguaje utilizando *p1\_ngram.sh*, que es fundamental para construir la probabilidad de las secuencias de palabras en el idioma objetivo. El modelo de traducción se entrena con *p1\_train\_model.sh*, empleando el corpus limpio y el modelo de lenguaje previamente generado.

Una vez que el modelo está entrenado, se procede con la traducción del conjunto de prueba utilizando *p1\_translate.sh*, seguido de la evaluación con *p1\_evaluate.sh*, que compara las hipótesis de traducción con las traducciones de referencia para calcular el score BLEU. Este último paso es crítico, ya que el BLEU es una métrica reconocida para evaluar la calidad de las traducciones automáticas, indicando cuán cercanas están las traducciones automáticas a las de calidad humana.

Los resultados del modelo base sin ajuste de pesos han arrojado un BLEU de 88.42, lo cual es notablemente alto y sugiere que las traducciones producidas por el sistema tienen una gran similitud con las traducciones de referencia. Los detalles del score BLEU indican un alto grado de precisión a nivel unigram (94.0), que disminuye gradualmente a medida que se consideran n-gramas de mayor longitud, lo cual es esperado en la evaluación de BLEU. La precisión a nivel de bigramas fue del 89.3%, trigramas 86.6% y tetrágramas 84.0%, mostrando que incluso en secuencias de cuatro palabras, el modelo mantiene una alta coherencia con las traducciones de referencia.

El valor BP (brevity penalty) de 1.000 indica que la longitud de las hipótesis de traducción es comparable a la de las referencias, lo cual es ideal, ya que significa que el modelo no está produciendo traducciones ni demasiado cortas ni excesivamente largas. Además, la ratio de longitud de 1.040 confirma que las traducciones automáticas tienen una longitud ligeramente mayor que las referencias, pero aún dentro de un rango aceptable. El alto score BLEU obtenido indica que el modelo base de MOSES, incluso sin ajuste de pesos, es capaz de producir traducciones de alta calidad.

### Valores más altos de iteraciones del MERT

En el ejercicio 2, el proceso experimentado ha introducido una variación significativa en comparación con el modelo original: el número de iteraciones de MERT, con el objetivo de optimizar los pesos del modelo de traducción.

El script para este ejercicio incluye pasos adicionales destinados a la optimización de pesos con MERT. En concreto, se ha creado un modelo para cada uno de los tres números distintos de iteraciones MERT: 10, 20 y 50. Esto permite comparar cómo afecta el número de iteraciones al rendimiento del modelo.

En cuanto a los resultados, hay una mejora incremental en los scores BLEU con el aumento de las iteraciones MERT:

* Con 10 iteraciones: El BLEU fue de 91.64. Este ya es un avance respecto al ejercicio 1, mostrando que incluso un número limitado de iteraciones de MERT puede mejorar la calidad de la traducción.
* Con 20 iteraciones: El BLEU aumentó ligeramente a 91.96. Esto sugiere que un número mayor de iteraciones permite afinar aún más los pesos del modelo para obtener traducciones de mayor calidad.
* Con 50 iteraciones: El BLEU alcanzó 91.99, que es el más alto de los tres, pero solo muestra una mejora marginal sobre 20 iteraciones.

Las diferencias en los resultados reflejan que, mientras que el número de iteraciones de MERT aumenta, hay una tendencia a la mejora en la calidad de las traducciones. Sin embargo, la mejora es menos notable entre 20 y 50 iteraciones, lo que puede indicar un punto de disminución de rendimientos donde iteraciones adicionales no resultan en mejoras significativas de BLEU.

Esto destaca la importancia de encontrar un equilibrio entre el tiempo de cómputo y los beneficios obtenidos en la precisión de la traducción. En este caso, incrementar el número de iteraciones más allá de 20 produce mejoras marginales, lo que sugiere que podría haber un punto óptimo de iteraciones MERT para este conjunto de datos particular.

### Variación en los valores de n-gramas

En el tercer ejercicio se ha llevado a cabo una modificación del modelo base de MOSES alterando los valores de n-gramas utilizados en la creación de los modelos de lenguaje. A diferencia de los anteriores ejercicios que mantenían fijo este parámetro, aquí se ha experimentado con modelos de bigrama y pentagrama, representados en los archivos *turist2.lm* y *turist5.lm* creados respectivamente. El script diseñado para esta parte del estudio replicó la estructura básica del ejercicio 1, pero introdujo la creación estos dos modelos de lenguaje distintos para observar el impacto que tiene el tamaño de n-grama en la calidad final de la traducción.

La preparación inicial de los datos y la limpieza del corpus siguieron siendo consistentes con los ejercicios anteriores. Sin embargo, en la etapa de n-grama, se generaron dos modelos de lenguaje separados, uno usando bigramas y otro de pentagramas, para evaluar cómo la complejidad del modelo de lenguaje influye en la traducción. Tras la construcción de los modelos de lenguaje, se procedió con la formación de los modelos de traducción y la optimización de pesos con MERT para cada uno de estos, creando así dos modelos de traducción diferenciados basados en la longitud de los n-gramas.

Los resultados obtenidos de esta fase han sido reveladores. El modelo que utilizó bigramas alcanzó un BLEU de 91.25, lo que denota un alto grado de precisión, pero con una ligera disminución comparado con el modelo base. Por otro lado, el modelo de pentagramas logró un BLEU de 92.57, superando al modelo base y mostrando que un modelo de lenguaje con una mayor complejidad puede capturar mejor las secuencias de palabras del idioma objetivo y, por ende, mejorar la calidad de la traducción.

La penalización por brevedad y la relación de hipótesis a referencia fueron consistentes en ambos modelos, indicando que la longitud de las traducciones generadas se mantuvo en un rango adecuado. Estos resultados sugieren que aumentar el número de n-gramas hasta un cierto punto puede beneficiar la precisión de las traducciones automáticas, siempre y cuando el sistema tenga suficientes datos para aprender de manera efectiva las secuencias de palabras más largas.

### Otras técnicas de suavizado

El ejercicio 5 abordó la experimentación con diferentes técnicas de suavizado en la construcción de modelos de lenguaje dentro del marco de MOSES. Las técnicas de suavizado son cruciales en el modelado de lenguaje estadístico, ya que ayudan a manejar la ocurrencia de n-gramas no vistos en el corpus de entrenamiento, asignando una probabilidad no nula a estos eventos raros o desconocidos. En este ejercicio, se compararon dos técnicas: el suavizado Good-Turing, especificado por los parámetros *-gt3min 1 -gt2min 1 -gt1min 1*, y el suavizado de Witten-Bell, indicado por el parámetro -*wbdiscount*.

El proceso de creación del modelo siguió los mismos pasos de preparación y limpieza de datos que los anteriores ejercicios. La distinción principal reside en la generación de los modelos de lenguaje con las opciones específicas de suavizado para cada técnica. Una vez generados, se procedió al entrenamiento de los modelos de traducción y a su respectiva optimización con MERT. Los modelos resultantes se utilizaron para traducir el conjunto de prueba y las salidas fueron evaluadas para obtener su correspondiente BLEU.

Los resultados de la evaluación mostraron que el modelo que utilizó el suavizado Good-Turing alcanzó un BLEU de 91.63, mientras que el modelo con suavizado de Witten-Bell obtuvo un BLEU de 91.17. Ambos modelos presentaron un rendimiento cercano al del modelo base, pero con sutiles diferencias atribuibles a las técnicas de suavizado aplicadas. La ligera ventaja en el score BLEU del modelo con suavizado Good-Turing puede sugerir que esta técnica de suavizado se alinea un poco mejor con la distribución del corpus utilizado para estos experimentos.

Estos resultados muestran la importancia de la selección de técnicas de suavizado en la construcción de modelos de lenguaje estadísticos. Aunque las diferencias en los scores BLEU no son sustancialmente grandes, subrayan cómo variaciones aparentemente menores en el procesamiento de los datos pueden influir en la calidad de la traducción automática. La penalización por brevedad y la relación de longitud entre la hipótesis de traducción y la referencia mantuvieron valores óptimos, demostrando que la longitud de las traducciones generadas no fue afectada negativamente por los métodos de suavizado aplicados.

### Moses monótono

El ejercicio 6 se centró en evaluar el rendimiento de un modelo de traducción automática estadística de MOSES configurado para realizar traducciones de manera monótona, es decir, sin permitir la reordenación de frases en la salida. Esta restricción se impone con el fin de analizar cómo la ausencia de reordenamiento influye en la calidad de la traducción en un modelo que, por defecto, considera la reordenación como un componente esencial para capturar la estructura sintáctica de las lenguas objetivo.

Para este fin, se siguió un proceso similar al de los ejercicios anteriores en cuanto a la preparación y limpieza de los datos, así como la creación de un modelo de lenguaje trigramático. La principal variación en la metodología vino dada en la etapa de traducción, donde se aplicó la opción *-distortion-limit 0* para forzar al sistema a traducir las frases de manera secuencial sin reordenamientos.

La evaluación de la salida traducida reflejó un score BLEU de 90.98, ligeramente inferior al obtenido en los ejercicios previos con modelos que permitían la reordenación. Este resultado era de esperar, ya que el español y el inglés difieren en su orden de palabras típico y la capacidad de reordenar frases permite a los sistemas de traducción adaptarse mejor a estas diferencias. La penalización por brevedad y la proporción de las longitudes se mantuvieron consistentes, lo que indica que la calidad de la traducción no se vio comprometida en términos de longitud de salida.

El ligero descenso en el BLEU sugiere que la reordenación juega un papel importante en la traducción entre pares de lenguas con diferencias sintácticas significativas. Sin embargo, el hecho de que el decrecimiento no sea drástico apunta a que, al menos para este conjunto de datos, la reordenación no es el único factor crítico para obtener traducciones de alta calidad.

## OpenNMT

### Variación del tamaño de representación de palabras

El ejercicio 1 de OpenNMT se basa en la variación del tamaño de representación de palabras, conocidas como *word embeddings*, en el modelo de traducción basado en redes neuronales Transformer. La hipótesis detrás del experimento es que el tamaño de estos vectores puede influir significativamente en la capacidad del modelo para capturar las sutilezas del lenguaje y, por lo tanto, en la calidad de la traducción resultante.

Para llevar a cabo el experimento, se configuraron tres diferentes tamaños de representaciones vectoriales: 32, 128 y 256 dimensiones. Cada configuración se especificó en su propio archivo YAML, que OpenNMT-py utiliza para entender los parámetros con los que debe ejecutar el entrenamiento. La homogeneidad en los demás parámetros garantizó que las variaciones en los resultados fueran atribuibles únicamente al cambio en el tamaño de los *embeddings*.

El proceso de experimentación comenzó con la creación de directorios para almacenar los modelos y resultados. Posteriormente, se ejecutaron scripts de entrenamiento para cada configuración de tamaño de *embedding*, generando modelos correspondientes. Después del entrenamiento, se tradujo un conjunto de pruebas utilizando cada uno de los modelos y se evaluó la calidad de las traducciones generadas con el script de evaluación.

Los resultados reflejaron una mejora incremental en el BLEU a medida que se aumentaba el tamaño del *embedding* hasta 128, pero con una ligera disminución al pasar a 256. La configuración de 128 dimensiones alcanzó un BLEU de 96.6, que fue el más alto entre los tres y sugiere un equilibrio óptimo entre la cantidad de información semántica que puede ser capturada y la eficiencia computacional. Por otro lado, el modelo con *embeddings* de 32 dimensiones obtuvo un BLEU de 95.9, y el de 256 dimensiones un BLEU de 95.6, indicando que un tamaño demasiado pequeño o grande puede no ser tan efectivo para este conjunto de datos específico.

Este patrón de resultados podría indicar que, aunque los *embeddings* más grandes tienen la capacidad de capturar más información, también pueden requerir conjuntos de datos más grandes para entrenarse de manera efectiva, o podrían introducir ruido que dificulte la generalización del modelo. Por el contrario, *embeddings* demasiado pequeños podrían no capturar suficientes matices del lenguaje, limitando la calidad de la traducción.

### Variación del número de capas

En el ejercicio 2 la atención se centró en la variación del número de capas en la arquitectura del modelo Transformer. Se entiende que las capas adicionales pueden permitir que la red capture relaciones más complejas en los datos, pero también pueden aumentar el riesgo de sobreajuste y requerir más recursos computacionales. Se crearon dos configuraciones distintas para el entrenamiento: una con 4 capas y otra con 8 capas en el *encoder* y el *decoder* del modelo Transformer.

El ajuste de configuración entre los dos experimentos se reflejó en los archivos *config4.yaml* y *config8.yaml*, que especificaron el número de capas como el parámetro variable entre los dos entrenamientos, manteniendo constantes todos los demás parámetros del modelo, como el tamaño del vector de palabras, la cantidad de cabezas de atención y los pasos de calentamiento, entre otros. Esto aseguró que cualquier diferencia en el rendimiento pudiera atribuirse directamente al número de capas del modelo.

Los resultados revelaron que el modelo con 4 capas superó al modelo con 8 capas, alcanzando un BLEU de 96.1 frente a un BLEU de 94.0. Esto sugiere que añadir capas adicionales no siempre conduce a una mejora en la calidad de la traducción y puede tener un efecto contraproducente, posiblemente debido a la complejidad añadida que no se justifica con la cantidad de datos de entrenamiento o puede ser debido a la dificultad para entrenar redes más profundas de manera efectiva.

El hecho de que el modelo de 4 capas haya logrado un mejor rendimiento podría indicar que la representación de los datos en este nivel de profundidad es suficiente para capturar las dependencias lingüísticas necesarias para la tarea de traducción con el corpus proporcionado. En cambio, el modelo de 8 capas, a pesar de tener potencialmente una mayor capacidad de modelado, no mejoró los resultados y de hecho disminuyó la calidad de la traducción según lo medido por BLEU.

Este fenómeno refleja la importancia de la selección adecuada de la arquitectura en los modelos de traducción automática y reafirma la noción de que más capas no siempre equivalen a mejores resultados.

# Comparativa de resultados

En la Ilustración 1, referente a los experimentos con MOSES, se observa que los ajustes en el modelo y las técnicas aplicadas influyen en la métrica BLEU. El modelo sin ajustes de pesos presenta un BLEU de 88.42, que sirve como base para comparar los incrementos que se generan al modificar el modelo.

Ilustración 1 - Resultados MOSES

La optimización con MERT muestra mejoras consistentes, alcanzando el mayor BLEU de 91.99 con 50 iteraciones, aunque la diferencia entre 20 y 50 iteraciones es mínima, lo que podría indicar un punto de rendimientos decrecientes. Los experimentos que alteran la complejidad del modelo de lenguaje, específicamente con bigramas y pentagramas, muestran un efecto positivo en la calidad de la traducción, siendo los pentagramas los que obtienen un BLEU más alto de 92.57.

Por otro lado, la aplicación de técnicas de suavizado como Good-Turing y Witten-Bell ofrece resultados ligeramente inferiores al modelo de pentagramas, pero aún superiores al modelo base. Finalmente, el modelo monótono, que restringe la reordenación de frases, presenta el BLEU más bajo de todos los experimentos con 90.98, sugiriendo que la capacidad de reordenar frases es un factor importante para la calidad de la traducción en MOSES.

Por otro lado, en la Ilustración 2 correspondiente a OpenNMT, la variación en los tamaños de los word embeddings muestra que un tamaño medio de 128 proporciona el mejor resultado con un BLEU de 96.6, mientras que los tamaños más pequeños y más grandes ofrecen resultados ligeramente inferiores. Esto puede indicar que existe un tamaño óptimo de embeddings que equilibra la cantidad de información semántica capturada y la generalización del modelo.

Ilustración 2 - Resultados OpenNMT

En cuanto a la cantidad de capas, se aprecia que un modelo más profundo no necesariamente resulta en una mejor traducción; de hecho, el modelo de 4 capas supera al de 8 capas con un BLEU de 96.1 frente a 94.0, lo que podría reflejar un sobreajuste o dificultades en el entrenamiento de modelos más complejos con la cantidad de datos disponibles.

Comparando ambas tecnologías, OpenNMT tiende a producir resultados de BLEU más altos en comparación con MOSES. Esto podría deberse a que los modelos basados en redes neuronales, como los que utiliza OpenNMT, son capaces de capturar relaciones contextuales más complejas y ofrecen un enfoque más moderno y potente para la traducción automática. Sin embargo, es importante destacar que la calidad de la traducción no solo depende de la tecnología utilizada, sino también de la configuración y optimización específica del modelo, así como del tipo y tamaño del corpus de entrenamiento.

En resumen, la elección entre MOSES y OpenNMT en un proyecto dependerá de los recursos disponibles, los objetivos específicos de la traducción y el contexto de uso. Mientras que OpenNMT podría ser preferible para obtener una calidad de traducción más alta en ciertos escenarios, MOSES sigue siendo una opción valiosa, especialmente cuando se consideran factores como la reordenación de frases o el uso de técnicas de suavizado específicas.

# Conclusiones

MOSES, al ser un sistema de traducción basado en reglas y estadísticas, demostró ser altamente configurable y permitió la exploración de diferentes técnicas y parámetros que afectan directamente la calidad de la traducción. A través de los experimentos realizados, se observó que ajustes como la optimización de pesos con MERT y la variación en el tamaño de los n-gramas pueden mejorar significativamente el BLEU. Sin embargo, también quedó claro que hay un punto a partir del cual aumentos adicionales en la complejidad del modelo o en el número de iteraciones de MERT no se traducen en mejoras significativas de calidad, lo que resalta la importancia de un balance entre la complejidad del modelo y la eficiencia computacional. Además, se demostró que la capacidad de reordenar frases es crítica para la calidad de las traducciones, especialmente en pares de idiomas con diferencias sintácticas significativas.

Por otro lado, OpenNMT, que utiliza una arquitectura de red neuronal, presentó resultados superiores en términos de BLEU, lo que sugiere una mejor capacidad para capturar y traducir el contexto y las sutilezas del lenguaje. La variación en el tamaño de los word embeddings y el número de capas de la red demostraron que existe un equilibrio óptimo que maximiza la calidad de la traducción. Los modelos de red neuronal demostraron ser menos dependientes de la ingeniería de características y más capaces de generalizar a partir de los datos de entrenamiento. No obstante, también se evidenció que modelos excesivamente complejos pueden sufrir de sobreajuste y que el entrenamiento de redes más profundas puede no siempre resultar en mejoras de calidad, destacando la necesidad de una selección cuidadosa de la arquitectura del modelo.

En términos de ventajas y desventajas, MOSES ofrece una gran flexibilidad y la posibilidad de ajuste fino, lo que puede ser muy valioso en situaciones donde se dispone de conocimiento lingüístico experto y tiempo para la experimentación. Sin embargo, puede requerir más esfuerzo para alcanzar la calidad que los sistemas basados en redes neuronales pueden proporcionar de manera más directa. OpenNMT, por su parte, se beneficia de los avances en aprendizaje profundo y puede alcanzar un rendimiento superior con menos configuración, pero requiere de recursos computacionales más significativos y puede ser menos transparente en términos de cómo las decisiones de traducción son tomadas por el modelo.

Este trabajo ha demostrado que la elección entre estas dos tecnologías de traducción automática debe ser informada por las necesidades específicas del proyecto, la naturaleza del par de idiomas involucrados, la disponibilidad de corpus de entrenamiento de alta calidad, y los recursos computacionales a mano. Cada tecnología tiene su lugar en el panorama de la traducción automática, y la comprensión de sus particularidades es esencial para cualquier investigador o profesional que busque aplicar estas herramientas en el mundo real.