TRABAJO DE FIN DE MÁSTER

2021-2022

MEJORA LÉXICA DEL MOTOR DE TRADUCCIÓN CATALÁN-GALLEGO DE APERTIUM

MÁSTER EN TRADUMÁTICA: TECNOLOGÍAS DE LA TRADUCCIÓN

FACULTAD DE TRADUCCIÓN E INTERPRETACIÓN

AUTORES

Enrique Orjales Otero
Patricia Toboso Sala

TUTOR/A

MARC RIERA IRIGOYEN

Barcelona, 1 de junio de 2022

Datos del TFG / Datos del TFG / *Dissertation data*

Título: Mejora léxica del motor de traducción catalán-gallego de Apertium

Títol: Millora lèxica del motor de traducció català-gallec d'Apertium

Title: Lexical Improvement of the Catalan-Galician Translation Engine of Apertium

Autor/a: Enrique Orjales Otero y Patricia Toboso Sala

Autor/a: Enrique Orjales i Patricia Toboso Sala

Author: Enrique Orjales Otero and Patricia Toboso Sala

Tutor: Marc Riera Irigoyen

Tutor: Marc Riera Irigoyen

Tutor: Marc Riera Irigoyen

Centro: Universidad Autónoma de Barcelona

Centre: Universitat Autònoma de Barcelona

Centre: Autonomous University of Barcelona (UAB)

Estudios: Máster oficial en Tradumática: Tecnologías de la Traducción

Estudis: Màster oficial en Tradumàtica: Tecnologies de la Traducció

Studies: Official master's degree in Tradumatics: Translation Technologies

Palabras clave / Paraules clau / *Keywords*

Apertium, traducción automática basada en reglas, lenguas minorizadas, motor de traducción catalán-gallego, mejora léxica.

Apertium, traducció automàtica basada en regles, llengües minoritzades, motor de traducció català-gallec, millora lèxica.

Apertium, rule-based machine translation, minoritised languages, Catalan-Galician translation engine, lexical improvement.

Resumen del TFM / Resum del TFM / *Abstract*

Este trabajo de fin de máster tiene como objetivo retomar y mejorar el motor de traducción automática basada en reglas de Apertium entre el par de lenguas minorizadas catalán-gallego. Concretamente, pretende mejorar la calidad lingüística del motor aumentando la cobertura léxica y disminuyendo los valores de las métricas WER y PER. Es por este motivo que la parte central del trabajo se basa en tres fases: la optimización del motor, las evaluaciones de este en su estado previo y posterior a nuestras modificaciones y la comparación junto con el análisis de los resultados obtenidos de las evaluaciones del motor en su estado inicial y final. Asimismo, otra fase crucial consiste en la explicación al detalle de la preparación de los recursos que se han necesitado para llevar a cabo este trabajo. En particular, estos abarcan la creación de un texto de referencia, la instalación de Apertium, la obtención de coberturas léxicas, la extracción de las métricas WER y PER y, por último, la ejecución de pruebas de vocabulario.

Aquest treball de fi de màster té com a objectiu reprendre i millorar el motor de traducció automàtica basada en regles d'Apertium entre el parell de llengües minoritzades català-gallec. Concretament, pretén millora la qualitat lingüística del motor augmentant la cobertura lèxica i disminuint els valors de les mètriques WER i PER. És per aquest motiu que la part central del treball es basa en tres fases: l'optimització del motor, les avaluacions d'aquest en el seu estat previ i posterior a les nostres modificacions i la comparació juntament amb l'anàlisi dels resultats obtinguts de les avaluacions del motor en el seu estat inicial i final. Així mateix, una altra fase crucial consisteix en l'explicació al detall de la preparació dels recursos que s'han necessitat per dur a terme aquest treball. En particular, aquests comprenen la creació d'un text de referència, la instal·lació d'Apertium, l'obtenció de cobertures lèxiques, l'extracció de les mètriques WER i PER i, per últim, l'execució de proves de vocabulari.

The main purpose of this master's dissertation is to resume and improve the rule-based machine translation engine of Apertium between the minoritised languages of Catalan and Galician. Specifically, it aims at improving the linguistic quality of this engine by increasing its lexical coverage as well as lowering the WER and PER metric scores. This is why the core of this paper is based on three stages: on the one hand, the optimisation of the engine; on the other hand, its evaluation before and after our modifications; and, finally, the comparison and analysis of the obtained results from the evaluations of the engine in its initial and current state. Furthermore, another crucial phase consists of the detailed explanation of the preparation of the resources needed to perform this dissertation. In particular, these resources encompass the creation of a reference text, the installation of Apertium, the extraction of both lexical coverages and WER and PER metrics, and, last but not least, the execution of vocabulary tests.

ÍNDICE

ÍNDICE DE TABLAS Y FIGURAS

I. Introducción

Hoy en día, la demanda de traducción automática (en adelante TA) es cada vez mayor debido, en gran parte, a la globalización (Armentano-Oller et al., 2007). Este tipo de traducción ha conseguido tener un papel clave en la actual era digital, puesto que uno de sus principales objetivos es ayudar a los traductores humanos a mecanizar el proceso de traducción (Sánchez Ramos y Rico Pérez, 2020; Sin-wai, 2015). De hecho, la TA «ocupa un lugar destacado en el mundo de la traducción profesional, ya que está relacionada con el tan ansiado concepto de productividad» (Sánchez Ramos y Rico Pérez, 2020, p. 1).

Armentano-Oller et al. (2007) añaden que, con el tiempo, ha aparecido un interés por desarrollar sistemas de TA que involucren lenguas minorizadas, declaración que nos lleva a nuestra propuesta de TFM: mejorar el motor de TA basada en reglas de Apertium, una plataforma de código abierto, entre el par de idiomas minorizados catalán-gallego. Nuestra intención es retomar este motor, que está aún en fase de desarrollo, y contribuir a su optimización en la medida de lo posible para su futura publicación.

1. Contextualización del trabajo

Este trabajo pretende aportar un marco teórico explicando el concepto de la TA, tanto su definición, como la evolución que ha experimentado desde sus comienzos hasta la actualidad. También se abordarán las diferentes métricas para evaluar motores de traducción y los diferentes tipos de TA que existen, aunque el objeto de estudio de este TFM es la TA basada en reglas. A mayores, también se mencionará el motor de Apertium, que se usará en el marco práctico, y el favor que hace a las lenguas minorizadas como el gallego y el catalán.

Por lo que respecta a la parte práctica del trabajo, esta se centra en cinco fases principales:

- la preparación de los recursos necesarios para llevar a cabo la parte práctica con éxito;
- el análisis del estado del motor de Apertium sin haber sido modificado;
- la optimización del motor mediante la selección de las palabras que se deberían incorporar en el diccionario bilingüe, con la intención de mejorar la cobertura léxica y la tasa de error del traductor;
- la evaluación del estado actualizado del motor después de haber implementado todos los cambios;
- la comparación de los resultados obtenidos tanto del estado inicial como final del motor.

2. Motivación, justificación y objetivos del trabajo

Nuestra intención de hacer este TFM en conjunto se debe al poco protagonismo y visibilidad que sufren el catalán y el gallego como lenguas minorizadas en el panorama lingüístico español. Por este motivo, pretendemos con nuestra propuesta establecer un puente entre el catalán y el gallego,

mejorando la calidad lingüística del motor actual de Apertium entre estas lenguas que se abandonó, aún en fase de desarrollo, a finales de 2015.

Por lo tanto, el objetivo de este trabajo es retomar el motor de traducción en cuestión, investigar en qué estado se encuentra y averiguar hasta qué punto podemos mejorarlo, marcando unos objetivos realistas.

Para mejorar la calidad del motor, tenemos pensado aumentar la cobertura léxica e intentar bajar los valores de las métricas WER y PER. En otras palabras, tenemos como objetivo mejorar la cobertura léxica y el porcentaje de error de las métricas, de manera que recuperaremos el trabajo ya hecho en este traductor y añadiremos palabras nuevas, ampliando así el diccionario bilingüe para que las reconozca. Asimismo, revisaremos y corregiremos algunas de las entradas ya existentes en el traductor para garantizar que se han introducido correctamente.

II. Marco teórico

El marco teórico de este trabajo consiste en cinco apartados principales. En el primer apartado se define la TA, un concepto clave en este trabajo. A continuación, en el segundo, se expone brevemente la historia de la TA. Seguidamente se presentan los diferentes tipos de TA y métricas que existen, es decir, los sistemas basados tanto en reglas como en corpus y los métodos de evaluación como WER, PER y BLEU, respectivamente. Finalmente, el último apartado está dedicado a la plataforma de Apertium donde aparece información no solo del motor de traducción de este sistema de TA basada en reglas, sino también de su relación con las lenguas minorizadas.

1. El concepto de la TA

Según Maučec y Donaj (2019) y Qun y Xiaojun (2015), la TA es un subcampo de la lingüística computacional (con influencia de la lingüística, la informática, la teoría de la informática, la inteligencia artificial y la estadística) que estudia los enfoques y el uso de programas informáticos para traducir texto o habla de una lengua natural a otra. Además, afirman que la esencia de la TA es la automatización, así como la aceleración, del proceso de traducción en su totalidad.

Asimismo, Armentano-Oller et al. (2007) declaran que «la traducción automática consiste en la obtención de un texto *equivalente* (esto es, que preserve el contenido) en una *lengua destino* a partir de un texto en una *lengua origen*» (p. 3). En su artículo, también hacen referencia a tres distintas aplicaciones o usos de la TA:

- el primer uso de la TA es la **asimilación** que hace referencia a una traducción cuya calidad es irrelevante y que es utilizada para obtener y transmitir una idea general sobre el contenido del texto original;
- el segundo uso se conoce como **comunicación**, que es bastante parecido a la asimilación. Esta aplicación de la TA se lleva a cabo en el momento en que individuos que hablan diferentes lenguas utilizan la TA para conversar;
- para acabar, la **diseminación** es el último uso que se presenta e implica una traducción de calidad, normalmente obtenida después del proceso de posedición, ya que esta va a ser divulgada públicamente.

Sin embargo, aún hoy en día, a pesar de haberse demostrado la eficacia y utilidad de la TA, su percepción pública se sigue viendo afectada negativamente a veces y esto se debe a la falta de conocimiento y apreciación que algunos tienen hacia la TA. Desafortunadamente, existen personas que no son capaces de darse cuenta y valorar lo mucho que la TA, sin ser perfecta, ha conseguido y cómo esta puede contribuir y facilitar el día a día de mucha gente en una infinidad de situaciones (Hutchins y Somers, 1992).

2. Historia de la TA

Los inicios de la TA se remontan a 1933, cuando empezaron en Francia y Rusia los primeros dos intentos de automatizar la traducción. Además, posteriormente, durante la Segunda Guerra

Mundial, «se empezaron a utilizar técnicas numéricas como mecanismos para conseguir la TA y poder así descifrar los distintos mensajes que se intercambiaban los servicios de inteligencia de los bandos en conflicto» (Sánchez Ramos y Rico Pérez, 2020).

Más tarde, se oficializó un intento de formalizar la TA en el 1949, año en el que Warren Weaver escribió un memorándum donde sugirió unas líneas de investigación para esta técnica de traducción. Asimismo, propuso que la manera de conseguir traducir automáticamente era mediante el uso de las técnicas criptográficas de la guerra, el análisis estadístico, la aplicación de la teoría de la información de Shannon y la exploración de la lógica subyacente y de las características universales del lenguaje (Hutchins y Somers, 1992). Poco después, empezaron las investigaciones en universidades como la de Georgetown que, colaborando con la multinacional de tecnología IBM, hizo la primera demostración pública del funcionamiento y la viabilidad de la TA, aunque con limitaciones gramaticales y léxicas (Hutchins, 2015). No obstante, estas restricciones no impidieron que la investigación de la TA fuera financiada por EE. UU. e influenciara a otros a hacer lo mismo como, por ejemplo, la Unión Soviética (Hutchins, 2001).

Desde los años cincuenta hasta los sesenta, se continuó la investigación de la TA de una manera más intensiva. De hecho, esta década se considera una de las más importantes del siglo XX por lo que respecta a la TA (Sánchez Ramos y Rico Pérez, 2020). Esta época vio el nacimiento de los tres enfoques básicos de la TA: el modelo de traducción directa, el modelo de interlingua y el modelo de transferencia. La TA ayudó no solo al desarrollo tanto de diccionarios automatizados como las técnicas de análisis sintáctico, sino también a la teoría lingüística (Hutchins y Somers, 1992). Sin embargo, sobre los años sesenta, los grupos de investigación no lograron su objetivo principal de crear un sistema capaz de hacer traducciones parecidas, si no iguales, a las que haría una persona. Es por ello que el comité ALPAC (*Automatic Language Processing Advisory Committee*) decidió hacer un informe en el que concluyó que la TA era menos eficiente que un traductor humano y que, por lo tanto, la TA debería únicamente investigarse con el propósito de ayudar a los traductores, creando con esta tecnología herramientas de apoyo (Hutchins, 2015).

En la etapa de los setenta, pese a la desilusión de no lograr lo pretendido en la década anterior, se podría decir que la TA renació, ya que algunos grupos siguieron investigando (Sánchez Ramos y Rico Pérez, 2020). Se centraron en los modelos de interlingua y transferencia en vez del modelo directo, aunque acabaron dándose cuenta de que realmente el modelo de interlingua no estaba a la altura del de transferencia (Hutchins, 2001). A los ojos de la comunidad de investigación de la TA, el sistema de transferencia era el modelo predilecto con el que iban a lograr avances significativos (Hutchins y Somers, 1992).

No obstante, durante la etapa de los ochenta hasta los noventa, los nuevos enfoques del modelo de interlingua se unieron al modelo de transferencia. La idea de los investigadores era que se debía ahondar en los procesos del lenguaje natural dentro de la IA (inteligencia artificial) para mejorar la calidad de la TA (Hutchins, 2015).

La década de los noventa también supuso la entrada de la TA al mercado. Entre otros factores, la productividad de los traductores humanos aumentó debido a la aparición de la *World Wide Web* y a la mejor capacidad de procesamiento de los ordenadores (Sánchez Ramos y Rico Pérez, 2020). Los traductores buscaban apoyo para poder traducir de manera más eficiente, siempre y cuando fueran ellos los que tuvieran el control a lo largo del proceso de traducción. Por ello, se crearon diferentes herramientas, pero las más notorias llegaron cuando se integraron en la mesa de trabajo

del traductor por parte de proveedores como, por ejemplo, Trados y STAR (Hutchins, 2001; Hutchins, 2015).

Una vez entrado el siglo XXI, la TA estadística alcanzó su máximo potencial. Hasta el momento, la TA no había cobrado especial importancia, puesto que solo se consideraba un complemento para aumentar la productividad del traductor. Esto cambió cuando, en 2006 y 2007, Google y Moses pusieron a disposición de los usuarios herramientas de TA, promoviendo así su consumo gratuito en línea (Sánchez Ramos y Rico Pérez, 2020).

La industria de la TA ha llegado a ser tan popular que su valor alcanzó los 250 millones de dólares en el año 2015. Además, las empresas, sobre todo de ámbito tecnológico como Google y Bing, utilizan la TA para mejorar e introducir al mercado nuevos productos (Way, 2018). Hasta la actualidad, la TA no ha hecho más que avanzar a pasos agigantados y, poco a poco, se está empezando a conseguir una TA de gran calidad, conocida como TA neuronal, que se asemeja cada vez más a la de un traductor profesional (Sánchez Ramos y Rico Pérez, 2020).

3. Tipos de TA

A lo largo del tiempo, se han definido y desarrollado diferentes enfoques para la TA; empezando por la TA basada en reglas, pasando por la TA basada en ejemplos y la TA estadística, hasta llegar a la TA neuronal (Maučec y Donaj, 2019). De hecho, «actualmente podría decirse que conviven principalmente tres tipos de traducción automática (TA) en el mercado: la que utiliza información lingüística (comúnmente denominada como TA basada en reglas …), la estadística … y la basada en redes neuronales», a pesar de que, «si bien [la TA basada en reglas] sigue teniendo cabida en la investigación, es quizás el [paradigma] que menos atención acapara, aunque no por ello se deba obviar» (Parra Escartín, 2018, p. 20-21). Otra manera de clasificar los diferentes tipos de TA es la distinción entre los dos grandes grupos de tecnologías de la traducción: por un lado, los sistemas de TA basados en reglas (apropiados sobre todo cuando se trata de traducir entre lenguas similares) y, por el otro, los sistemas de TA basados en corpus (que incluyen tanto la TA basada en ejemplos, como la estadística y la neuronal) (Forcada, 2009; Maučec y Donaj, 2019).

3.1 Sistemas de TA basados en reglas

De acuerdo con Armentano-Oller et al. (2007), en el primer gran grupo, es decir, en los sistemas basados en reglas:

> Los datos lingüísticos (básicamente diccionarios monolingües, diccionarios bilingües y reglas de transferencia, que recogen las transformaciones estructurales necesarias para pasar de una lengua a otra) se recopilan manualmente y se codifican adecuadamente para que puedan ser usados por el motor de traducción (p. 5).

Además, argumentan que, dentro de los sistemas basados en reglas, los de TA por **transferencia** son los más habituales, que se componen de tres fases:

- **análisis**, fase monolingüe que produce una representación intermedia de una frase en la lengua origen, dependiente de esta;

- **transferencia**, fase bilingüe que convierte la representación intermedia en otra que es dependiente de la lengua destino, en vez de la lengua origen;
- **generación**, fase monolingüe que produce, a partir de la representación intermedia anterior, la frase en lengua destino.

Los otros sistemas basados en reglas son dos: los **directos**, también conocidos como sistemas basados en diccionarios en que un conjunto de palabras en la lengua origen se traduce directamente a una frase en la lengua destino, y los de **interlingua** (Qun y Xiaojun, 2015). Este último sistema es muy parecido al de transferencia, pero, en vez de crear representaciones intermedias dependientes a la lengua origen y destino, la representación intermedia abstracta que se produce es neutral y universal para todas las lenguas de origen y destino (Maučec y Donaj, 2019).

3.2 Sistemas de TA basados en corpus

Por lo que respecta a los sistemas basados en corpus (el segundo gran grupo de tecnologías de traducción), estos aprenden a traducir automáticamente a partir de enormes cantidades de corpus bilingües donde el texto en una lengua ha sido alineado con su traducción en la otra (Forcada, 2009). En este gran grupo se encuentran:

- la **TA basada en ejemplos**, que consiste en utilizar traducciones existentes como ejemplos para facilitar la traducción de nuevas frases (Wong Tak-ming y Webster, 2015);
- la **TA estadística**, que, como argumentan Yang y Min (2015), es un paradigma basado en un modelo probabilístico para generar traducciones y acaba creando «todas las traducciones posibles y propone la más probable» (Parra Escartín, 2018, p. 23);
- la **TA neuronal**, que hace uso de una gran red de neuronas artificiales, basada en las representaciones vectoriales de las palabras, para aprender y llevar a cabo la TA (Maučec y Donaj, 2019; Parra Escartín, 2018).

Por último, en este apartado también debería mencionarse la existencia de **sistemas híbridos** que intentan beneficiarse tanto de los sistemas basados en reglas como de los basados en corpus. Existen dos grupos de TA híbrida que se guían, o bien por la TA basada en reglas, o bien por la TA estadística e incorporan el otro tipo de TA en alguna fase del proceso de traducción para intentar obtener un mejor resultado final (Maučec y Donaj, 2019).

4. Métricas

Conforme ha pasado el tiempo, el uso de la TA ha aumentado drásticamente en una gran variedad de campos, por lo tanto, también ha sido preciso que su calidad fuera a la par. Para ello, se necesita un método de evaluación para poder juzgar el resultado de una TA. Existen dos métodos que difieren en quién o qué evalúa: la **evaluación manual** y la **automática** (Maučec y Donaj, 2019).

Por un lado, según Sánchez Ramos y Rico Pérez (2020), la evaluación manual es una corrección de textos realizada por profesionales de la traducción. Es un método costoso, lento y se caracteriza por su alta subjetividad a la hora de evaluar el resultado de la TA, puesto que cada corrector tiene

su propia opinión respecto a la calidad que pueda tener una traducción. Asimismo, exponen que la evaluación automática consiste en el empleo de unas métricas que comparan las traducciones hechas por el motor de TA con una traducción de referencia, efectuada por un traductor profesional. De hecho, que la evaluación automática se base y dependa de traducciones humanas es sin duda una de sus limitaciones más evidentes (Elliott et al., 2004).

Además, se debe tener presente que las métricas de la evaluación automática nos impiden obtener un análisis detallado sobre la naturaleza de los errores de traducción, a diferencia de la evaluación manual (Popovic et al., 2006). No obstante, pese a que esta información pormenorizada puede resultar útil en ciertas ocasiones y, por lo tanto, la evaluación manual destacaría en estos casos; en otros, podría primar la rapidez y productividad al menor coste y, por consecuente, la evaluación automática sería más conveniente.

Tres de las métricas que se suelen utilizar para evaluar las TA de manera automática son:

- **WER** (*Word Error Rate*): método que evalúa la TA dependiendo del número de cambios que han tenido que llevarse a cabo, cuantos menos mejor, con el fin de que se parezca lo máximo posible a la traducción de referencia (Maučec y Donaj, 2019). Una desventaja de esta métrica es que penaliza la reordenación de palabras.
- **PER** (*Position-Independent Error Rate*): métrica basada en WER. Sin embargo, por lo que respecta a este método, la posición de las palabras es irrelevante con tal de que estas estén bien traducidas. La desventaja es que la traducción, pese a que según PER esté correcta, es posible que en realidad no sea el caso debido a que el orden de las palabras puede ser incorrecto (Yakovyna y Masyukevych, 2013).
- **BLEU** (*Bilingual Evaluation Understudy*): métrica frecuentemente usada para corregir TA. Esta se rige por la proximidad que la TA pueda tener con el texto de referencia, que se calcula gracias a una comparativa entre el texto traducido y la traducción de referencia. La evaluación se calcula teniendo en cuenta el número de n-gramas comunes, el número total de palabras traducidas por la TA y la longitud del texto de referencia. La métrica BLEU también tiene algunas desventajas, puesto que la métrica puede otorgar un 0 % a una traducción válida de la TA por el mero hecho de no acercarse a la traducción de referencia. Asimismo, no es demasiado fiable por lo que respecta a la calidad semántica, con sinónimos ni en oraciones individuales, ya que fue ideada para tratar con grandes corpus (Sánchez Ramos y Rico Pérez, 2020).

5. Apertium

Apertium es un sistema de TA basado en reglas de transferencia superficial, desarrollado en la Universitat d'Alacant, que surgió en 2004 como parte de un proyecto fundado por el Ministerio de Industria, Comercio y Turismo de España (Forcada, 2015). El mismo sitio web principal de Apertium[1] define esta plataforma como una plataforma de TA libre y de código abierto, por lo que «[el traductor automático] puede ser usado, copiado, estudiado, modificado y redistribuido con la única restricción de que el código fuente ha de estar siempre disponible» (Armentano-Oller et al., 2007, p. 6).

[1] https://apertium.org/

En un principio, la plataforma Apertium iba destinada y se centraba en pares de lenguas relacionadas, pero, con el tiempo, amplió su enfoque para incluir y tratar también pares de lenguas más distantes. La plataforma Apertium proporciona tres componentes (META-NET, 2013):

- un **motor**, independiente de la lengua, que utiliza una tecnología muy sencilla para la TA;
- **datos lingüísticos**, en un formato XML especificado, para una cantidad de pares de lenguas cada vez más grande (muchas de las cuales son lenguas pequeñas);
- **herramientas** para gestionar estos datos lingüísticos y convertirlos en el formato que el motor utiliza.

5.1 El motor de traducción de Apertium

El motor de Apertium consiste en una cadena de módulos que procesan el texto transformándolo a medida que pasa por cada uno de ellos. Los diferentes módulos que forman este sistema de TA de transferencia superficial son los siguientes (Armentano-Oller et al., 2006; Forcada, 2009):

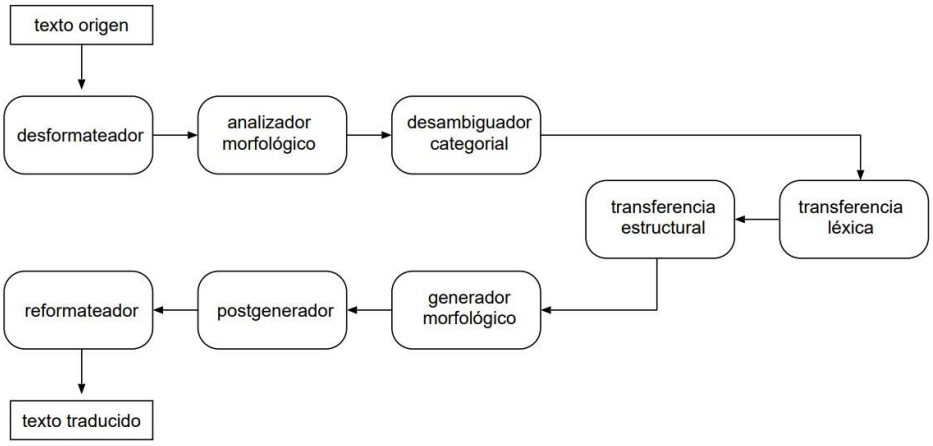

Figura 1: Cadena de módulos de un sistema de TA construido con Apertium (adaptado de Armentano-Oller et al., 2007)

1. El **desformateador** separa el texto origen del formato del documento en el que se encuentra, que es encapsulado para que el resto de los módulos no detecten esta información de formato.
2. El **analizador morfológico** segmenta el texto en las unidades léxicas que aparecen en este y ofrece las diferentes opciones o análisis posibles para cada una de ellas. Esto incluye el lema (también conocido como la forma base que aparece en los diccionarios), la categoría léxica y la flexión morfológica.
3. El **desambiguador categorial** escoge uno de los análisis posibles para aquellas unidades léxicas que tengan más de uno, teniendo en cuenta el contexto en que se encuentra la unidad léxica. En este módulo se utiliza un modelo estadístico.
4. La **transferencia léxica**, mediante el uso de un diccionario bilingüe, produce un equivalente en la lengua destino para cada unidad léxica de la lengua origen.
5. La **transferencia estructural** encuentra patrones de palabras (en otras palabras, sintagmas) y detectar las reglas que deben aplicarse a cada patrón.

6. El **generador morfológico** flexiona adecuadamente la unidad léxica en la lengua destino.
7. El **postgenerador** se encarga de las operaciones ortográficas necesarias en la lengua destino como, por ejemplo, el uso de contracciones y apostrofes.
8. El **reformateador** devuelve al texto la información de formato que el desformateador había encapsulado y ocultado al resto de la cadena de módulos. De esta manera, el texto traducido se le aplica inmediatamente el mismo formato que el del texto origen.

5.2 Apertium y las lenguas minorizadas

Desde el principio, la comunidad de Apertium estaba muy motivada por la necesidad de proporcionar una TA que fuera no solo gratuita, sino fácilmente accesible para las lenguas más pequeñas, en concreto las de España inicialmente (META-NET, 2013). De hecho, Parra Escartín (2018) argumenta que «Apertium se ha convertido en un gran catalizador para el desarrollo de motores con lenguas minoritarias» (p. 22). Esto se debe a que esta plataforma de TA basada en reglas tiene un sistema que resulta de gran utilidad, especialmente cuando se tratan pares de lenguas que no disponen de grandes cantidades de datos, ya que no necesita un corpus extenso para entrenar su motor (a diferencia de la TA estadística y la TA neuronal). Khanna et al. (2021) añaden que Apertium se ha convertido en un lugar clave por lo que respecta a la creación de recursos para las lenguas minorizadas de Europa y ha demostrado su potencial como plataforma tecnológica lingüística de apoyo para todas las lenguas del mundo.

De acuerdo con Cronin (1995), González et al. (2006) y Kuusi (2017), tanto la traducción como la TA son fundamentales para el desarrollo y la preservación de las lenguas minorizadas y esto es precisamente lo que Apertium parece ofrecer. Incluso se podría decir que este es uno de los aspectos más característicos, únicos y especiales de la plataforma de Apertium.

III. Metodología

En esta sección se presentan los diferentes pasos que se llevaron a cabo para intentar mejorar el motor aumentando la cobertura léxica, mediante la incorporación de los términos más comunes en ambos idiomas que el traductor no reconocía. Los cuatro apartados en que se ha dividido esta parte más práctica del TFM son: la preparación del entorno de trabajo, el funcionamiento del motor antes de ser actualizado, la mejora del motor de traducción y, por último, el funcionamiento de este motor una vez modificado.

1. Preparación del entorno de trabajo

Antes de empezar a modificar el motor y analizar su mejora, se tuvieron que efectuar una serie de procesos previos para que este TFM se pudiera desarrollar, como son: primero, la creación de un texto de referencia; seguidamente, la instalación de Apertium en un ordenador personal; y, finalmente, la obtención y análisis de las coberturas léxicas, las métricas WER y PER y las pruebas de vocabulario *testvoc*.

Este último proceso proporciona información clave tanto para decidir los cambios a implementar en el traductor, concretamente en el diccionario bilingüe, como para poder estudiar su progreso y mejora. Por esta razón, este paso se realizó dos veces: una antes de implementar los cambios de este TFM en el diccionario, para así evaluar el estado inicial del motor, y la otra después de haberlos implementado, para valorar el estado final de este.

1.1 Creación de un texto de referencia

A la hora de elaborar un texto de referencia, fue importante tener en consideración tres factores: este estaría compuesto de diferentes fragmentos extraídos de distintos escritos de ámbito general; además, estos escritos tenían que provenir de fuentes mínimamente fiables; y, ante todo, los escritos debían tener un equivalente en la otra lengua minorizada.

Teniendo todos estos criterios en cuenta, se crearon dos textos de referencia en formato TXT de alrededor de 5500 palabras cada uno, con los nombres *ca.txt* y *gl.txt*. Estos textos alineados se formaron a partir de diferentes entradas de la Wikipedia en catalán y gallego, que comprendían diversas temáticas, desde biografías, ciudades y alimentos hasta arquitectura, literatura e historia.

1.2 Instalación de Apertium

Por lo que respecta a la instalación de Apertium, una opción válida es utilizar el sistema operativo Windows WSL (Windows Subsystem for Linux), ya que Apertium necesita un sistema operativo que cuente con las instrucciones de Unix/Linux para la compilación de los datos lingüísticos y la ejecución de los módulos.

En el caso de que Windows WSL no estuviera ya descargado en el ordenador, se debería acceder a Windows Powershell desde el explorador de Windows e introducir el comando *wsl --install*. De

esta manera se instalaría, por defecto, la distribución de Ubuntu. También es recomendable descargarse la aplicación Windows Terminal de Microsoft Store, desde donde se puede trabajar con Ubuntu.

El primer paso para instalar Apertium es introducir el siguiente comando, obtenido de la wiki de Apertium[2].

curl -sS https://apertium.projectjj.com/apt/install-nightly.sh | sudo bash

Con esta orden, dentro de Ubuntu se han configurado las direcciones web desde donde se pueden bajar los programas que forman Apertium y, así, Ubuntu puede encontrarlos automáticamente e instalarlos con más facilidad.

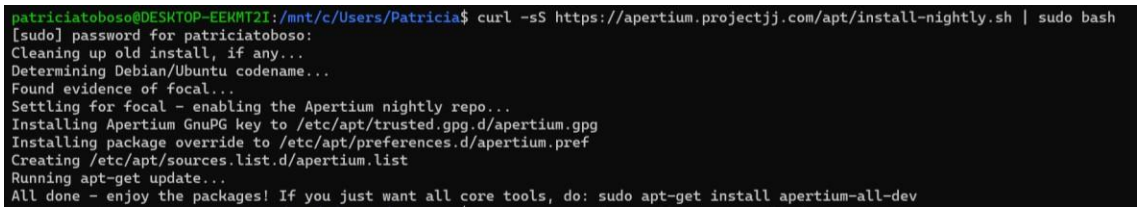

Figura 2: Comando para configurar las direcciones web de los programas de Apertium

El siguiente bloque de código que se debería emplear es:

sudo apt-get -f install apertium-all-dev

Esta orden instala de manera automática todos los paquetes que se necesitan para tener Apertium en un ordenador personal:

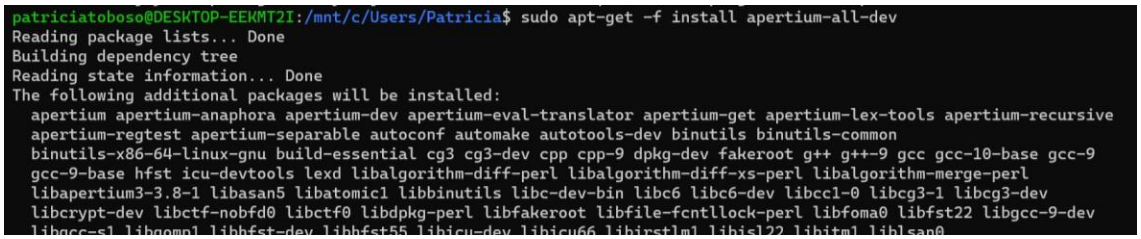

Figura 3: Comando para instalar todos los paquetes de Apertium

Una vez finalizada la orden, se debería comprobar que se ha instalado correctamente escribiendo *apertium* en el terminal.

[2] https://wiki.apertium.org/wiki/Install_Apertium_core_using_packaging

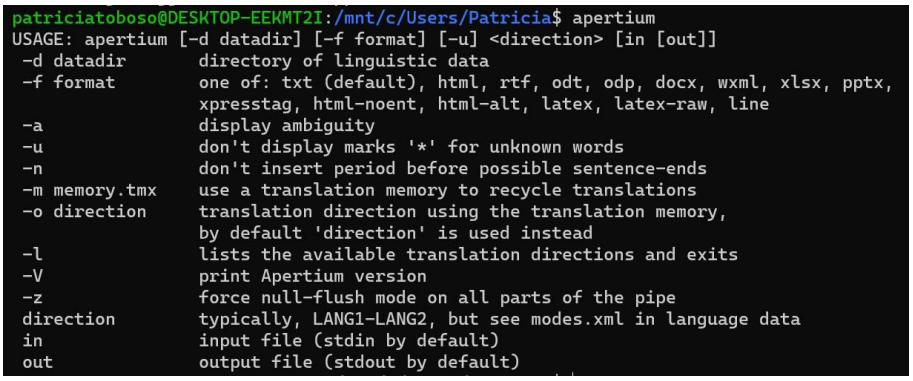

Figura 4: Comando para comprobar la correcta instalación de Apertium

Acto seguido, por un lado, en la carpeta *Documentos* del ordenador, se tendría que crear una carpeta llamada *apertium* y, por el otro, en GitHub, se deberían descargar los siguientes tres repositorios de Apertium: *apertium-cat-glg*, *apertium-cat* y *apertium-glg*[3].

El repositorio *apertium-cat-glg* contiene datos tanto de catalán como de gallego, mientras que *apertium-cat* y *apertium-glg* comprenden únicamente datos de catalán y gallego, respectivamente. Es decir, los datos bilingües de este traductor se encuentran en el primer repositorio y los datos monolingües en los otros dos.

A continuación, en el terminal, se debería introducir el comando *git clone* tres veces, seguido de las URL de cada uno de los tres repositorios previamente mencionados.

git clone https://github.com/apertium/apertium-cat-glg

git clone https://github.com/apertium/apertium-cat

git clone https://github.com/apertium/apertium-glg

[3] Estos recursos están disponibles en la sección *Repositories* del perfil de Apertium en GitHub (https://github.com/apertium), que contiene todos los repositorios y pares de idiomas instalados. No obstante, como no teníamos permisos para modificar estos archivos, Marc Riera, nuestro tutor, creó una copia en su perfil (https://github.com/MarcRiera) y nos concedió permisos de edición.

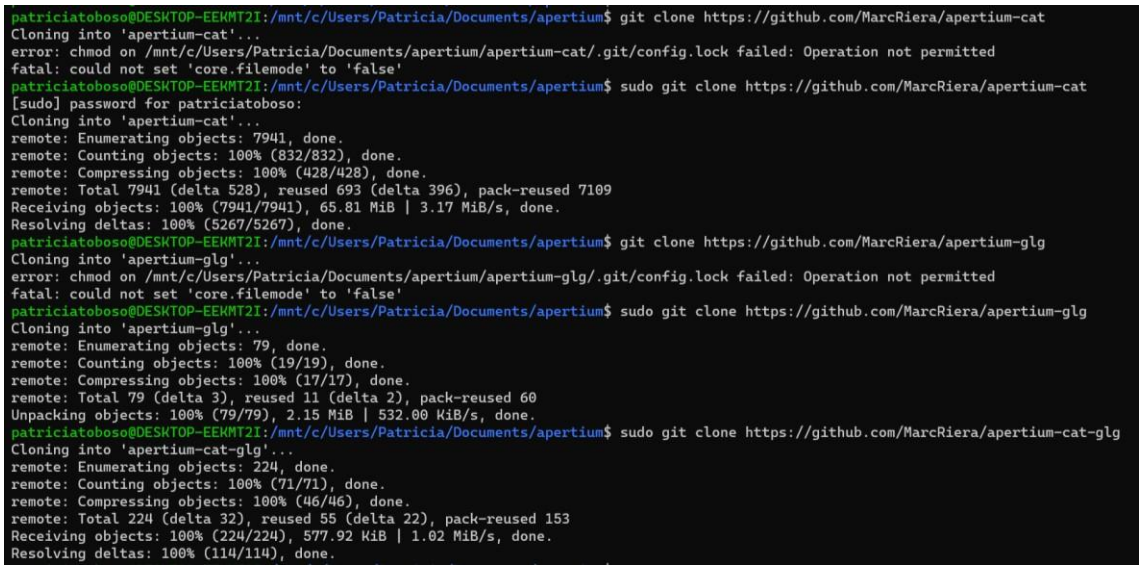

Figura 5: Comandos para clonar los repositorios «apertium-cat», «apertium-glg», «apertium-cat-glg»

A estas alturas, en el ordenador, concretamente en la carpeta *apertium* que se había creado anteriormente dentro de *Documentos*, se deberían haber creado tres carpetas nuevas con los nombres *apertium-cat*, *apertium-glg* y *apertium-cat-glg*, con sus datos correspondientes.

De momento, el traductor no podría ejecutarse, ya que el código de este estaría sin compilar, motivo por el cual los datos se podrían modificar libremente.

A la hora de compilar el programa, se le tendría que indicar al ordenador que los datos de catalán, gallego y del traductor catalán-gallego se encuentran en las carpetas *apertium-cat*, *apertium-glg* y *apertium-cat-glg*, respectivamente. Desde dentro de cada una de las tres carpetas se debería introducir, por un lado, el primer código para las carpetas de catalán y gallego y, por el otro, el segundo para la carpeta del traductor catalán-gallego:

1. *./autogen.sh*
2. *./autogen.sh --with-lang1=../apertium-cat --with-lang2=../apertium-glg*[4]

[4] Estos comandos, extraídos de la wiki de Apertium, se pueden encontrar en el siguiente enlace: https://wiki.apertium.org/wiki/Install_language_data_by_compiling

```
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium$ cd apertium-cat
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat$ ./autogen.sh
configure.ac:4: installing './install-sh'
configure.ac:4: installing './missing'
Makefile.am: installing './INSTALL'
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for APERTIUM... yes
checking for LTTOOLBOX... yes
checking for CG3... yes
checking for REGTEST... yes
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating apertium-cat.pc
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat$ cd ..
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium$ cd apertium-glg/
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-glg$ ./autogen.sh
configure.ac:4: warning: AM_INIT_AUTOMAKE: two- and three-arguments forms are deprecated.  For more info, see:
configure.ac:4: https://www.gnu.org/software/automake/manual/automake.html#Modernize-AM_005fINIT_005fAUTOMAKE-invocation
configure.ac:4: installing './install-sh'
configure.ac:4: installing './missing'
Makefile.am: installing './INSTALL'
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether ln -s works... yes
checking for gawk... (cached) gawk
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for APERTIUM... yes
checking for LTTOOLBOX... yes
checking for cg-comp... /usr/bin/cg-comp
checking for REGTEST... yes
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating apertium-glg.pc
```

Figura 6: Comandos para compilar los datos de catalán y gallego, por separado

```
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium$ cd apertium-cat-glg/
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg$ ./autogen.sh --with-lang1=../apertium-cat --with-lang2=../apertium-glg
configure.ac:4: warning: AM_INIT_AUTOMAKE: two- and three-arguments forms are deprecated.  For more info, see:
configure.ac:4: https://www.gnu.org/software/automake/manual/automake.html#Modernize-AM_005fINIT_005fAUTOMAKE-invocation
configure.ac:4: installing './install-sh'
configure.ac:4: installing './missing'
Makefile.am: installing './INSTALL'
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether ln -s works... yes
checking for gawk... (cached) gawk
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for APERTIUM... yes
checking for LTTOOLBOX... yes
checking for CG3... yes
checking for APERTIUM_LEX_TOOLS... yes
checking for gzcat... no
checking for zcat... /usr/bin/zcat
Using apertium-cat from ../apertium-cat
Using apertium-glg from ../apertium-glg
checking for REGTEST... yes
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
```

Figura 7: Comando para compilar los datos del traductor catalán-gallego

Si se quisiera compilar de nuevo el par de idiomas después de haber realizado cualquier cambio, simplemente se tendría que dar la orden *make langs* desde dentro de la carpeta del traductor. Este comando compilaría primero la carpeta de catalán, después la de gallego y, por último, la del traductor catalán-gallego.

Figura 8: Comando para compilar el par de lenguas después de introducir algún cambio

Compilarlo todo por primera vez es un proceso lento debido a que el programa tiene que hacerlo desde cero. En cambio, cuando se vuelve a compilar después de haberse modificado algún elemento, es más rápido puesto que solamente se compila la parte modificada.

Seguidamente, se debería descargar un programa que se utiliza no solo para conectarse desde Windows a Apertium una vez ya compilado, sino también para probar el funcionamiento del traductor. Para ello, se tendría que abrir un archivo llamado *apertium-viewer.html* que se encuentra en la carpeta *tools* del siguiente enlace: https://github.com/apertium/apertium-apy. A continuación, se tendría que seleccionar la opción de visualización *Raw* del archivo[5] para, posteriormente, poder descargar este HTML y guardarlo en la carpeta *apertium*, creada al principio dentro de la carpeta *Documentos* del ordenador.

Para probar el programa, se debería abrir el HTML *apertium-viewer.html* desde la carpeta *apertium-cat-glg*, donde están los datos del par de idiomas, y ejecutar la siguiente orden en la terminal:

apertium-apy .

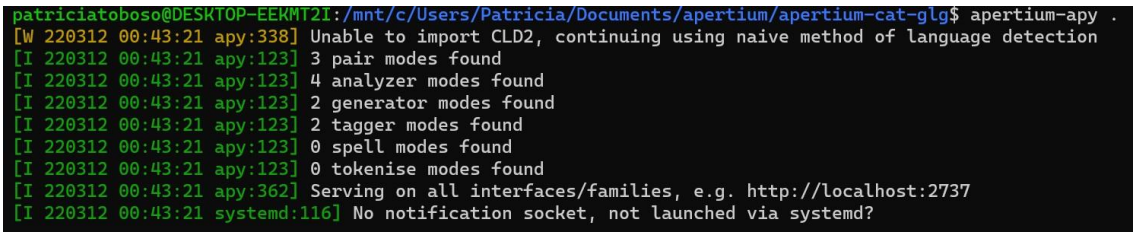

Figura 9: Comando que nos facilita el número de puerto necesario para acceder al traductor en el HTML

Como resultado, entre otra información, la terminal debería proporcionar un número de puerto (2737) y, tras haberlo introducido en el apartado *Enter port* del HTML, el programa ya se tendría que poder usar. Este programa genera una traducción final en la lengua destino, como lo haría el traductor en la página web oficial, pero además muestra paso por paso el proceso de la traducción, así como la estructura tanto externa como interna de las palabras que se quieren traducir.

Por ejemplo, al traducir «casa» y «cases» del catalán al gallego, así es como aparece la codificación interna de estas palabras en medio del proceso de traducción:

[5] https://raw.githubusercontent.com/apertium/apertium-apy/master/tools/apertium-viewer.html

```
apertium-tagger,-z,-g,/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg/cat-glg.prob
^casa<n><f><sg>$^.<sent>$[]
```

Figura 10: Estructura interna de la palabra «casa»

```
apertium-tagger,-z,-g,/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg/cat-glg.prob
^casa<n><f><pl>$^.<sent>$[]
```

Figura 11: Estructura interna de la palabra «casas»

1.3 Obtención de las coberturas léxicas

La cobertura léxica corresponde al porcentaje de léxico que el motor reconoce y se calcula en relación con un corpus. En este trabajo, se descargó la Wikipedia en catalán y gallego como corpus de referencia, ya que es suficientemente grande, desde el siguiente enlace: https://dumps.wikimedia.org/backup-index-bydb.html. Por un lado, para la Wikipedia en catalán, dentro de *cawiki*, se seleccionó el archivo *.xml* debajo de *Articles, templates, media/file descriptions, and primary meta-pages, in multiple bz2 streams, 100 pages per stream*. Por el otro, para la Wikipedia en gallego, dentro de *glwiki*, se volvió a escoger el archivo *.xml* debajo de la misma sección que en la Wikipedia en catalán.

Como los corpus de la Wikipedia descargados eran documentos XML, se utilizó la herramienta *WikiExtractor* para eliminar el formato del texto y convertirlo a TXT. Para ello, se usaron los siguientes comandos:

python3 WikiExtractor.py --infn cawiki-20220301-pages-articles-multistream.xml.bz2

python3 WikiExtractor.py --infn glwiki-20220301-pages-articles-multistream.xml.bz2

Una vez obtenidos los corpus de la Wikipedia en TXT, se crearon dos listas, una para cada idioma, de todas las palabras del corpus ordenadas por frecuencia, con el fin de traducir las listas en vez de todo el corpus.

Para crear las listas de frecuencia y calcular las coberturas léxicas, se emplearon los siguientes *scripts*[6]:

make-freqlist.sh

freqlist-coverage.sh

[6] Estos *scripts* se obtuvieron del enlace https://wiki.apertium.org/wiki/Calculating_coverage#Faster_coverage_testing_with_frequency_lists.

En el *script freqlist-coverage.sh*, se hizo un cambio en la línea 18 que presentaba la siguiente línea de código: *if(++printed<10) print $2,$3*. El valor 10 hacía referencia al número de palabras más utilizadas que se obtendrían en la cobertura léxica resultante. Para este trabajo, se aumentó la cifra a 2000 para que se generara una lista de los términos más comunes del corpus considerablemente más extensa.

En definitiva, para crear las listas de frecuencia y calcular las coberturas léxicas, se introdujeron, respectivamente, los siguientes códigos en la terminal:

> ***cat*** *<corpus>* ***.txt | ./make-freqlist.sh > llista.txt***
>
> ***<*** *<freqlist>* ***.txt ./freqlist-coverage.sh -d*** *<carpeta_motor> <direcció>* ***-morph > cobertura.txt***

A modo de ejemplo, los códigos que se emplearon para este trabajo en particular fueron:

> *cat wiki_ca.txt | ./make-freqlist.sh > llista_ca.txt*
>
> *< llista_ca.txt ./freqlist-coverage.sh -d ./apertium-cat-glg cat-glg-morph > cobertura_ca.txt*
>
> *cat wiki_gl.txt | ./make-freqlist.sh > llista_gl.txt*
>
> *< llista_gl.txt ./freqlist-coverage.sh -d ./apertium-cat-glg glg-cat-morph > cobertura_gl.txt*

Como observación, es conveniente recordar que, al trabajar desde Windows con WSL, es posible que se produzca algún error debido a que los finales de línea entre Unix/Linux y Windows son diferentes. Para solucionar este problema, se pueden editar los *scripts*, por ejemplo, con el blog de notas avanzado de Notepad++ siguiendo la ruta *Editar > Conversión fin de línea > Convertir a formato UNIX*. Desafortunadamente, aun haciendo estas modificaciones, puede que no sea suficiente y no se logren obtener ni las listas de frecuencia ni las coberturas léxicas con Windows. De manera que, para conseguir estos documentos, es probable que se tenga que recurrir a un ordenador cuyo sistema operativo sea Linux.

1.4 Extracción de las métricas WER y PER

Con el fin de obtener las métricas WER y PER fue preciso instalar un programa llamado *apertium-eval-translator*, que calcula estas métricas mediante la examinación y comparación de una traducción hecha por el motor de Apertium con una traducción humana que funcione como texto de referencia. Para ello, el primer paso fue introducir el siguiente código dentro de la carpeta del diccionario bilingüe:

> *sudo apt-get install apertium-eval-translator*

Figura 12: Instalación del programa «apertium-eval-translator»

A continuación, para traducir los textos de referencia *gl.txt* y *ca.txt* al otro idioma con TA, se ejecutaron las órdenes:

cat gl.txt | apertium -d . glg-cat > ca_apertium.txt

cat ca.txt | apertium -d . cat-glg > gl_apertium.txt

```
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg$ cat gl.txt | apertium -d . glg-cat > ca_apertium.txt
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg$ cat ca.txt | apertium -d . cat-glg > gl_apertium.txt
```

Figura 13: Comando que traduce los textos de referencia con el motor de Apertium

Una vez introducidos estos comandos, se generaron en la carpeta *apertium-cat-glg* los archivos resultantes *ca_apertium.txt* y *gl_apertium.txt*. Seguidamente, se ejecutaron estos otros códigos desde el terminal:

apertium-eval-translator -test gl-apertium.txt -ref gl.txt

apertium-eval-translator -test ca-apertium.txt -ref ca.txt

Estas instrucciones calculan las métricas WER y PER y, para ello, necesitan dos documentos: el texto de referencia original con traducción humana en un idioma y el texto de referencia traducido por Apertium automáticamente a este mismo idioma desde el otro idioma del par de lenguas.

```
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg$ apertium-eval-translator -test gl_apertium.txt -ref gl.txt
Test file: 'gl_apertium.txt'
Reference file 'gl.txt'

Statistics about input files
-------------------------------------------------------
Number of words in reference: 5355
Number of words in test: 5315
Number of unknown words (marked with a star) in test: 483
Percentage of unknown words: 9.09 %

Results when removing unknown-word marks (stars)
-------------------------------------------------------
Edit distance: 1714
Word error rate (WER): 32.01 %
Number of position-independent correct words: 4240
Position-independent word error rate (PER): 20.82 %

Results when unknown-word marks (stars) are not removed
-------------------------------------------------------
Edit distance: 1984
Word Error Rate (WER): 37.05 %
Number of position-independent correct words: 3962
Position-independent word error rate (PER): 26.01 %

Statistics about the translation of unknown words
-------------------------------------------------------
Number of unknown words which were free rides: 270
Percentage of unknown words that were free rides: 55.90 %
```

Figura 14: Métricas WER y PER obtenidas con el programa «apertium-eval-translator» para la dirección catalán-gallego

```
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg$ apertium-eval-translator -test ca_apertium.txt -ref ca.txt
Test file: 'ca_apertium.txt'
Reference file 'ca.txt'

Statistics about input files
---------------------------------------------------------
Number of words in reference: 5508
Number of words in test: 5591
Number of unknown words (marked with a star) in test: 504
Percentage of unknown words: 9.01 %

Results when removing unknown-word marks (stars)
---------------------------------------------------------
Edit distance: 2055
Word error rate (WER): 37.31 %
Number of position-independent correct words: 4258
Position-independent word error rate (PER): 24.20 %

Results when unknown-word marks (stars) are not removed
---------------------------------------------------------
Edit distance: 2270
Word Error Rate (WER): 41.21 %
Number of position-independent correct words: 4039
Position-independent word error rate (PER): 28.18 %

Statistics about the translation of unknown words
---------------------------------------------------------
Number of unknown words which were free rides: 215
Percentage of unknown words that were free rides: 42.66 %
```

Figura 15: Métricas WER y PER obtenidas con el programa «apertium-eval-translator» para la dirección gallego-catalán

A parte de las métricas WER y PER, las Figuras 14 y 15 presentan el siguiente contenido:

- el número de palabras que contiene el texto de referencia;
- el número de palabras traducidas por Apertium;
- el número de palabras desconocidas que se han detectado;
- el porcentaje de estas palabras desconocidas;
- las métricas WER y PER cuando las palabras desconocidas se eliminan de la evaluación;
- las métricas WER y PER cuando las palabras desconocidas siguen presentes en la evaluación.

1.5 Ejecución de las pruebas de vocabulario «testvoc»

La prueba de vocabulario tiene la función de detectar y marcar la cantidad de palabras que dan error y, por lo tanto, no generan una traducción válida en ambas direcciones del motor. En otras palabras, *testvoc* expande el diccionario, prueba cada posible análisis de todas las entradas, las intenta traducir una por una y comprueba si hay errores o problemas de etiquetado.

La ejecución de estas pruebas de vocabulario es fundamental, ya que se espera que el par de idiomas que se publique no presente errores en el momento en que se ejecute en la página web.

Para realizar estas pruebas, se tuvo que acceder a la carpeta */dev/testvoc* dentro de los datos bilingües y ejecutar el script *testvoc.sh*:

```
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg$ cd dev/testvoc/
patriciatoboso@DESKTOP-EEKMT2I:/mnt/c/Users/Patricia/Documents/apertium/apertium-cat-glg/dev/testvoc$ ./testvoc.sh
```

Figura 16: Comando para ejecutar el script «testvoc.sh» y obtener los resultados de las pruebas automáticamente

En las pruebas *testvoc* aparecen una serie de columnas, por ejemplo, la primera (*POS*) y la segunda (*Total*) hacen referencia a las diferentes categorías gramaticales que el programa ha detectado en el texto y el número total de palabras que pertenecen a cada categoría gramatical. Además, *Clean* contiene el número de entradas que han sido procesadas correctamente y, en *Clean %*, aparece el porcentaje correspondiente de palabras bien procesadas respecto al total que aparecía en la segunda columna. Finalmente, la columna *With #* señala el número de errores que se generan cuando la traducción se lleva a cabo.

```
================================================
POS      Total    Clean    With @   With #   Clean %
```

Figura 17: Columnas de las pruebas de vocabulario «testvoc»

2. Evaluación del estado previo del motor

Como se ha mencionado previamente en el apartado anterior (*1. Preparación del entorno de trabajo*), el estado inicial del motor se evaluó teniendo en cuenta los datos obtenidos de las coberturas léxicas, las métricas WER y PER, así como las pruebas de vocabulario *testvoc*.

Por lo que respecta a las coberturas léxicas obtenidas del corpus general de la Wikipedia, como se puede apreciar en las Figuras 18 y 19, los porcentajes de estas fueron bastante altos para ambos idiomas. Sin embargo, se reconocieron más palabras en gallego, puesto que el porcentaje de la cobertura léxica de este idioma resultó ser ligeramente superior (87,05 %) comparado con el del catalán (84,24 %).

```
Top unknown tokens:
3599648 ^l/*l$ ^./.<sent>$
3569777 ^d/*d$ ^./.<sent>$
639766 ^s/*s$ ^./.<sent>$
516399 ^L/*L$ ^./.<sent>$
241651 ^des/*des$ ^./.<sent>$
232813 ^qual/*qual$ ^./.<sent>$
141123 ^quals/*quals$ ^./.<sent>$
125358 ^Barcelona/*Barcelona$ ^./.<sent>$
106063 ^mediana/*mediana$ ^./.<sent>$
84.2371 % known of total 228405107 tokens
```

Figura 18: Cobertura léxica del catalán en fase inicial

```
Top unknown tokens:
42284 ^moi/*moi$ ^./.<sent>$
25052 ^hai/*hai$ ^./.<sent>$
23612 ^The/*The$ ^./.<sent>$
22100 ^lle/*lle$ ^./.<sent>$
19868 ^I/*I$ ^./.<sent>$
19195 ^quen/*quen$ ^./.<sent>$
18815 ^través/*través$ ^./.<sent>$
18551 ^II/*II$ ^./.<sent>$
17930 ^C/*C$ ^./.<sent>$
87.0482 % known of total 51713361 tokens
```

Figura 19: Cobertura léxica del gallego en fase inicial

En cuanto a las métricas WER y PER, obtenidas gracias al programa *apertium-eval-translator*, los resultados muestran que el porcentaje de palabras desconocidas en ambos idiomas fue bastante

bajo. Esto se puede explicar debido a los elevados porcentajes de las coberturas léxicas que fueron superiores a 80 % (Figura 18 y Figura 19).

Asimismo, en las Figuras 20 y 21, se puede observar que, cuando las palabras desconocidas no se eliminaron, los índices de error subieron, ya que estas palabras que no reconocía el motor estaban dentro de las evaluaciones. Las métricas PER fueron considerablemente más bajas (aproximadamente un 11 %) en comparación con las WER, debido a que la métrica PER no penaliza la reordenación de palabras. Aun así, tal y como se puede apreciar a continuación, las métricas WER y PER presentaban un porcentaje medio-bajo incluso cuando las palabras desconocidas no estaban presentes en la evaluación.

```
Test file: 'gl_apertium.txt'
Reference file 'gl.txt'

Statistics about input files
---------------------------------------------------------
Number of words in reference: 5355
Number of words in test: 5315
Number of unknown words (marked with a star) in test: 589
Percentage of unknown words: 11.08 %

Results when removing unknown-word marks (stars)
---------------------------------------------------------
Edit distance: 1752
Word error rate (WER): 32.72 %
Number of position-independent correct words: 4203
Position-independent word error rate (PER): 21.51 %

Results when unknown-word marks (stars) are not removed
---------------------------------------------------------
Edit distance: 2106
Word Error Rate (WER): 39.33 %
Number of position-independent correct words: 3839
Position-independent word error rate (PER): 28.31 %

Statistics about the translation of unknown words
---------------------------------------------------------
Number of unknown words which were free rides: 354
Percentage of unknown words that were free rides: 60.10 %
```

Figura 20: Test de evaluación de catalán a gallego en fase inicial

```
Test file: 'ca_apertium.txt'
Reference file 'ca.txt'

Statistics about input files
---------------------------------------------------------
Number of words in reference: 5508
Number of words in test: 5590
Number of unknown words (marked with a star) in test: 597
Percentage of unknown words: 10.68 %

Results when removing unknown-word marks (stars)
---------------------------------------------------------
Edit distance: 2051
Word error rate (WER): 37.24 %
Number of position-independent correct words: 4263
Position-independent word error rate (PER): 24.09 %

Results when unknown-word marks (stars) are not removed
---------------------------------------------------------
Edit distance: 2337
Word Error Rate (WER): 42.43 %
Number of position-independent correct words: 3969
Position-independent word error rate (PER): 29.43 %

Statistics about the translation of unknown words
---------------------------------------------------------
Number of unknown words which were free rides: 286
Percentage of unknown words that were free rides: 47.91 %
```

Figura 21: Test de evaluación de gallego a catalán en fase inicial

Según *testvoc*, en la fase inicial, los datos de la dirección catalán-gallego revelaron unos porcentajes muy elevados en todas las categorías gramaticales, salvo en los nombres propios (*np*). Los resultados de esta categoría no correspondían con el contenido del diccionario bilingüe, puesto que la sección de nombres propios ya era bastante extensa inicialmente. No obstante, según la Figura 22, solamente se detectaron un total de 342 entradas como nombres propios y el 95,32 % de estas no se pudieron procesar correctamente por la prueba de vocabulario. Por lo tanto, se dedujo que esta sección del diccionario debía presentar algún tipo de problema de etiquetado.

```
dilluns, 20 de juny de 2022, 10:26:04 CEST
================================================
POS     Total   Clean   With @  With #  Clean %
vblex   189429  189429  0       0       100
n       17309   17297   0       12      99.93
adj     12143   11629   0       514     95.76
adv     1414    1405    0       9       99.36
np      342     16      0       326     4.68
```

Figura 22: Prueba de vocabulario en la dirección catalán-gallego en fase inicial

En la dirección gallego-catalán, los datos de la prueba de vocabulario también fueron bastante óptimos, puesto que todas las categorías gramaticales tenían porcentajes altos. El porcentaje más bajo fue el de los adjetivos (*adj*), ya que una cuarta parte de estos dieron error, concretamente un 23,91 %.

```
dilluns, 20 de juny de 2022, 11:06:35 CEST
================================================
POS     Total   Clean   With @  With #  Clean %
vblex   584106  544782  0       39324   93.26
n       17751   17300   0       451     97.46
adj     15922   12102   0       3820    76.01
np      1732    1647    0       85      95.09
adv     1481    1440    0       41      97.23
```

Figura 23: Prueba de vocabulario en la dirección gallego-catalán en fase inicial

Al comparar los porcentajes de ambas direcciones, se puede apreciar que los de la dirección gallego-catalán fueron ligeramente inferiores, a excepción de los nombres propios que, como se mencionó anteriormente, lo más probable es que se deba a un error de etiquetado.

3. Optimización del motor

La parte más práctica de este trabajo consistió fundamentalmente en la implementación, en el diccionario bilingüe, de los términos más comunes en ambos idiomas que el motor no reconociera. Para ello, primero de todo se tuvieron que seleccionar las palabras de la cobertura léxica de la Wikipedia que se deberían introducir al diccionario *apertium-cat-glg.cat-glg.dix*. Acto seguido, por un lado, se revisó que estas no estuvieran dentro del diccionario para evitar que hubiera entradas duplicadas y, por el otro, en el caso de que ya aparecieran en el diccionario, se comprobó que no tuvieran errores ni de generación o etiquetas ni de equivalencia de palabras entre ambos idiomas.

Concretamente, en este TFM se creó un listado de aproximadamente 2000 palabras extraídas de la cobertura léxica (1000 términos por idioma) de las cuales se repitieron unas 50 palabras del gallego y 300 del catalán. Así que se acabó añadiendo un total de unos 950 y 700 términos nuevos en gallego y catalán, respectivamente, a la vez que se revisó que las 50 y 300 entradas ya presentes en el diccionario no tuvieran ningún error.

En otras palabras, como hubo términos repetidos, no se pudo añadir la totalidad de las 2000 palabras. Sin embargo, se aprovechó esta oportunidad para revisar y corregir estas palabras ya existentes en el diccionario, en el caso de ser necesario, para que el motor las pudiera identificar. Los cambios que se llevaron a cabo fueron:

- activar entradas que estaban bien, pero aparecían como comentarios (Figura 24);
- dejar a modo de nota traducciones literales o mal traducidas para que no fueran procesadas por el motor (Figura 25);
- corregir equivalentes de términos incorrectos (Figura 26) y entradas duplicadas con diferentes marcas de género o número (Figura 27);
- añadir el código de espacio en algunos términos que contienen más de una palabra (Figura 28);
- incorporar restricciones de dirección en algunas palabras que aparezcan en más de una entrada con diferentes equivalentes (Figura 29).

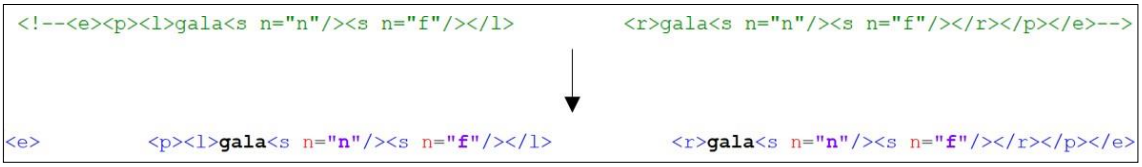

Figura 24: Entrada activada que previamente aparecía como comentario

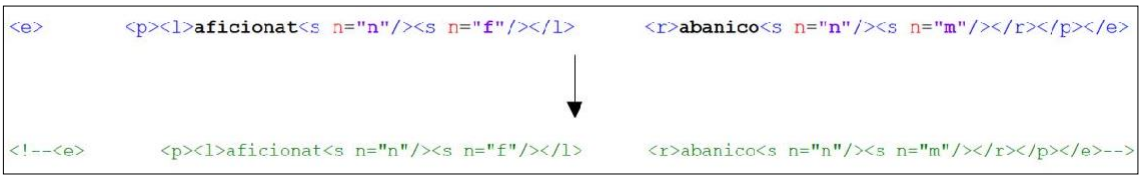

Figura 25: Entrada inhabilitada que aparece a modo de nota

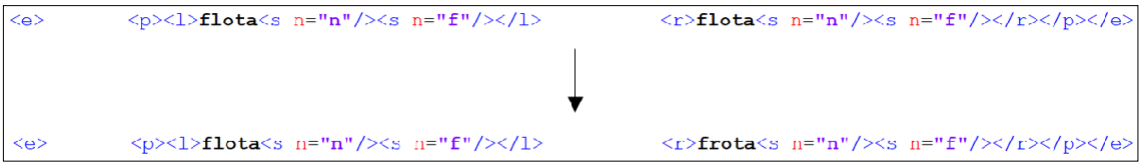

Figura 26: Entrada corregida con el equivalente correcto en gallego

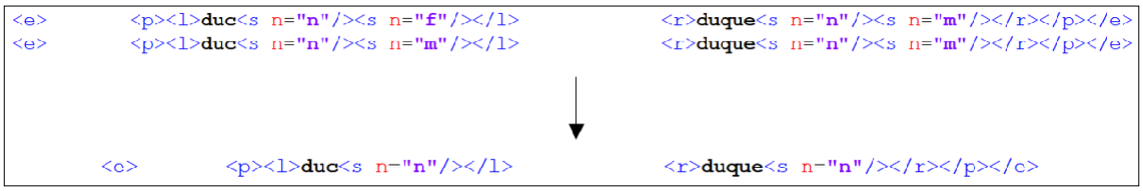

Figura 27: Entrada corregida que estaba duplicada y presentaba diferentes marcas de género

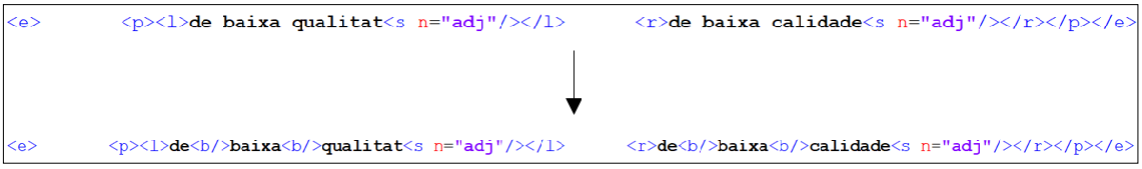

Figura 28: Entrada modificada para que contenga el código de espacio entre palabras

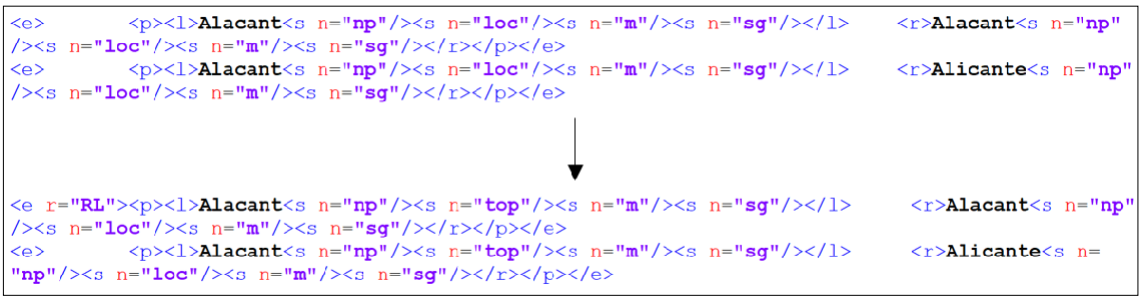

Figura 29: Entrada con restricción de dirección «RL» (de derecha a izquierda) incorporada para que así el término «Alacant» solo se traduzca como tal del gallego al catalán y no viceversa

Asimismo, dentro del diccionario bilingüe, se detectaron ciertos problemas con el contenido de las etiquetas. La primera modificación que se efectuó fue sustituir el símbolo de los topónimos *loc*, que se empleaba anteriormente para hacer referencia a estos, por *top*, símbolo utilizado actualmente para marcarlos.

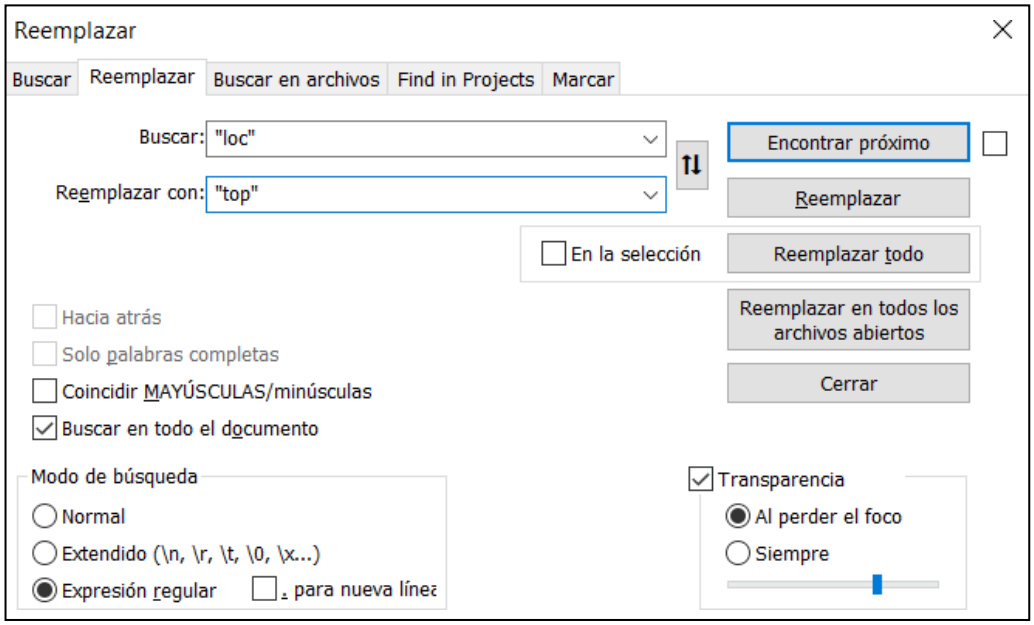

Figura 30: Cuadro de diálogo «Reemplazar» de Notepad++ para sustituir «loc» por «top» en todo el documento

Sin embargo, al echar un vistazo a los diccionarios monolingües de cada idioma, se pudo observar que, así como en el diccionario monolingüe catalán los topónimos fueron catalogados como *top*, en el gallego los clasificaron como *loc*. Así que, para poder aprovechar los diccionarios monolingües, el enfoque que se adoptó consistió en modificar el diccionario bilingüe, en vez de los monolingües, de manera que los topónimos en catalán tuvieran la etiqueta *top* y en gallego *loc*.

Para ello, se generó una expresión regular para buscar todos los topónimos en la parte gallega del diccionario y reemplazar la etiqueta *top* por *loc*. Se hizo una búsqueda con expresiones regulares desde *<r>* hasta *<s n="top"/>*, concretamente *<r>(.*)<s n="np"/><s n="top"/>*, y se reemplazó por *<r>\1<s n="np"/><s n="loc"/>*.

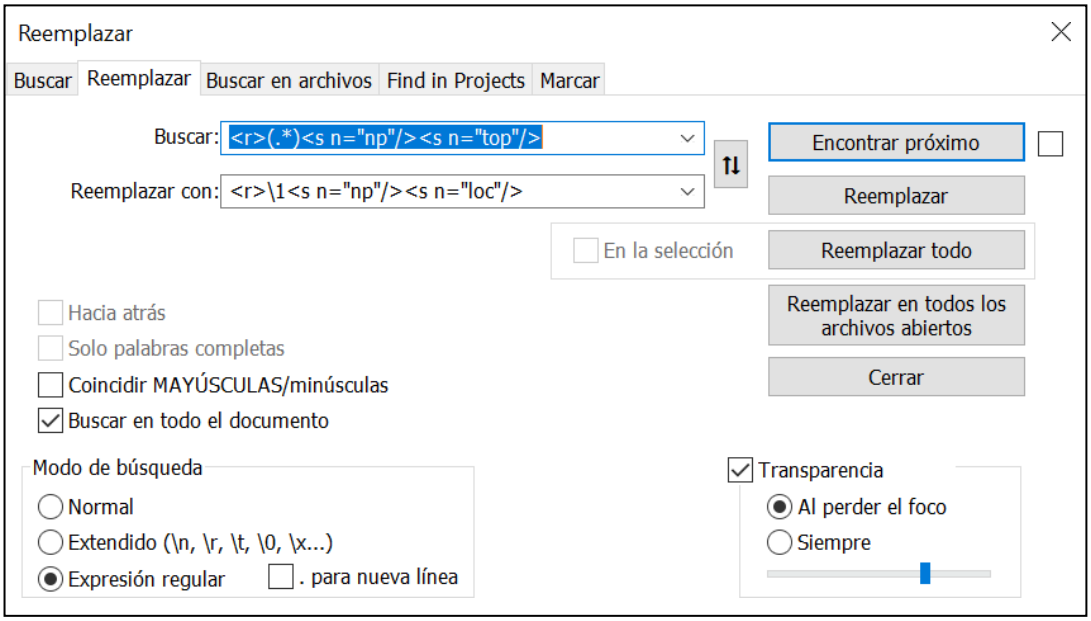

Figura 31: Expresión regular para que apareciera la etiqueta «loc», en vez de «top», en los topónimos gallegos

El resultado después de utilizar la expresión regular fue el siguiente:

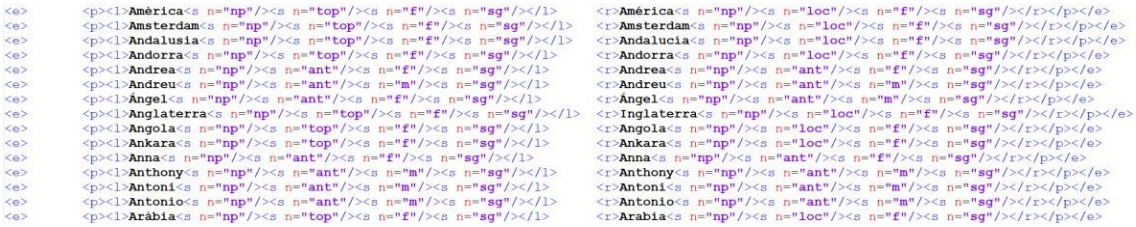

Figura 32: Fragmento extraído del apartado de topónimos del diccionario

Además, al principio del documento, aparece una lista de todas las etiquetas permitidas en el diccionario, de modo que se añadió *loc* a la lista.

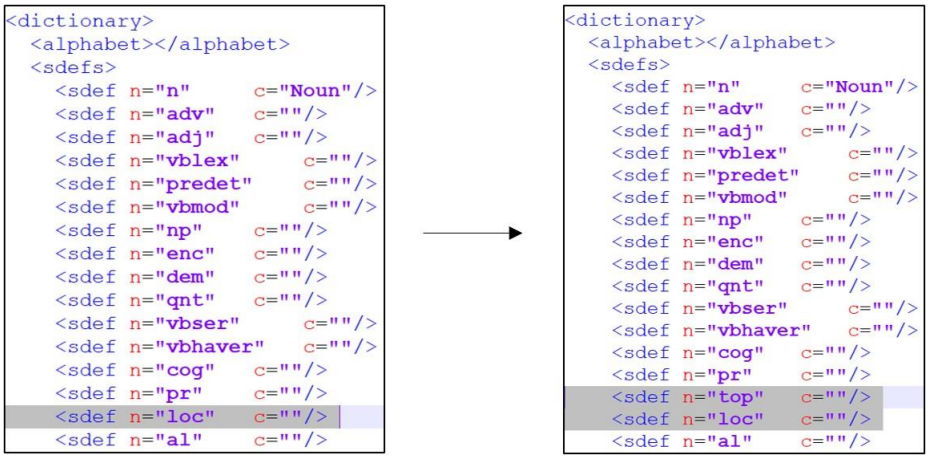

Figura 33: Lista actualizada de las etiquetas permitidas en el diccionario bilingüe

En toda la sección de nombres propios se detectó otro problema, puesto que la parte catalana de las entradas no contenía información ni de género ni de número.

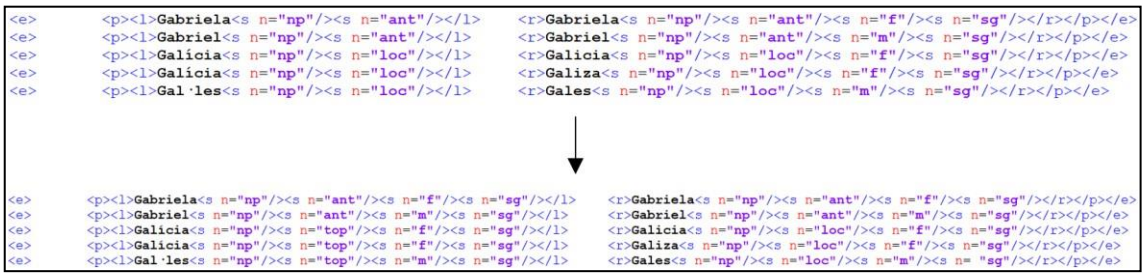

Figura 34: Entradas de la sección de nombres propios con la información de género y número añadida en la parte catalana del diccionario

Asimismo, en algunas entradas, el tipo de nombre propio que se había asignado en un idioma no correspondía con su equivalente en el otro. Como se trataba de un error, se duplicaron estas entradas y, como se puede observar a continuación, se atribuyó un tipo de nombre propio a cada una de ellas:

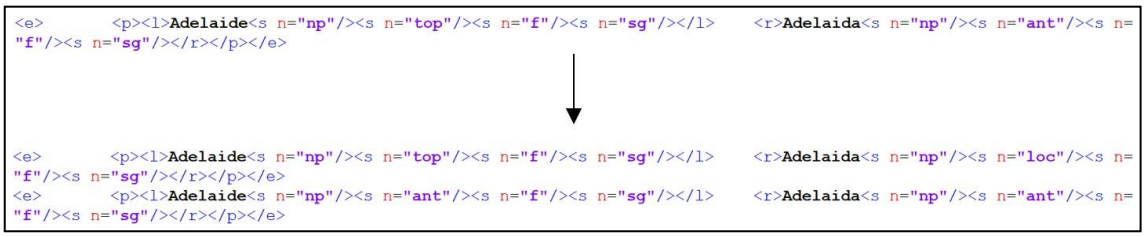

Figura 35: Entrada «Adelaide-Adelaida» duplicada para que el diccionario la pueda reconocer tanto como topónimo como antropónimo

No obstante, estas discrepancias o inconsistencias entre idiomas y omisiones de etiquetas no fueron exclusivas de los nombres propios. A lo largo de todas las secciones del diccionario se encontraron errores en las asignaciones de, sobre todo, género, número e, incluso, paradigmas.

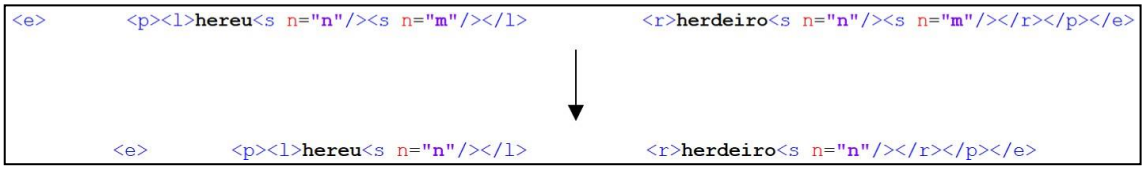

Figura 36: Entrada corregida debido a una asignación incorrecta de etiquetas, en este caso, de género

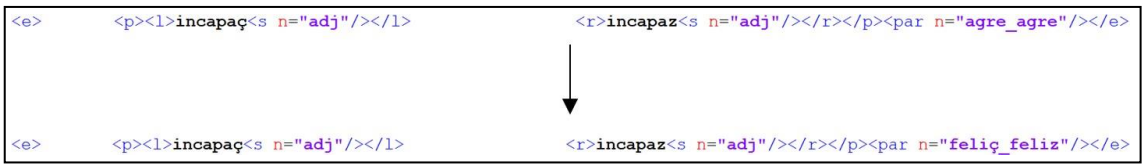

Figura 37: Entrada corregida debido a que se le asignó un tipo de paradigma erróneo

Para automatizar el proceso de introducción de las entradas nuevas en el diccionario bilingüe y siguiendo el formato de Apertium, se empleó la fórmula CONCAT de Excel (no confundir con CONCATENATE o CONCATENAR), que permite juntar de manera automática los datos de diferentes celdas y trasladarlos como una unidad a una única celda. De manera que, primero, se seleccionó una línea de código de referencia del diccionario, concretamente una entrada que tuviera el máximo número de etiquetas posibles[7]. A continuación, se extrajo la información variable (es decir, el término en sí y las marcas de categoría gramatical, género y número) y se incorporó esta información en celdas separadas. Una vez separado el código en las diferentes celdas e introducida la información variable correspondiente para cada término, se usó la fórmula CONCAT para unir todas las piezas del código.

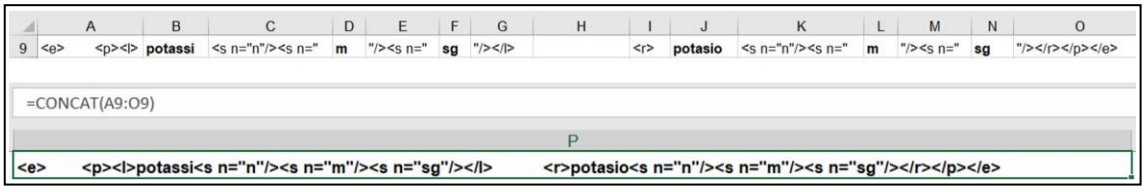

Figura 38: Proceso seguido para crear entradas automatizadas con el formato de Apertium

Una vez realizados todos los cambios del diccionario bilingüe en el ordenador, fue el momento de subir la versión nueva de este a GitHub, pero antes se tuvo que compilar. Para ello, el diccionario debía estar dentro de la carpeta *apertium-cat-glg* para, a continuación, acceder a esta desde el terminal de Linux y ejecutar el comando que se utilizó al compilar el diccionario por primera vez: *make langs*. Esta instrucción además fue de gran ayuda porque proporciona información concreta sobre las líneas del documento que contienen algún error de formato, así que se subió con certeza una versión actualizada del diccionario sin fallos.

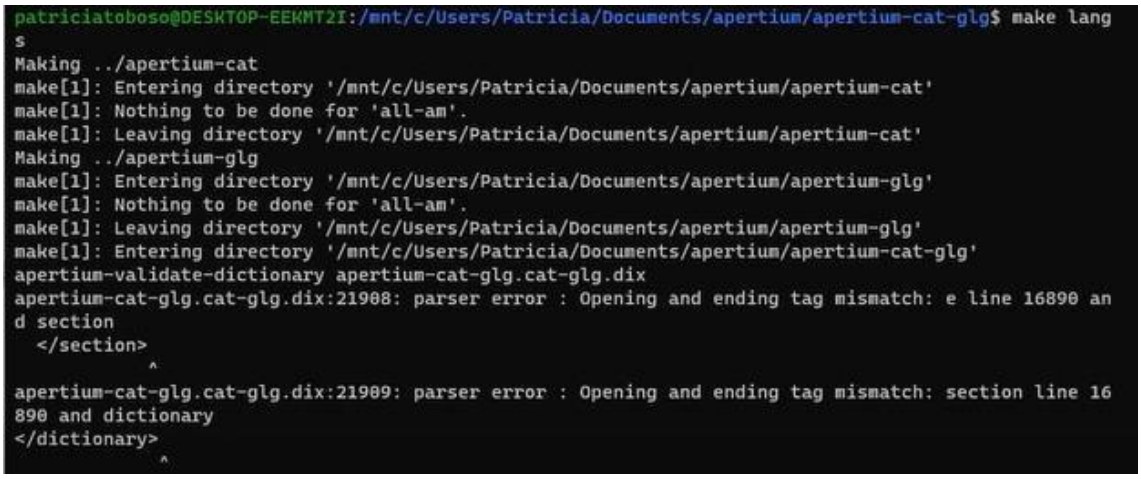

Figura 39: Comando utilizado para compilar el diccionario bilingüe y detectar si hay problemas de formato en el documento

[7] El número de etiquetas que debe tener una palabra varía dependiendo de la categoría gramatical y la morfología de esta. En el siguiente enlace de la página Wiki de Apertium se puede encontrar información acerca de los diferentes símbolos utilizados por el motor: https://wiki.apertium.org/wiki/List_of_symbols.

Tras haber compilado el diccionario y antes de subirlo a GitHub, en el terminal desde la carpeta donde estaba el código (*apertium-cat-glg*), se ejecutaron los comandos *git status* y *git diff*. Como resultado, con el primero, se comprobó que efectivamente el diccionario bilingüe se había modificado y, con el segundo, se pudieron observar todos los cambios efectuados en este archivo[8].

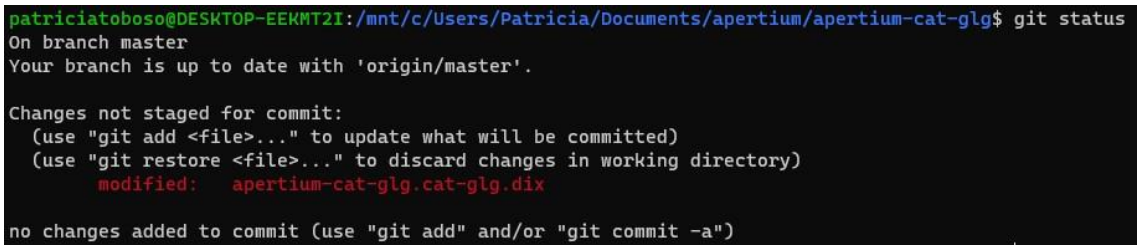

Figura 40: Información, obtenida mediante el comando «git status», acerca de los cambios que se han llevado a cabo en la carpeta «apertium-cat-glg» marcados en rojo

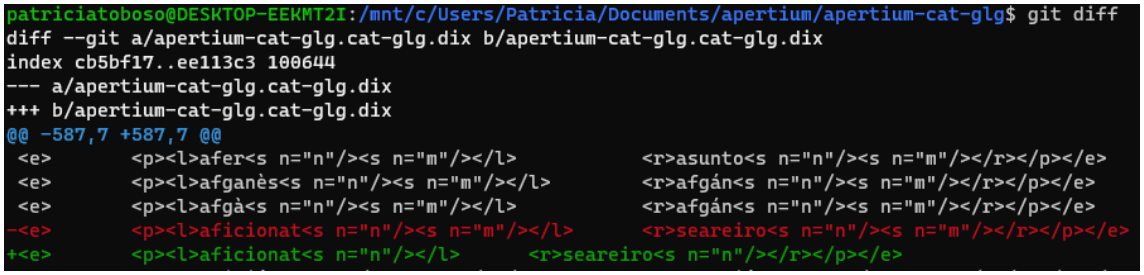

Figura 41: Información detallada, obtenida mediante el comando «git diff», acerca de todas las modificaciones que se han llevado a cabo dentro del diccionario bilingüe

A estas alturas, los cambios seguían pendientes de ser subidos a GitHub, de modo que se usó *git add .*, comando que afecta a todo el contenido de la carpeta. Aun así, como solo se habían efectuado cambios en el diccionario, únicamente se añadió este documento.

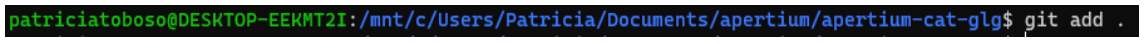

Figura 42: Comando utilizado para subir a GitHub los cambios pendientes

Al volver a usar la orden *git status*, esta vez el diccionario apareció en color verde, en vez de en rojo, conforme este sería el documento que se publicaría.

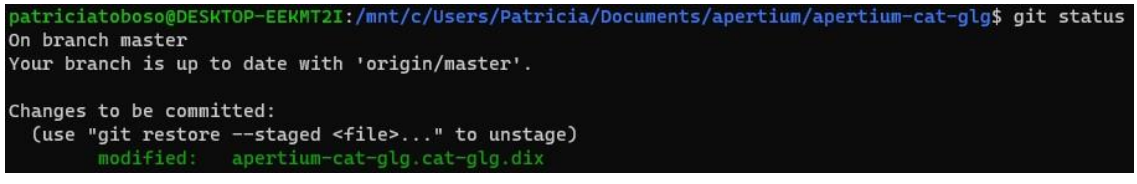

Figura 43: Información, obtenida mediante el comando «git status», que muestra en verde los cambios a publicarse

[8] Toda esta información se almacena en GitHub para que, en cualquier momento, se puedan consultar los cambios en sí que se han realizado, junto con información sobre la fecha de modificación y autor de estos.

El siguiente paso fue confirmar los cambios y, para ello, se ejecutó la orden *git commit -m* seguido de un mensaje, que indicara la naturaleza de los cambios realizados, entre comillas. No obstante, antes se tuvo que configurar una cuenta de GitHub e introducir una dirección de correo y nombre.

Figura 44: Comando que se utiliza para confirmar los cambios

Tras haber configurado la cuenta de GitHub, se pudo introducir el código *git commit -m* junto al mensaje entre comillas.

Figura 45: Código que permite añadir un mensaje a modo de resumen para que los cambios tengan un nombre

Puesto que este trabajo está compuesto por dos personas y solamente se reflejaría el nombre de una en el historial de cambios de GitHub como autor/a, se modificó el mensaje original para añadir ambos nombres en este con el código *git commit --amend -m*:

Figura 46: Comando para modificar el mensaje de los cambios

Para verificar que aparecía la información correcta en el historial de últimos cambios, se ejecutó la orden *git log*, que muestra el autor de los cambios junto con la fecha y hora de estos.

Figura 47: Historial de cambios de la carpeta en GitHub

Después de confirmar los cambios, se utilizó *git push* para subirlos a GitHub. Este código pide un nombre de usuario de GitHub y una contraseña temporal que se generó a través de los siguientes pasos en la plataforma de GitHub: *Settings > Developer settings > Personal access tokens*. Se tuvo que configurar la caducidad de la contraseña, rellenar el campo *Note* y marcar la casilla general *repo*. Una vez generada la contraseña, se ejecutó *git push* y el diccionario bilingüe se subió exitosamente.

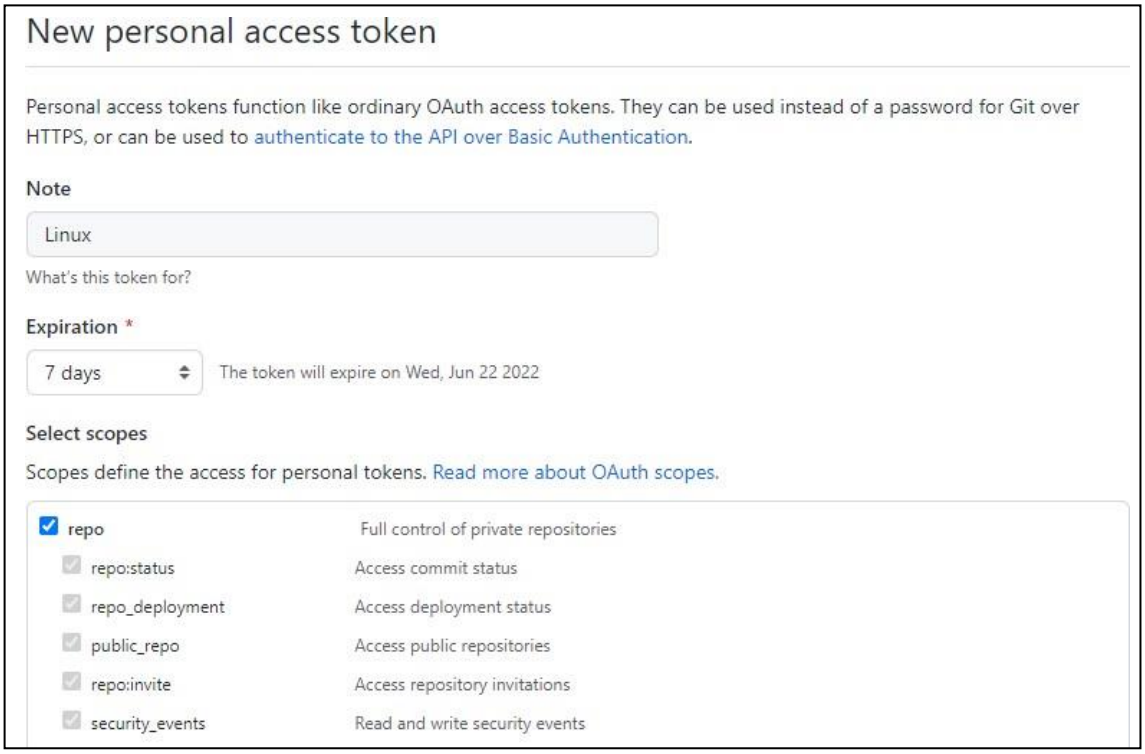

Figura 48: Configuración previa a generar una contraseña temporal en GitHub

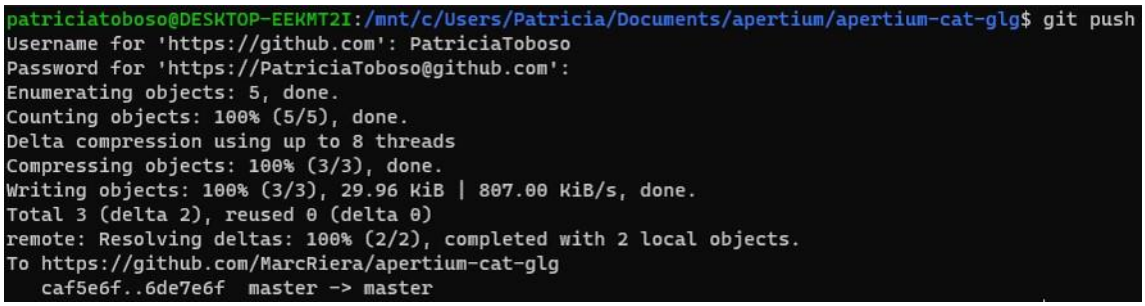

Figura 49: Comando «git push», utilizado para subir los cambios efectuados a GitHub

Además, al acceder a la carpeta *apertium-cat-glg* en GitHub, se pudo comprobar que el archivo se subió correctamente.

Figura 50: Evidencia conforme el diccionario se subió con éxito en la cuenta de Apertium de GitHub

Desde la plataforma, con tan solo clicar en el mensaje *Millora lèxica TFM Enrique Orjales i Patricia Toboso* que aparece en la Figura 50, se puede observar el registro de los cambios llevados a cabo, con el número de líneas añadidas y eliminadas respecto a la versión anterior. De hecho, al

hacer clic en *load diff*, se muestran todas las modificaciones realizadas en el diccionario bilingüe en verde y rojo.

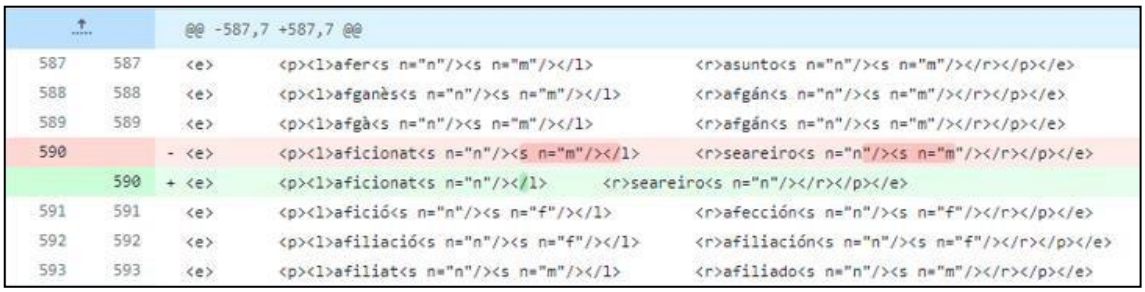

Figura 51: Registro de cambios realizados en el diccionario bilingüe de GitHub

Es importante recordar que todos los cambios mencionados en este apartado han sido subidos e incorporados a Apertium y, por lo tanto, están disponibles en el repositorio del código fuente de Apertium[9].

4. Evaluación del estado actualizado del motor

Una vez subidos todos los cambios del diccionario bilingüe a GitHub, con el fin de valorar el estado final del motor, se han vuelto a obtener y analizar los mismos datos que en la evaluación inicial, es decir, las coberturas léxicas, las métricas WER y PER y las pruebas de vocabulario *testvoc*.

En las coberturas léxicas, como se puede apreciar en las Figuras 52 y 53, las cifras han sido muy elevadas, la gallega siendo un 1,54 % más alta que la catalana.

```
Top unknown tokens:
3599648 ^l/*l$ ^./.<sent>$
3569777 ^d/*d$ ^./.<sent>$
639766 ^s/*s$ ^./.<sent>$
516399 ^L/*L$ ^./.<sent>$
241651 ^des/*des$ ^./.<sent>$
232813 ^qual/*qual$ ^./.<sent>$
141123 ^quals/*quals$ ^./.<sent>$
102074 ^n/*n$ ^./.<sent>$
73723 ^II/*II$ ^./.<sent>$
86.6151 % known of total 228405107 tokens
```

Figura 52: Cobertura léxica del catalán en fase final

[9] https://github.com/apertium/apertium-cat-glg/commit/6c8b6b65d4ed7b6573d5cfc3514973355192fe62

```
Top unknown tokens:
42284 ^moi/*moi$ ^./.<sent>$
25052 ^hai/*hai$ ^./.<sent>$
23612 ^The/*The$ ^./.<sent>$
22100 ^lle/*lle$ ^./.<sent>$
19868 ^I/*I$ ^./.<sent>$
19195 ^quen/*quen$ ^./.<sent>$
18815 ^través/*través$ ^./.<sent>$
18551 ^II/*II$ ^./.<sent>$
17930 ^C/*C$ ^./.<sent>$
88.156 % known of total 51713361 tokens
```

Figura 53: Cobertura léxica del gallego en fase final

Con las evaluaciones de las métricas WER y PER, reflejadas en las Figuras 54 y 55, se ha podido discernir que los porcentajes de palabras desconocidas del gallego (8,07 %) y el catalán (9 %) han resultado ser considerablemente bajos. Otro dato reseñable de estas figuras es que, en ambas direcciones (catalán-gallego y gallego-catalán), dependiendo de si se han quitado o mantenido las palabras desconocidas en la evaluación, las métricas han disminuido o aumentado, respectivamente.

```
Test file: 'gl_apertium.txt'
Reference file 'gl.txt'

Statistics about input files
---------------------------------------------------------
Number of words in reference: 5355
Number of words in test: 5315
Number of unknown words (marked with a star) in test: 429
Percentage of unknown words: 8.07 %

Results when removing unknown-word marks (stars)
---------------------------------------------------------
Edit distance: 1702
Word error rate (WER): 31.78 %
Number of position-independent correct words: 4252
Position-independent word error rate (PER): 20.60 %

Results when unknown-word marks (stars) are not removed
---------------------------------------------------------
Edit distance: 1948
Word Error Rate (WER): 36.38 %
Number of position-independent correct words: 3999
Position-independent word error rate (PER): 25.32 %

Statistics about the translation of unknown words
---------------------------------------------------------
Number of unknown words which were free rides: 246
Percentage of unknown words that were free rides: 57.34 %
```

Figura 54: Test de evaluación de catalán a gallego en fase final

```
Test file: 'ca_apertium.txt'
Reference file 'ca.txt'

Statistics about input files
-------------------------------------------------------
Number of words in reference: 5508
Number of words in test: 5591
Number of unknown words (marked with a star) in test: 503
Percentage of unknown words: 9.00 %

Results when removing unknown-word marks (stars)
-------------------------------------------------------
Edit distance: 2055
Word error rate (WER): 37.31 %
Number of position-independent correct words: 4258
Position-independent word error rate (PER): 24.20 %

Results when unknown-word marks (stars) are not removed
-------------------------------------------------------
Edit distance: 2270
Word Error Rate (WER): 41.21 %
Number of position-independent correct words: 4039
Position-independent word error rate (PER): 28.18 %

Statistics about the translation of unknown words
-------------------------------------------------------
Number of unknown words which were free rides: 215
Percentage of unknown words that were free rides: 42.74 %
```

Figura 55: Test de evaluación de gallego a catalán en fase final

Finalmente, con los datos obtenidos por los *testvoc* (Figuras 56 y 57), se puede observar que los valores de ambas direcciones han sido generalmente muy altos. No obstante, hay algunas cifras que, en vez de oscilar entre 90 % y 100 %, están a un 75 % aproximadamente. Este es el caso de los nombres propios en la dirección catalán-gallego, una cuarta parte de los cuales no se han reconocido por el motor, y de los adjetivos en la dirección gallego-catalán, que un 23,99 % han presentado algún error.

```
dilluns, 20 de juny de 2022, 11:00:26 CEST
=================================================
POS     Total   Clean   With @  With #  Clean %
vblex   189654  189504  0       150     99.92
n       17881   17474   0       407     97.72
adj     12429   11767   0       662     94.67
np      1740    1305    0       435     75
adv     1421    1411    0       10      99.29
```

Figura 56: Prueba de vocabulario en la dirección catalán-gallego en fase final

```
dilluns, 20 de juny de 2022, 11:06:35 CEST
=================================================
POS     Total   Clean   With @  With #  Clean %
vblex   584106  544782  0       39324   93.26
det     397305  367380  0       29925   92.46
n       17751   17300   0       451     97.46
adj     15922   12102   0       3820    76.01
np      1732    1647    0       85      95.09
adv     1481    1440    0       41      97.23
```

Figura 57: Prueba de vocabulario en la dirección gallego-catalán en fase final

IV. Resultados

La finalidad de este apartado es llevar a cabo una comparación, lo más detallada posible, del estado previo y actualizado del motor de traducción catalán-gallego de Apertium para así conocer el progreso que realmente se ha conseguido con la aportación de este TFM.

A la hora de comparar el estado inicial y final del motor, se puede observar que las coberturas léxicas de la Wikipedia en catalán y gallego han mejorado, ya que los porcentajes han subido un 2,36 % y un 1,11 % para cada idioma. Seguramente esto se debe principalmente a tres factores: por un lado, la incorporación de los 1000 términos más comunes de cada idioma en el diccionario bilingüe, obtenidos de la lista de la cobertura léxica; por otro, la corrección de la sección de nombres propios, que afectó a más de 900 entradas del diccionario; y, finalmente, el hecho de que los valores altos de estas coberturas léxicas, en ambos estados, dificultan que los porcentajes aumenten de manera rápida y significativa.

COBERTURA LÉXICA			
	Estado inicial	Estado final	Diferencia
Catalán	84,24 %	86,62 %	↑ 2,36 %
Gallego	87,05 %	88,16 %	↑ 1,11 %

Figura 58: Comparación del estado inicial y final de las coberturas léxicas en catalán y gallego

Por lo que respecta a las palabras desconocidas de los textos de referencia en catalán y gallego, se partió de unos valores iniciales de 11,08 % y 10,68 % y, después de añadir los nuevos términos de la lista de la cobertura al diccionario bilingüe, los porcentajes actuales han disminuido un 3,01 % en catalán y 1,68 % en gallego. Actualmente, el motor desconoce un 8,07 % de palabras en catalán y un 9 % de palabras en gallego.

PORCENTAJE DE PALABRAS DESCONOCIDAS			
	Estado inicial	Estado final	Diferencia
Catalán	11,08 %	8,07 %	↓ 3,01 %
Gallego	10,68 %	9 %	↓ 1,68 %

Figura 59: Comparación de los porcentajes de palabras desconocidas en el estado inicial y final de ambos idiomas

Este mismo patrón se puede percibir en los valores WER y PER obtenidos, puesto que, en general, los porcentajes de estas métricas han acabado disminuyendo, sobre todo si las palabras desconocidas no han sido eliminadas de las evaluaciones (Figuras 60 y 61).

Concretamente, en la dirección catalán-gallego, tanto los porcentajes de WER como los de PER han bajado, respectivamente, un 0,94 % y 0,91 %, sin palabras desconocidas, y un 2,95 % y 2,99 %, con palabras desconocidas.

		Estado inicial	Estado final	Diferencia
Sin palabras desconocidas	WER	32,72 %	31,78 %	↓ 0,94 %
	PER	21,51 %	20,60 %	↓ 0,91 %

		Estado inicial	Estado final	Diferencia
Con palabras desconocidas	WER	39,33 %	36,38 %	↓ 2,95 %
	PER	28,31 %	25,32 %	↓ 2,99 %

Figura 60: Comparación de las métricas WER y PER de la dirección catalán-gallego en la fase inicial y final

En la dirección gallego-catalán, sin embargo, al eliminar las palabras desconocidas, curiosamente el porcentaje actual ha aumentado, aunque mínimamente, 0,07 % de WER y 0,31 de PER. No obstante, este resultado no es alarmante y menos aun teniendo en cuenta que, manteniendo las palabras desconocidas en la evaluación, las métricas WER y PER han mejorado su porcentaje un 1,22 y 1,25, respectivamente. Es posible que las métricas no hayan disminuido más porque, pese a haber añadido nuevas entradas en el diccionario, las entradas de los diccionarios podrían, por ejemplo, tener algún error de etiquetado o directamente no coincidir con los términos exactos que aparecen en el texto de referencia.

		Estado inicial	Estado final	Diferencia
Sin palabras desconocidas	WER	37,24 %	37,31 %	↑ 0,07 %
	PER	24,09 %	24,20 %	↑ 0,31 %

		Estado inicial	Estado final	Diferencia
Con palabras desconocidas	WER	42,43 %	41,21 %	↓ 1,22 %
	PER	29,43 %	28,18 %	↓ 1,25 %

Figura 61: Comparación de las métricas WER y PER de la dirección gallego-catalán en la fase inicial y final

Además, es importante mencionar que, como se puede observar en la Figura 59, el hecho de que el motor actualmente reconozca más palabras explica que, en ambas direcciones, la diferencia entre el estado inicial y final de las métricas sea más pronunciada al mantener las palabras desconocidas en las evaluaciones, en vez de eliminarlas (Figuras 60 y 61).

Finalmente, de las pruebas de vocabulario *testvoc* cabe destacar que los resultados se han mantenido, pese a verse ligeramente afectados por pequeñas variaciones. Esto se debe principalmente a que el trabajo que se ha efectuado en este TFM se ha centrado básicamente en el diccionario bilingüe. De manera que, seguramente, la mayoría de los errores que han aparecido en estas pruebas de vocabulario se podrían solucionar revisando y actualizando los diccionarios monolingües.

Sin embargo, es crucial comentar la notable mejora del 70 % que se ha realizado por lo que respecta a los nombres propios en la dirección catalán-gallego (Figura 62). Lo más probable es que los errores de etiquetado, previamente mencionados en el apartado *3. Optimización del motor*, no permitieran que la mayoría de los nombres propios fueran siquiera reconocidos como tales por el motor. Además, no se puede ignorar la incorporación de aproximadamente 1000 nombres propios que se llevó a cabo en el diccionario bilingüe gracias a este TFM. El número total de estos

ha aumentado de 342 a 1740 y el porcentaje de nombres propios con errores pasó de ser inicialmente un 95,32 % a ser actualmente un 25 %.

TESTVOC				
		Estado inicial	Estado final	Diferencia
Catalán-gallego	vblex	100 %	99,92 %	↓ 0,08 %
	n	99,93 %	97,72 %	↓ 2,21 %
	adj	95,76 %	94,67 %	↓ 1,09 %
	adv	99,36 %	99,29 %	↓ 0,07 %
	np	4,68 %	75 %	↑ 70,32 %

TESTVOC				
		Estado inicial	Estado final	Diferencia
Gallego-catalán	vblex	93,26 %	93,26 %	-
	n	97,59 %	97,26 %	↓ 0,33 %
	adj	76,09 %	76,01 %	↓ 0,08 %
	adv	97,22 %	97,23 %	↑ 0,01 %
	np	98,62 %	95,09 %	↓ 3,53 %

Figura 62: Comparación de los resultados de las pruebas de vocabulario «testvoc» entre el estado inicial y final en ambas direcciones

V. Conclusiones

El presente trabajo cierra con esta sección en la que expondremos las conclusiones de los resultados obtenidos, además de las observaciones que consideramos pertinentes para continuar mejorando el motor catalán-gallego de Apertium y abrir futuras líneas de investigación.

El principal objetivo que definimos cuando nos planteamos este trabajo en un principio fue el de mejorar la calidad lingüística del motor de traducción de Apertium del par de idiomas catalán-gallego. A nuestro parecer, aunque la mayoría de los resultados que hemos obtenido puedan parecer sutiles, al fin y al cabo, muestran una clara mejoría en la calidad del motor con un aumento de la cobertura léxica y disminución de los valores de las métricas WER y PER a nivel general. Además, se debe tener en cuenta que esta mejora es de carácter léxico y, por lo tanto, no es tan significativa como lo serían otras (como, por ejemplo, las verbales), ya que las mejoras léxicas están más limitadas de por sí. Desafortunadamente, también se deben tener presentes las limitaciones de tiempo de este TFM, razón por la cual no hemos podido aplicar a la totalidad del diccionario los cambios efectuados.

A lo largo de la elaboración de este trabajo, nuestra intención ha sido la de relatar los pasos que hemos seguido en todo momento. Así, cualquier persona podrá orientarse con este TFM no solo si quisiera retomar el progreso llevado a cabo, sino también en el caso de necesitar una referencia para recrear este trabajo con otros pares de idiomas.

Como observación final, nos llamó especialmente la atención que algunas de las palabras obtenidas de las listas de las coberturas léxicas, sobre todo del catalán, estuvieran ya presentes dentro del diccionario. Hemos llegado a la conclusión que esto podría deberse a que, o bien las etiquetas de estas palabras tenían algún error, o bien las entradas no coincidían de manera exacta en el diccionario bilingüe y en el monolingüe. Esto confundiría al motor y justificaría que no fuera capaz de interpretar ni reconocer estas palabras.

Por último, cabe resaltar el hecho de que hemos trabajado en este TFM dos personas cuyas lenguas maternas son el catalán, por un lado, y el gallego, por el otro. Por este motivo en particular quisimos centrarnos en el diccionario bilingüe, puesto que conocemos ambas lenguas minorizadas y consideramos que esta era nuestra ventaja y lo que hacía de nuestra aportación una contribución beneficiosa y significativa.

VI. Bibliografía

Apertium-cat-glg. GitHub. Recuperado el 27 de junio de 2022 de https://github.com/apertium/apertium-cat-glg

Armentano-Oller, C., y Forcada, M. L. (2006). Open-source machine translation between small languages: Catalan and Aranese Occitan. *Strategies for Developing Machine Translation for Minority Languages*, 51-54. https://www.isca-speech.org/archive_open/saltmil/SALTMIL2006_procs2006.pdf

Armentano-Oller, C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Montava Bleda, M. A., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G. y Sánchez-Martínez, F. (2007). Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática. https://rua.ua.es/dspace/bitstream/10045/27531/1/armentano07.pdf

Cronin, M. (1995). Altered States: Translation and Minority Languages. TTR: *Traduction, terminologie, rédaction*, *8*(1), 85-103. https://doi.org/10.7202/037198ar

Elliott, D., Hartley, A. y Atwell, E. (2004). A Fluency Error Categorization Scheme to Guide Automated Machine Translation Evaluation. En R. E. Frederking y K. B. Taylor (Eds.), *Machine Translation: From Real User to Research* (pp. 64-73). Springer. https://doi.org/10.1007/978-3-540-30194-3_8

Enlace del código fuente de motor de traducción catalán-gallego de Apertium: https://github.com/apertium/apertium-cat-glg

Forcada, M. L. (2009). Apertium: traducció automàtica de codi obert per a les llengües romàniques. *LinguaMÁTICA*, *1*(1), 13-23. https://linguamatica.com/index.php/linguamatica/article/view/18

Forcada, M. L. (2015). Open-source machine translation technology. En C. Sin-wai (Ed.), *The Routledge Encyclopedia of Translation Technology*. Routledge.

González, J., Lagarda, A. L., Navarro, J. R., Eliodoro, L., Giménez, A., Casacuberta, F., de Val, J. M. y Fabregat, F. (2006) SisHiTra: A Spanish-to-Catalan hybrid machine translation system. *Strategies for Developing Machine Translation for Minority Languages*, 69-73. https://www.isca-speech.org/archive_open/saltmil/SALTMIL2006_procs2006.pdf

Hutchins, W. J. (2001). Machine translation over fifty years. *Histoire Épistémologie Langage*, *23*(1), 7-31. https://doi.org/10.3406/hel.2001.2815

Hutchins, W. J. (2015). Machine translation: history of research and applications. En C. Sin-wai (Ed.), *The Routledge Encyclopedia of Translation Technology*. Routledge.

Hutchins, W. J. y Somers, H. L. (1992). *An Introduction to Machine Translation*. Academic Press.

Khanna, T., Washington, J. N., Tyers, F. M., Bayatli, S., Swanson, D. G., Pirinen, T. A., Tang, I. y Alòs i Font, H. (2021) Recent advances in Apertium, a free/open-source rule-based machine translation platform for low-resource languages. *Machine Translation*, *35*, 475-502. https://doi.org/10.1007/s10590-021-09260-6

Kuusi, P., Kolehmainen, L. y Riionheimo, H. (2017). Introduction: Multiple Roles of Translation in the Context of Minority Languages and Revitalisation. *trans-kom*, *10*(2), 138-163. http://www.trans-kom.eu/bd10nr02/trans-kom_10_02_02_Kuusi_Kolehmainen_Riionheimo_Introduction.20171206.pdf

Maučec, M. S. y Donaj, G. (2019). Machine Translation and the Evaluation of Its Quality. En A. Sadollah, y T. S. Sinha (Eds.), *Recent Trends in Computational Intelligence*. IntechOpen. https://doi.org/10.5772/intechopen.89063

META-NET. (2 de diciembre de 2013). *Mikel Forcada: Free/Open-Source Machine Translation: The Apertium Platform. Translingual Europe 2010* [Archivo de Vídeo]. YouTube. https://www.youtube.com/watch?v=QUjxagyYJKg

Parra Escartín, C. (2018). ¿Cómo ha evolucionado la traducción automática en los últimos años? *La Linterna Del Traductor*, (16), 20-27. http://www.lalinternadeltraductor.org/pdf/lalinterna_n16.pdf

Popovic, M., Ney, H., Gispert, A., Marino, J. B., Gupta, D., Federico, M., Lambert, P. y Banchs, R. (2006). Morpho-syntactic Information for Automatic Error of Statistical Machine Translation Output. *NAACL 2006 Workshop on Statistical Machine Translation*, 1-6. https://aclanthology.org/W06-3101.pdf

Qun, L. y Xiaojun, Z. (2015). Machine translation: general. En C. Sin-wai (Ed.), *The Routledge Encyclopedia of Translation Technology*. Routledge.

Sánchez Ramos, M. del M. y Rico Pérez, C. (2020). *Traducción automática: conceptos clave, procesos de evaluación y técnicas de posedición*. Editorial Comares.

Sin-wai, C. (2015). Computer-aided translation: major concepts. *The Routledge Encyclopedia of Translation Technology*. Autoedición.

Way, A. (2018). Quality Expectations of Machine Translation. En J. Moorkens, S. Castilho, F. Gaspari, y S. Doherty (Eds.), Transla*tion Quality Assessment: From Principles to Practice* (Vol. 1, pp. 157-178). Springer. https://doi.org/10.1007/978-3-319-91241-7_8

Wong Tak-ming, B. y Webster, J. J. (2015). Example-based machine translation. En C. Sin-wai (Ed.), *The Routledge Encyclopedia of Translation Technology*. Routledge.

Yakovyna, V. y Masyukevych, V. (2013). Review and analysis of machine translation quality evaluation metrics. *Series of Computer Sciences and Information Technologies*, *771*. https://science.lpnu.ua/sites/default/files/journal-paper/2017/may/2369/yakovinamasyukevicheng.pdf

Yang, L. y Min, Z. (2015). Statistical machine translation. En C. Sin-wai (Ed.), *The Routledge Encyclopedia of Translation Technology*. Routledge.