Programari per a la personalització de motors de TA. Anàlisi de productes

Treball de Fi de Màster

Autores: Núria Paillissé Vilanova i Coral Suero González
Director: Adrià Martín Mor
Màster en Tradumàtica: Tecnologies de la Traducció
Facultat de Traducció i Interpretació, UAB

Dades del TFM

Títol (ca): *Programari per a la personalització del motors de TA. Anàlisi de productes*

Título (es): *Software para la personalización de motores de TA. Análisis de productos*

Title (en): *Software for engine customization of MT. Product analysis*

Autores: Núria Paillissé Vilanova i Coral Suero González

Tutor: Adrià Martín-Mor

Centre: Facultat de Traducció i Interpretació

Estudis: Màster en Tradumàtica: Tecnologies de la Traducció

Curs acadèmic: 2017-2018

Paraules clau (ca)

Traducció automàtica, programari per a la personalització de motors de TA, català francès, xinès

Resum (ca)

L'objectiu d'aquest Treball de Fi de Màster és analitzar programari que permet la personalització de motors de TA. Amb aquest propòsit s'han escollit 6 eines diferents (Machine Translation Training Tool, ModernMT, MTradumàtica, LetsMT, KantanMT i Microsoft Translation Hub) i s'ha investigat la seva instal·lació i el seu entrenament. A més, es proporciona una explicació teòrica sobre els sistemes de TA, la qualitat en TA i el seu estat de la qüestió. De la mateixa manera, es descriuen amb precisió els recursos que s'han utilitzat durant el treball, les característiques principals de les eines, la creació dels dos corpus i la instal·lació i l'entrenament del programari. Amb els motors ja entrenats s'ha preparat un resum de les característiques més importants de cada eina i una anàlisi de la qualitat de la TA. A banda d'aquests resultats, aquest treball subratlla els mètodes relatius a la instal·lació i a l'entrenament dels programes.

Paraules clau (es)

Traducción automàtica, software para la personalización de motores de TA, catalán, francés, chino

Resum (es)

El objetivo de este Trabajo de Fin de Máster es analizar software que permite la personalización de motores de TA. Con este propósito se han escogido 6 herramientas distintas (Machine Translation Training Tool, ModernMT, MTradumàtica, LetsMT, KantanMT y Microsoft Translation Hub) y se ha indagado acerca de su instalación y de su entrenamiento. Además, se han creado dos corpus (chino-catalán y francés-catalán) para entrenar los motores de estos programes con combinaciones lingüísticas concretas. Así, se proporciona una explicación teórica acerca de los sistemas de TA, de la calidad en TA y de su estado de la cuestión. De la misma manera, se describen con precisión los recursos que se han utilizado a lo largo de este trabajo, las características principales de las herramientas, la creación de los dos corpus y la instalación y el entrenamiento del software. Con los motores ya entrenados se ha preparado un resumen de las características más relevantes de cada herramienta y un análisis de la calidad de la TA. A parte de estos resultados, este trabajo subraya los métodos relativos a la instalación y al entrenamiento de los programas.

Keywords (en)

Machine Translation, engine customization of MT, Catalan, French, Chinese

Abstract (en)

The aim of this Master's Degree Dissertation is analyzing software that allows engine customization of Machine Translation. With this purpose 6 different tools (Machine Translation Training Tool, ModernMT, MTradumàtica, LetsMT, KantanMT and Microsoft Translation Hub) have been chosen and their installation and their engine training have been explored. Moreover, two corpus (chinese-catalan and french-catalan) have been created in order to train the engines with specific linguistic combinations. A theoretical explanation about MT systems, quality in MT and its state-of-the-art is provided. Likewise, the resources used during this dissertation, the tools' main features, every single step of the creation of both corpus and the installation and the training of the software is described. Once the engines have been trained, a summary of the most significant features of each tool and the analysis of the quality of the MT are given. Apart from these results, this dissertation highlights the methods of the installation and the training.

Índex

Índex d'il·lustracions

Índex de scripts

Índex de taules

1.Introducció

Creiem que elaborar un Treball de Fi de Màster com aquest no només ha suposat un repte des del punt de vista acadèmic, també ho ha fet des del punt de vista personal. El fet d'haver hagut de treballar en parella i, totes les complicacions que això suposa, ens ha obligat a fer un treball apte per als lectors, però, sobretot, apte per a la persona amb qui hem treballat.

Des del punt de vista de les nostres motivacions, ens cridava l'atenció el fet d'analitzar programes de Traducció Automàtica (TA). En primer lloc, per l'impacte que aquestes tecnologies tenen (i tindran) en el flux dels projectes de traducció, dades que encara no coneixem amb seguretat, però que afecten de manera directa la nostra professió. En segon, perquè volíem saber què ofereix el mercat, en quin grau podem personalitzar aquests programes i, sobretot, quines diferències essencials tenen entre ells. És a dir, volíem conèixer com en podríem treure profit en un futur i, sense una anàlisi prèvia de les eines, això és una tasca impossible.

La segona gran motivació era treballar amb llengües que dominem i que fem servir al nostre entorn professional. Les raons principals eren, deixant de banda l'anàlisi dels resultats de l'entrenament, saber amb seguretat la quantitat i qualitat dels corpus existents al web avui dia, per, com hem comentat al paràgraf anterior, saber si amb les eines existents, ens podríem beneficiar de l'ús d'aquests programes.

Finalment, volem remarcar que aquest treball ha sigut l'excusa perfecta per posar en pràctica molts dels coneixements que hem anat adquirint al llarg del curs. Així, per exemple hem hagut de treballar amb la codificació de fitxers, l'ús d'expressions regulars, la transformació de fitxers, l'ús de sistemes operatius gràcies a màquines virtuals, conceptes teòrics sobre Traducció Automàtica Estadística (TAE), les mètriques d'avaluació automàtica, etc.

2.Objectius: Hipòtesis i preguntes

Aquest treball té dos objectius principals: per una banda, analitzar sis programes de TA que permeten la personalització dels seus motors (Machine Translation Training Tool, ModernMT, MTradumàtica, LetsMT, KantanMT i Microsoft Translation Hub). Per fer-ho, es volen classificar segons les seves característiques i se'n vol estimar el rendiment gràcies a l'avaluació automàtica dels resultats de les seves traduccions, ja que, aquest treball no pretén classificar amb precisió els diferents errors de cada programa.

Per una altra banda, es pretén treballar amb dos corpus de combinacions lingüístiques del nostre entorn professional (xinès-català i francès-català). Aquestes combinacions lingüístiques tan diferents permeten, a més, comparar els resultats dels productes de traducció entre llengües properes (corpus de francès-català) i llunyanes (xinès-català).

Pel que fa a les hipòtesis, preveiem dificultats amb la cerca i elaboració del corpus xinès-català, bé per manca de recursos, bé per les normes de segmentació d'aquesta llengua. També creiem que la

instal·lació i entrenament de programes que no disposen d'interfície gràfica ens poden generar conflictes, atesa la nostra manca de coneixements en programació. Finalment, i per al propòsit d'aquest treball, preveiem problemes si hi ha motors que es troben restringits a certes combinacions lingüístiques, ja que, com hem exposat, un dels nostres objectius és comparar els resultats de dues combinacions concretes. En aquests casos, els programes es classificaran, però no se n'avaluaran els resultats de la traducció.

3.Marc teòric

En aquest marc teòric volem comprendre els conceptes teòrics bàsics del nostre treball, és a dir, tota la informació sense la qual és impossible entendre el funcionament i l'anàlisi de les eines d'aquest TFM. Encara que cada programari té les seves particularitats, creiem que els continguts d'aquest apartat són comuns i es poden aplicar a tots els mecanismes que analitzem.

Per aquest motiu, descriurem què és la traducció automàtica, els tipus de TA existents avui dia (fent èmfasi en l'estadística i la neuronal, ja que les eines que analitzem utilitzen aquests sistemes) i la qualitat en traducció. També parlarem dels recursos per analitzar-la (gairebé totes les eines que tractem inclouen mecanismes d'avaluació del producte traduït) descriurem alguns dels mètodes per millorar els resultats en TA i resumirem el seu estat de la qüestió.

3.1.Què és la TA?

Si pensem en la definició més global de TA, aquella que podria donar un usuari no especialista, segurament ens referirem al "procés de traducció, mitjançant un sistema informàtic (compost per ordinadors i programes), de textos informatitzats escrits en la llengua origen a textos informatitzats escrits en la llengua meta" (Ginestí i Forcada 2009, p. 43). Aquest mecanisme, però, no només ha de vetllar per aconseguir transmetre el significat d'un text escrit en llengua natural, sinó que ha de "recrear su significado en otra lengua, teniendo en cuenta la inflexión, el registro idiomático y el orden de las palabras" (EU Law and Publications, s.d.). Malauradament, tal com afirma Sánchez (2017), si es fa servir TA, cal "descartar el procesamiento de algunos aspectos de la estructura del discurso".

Un altre punt destacable que cal considerar quan s'utilitza TA és la seva finalitat, que pot ser merament informativa o d'*assimilació* (Ginestí i Forcada, 2009, p. 44), i que, en la indústria, és comparable al que Aranberri (2014, 472) descriu com a *comunicación interna*, o si té un destí publicable, anomenat de *disseminació* segons Ginestí i Forcada (2009, p. 44) o de *comunicación externa* segons Aranberri (2014, 472). Així, la diferència principal entre aquestes dues modalitats de TA no és només l'objectiu de la traducció sinó el seu destinatari final (per exemple, els usuaris d'una empresa que necessiten informació interna o el lector d'un manual d'instruccions d'una rentadora).

3.2.Sistemes de TA

Actualment, existeixen diverses aproximacions de TA i la seva classificació pot variar segons l'autor. Per aquest treball, però, en seguirem la de Casacuberta i Peris (2017, p. 66) que diferencien entre Traducció Automàtica Basada en Regles (TABR), basada en corpus i neuronal. D'aquesta manera, encara que el seu objectiu principal sigui el mateix (aconseguir la millor traducció possible) els mètodes i recursos que utilitzen són totalment diferents.

Pel que fa a la TABR és l'única de totes les existents que funciona gràcies a coneixements purament lingüístics. Encara que el seu desenvolupament és més lent i, per tant, més car, que el d'un sistema estadístic, sembla ser una bona opció per les llengües en què hi ha menys accés a corpus bilingües (Ginestí i Forcada, 2009, p. 48). Aquest tipus de sistemes, a més, són "els que millor qualitat de traducció proporcionen [...] especialment entre llengües properes (Ginestí i Forcada, 2009, p. 48) i es classifiquen, al seu torn, en tres subgrups:

1. **Traducció directa**: van ser els primers sistemes de TABR. Es basaven en grans corpus lèxics monolingües i bilingües que traduïen paraula per paraula de manera literal, ja que, "es pensava que tot era qüestió de tenir grans lèxics amb els quals poder traduir ràpidament les paraules de les frases del text" (Alonso, 2007, p. 26). Així doncs, "la quantitat de coneixement lingüístic (informació morfosintàctica) inclosa en aquests programes és molt limitada" (Alonso, 2007, p. 27) i l'anàlisi que es feia de les oracions era, o bé molt simple (procés conegut com a *shallow parsing*) o bé inexistent.
2. **Sistemes basats en transferència**: es basen en tres fases fonamentals (anàlisi, transferència i generació). Durant la primera s'analitza el text sintàcticament, semànticament i morfològicament. A partir d'aquesta informació, es tria la millor traducció de cada paraula, tot tenint en compte que "aquesta pot portar associada una transformació estructural que s'ha d'aplicar sobre la frase en la llengua de destinació" (Alonso, 2007, p. 28). Finalment, es genera la traducció en la llengua desitjada.
3. **Sistemes d'interlingua**: són la millora dels sistemes anteriors. Desapareix la fase de transferència i es duu a terme una "representació formal del significat de la frase en forma de xarxa semàntica" (Alonso, 2007, p. 29). La traducció que es genera és la que millor representa aquests conceptes semàntics i les relacions que s'estableixen entre ells (Alonso, 2007, p. 29).

   Quant a la TA basada en corpus, a diferència de l'anterior, no es basa en coneixements lingüístics per aconseguir els seus objectius. Aquests sistemes "«aprenen a traduir» (per exemple usant complexos models estadístics) a partir d'enormes corpus de textos bilingües on milions de frases en una llengua s'han alineat amb les seves traduccions en l'altra llengua" (Ginestí i Forcada, 2009, p. 48).

   Encara que n'hi ha de dos tipus, la traducció automàtica basada en exemples (Example-based Machine Translation, EBMT) i la TAE, en aquest treball s'explicarà amb més deteniment la segona, ja que és la que incorporen les eines que analitzem:

1. **Basada en exemples**: tradueix imitant els exemples trobats a corpus bilingües, amb els quals s'entrenen els motors. Per tal de millorar la qualitat de les traduccions, utilitza diccionaris i diccionaris bilingües. Els primers "inclouen oracions d'exemple per als verbs, de les quals es poden extreure les estructures argumentals" mentre que els bilingües "incorporen informació sobre les relacions de significat entre paraules: sinònims, antònims i relacions conceptuals" (Simon, 2017. p. 8). La plataforma TAUS (2016) la compara amb l'aprenentatge per analogia.

2. **Estadística**: tradueix utilitzant "a learning algorithm to a large body of previously translated text, known variously as a parallel corpus, parallel text, bitext, or multitext" (Lopez, 2008, p. 2) a més de corpus monolingües en la llengua de destí.

Cal destacar que, tot i les diferències entre els sistemes basats en regles i els basats en corpus, existeixen versions híbrides o mixtes com ara "sistemes estadístics que incorporen algun tipus de coneixement lingüístic" o "sistemes de regles que incorporen algun tipus de coneixement no lingüístic" (Ginestí i Forcada, 2009, p. 48). Aquest tipus de sistemes es poden enfocar de maneres diverses. Sánchez (2017), proposa exemples molt concrets, com ara enriquir models de traducció estadístics (vegeu l'apartat 3.2.1.1.Entrenament) amb dades generades utilitzant regles.

Finalment, el darrer tipus de traducció automàtica que expliquem en aquest treball és la Traducció Automàtica Neuronal (TAN). Aquest sistema té com a característica principal que "las palabras y las frases son representadas de forma numérica mediante vectores [...]. Este hecho ha permitido el uso de potentes técnicas de aprendizaje automático (ML del inglés "machine learning") como las redes neuronales" (Casacuberta i Peris, 2017, p. 68). Aquestes *xarxes neuronals* es troben relacionades entre si com les d'un cervell humà i gestionen tot el volum d'informació de la llengua d'arribada i de partida. Aquest tipus de TA, però, presenta problemes amb les paraules que no pot reconèixer, és un sistema lent i pot arribar a eliminar mots en la llengua de destí (Wu et al., 2016, p. 2).

3.2.1.Traducció Automàtica Estadística

Com ja hem comentat a l'apartat anterior *(*vegeu l'apartat 3.2.Sistemes de TA), els coneixements lingüístics no es tenen en compte en aquest tipus de mecanismes i és necessària una gran quantitat de corpus monolingües i bilingües segmentats i alineats per aconseguir els resultats desitjats. Això implica que els resultats en aquest tipus de traducció semblen naturals de la llengua d'arribada, ja que es basen en traduccions reals, però, a la vegada, corren el risc de generar traduccions no gaire *fidels* al text original (Sánchez, 2017). Per aquest motiu, Sánchez (2017) ens recomana "prestar atención a la oración en LO, incluso cuando la oración no parece contener errores".

Un altre apunt destacable de la TAE actual és que, diferència dels primers intents de traducció automàtica estadística, en què es traduïa paraula per paraula (pensem en el sistema Candide de IBM), avui dia la TAE utilitza *phrases*, segments bilingües d'una longitud variable. Aquest tret permet que aquest tipus de TA també pugui traduir expressions multiparaula i col·locacions (Sánchez, 2017).

Quant al mètode, "SMT procedure is divided into three important processes: training, tuning and decoding." (Pérez, 2016, p. 9). El *training* o *entrenament* és el procés mitjançant el qual "translation correspondences are inferred between the two languages by analysing co-occurrences of words and segments" (Martín-Mor, 2016, p. 29). Així, quan es tradueix un text, el motor sempre escollirà l'opció de traducció més probable segons els corpus amb què s'hagi entrenat. Per aquesta raó, aquest tipus de traducció "fails when it is presented texts that are not similar to material in the training corpora" (TAUS, 2016).

Pel que fa al *tuning* o (*optimització* en català) consisteix en un canvi del pes dels models de traducció obtinguts a la fase anterior (vegeu l'apartat 3.2.1.1.Entrenament) per tal de millorar els resultats del motor que s'ha entrenat. Martín-Mor i Peña-Irles (2017, p. 47) descriuen aquest procés de la manera següent:

> L'optimització (o tuning) és un procés que determina automàticament els valors òptims d'una sèrie de paràmetres per tal que el motor generi "the best possible translations" (Koehn, 2016, p. 12). L'optimització consisteix en la traducció automàtica de milers de frases d'un subconjunt dels models (anomenat development o tuning set), la comparació amb les traduccions humanes de referència i l'ajustament automàtic dels valors de cada paràmetre per tal de millorar la qualitat del motor, mesurada mitjançant mètriques automàtiques com ara BLEU (Papineni et al., 2002).

El darrer pas del flux de treball als sistemes TAE és la fase coneguda com a *decoding*, que "is in charge of the last link of the chain: the translation phase" (Pérez, 2016, p. 22) i l'objectiu de la qual és, donades totes les possibles traduccions, trobar la més probable segons el motor "by searching over the space of all possible translations, scoring each translation with the different models in order to compute the final score and finally choosing the translation that has obtained the best overall score during the process" (Pérez, 2016, p. 22).

3.2.1.1.Entrenament

L'objectiu principal d'aquesta fase és obtenir, a partir dels corpus, els models estadístics que s'utilitzaran durant la fase de traducció. Aquests models són mecanismes probabilístics que intenten modelar un aspecte de la traducció (Sánchez, 2017) i que puntuen, tenint en compte aspectes concrets, les diferents opcions de traducció existents. Finalment, la hipòtesi guanyadora és la que ha aconseguit, d'entre totes les notes dels models, la valoració més alta.

A continuació, s'explicarà i s'exemplificarà de manera detallada en què consisteix cadascun d'aquests models i els seus objectius principals.

Model(s) de traducció

L'objectiu principal d'aquests models és calcular la probabilitat de traducció dels segments (*phrases*) bilingües. Tot i que a la literatura de referència es tracten aquests models en singular, també es destaca la necessitat de "dos models, un per a cada direcció de traducció, atès que la traducció no és simètrica" (Peña-Irles i Martín-Mor, 2017, p. 19). Per aquest motiu, en aquest treball es tractaran aquests models en plural, com ja han fet altres autors de treballs similars (pensem en Pérez, 2016).

Per aconseguir aquesta probabilitat, però, cal, en primer lloc, alinear les paraules en llengua origen i en llengua destí utilitzant models de traducció basats en paraules. Aquests permeten descobrir l'estructura d'alineament entre paraules i obtenir diccionaris bilingües probabilístics (Sánchez, 2017) sense necessitat de cap coneixement lingüístic. Hearne i Way (2011) donen una bona definició d'aquesta tasca en concret:

> The SMT approach considers all possible alignments between each sentence pair and works out which ones are the most likely. In order to work out the probability of a particular alignment, several factors are taken into consideration; the most important is the probability that the aligned words correspond to each other in meaning, at least in this specific context (p. 9).

A continuació, el model extreu els parells de segments bilingües tot tenint en compte les alineacions de paraules que acaba de fer:

Taula 1. Exemple de parells de segments bilingües xinès-català

我	Jo
喜欢	Agradar
巧克力	xocolata
吃	Menjar
面条	fideus
下星期	La setmana que ve
去	Anar
打算	planejar
北京	Pequín
哥哥	Germà gran
上海	Xangai
我喜欢巧克力	M’agrada la xocolata
我吃面条	Menjo fideus
我下星期打算去北京	La setmana que ve planejo anar a Pequín
我哥哥去上海	El meu germà va a Xangai

Finalment, el model calcula la probabilitat dels segments en totes dues direccions. Aquest càlcul és molt simple: es divideix el nombre de vegades que es detecten aquests segments connectats pel nombre de vegades que apareixen a la llengua d'origen. Com més proper sigui el resultat a 1, més probable serà aquesta traducció.

Pérez (2016) destaca la importància de la coherència durant la fase de l'alineació. Així, totes les paraules han d'alinear-se amb una paraula com a mínim i "with more complex sentences where a word in the source sentence is aligned to two or more words in the target language or vice versa, the consistency concept remains the same" (p. 12).

Model (s) per a la ponderació lèxica

Aquests models utilitzen, com el model anterior, un diccionari bilingüe probabilístic. En aquest cas, però, el seu objectiu és determinar la fiabilitat de segments infreqüents (Sánchez, 2017), ja que, els models anteriors, proporcionen probabilitats molt altes als segments que s'han vist poques vegades als corpus (i que, per tant, cal dividir entre menys vegades). D'aquesta manera, un segment que només s'ha vist tres cops al corpus obtindrà una probabilitat molt més alta que un que s'ha vist set cops (encara que, realment, sigui menys freqüent). L'operació exemplificada que realitzen aquests models seria la següent:

> lex (Menjo a Barcelona | Je mange à Barcelone) = (w (Menjo | Je) + (w (Menjo | mange)) / 2 x w (a | à) x w (Barcelona | Barcelona)

Així, per conèixer la fiabilitat de l'oració "Menjo a Barcelona" | "Je mange à Barcelone" s'alinearien les oracions paraula per paraula. *Menjo* s'hauria alineat amb *Je* i *mange*, és a dir, caldria sumar les vegades que *Menjo* s'hauria alineat amb *Je* i que *Menjo* s'hauria alineat amb *mange* al corpus i dividir-ho per dos (el nombre d'alineacions a la llengua destí). A continuació, caldria multiplicar aquest resultat pel de les operacions realitzades amb à | a i Barcelona | Barcelone.

Cal afegir que, de la mateixa manera que amb el cas anterior (vegeu l'apartat Model(s) de traducció), la traducció no és simètrica i cal fer aquestes operacions en les dues direccions de la traducció.

Model de llengua

El propòsit d'aquest model és calcular la fiabilitat d'una frase en llengua d'arribada, ja que, com que el propòsit dels models anteriors és trobar la millor traducció possible d'un segment, a vegades, es pot produir una manca de fluïdesa en la llengua de destí (Pérez, 2016, p. 15). Per aconseguir-ho, calcula la freqüència de *n-grames* (conjunt de paraules de longitud variable) en una gran quantitat de corpus monolingües. És a dir, es calcula que probable és que una paraula aparegui darrere d'una altra en un conjunt concret de paraules.

Aquest càlcul, com els casos anteriors, resulta força simple i, normalment, es realitza amb cinc *n-grames*. Així, si volguéssim saber que fluida és l'oració *El MWC ha sigut un èxit* utilitzant cinc n-

grames hauríem de dividir el nombre de vegades que apareix al corpus monolingüe *ha* pel nombre de vegades que apareix al corpus *El MWC* i multiplicar-ho pel resultat de la següent divisió (*sigut | El MWC ha*). Caldria fer-ho successivament fins arribar a *un | El MWC ha estat*. A partir d'aquest n-grama (el cinquè), la divisió seria la següent *èxit | MWC ha estat un*. L'operació, doncs, seria la següent:

> *p* (El MWC ha sigut un èxit) = *p* (*El*) x *p* (*MWC*) x *p* (*ha | El MWC*) x *p* (*sigut | El MWC ha*) x *p* (*un | El MWC ha sigut) x p* (è*xit | MWC ha sigut un*)

Aquest model presenta, però, alguns problemes. Per exemple, "it cannot discriminate between well-formed sentences and those with incorrect word order" (Hearne i Way, 2011, p. 5). A més, segons aquests autors el model sempre assigna probabilitats més altes a les oracions més curtes, ja que, "the fewer words there are, the fewer probabilities need to be multiplied together" (p. 5).

Models de reordenament: basat en distàncies i lexicalitzat

Aquests dos models condicionen l'ordre dels segments en llengua de destí. El primer calcula el nombre de paraules que s'han omès durant la traducció quan hi ha hagut moviment de posició de paraules als segments traduïts. El segon, però, aprèn de les alineacions que s'han produït als models de traducció i puntua "l'ordre en què les paraules alineades apareixen en el text de destinació" (Peña-Irles i Martín-Mor, 2017, p. 21). Això vol dir que el primer no té cap implicació lingüística és, a dir, "no tiene en cuenta los segmentos concretos que se reordenan" (Sánchez, 2017).

Models de penalització: paraules i segments

El propòsit d'aquests models és assegurar una correcta llargària dels segments en llengua d'arribada. Per exemple, el de paraules mesura el nombre de mots als segments traduïts i s'utilitza per evitar traduccions massa curtes *(*vegeu l'apartat Model de llengua).

D'altra banda, el de segments s'utilitza per "incentivar el uso de segmentos largos (menos pares de segmentos bilingües)" (Sánchez, 2017).

3.2.1.2. Pes dels models i *optimització*

En total, durant la fase d'entrenament dels corpus es poden utilitzar de 8 a 15 models diferents. Pérez (2016) ens ofereix al seu treball un exemple d'entrenament amb 14 models (un de llengua, dos de traducció, dos de ponderació lèxica, set de reordenament i dos de penalització).

Imatge 1. Models i puntuació de Pérez (2016, p. 34)

m	λ_m	h_m
1	0.124094	log (P (S | T))
2	**0.039016**	**log (P (T | S))**
3	**0.194958**	**log (P(T))**
4	0.064472	log P(monotone backward | S, T)
5	-0.0409665	log P(monotone forward | S, T)
6	0.0285894	log P(swap backward | S, T)
7	0.0148673	log P(swap forward | S, T)
8	0.126	log P(discont. backward | S, T)
9	-0.0227287	log P(discont. forward | S, T)
10	-0.0128585	distortion (S | T)
11	0.00110911	log (lex(S | T))
12	0.0277224	log (lex(T | S))
13	-0.286949	wordcount (T)
14	0.0155945	phrasecount (S, T)

Com es pot veure a la imatge (l'apartat destacat en color), cada model té un pes (rellevància) diferent, ja que "it has been shown that not all models are equally important and, consequently, each one is assigned a different weight" (Pérez, 2016, p. 21). Aquests pesos, però, es poden modificar per millorar la qualitat de les traduccions (procés conegut com a *optimització*). Per fer-ho, s'utilitza una mètrica d'avaluació automàtica (vegeu l'apartat 3.3.Qualitat i TA) i "a small amount of parallel data, separate from the training data" (Koehn, 2018, p. 39) anomenat *tuning set*.

Amb aquests dos recursos, s'utilitza un sistema automàtic iteratiu de *traducció + avaluació+ canvi als pesos* que permet constituir el sistema més productiu possible.Tot i això, tal com assenyala Pérez (2016, p. 22) aquest procediment implica una gran quantitat de temps i recursos i s'utilitzen mètodes heurístics per agilitzar el procés.

3.2.1.3. *Decoding*

Preparació dels textos

La darrera fase del procés de TAE és la mateixa traducció. Després d'haver ajustat el pes dels models durant la fase anterior, ara només cal preparar els textos i traduir-los amb aquests paràmetres. Les eines que utilitzen "Moses, a statistical machine translation system that allows you to automatically train translation models for any language pair" (Koehn, 2018, p. 11), però, ja incorporen les funcions de neteja, segmentació i *truecasing* necessàries per començar a traduir:

> MTradumàtica, com Moses (Koehn, 2016: 36), duu a terme els processos de segmentació, truecasing i neteja dels corpus. Segmentar vol dir separar amb espais les paraules dels signes de puntuació. En altres paraules, aïllar la puntuació permet incrementar les probabilitats d'obtenir coincidències amb els futurs textos que es traduiran automàticament. El procés de truecasing, en canvi, consisteix a determinar la caixa més probable de cada paraula, majúscules o minúscules. La neteja consisteix en la

> supressió de les frases llargues i mal alineades dels corpus amb l'objectiu de minimitzar els problemes en la fase d'entrenament. (Martín-Mor, 2017, p. 103).

Cal tenir en compte que les accions de neteja, segmentació i *truecasing* s'han de dur a terme en sentit contrari, és a dir, un cop realitzada la traducció, cal retornar els textos al seu format original. Així, tant de manera prèvia com posterior a la traducció, aquestes accions es duen a terme de forma automàtica en aquest tipus d'eines.

Traducció

Com ja s'ha vist als apartats anteriors, durant la traducció es busca la hipòtesi de traducció, d'entre totes les possibles, que té una puntuació més alta segons els diferents models. Pérez (2016, p. 23), però, remarca que, per dos motius ben diferents, la traducció que té la millor puntuació no és necessàriament la que el traductor humà consideraria la *millor*:

> Why there is no guarantee that the best translation possible is found? Koehn (2010: 155-156) explains that there are two types of errors that prevent this. One is the search error, for which SMT systems are not able to explore the whole search space in order to find the best translation; and the other is the model error, where the translation with highest probability according to a specific model might not be a good translation at all.

3.2.2.Traducció neuronal

Aquest sistema de TA, com la TAE, utilitza enormes quantitats de parells de segments bilingües (centenars de milers o milions d'unitats de traducció) per a dur a terme les traduccions. En aquest aspecte, ambdós tipus de traducció automàtica són similars. Forcada (2017, p. 2) destaca, però, que la TAN fa servir una aproximació computacional completament diferent: les xarxes neuronals artificials.

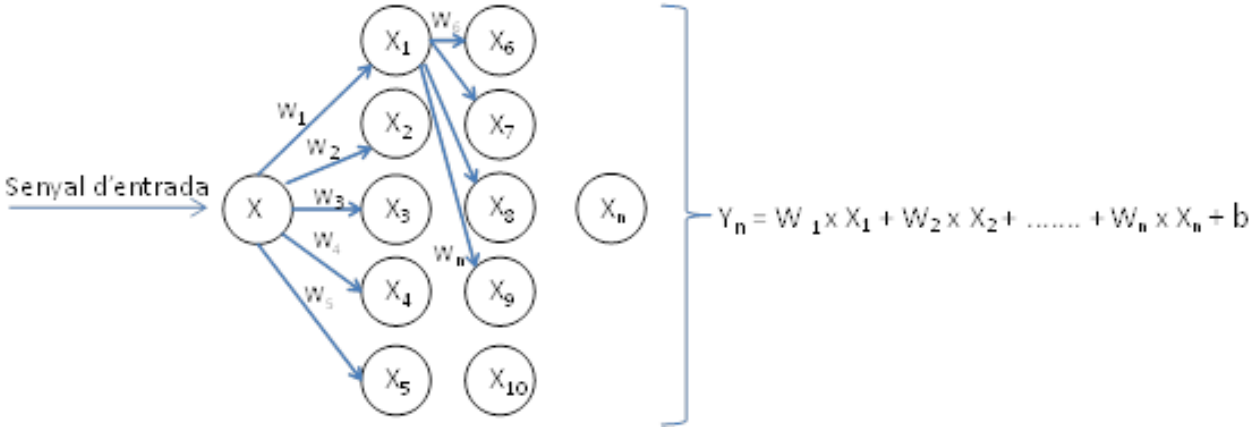
Aquestes xarxes, que Sánchez (2017) defineix com "unidades interconectadas que se asemejan a las neuronas del cerebro" s'assemblen a l'organ humà tant en la manera i el grau en què s'activen, com en la informació que transmeten en resposta a estímuls que reben d'altres unitats i en la força d'aquestes connexions que permeten canalitzar els impulsos (Forcada, 2017,p.2).

3.2.2.1.Funcionament bàsic de les unitats neuronals artificials

Les unitats neuronals que es fan servir a la TAN treballen en dos passos o etapes. Per entendre el seu funcionament ens hem d'imaginar una neurona o node, *x*, connectada a *N* neurones o nodes (x_n) i cadascuna d'aquestes a altres nodes formant capes successives (vegeu imatge 2, elaboració pròpia). En el primer pas, les activacions o senyals que rep cada node de les de la capa anterior es multipliquen per un valor (w_n), que representa el tipus (positiu o negatiu) i la intensitat del senyal que rep cada unitat i, posteriorment, se sumen (Forcada, 2017, p.3). Al segon pas, el valor de *y* trobat per a cada

node es converteix, aplicant una funció matemàtica anomenada d'*activació*, en el senyal que s'enviarà al següent node. En les arquitectures neuronals l'activació d'una única unitat "do not make sense by themselves, but rather when grouped with the activations of the other neurons" (Forcada, 2017, p.3).

Imatge 2. Relació entre els nodes (elaboració pròpia)



Aquesta imatge conté un esquema simplificat del funcionament de les xarxes neuronals. Al model, encara que el nombre de capes pot ser molt gran, se'n representen només tres: la que rep el senyal d'entrada, la capa intermèdia i la posterior. Cada node d'una capa suma els senyals dels nodes anteriors ponderat segons la intensitat i tipus de senyal (w). Per tant, per a cada unitat o node s'acaba trobant un valor de *y* que es converteix mitjançant una funció matemàtica en el senyal que rebran els nodes de la capa següent. Al final s'ha d'arribar a un únic node que ha rebut informació de tots els anteriors i del qual ha de sortir el senyal resultant.

3.2.2.2.Ús de les xarxes neuronals en la traducció automàtica neuronal

Per a la traducció de textos d'una llengua d'origen a una llengua de destí els senyals d'entrada de la xarxa són les paraules o frases de la primera i, el senyal de sortida, una traducció el més fidel i correcta possible. No obstant això, el context afecta les traduccions de paraules i frases. La traducció neuronal situa cada paraula en un punt d'un espai *N-dimensional* on les paraules es representen en el seu context i, per tant, inclou les relacions entre elles. Aquestes paraules es representen amb vectors que contenen, al seu torn, un conjunt de valors de la xarxa neuronal corresponents als diferents valors de *w* (vegeu l'apartat 3.2.2.1.Funcionament bàsic de les unitats neuronals artificials) i que representarà l'estat d'excitació de cada node. Cada punt és un significat de la paraula en l'oració corresponent. Així, les paraules que s'assemblen entre elles es situen properes en aquest espai, en què, en un principi, hi han d'enquibir totes les possibilitats de traducció. (Forcada, 2017, p. 6-7).

Per aconseguir una bona traducció les xarxes neuronals s'han d'entrenar, és a dir, la xarxa neuronal ha de ser capaç de seleccionar els pesos i força dels senyals de cada node per obtenir els resultats més adients. Això requereix "large training corpora, typically as large as those used in good old SMT" (Forcada, 2017, p.5) i implica, des d'un punt de vista computacional, una elevada potència de computació (és a dir, ordinadors amb processadors avançats). Durant el procés d'entrenament, els

pesos (valors de *w*) es van modificant de manera que els valors que descriuen com de lluny els senyals de sortida o traduccions es troben del correcte es facin el més petits possible. Per a cada possible paraula en una frase existeix una probabilitat més o menys elevada de què sigui la correcta en la llengua de destí. El sistema s'entrena de tal manera que assigna la màxima probabilitat a la frase en el seu conjunt considerant totes les probabilitats de les parelles de traducció possibles entre la frase en llengua d'origen i llengua de destí. Forcada (2017) afirma el següent sobre aquest tipus de sistema:

> Most NMT systems are built and trained in such a way that they resemble a text completion device (analogous to the word prediction feature of smartphone keyboards) which is informed by a representation of the source sentence, or, more specifically, by representations of each of the words of the source sentence in their context, built by the encoder part of the system. As a text completion device, a part of the system called the decoder provides, at each position of the target sentence being built, and for every possible word in the target vocabulary, the likelihood that the word is a continuation of what has already been produced. The best translation is usually built by picking the most likely word at each position (p.5).

Relacionat amb la completació de textos, sovint les xarxes neuronals aprenen de textos monolingües a reproduir una determinada paraula en un context específic a partir de poques paraules que poden ser presents a la seva esquerra i a la seva dreta. O a la inversa, aprenen a predir quines paraules poden ser presents a esquerra i dreta d'una paraula determinada. Aquests processos s'anomenen incrustacions o insercions *(embedddings*), centrals, en el primer cas exposat, o pre- i post-posades, en el segon cas. Les incrustacions permeten que paraules semànticament similars presentin representacions similars en forma de vectors la qual cosa permet fer combinacions *aritmètiques* de vectors que permeten trobar la paraula correcte en el text a traduir (Forcada, 2017, p. 6).

3.3.Qualitat i TA

Com hem vist anteriorment (vegeu l'apartat 3.1.Què és la TA?) els diferents objectius amb què s'utilitza la traducció automàtica exigeixen diferents tipus de resultats i, per tant, es necessiten *qualitats* diferents. És a dir, en el cas de la TA d'assimilació "els errors en la traducció no són tan importants si s'aconsegueix transmetre el sentit general del text" (Ginestí i Forcada, 2009, p. 44). En canvi, en el cas de la traducció de disseminació, "el text traduït automàticament l'ha de revisar i corregir, o com se sol dir, posteditar, una persona especialitzada" (Ginestí i Forcada, 2009, p. 44).

En primer lloc, però, cal definir què és la qualitat en TA. Segons Görög (2014, p. 444) "quality is when the user or customer is satisfied". És a dir, una traducció de qualitat compleix amb els requisits marcats per un client o usuari. Per aquesta mateixa raó, cal anar amb molt de compte sobre com s'avalua la qualitat d'una traducció, ja que "there is no one-size-fits-all approach to translation quality evaluation (QE)" (Görög, 2014, p. 444). El principal problema, segons O'Brien (2012, p. 55) és el següent:

> Little consideration was given to multiple variables such as content type, communicative function, end user requirements, context, perishability, or mode of translation generation (whether the translation is

> created by a qualified human translator, unqualified volunteer, machine translation system or a combination of these).

Tot i això, en l'àmbit de la traducció automàtica *(*vegeu l'apartat 3.3.1.Mètodes d'avaluació automàtica de TA), s'han desenvolupat diversos mètodes o eines que permeten una avaluació, més ràpida i barata que la humana, d'aspectes concrets del text. Així, per exemple, es pot avaluar, si els canvis de pes als models dels motors de traducció estadístics són positius per a la traducció final o si val la pena fer servir TA i postedició, per a una traducció determinada.

A més, per suplir una necessitat *avaluadora*, que, a més, tingui en compte el context de la traducció, la indústria ha començat a oferir recursos que permeten una gran personalització segons el tipus d'avaluació (pensem en el Dynamic Quality Framework, de TAUS). També s'han desenvolupat tècniques per a traductors i posteditors que permeten identificar i classificar els tipus d'errors en TA (com ara el Multidimensional Quality Metrics) i, per tant, posteditar segons les nostres necessitats (vegeu l'apartat 3.3.2.Millorant la qualitat en TA).

3.3.1.Mètodes d'avaluació automàtica de TA

Existeixen molts mètodes d'avaluació automàtica de TA i, segons l'autor, es poden classificar en grups molt diversos. En aquest TFM, però, utilitzarem la classificació que proposa Babych (2014) de recursos *reference proximity* o *performance based-methods* l'objectiu principal dels quals és "to compute numerical scores, which characterize the 'quality', or the level of performance of specific Machine Translation systems" (Babych, 2014, p. 465).

3.3.1.1.Reference proximity

El primer grup compren tots els mètodes que comparen el producte de TA amb una *gold-standard human reference* (una traducció humana de referència que no s'ha utilitzat durant la fase d'entrenament de motors amb TAE). Aquestes tècniques calculen la *distància* entre els dos productes i, com més s'aproximin els resultats de la TA a la referència, millor puntuació rep aquesta traducció.

Alguns exemples de mètodes coneguts que pertanyen a aquest grup i que, a més, incorporen algunes de les eines que analitzem, són aquests:

1. **WER (Word Error Rate)**: es considera massa simple per avaluar correctament la TA. Calcula el nombre mínim de paraules inserides, esborrades i substituïdes per transformar una frase d'una llengua a una altra. El problema és que penalitza el reordenament de paraules a les frases quan, en molts casos, això no afecta negativament les traduccions. (Babych, 2014, p. 466).
2. **TER (Translation Error Rate)**: és la millora del sistema anterior, ja que no penalitza el moviment de paraules del producte de TA. Calcula la divisió entre el nombre d'edicions (paraules inserides, esborrades o moviments de seqüències de paraules) i la mitjana de

paraules per frase de les traduccions de referència. Com més proper sigui el resultat a 0, millor valorada serà la traducció.

3. **BLEU (Bilingual Evaluation Understudy)**: calcula el solapament de *n-grames* entre el resultat de TA i la traducció de referència. Ho fa dividint el nombre de paraules comunes de la TA i la traducció de referència pel nombre de paraules de la traducció de referència. Com més proper sigui el resultat a 1, millor valorada serà la TA.
4. **METEOR:** és una modificació de la mètrica BLEU. A més del càlcul de diferència entre la quantitat de paraules entre frases de TA i de referència, inclou el càlcul del *recall*, "the score for avoiding 'under-generation': the words which are in the reference, but not in MT ouput" Babych (2014, p.466). D'aquesta manera, es penalitzen les paraules que s'han vist a les traduccions de referència però que no apareixen al producte de la TA.

Un apunt que volem destacar és que totes aquestes mètriques funcionen a nivell de frase i d'alineació de paraules o n-grames. És a dir, s'avalua la similitud entre frases (de la mateixa manera que la TA tradueix a nivell de frase, com ja comentàvem a l'apartat 3.1.Què és la TA?) i tot i que creiem que, com afirma Babych (2014, p. 465), el món de l'avaluació automàtica de TA ha crescut en paral·lel al de la TA, encara cal millorar i fer recerca perquè aquest tipus d'avaluació tingui en compte el conjunt del text.

3.3.1.2.Performance based-methods

Aquest segon grup no es basa en les comparacions amb traduccions de referència. La seva idea principal és mesurar "how well someone can carry out a task on the basis of a degraded MT output. Different quantitative measures of performance for the task are taken to characterize the quality, or usability of MT output." (Babych, 2014, p. 467). Per exemple, un mètode automàtic que permetés saber si és fàcil, o no, entendre les idees principals d'un text.

El principal inconvenient d'aquestes eines és que es troben restringides a un tipus de text concret, normalment instructiu i, per aquest motiu, el programari que analitzem no incorpora aquest tipus de mètodes.

3.3.2.Millorant la qualitat en TA

Aquest breu apartat descriu alguns dels recursos que els usuaris de TA utilitzen amb el propòsit de millorar-ne els resultats. Són els següents:

3.3.2.1.La preedició de textos

Encara que no sembla ser un recurs implantat al flux de treball habitual (la manca d'articles acadèmics i de recerca n'és la conseqüència) la preedició de textos abans de fer servir TA és una de les opcions disponibles per millorar-ne els resultats, ja que permet evitar alguns problemes habituals al producte (TAUS, 2017). Amb aquesta filosofia, també es podria fer servir TA amb textos escrits en llenguatge

controlat lexicament i gramaticalment (Polo, 2012, p. 195) que facilitin "el procesamiento del lenguaje natural para aplicaciones como la traducción automàtica y la recuperación de información".

3.3.2.2.La postedició de TA

La postedició és un dels recursos existents per millorar els resultats de la TA. Yuste (2012, p. 158) el defineix de la manera següent:

> La posedición (PE) es una actividad de edición y corrección lingüística vinculada a la traducción automàtica (MT, en sus siglas inglesas). En términos generales, se requiere que la persona encargada de realizar la tarea de PE realice una serie de modificaciones en el orden de los elementos de la frase o ciertas mejoras lingüísticas de modo tal que el texto que devuelva tenga la misma calidad de traducción que si se hubiera traducido de manera exclusivamente humana desde el principio

El tipus de postedició que es realitza, però, depèn de l'objectiu final de la traducció (vegeu l'apartat 3.1.Què és la TA?) i, per tant, no totes les revisions han de ser iguals i han de tenir en compte els mateixos tipus d'errors:

> La posedición puede aplicarse en distintos grados y es precisamente esta flexibilidad la que dota a la TA de capacidad de adaptación a las diferentes exigencias del mercado. Originariamente se distinguen la posedición parcial (light post-editing) y la posedición completa (full post-editing). La primera consiste en realizar los cambios necesarios e imprescindibles para que un texto pueda ser comprendido. [...] Por su parte, la posedición completa tiene como objetivo eliminar todo error de la TA y conseguir una traducción de alto nivel, a la par de la traducción tradicional manual. (Aranberri, 2014. p. 472)

Amb el propòsit d'ajudar en aquesta tasca, la plataforma TAUS (https://www.taus.net/academy/best-practices/postedit-best-practices/machine-translation-post-editing-guidelines) ofereix una sèrie de directrius per a postedicions *light* i *full*, què permeten saber que cal corregir per a què una traducció es consideri *good enough* (equiparable a la d'assimilació) o *equal human translation* (comparable a la de disseminació).

A més de l'objectiu amb què es revisa, però, també cal tenir en compte *com* es duu a terme aquesta tasca. En aquest sentit, Martín-Mor, Piqué i Sánchez-Gijón. (2016, p. 67) classifiquen el tipus de postedició entre *clàssica* (que no requereix cap mena de programari específic) i *integrada* (que es realitza mitjançant eines de traducció assistida i memòries de traducció):

> L'únic requisit per fer una PE clàssica és disposar d'un entorn que permeti l'edició [...] La segona de les opcions és la que denominem *PE integrada*. El traductor rep el text original i la traducció en brut juntament amb l'MT. Per dur a terme una PE que integri TA i TAO és imprescindible treballar amb un entorn TAO (Martín-Mor, Piqué i Sánchez-Gijón, 2016, p. 67).

3.3.2.3.L'ús de corpus del mateix domini (TAE i TABR)

Un altre dels mètodes més coneguts per millorar els resultats en traducció automàtica és l'ús de corpus del mateix domini per a l'entrenament dels motors. Això és més productiu, segons Alonso (2007, p. 25), perquè els programes d'ordinador encara no són capaços de comprendre la informació relacionada amb el *coneixement del món* i per tant "si no ens restringim a un domini molt específic [...], estem parlant, no solament de totes les característiques de totes les possibles entitats del món (éssers vius, objectes i conceptes), sinó de totes les possibles relacions rellevants entre aquestes entitats." Un bon exemple d'aquesta manca de *coneixement del món* seria la traducció que proposa aquest autor de l'oració *Els pingüins neden però no volen*, i que Google Translate tradueix com *Los pingüinos nadan pero no quieren*.

Així, si es tradueix un text d'àmbit farmacèutic en un motor entrenat amb corpus del mateix àmbit, els resultats de TA milloren en comparació amb el mateix text traduït amb un motor entrenat amb corpus d'àmbit jurídic. Així, "performance declines when the training data is in a different domain from the test data" (Haddow i Koehn, 2012, p. 422).

Tanmateix, aquests autors ofereixen la possibilitat, tant en TABR com en TAE, de complementar l'entrenament de motors amb corpus externs al domini i d'assignar diferents pesos (rellevància) a cada corpus. D'aquesta manera, els corpus no pertanyents a l'àmbit no tindrien un impacte negatiu en el producte de la TA, però podrien ajudar en la traducció de paraules menys freqüents.

3.4.TA: estat de la qüestió

Durant els darrers anys, l'evolució de la traducció automàtica i la millora dels sistemes existents han permès l'increment d'aquesta tecnologia als fluxos de treball i la demanda del mercat d'aquest tipus de traducció:

> El avance de estos últimos años en el campo de la traducción automàtica (TA) ha hecho posible su adopción en la industria. Y no sólo porque la calidad de los sistemas punteros haya mejorado considerablemente, sino también porque las empresas del sector han reconocido que incluso una TA imperfecta puede ser útil para satisfacer las demandas actuales del mercado de la traducción. (Aranberri, 2014, p. 471).

Aquest augment de l'ús de TA, però, no sembla correspondre's amb l'ús de TA en l'àmbit estatal. Els resultats del projecte ProjecTA, que buscava conèixer l'ús d'aquest tipus de traducció en les empreses del sector, conclou amb unes dades molt desencoratjadores. Els resultats d'aquest projecte afirmen que només la meitat d'empreses de les enquestades fa servir TA i que, gairebé la meitat d'aquestes, la fa servir en menys d'un 10% dels projectes que realitza. A més, només un 16% té un sistema de TA propi (Torres et al., 2016, p. 27-28).

Pel que fa als sistemes preferents, varien segons el període, per exemple, el 2009, Ginestí i Forcada afirmaven el següent:

> D'altra banda, podríem dir que, en general, els sistemes basats en regles són els que millor qualitat de traducció proporcionen avui dia, especialment entre llengües properes, i els més utilitzats en els sistemes comercials tot i que en els últims anys els sistemes estadístics han millorat considerablement i comencen a oferir bons resultats. (p. 48)

Mentre que, el 2016, la perspectiva era força diferent:

> Consequently, it can be argued that SMT has become nowadays one of the main focus in machine translation (MT) research and that it has gained an important share in the MT market, although new approaches such as neuronal machine translation are also gaining momentum. (Pérez, 2016, p. 7)

Així doncs, durant els darrers anys s'ha viscut un notable l'interès en l'ús de TAE, que el 2009 encara no era la TA preferent dels sistemes comercials. Tanmateix, cal tenir present l'aparició al mercat de la TAN, que ja s'ha introduït en alguns dels sistemes de traducció automàtica més reconeguts (pensem en Google Translate o Microsoft).

4.Metodologia

Aquest apartat pretén explicar, des d'un punt de vista pràctic, les passes prèvies als resultats d'aquest treball. És a dir, quina informació coneixíem sobre les eines abans de començar a treballar-hi, com hem aconseguit constituir els corpus i, finalment, com hem instal·lat i entrenat els diferents programes que analitzem.

Per aquest motiu, hem considerat que s'havien de tractar temes com la cerca i obtenció dels diferents corpus, la seva elaboració, la instal·lació de les eines i l'entrenament dels seus motors. Aquest apartat pretén, a més, fer èmfasi en els recursos que hem utilitzat durant aquesta fase (tècniques, documentació, etc.).

4.1.Programari de l'anàlisi

Les eines que analitzem en aquest treball presenten característiques diverses (es limiten a certes combinacions lingüístiques o no, són de codi obert o privatives, etc.) Així, per als propòsits d'aquest Treball de Fi de Màster, és necessari descriure-les destacant-ne els elements amb què es realitzarà la comparació i anàlisi finals.

4.1.1.Machine Translation Training Tool (MTTT)

La primera de les eines amb què tractem es defineix com una eina de codi obert, amb versió *desktop* (per a Windows i Linux) i web i indicada per a l'ensenyament i pràctica de traducció automàtica (Bouillon et al., 2017, p. 2). Es tracta de programari que utilitza TAE, en concret, Moses (vegeu l'apartat 3.2.1.3. *Decoding*).

Pel que fa al flux de treball de l'eina, les dues versions n'ofereixen un de molt similar que s'inicia amb la preparació dels corpus i l'entrenament del motor. A continuació, es duu a terme la traducció *per se* i una avaluació automàtica de la traducció (que l'eina realitza gràcies a la incorporació de la mètrica BLEU). El darrer pas és la postedició del producte resultant (LaFuente, 2017).

La informació oficial d'aquesta eina remarca que disposa d'interfície gràfica, cosa que no totes les eines de codi obert disponibles actualment ofereixen (Bouillon et al., 2017, p. 2). També destaca que cap eina que compleixi aquestes característiques permet integrar "the whole MTPE workflow" (Bouillon et al., 2017, p. 2).

Quant a la informació necessària per a l'entrenament del seu motor, no s'ha trobat documentació concreta sobre el mínim de corpus necessari per realitzar aquesta tasca, tot i que, com assenyalen Gavrila i Vertan (2011, p.1):

> State-of-the-art literature tends to share the opinion that the larger the data, the better the results. (Suresh, 2010) shows that a larger corpus size for training increases the quality of a Moses-based SMT system.

En aquest sentit, tampoc s'ha trobat documentació sobre quins formats admet. Tot i això, en tractar-se del mateix sistema TAE que MTradumàtica, que utilitza "the usual Moses text format" (Doğru, Martín-Mor i Ortiz-Rojas, 2017, p.2) caldria utilitzar corpus amb aquest format per entrenar els motors i realitzar les traduccions, és a dir, documents de text pla amb oracions separades per salts de línia.

Finalment, i pel que fa a les llengües amb què treballa, aquesta eina només permet crear combinacions lingüístiques que continguin francès, anglès i alemany (vegeu l'apartat 4.5.1.Instal·lació de Machine Translation Training Tool).

4.1.2.Modern MT (MMT)

ModernMT es presenta com una eina de codi obert, amb versió d'escriptori (per a Ubuntu), neuronal i enfocada a empreses (MMT, 2018). Així, en un principi, és una eina que nosaltres, amb la capacitat dels nostres ordinadors personals, no hauríem de poder instal·lar (vegeu l'apartat 3.2.2.2.Ús de les xarxes neuronals en la traducció automàtica neuronal). Tot i això, gràcies al procés de compilació del codi font (vegeu l'apartat 4.5.2.Instal·lació de ModernMT) finalment sí que s'ha pogut instal·lar i serà present durant l'anàlisi final del treball.

Dues de les seves característiques principals són que no cal entrenar el sistema per utilitzar-la, ja que "ModernMT is incremental [...] Forget training time and trial & error sessions" (MMT, 2018). Així doncs, només cal implementar les nostres memòries per fer funcionar l'eina:

> With MMT you don't need anymore to train multiple custom engines, you can push all your data to a single engine that will automatically and in real-time adapt to the context you provide. MMT aims to deliver the quality of a custom engine and the low sparsity of your all data combined. (MMT, 2018)

En aquest sentit, ens sembla sorprenent que ni la pàgina oficial ni a la documentació de l'eina a Github no s'esmenti quina mena de formats accepta (o no). Només es parla del format XML, que sí que contempla entre els formats amb què tracta, i de "standard sentence aligned corpora, in the format of either TMX files or in couples of parallel files representing memories" (MMT, 2018).

Aquesta eina, a diferència de l'anterior, no disposa d'interfície gràfica i, per tant, el seu ús es troba condicionat al coneixement del sistema de comandes del terminal. Pel que fa al seu flux de treball incorpora una acció *evaluate* que permet realitzar una avaluació automàtica de la traducció gràcies a la mètrica BLEU, analitza la velocitat amb què s'ha traduït el text i puntua la postedició del producte de la traducció.

Finalment, pel que fa a les combinacions lingüístiques que accepta, cal destacar que la llengua d'origen és sempre l'anglès i que només n'ofereix nou de diferents (MMT, 2018).

4.1.3.MTradumàtica

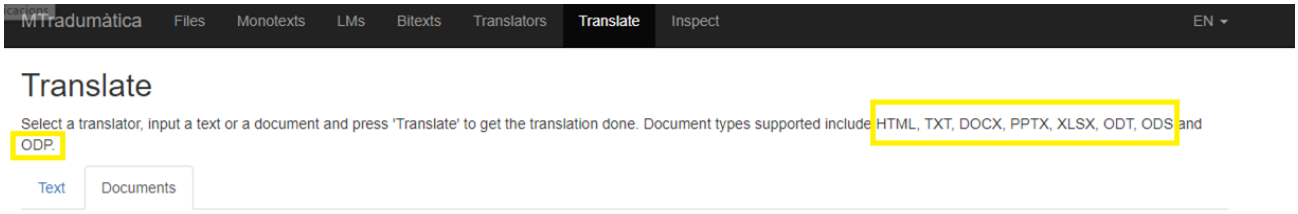
MTradumàtica és una eina molt semblant a MTTT. Es tracta d'un recurs que disposa de versió web i versió d'escriptori (per a Ubuntu), de codi obert, amb interfície gràfica i que utilitza Moses com a sistema de traducció. És una eina desenvolupada pel grup Tradumàtica de la Universitat Autònoma de Barcelona que, de la mateixa manera que la primera, també es pot presentar com un recurs pedagògic:

> L'objectiu de l'eina és proporcionar a investigadors no experts en tecnologia una aplicació web que permeti crear un motor de traducció amb Moses. L'aplicació vol servir, també, com a prova de concepte per a les empreses de traducció i els posteditors que vulguin posar a prova un flux de treball amb motors propis de TA, sense oblidar el vessant didàctic en el marc de la docència de processos de TA adreçada a estudiants de traducció (Martín-Mor i Piqué 2017, p. 102).

Quant al seu flux de treball, "la interfície del programa presenta un procés lineal de sis passos (set, si es té en compte la funció Inspect" (Martín-Mor i Piqué, 2017, p. 104) que s'inicia amb la càrrega de fitxers i finalitza amb la traducció *per se*. A més, "al llarg de tot el procés, la barra superior mostra a l'usuari en quin pas es troba" (Martín-Mor i Piqué, 2017, p. 104).

Pel que fa als formats amb què treballa, la documentació assenyala que els fitxers per a l'entrenament del motor han de trobar-se en format Moses i que "mentre no s'implementi en MTradumàtica una funció per a la conversió del conegut estàndard TMX a format Moses, es pot recórrer a programes com Okapi Rainbow" (Martín-Mor i Piqué, 2017, p. 105). Tot i això, per a la traducció, accepta els següents formats:

Imatge 3. Formats de traducció amb MTradumàtica (MTradumàtica, s.d.)



Actualment, a la versió web es poden consultar motors ja entrenats amb combinacions lingüístiques diverses (català-rus, anglès-turc, anglès-sard, etc.). Tot i això i encara que, en teoria, les eines que permeten la personalització de motors haurien d'acceptar-ne qualsevol, es pot afirmar que, aquest programa es troba restringit a una llista concreta de llengües (vegeu Annex 3). De la mateixa manera, a la documentació no s'especifica si hi ha un mínim requerit de corpus per a les fases d'entrenament del motor.

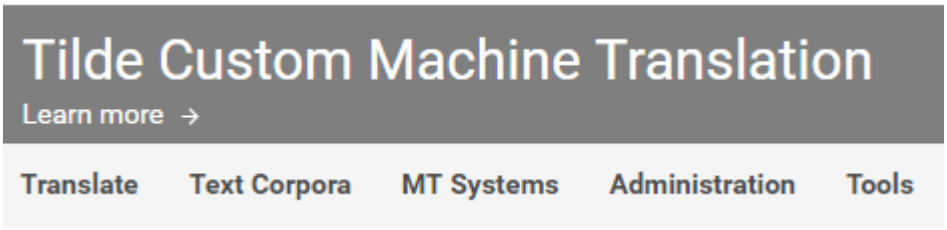
4.1.4.LetsMT

Aquesta eina es defineix a la documentació oficial com un recurs de TAE de codi obert que utilitza Moses i que, a més, és "cloud-based" (Vasiļjevs, Skadiņš i Tiedemann, 2012, p. 1). Encara que a la informació trobada s'especifica que "the LetsMT! project partners are companies TILDE (coordinator), Moravia, and SemLab, and the Universities of Edinburgh, Zagreb, Copenhagen, and Uppsala" (Vasiļjevs, Skadiņš i Tiedemann, 2012, p. 1), la pàgina oficial (http://letsmt.com) redirigeix directament a la de l'empresa TILDE, la coordinadora del projecte.

Un altre apunt destacable és que aquesta és una de les poques eines que analitzem (juntament amb KantanMT) que és de pagament. Per als propòsits d'aquest TFM, però, hem fet servir la versió de prova de 30 dies.

Pel que fa al flux de treball, com en el cas de MTradumàtica, es troba separat i diferenciat per pestanyes o opcions concretes:

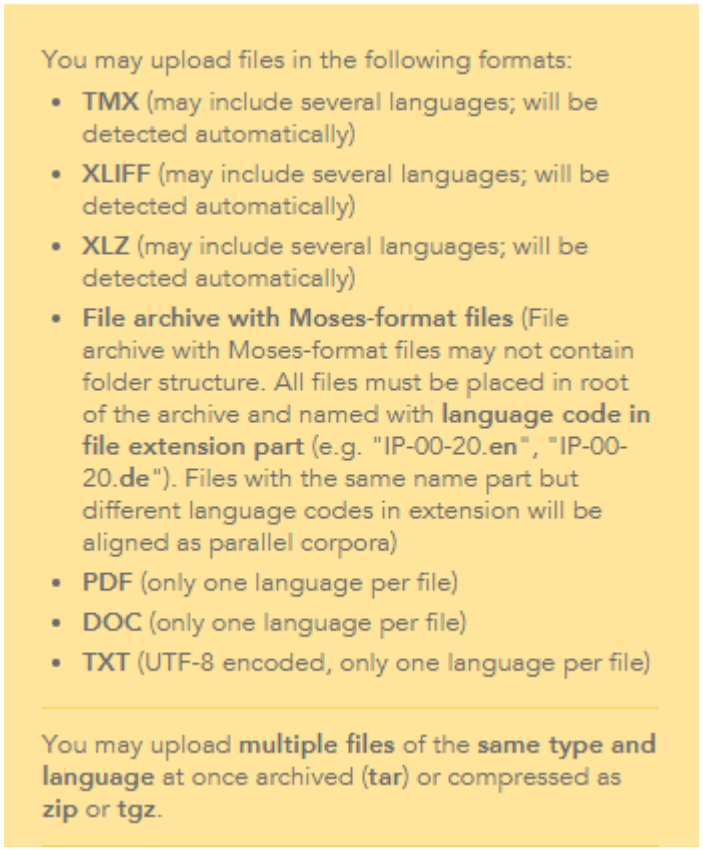
Imatge 4. Flux de treball amb LetsMt (TILDE MT, s.d.)



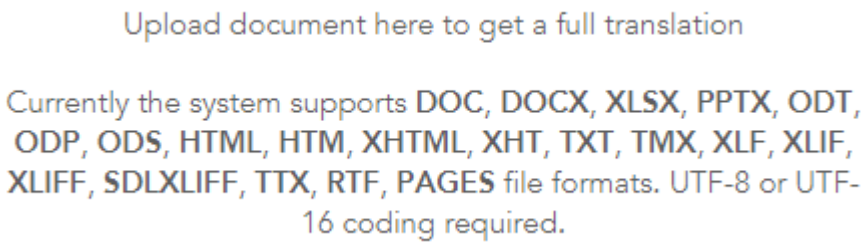
Així doncs, hi ha una primera opció que ens permet traduir, una segona pestanya (*Text Corpora*) que ens permet pujar els textos paral·lels i monolingües i una darrera (*MT Systems*) on s'entrenen els motors. La interfície també permet administrar els projectes i ofereix recursos com la mètrica BLEU o la integració del recurs en eines de traducció assistida per ordinador.

Quant als formats, cal fer distinció entre els formats que accepta l'eina per a l'entrenament de corpus, per a la traducció i per a la mètrica BLEU (en aquest cas, només accepta TXT):

Imatge 5. Formats per a l'entrenament de motors amb LetsMT (TILDE MT, s.d.)



Imatge 6. Formats per a la traducció amb LetsMt (TILDE MT, s.d.)



Finalment, i pel que fa a les restriccions, LetsMt recomana un mínim d'un milió de frases de corpus paral·lel per a l'entrenament dels motors (Tilde MT, s.d.). A més, les combinacions lingüístiques que permet fer es troben limitades a una llista concreta que no permet personalització (vegeu Annex 4).

4.1.5.KantanMT

KantanMT és una eina de pagament, amb versió en línia i privativa que permet entrenar TAE i TAN (encara que, per a aquest treball, se'ns ha concedit accés limitat a la versió per a docents, que només permet entrenar TAE). Quant al flux de treball, l'eina el divideix entre *engines* (on es creen els motors), *training* (on es pugen els arxius per entrenar), *deploy* (que requereix el temps d'entrenament) i *translate* (on es pot traduir en línia o pujar els arxius corresponents). A més, permet realitzar avaluacions automàtiques de la traducció gràcies a una sèrie de mètriques com ara TER, BLEU, F-measure, etc.

Pel que fa als formats que accepta, cal distingir, un cop més, entre els formats per a l'entrenament dels motors i els formats per a la traducció:

Taula 2. Formats de KantanMt

Formats per a l'entrenament	**Formats per a la traducció**
TMX	XLIFF
TXT	TTX
XLSX	TXML
TBX	TMX
DOCX	EXP
ZIP	XLZ
GZ	MQXLZ
PDF	.sub.trg
	XLF - CAD
	DOCX
	PDF
	ODT
	DITA
	XML
	INX
	IDML
	HTML
	SVG
	NovaDoc
	MonTag XML
	AborText XML
	TXT
	XLSX

Finalment, en relació a les restriccions d'aquesta eina, aquesta aplicació no necessita un mínim de corpus per a l'entrenament dels seus motors, però sí que conté una llista limitada (i que no permet personalització) de llengües amb les quals treballar.

4.1.6.Microsoft Translator Hub

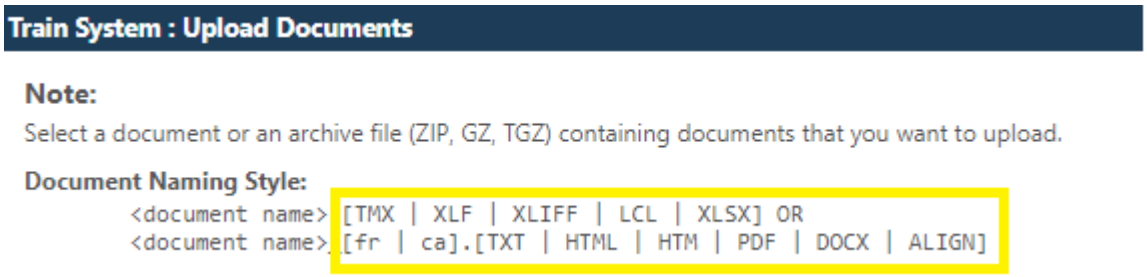
El translator Hub és una extensió gratuïta de Microsoft Translator (Translator Hub - Microsoft Translator, s.d.) amb versió web i privativa. Així, per utilitzar-la, només cal tenir un compte de correu electrònic amb Microsoft (Outlook). Pel que fa al seu sistema de TA, sembla que utilitza TAE i TAN, segons la pàgina oficial (Translator Hub - Microsoft Translator, s.d) i que els nostres corpus es combinen amb "Microsoft's vast language knowledge to generate translation systems combining the best of both our large scale training data and your industry specific one" (Translator Hub - Microsoft Translator, s.d.).

El flux de treball d'aquesta eina és força particular, ja que, després d'entrenar els motors i avaluar-ne els resultats amb la mètrica BLEU, l'eina ens demana realitzar un *request Deployment*, que un cop realitzat, permet fer proves de traducció i compartir-les:

> After a set of systems have been trained, go to the Project Details page and select one with a good BLEU score. You may want to consult with reviewers before deciding that the quality of translations is suitable for deployment. If you have not already associated your Translator Text API subscription when you created your workspace, your training won't be deployed, and you will see a message to associate your Translator API subscription by clicking the Settings tab and associating your subscription. In the Request Deployment page of the selected system, click on Request Deployment. (Microsoft Corporation, 2018, p. 44)

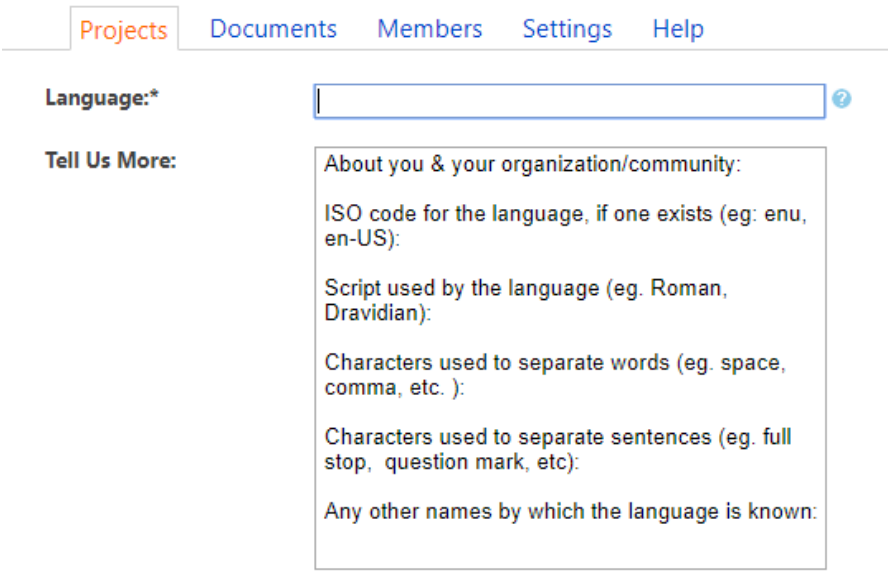
Quant als formats que accepta, cal distingir altre cop entre formats per a la traducció (que són, segons la pàgina oficial, TMX, XLIFF, TXT, HTML, DOCX, XLSX i PDF) i formats per a l'entrenament amb corpus:

Imatge 7. Formats per a l'entrenament de motors amb Microsoft Translator Hub (s.d.)



Finalment, cal destacar que es recomana "a minimum of 10,000 parallel sentences for full trainings" (Microsoft Corporation, 2018, p. 10) per entrenar aquesta eina i que, en cas de no disposar d'alguna llengua en concret a la llista que ens proposa, permet enviar un missatge especificant-ne les característiques principals:

Imatge 8. Especificacions per sol·licitar llengua (Microsoft Translator Hub, s.d.)



4.2.Eines per a l'elaboració del treball

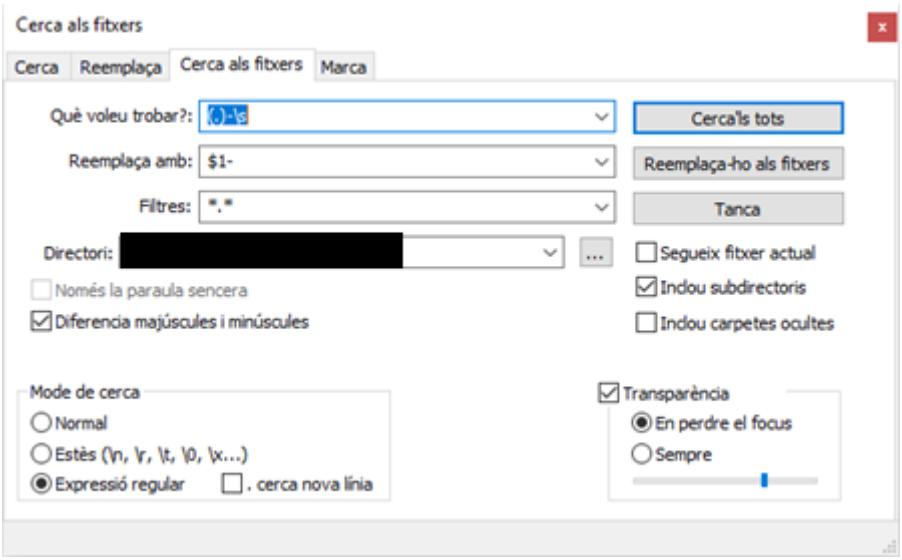
Aquest segon apartat de la metodologia pretén descriure breument alguns dels recursos que s'han hagut d'utilitzar al llarg del treball per dur a terme les tasques relacionades amb l'elaboració dels corpus, la instal·lació de les màquines i l'entrenament dels motors.

4.2.1.Editors de text

Els editors de text són "programes per a l'edició de text pla, és a dir, sense format [...], molt utilitzats per editar llenguatges d'etiquetes, com ara l'HTML o el TMX" (Martín-Mor, Piqué i Sánchez-Gijón, 2016, p. 110). Aquest tipus d'eines han sigut indispensables per al tractament dels arxius en format Moses i TMX utilitzats per a l'entrenament dels motors del programari (vegeu l'apartat 4.4.Elaboració dels corpus) o per a la creació de l'script necessari per a les cerques a la pàgina de traduccions col·laborativa (vegeu l'apartat 4.4.1.Creació de documents amb els resultats de la pàgina col·laborativa), ja que, a més de ser una eina bàsica en el món de la programació (Aprender a programar: introducción y conceptos básicos, s.d.) també es classifiquen com a eines de processament de textos (Martín-Mor, Piqué i Sánchez-Gijón, 2016, p. 109).

Avui dia existeixen una gran quantitat d'editors de text diferents (Notepad++, SublimeText, Brackets, etc.) amb funcions i opcions molt diverses (per exemple, Notepad++ permet fer *cerca i reemplaça* a tots els arxius d'un mateix directori) que permeten que els usuaris puguin escollir l'eina segons les seves necessitats:

Imatge 9. Funció *Cerca als fitxers* de Notepad++



4.2.2.Sistemes operatius i màquines virtuals

Durant la fase d'instal·lació del programari vam haver de treballar amb diferents sistemes operatius, és a dir, amb diferents conjunts de programes que controlen el funcionament d'un ordinador (Martín-Mor, Piqué i Sánchez-Gijón, 2016, p. 89) al nostre entorn de treball. Aquest problema s'origina en el fet que utilitzem Windows als nostres ordinadors personals i tant ModernMT com MTradumàtica requereixen un sistema operatiu Ubuntu per funcionar. Així doncs, per tal de poder instal·lar aquest sistema de manera senzilla podíem, o bé utilitzar el recurs d'una màquina virtual, o bé recórrer a una partició del disc.

Una màquina virtual és, segons Catalunya, I. O. (s.d.) "una màquina que simula el funcionament d'una màquina real sobre la qual es poden instal·lar sistemes operatius, aplicacions informàtiques, navegar de manera segura per Internet, utilitzar diversos dispositius [...]". Així, a diferència d'una màquina física que té elements físics o components (Catalunya, I. O., s.d.) la màquina virtual *simula* la real, ja sigui a través de programari de virtualització (com ara VirtualBox, VMware, QEMU, etc.) o utilitzant una memòria USB.

Pel que fa al sistema en qüestió, l'Ubuntu, es tracta d'una versió o distribució de GNU/Linux pensada per a "aquellos que estan acostumbrados a otros sistemas operativos de escritorio como Windows o Mac OS X" (SL, U. T., s.d.) i considerada com el sistema lliure més popular i reconegut, amb més de 20 milions d'usuaris arreu del món (Ubuntu, s.d.). A més, incorpora un seguit de programari lliure de sèrie, com ara el navegador Mozilla Firefox, el paquet ofimàtic LibreOffice, l'editor d'imatges GIMP, etc. que es poden complementar amb l'enorme catàleg de programari lliure existent (SL, U. T., s.d.).

4.3.Cerca i obtenció de corpus

En aquest capítol s'analitza el procés que hem seguit les integrants del treball per obtenir els recursos necessaris per a l'entrenament dels motors dels programes. Així doncs, només s'explicarà com s'han obtingut els recursos, ja que, les seves modificacions, es resumeixen a l'apartat 4.4. (vegeu l'apartat 4.4.Elaboració dels corpus).

4.3.1.Corpus bilingües

Aquest apartat resumeix la cerca i obtenció dels corpus bilingües en les dues combinacions que treballem. Així, encara que per al corpus xinès-català s'han utilitzat mètodes alternatius per poder ampliar-ne el contingut (vegeu l'apartat 4.4.1.Creació de documents amb els resultats de la pàgina col·laborativa i l'apartat 4.4.2.Transformació de formats (Moses i TMX)) l'obtenció d'alguns d'ells és comú a les dues combinacions amb què treballem.

4.3.1.1.Opus Corpus

El primer recurs que vam consultar per a la cerca de corpus amb unes combinacions lingüístiques tan concretes va ser Opus (http://opus.nlpl.eu/). Aquesta pàgina "provides tools for processing parallel and monolingual data as well as several interfaces for searching the data, which makes it a unique resource for various research activities" (Tiedemann, 2012, p. 5) i proporciona, de manera gratuïta, una gran quantitat de recursos en format TMX i Moses amb combinacions lingüístiques i orígens diversos (el que suposa textos d'àmbits diferents). En aquest punt, però, encara no teníem clar l'àmbit amb què hauríem d'entrenar els motors, ja que desconeixíem la quantitat i tipus de recursos disponibles per a la combinació xinès-català. Els resultats en la recerca amb aquesta combinació van ser els següents:

Taula 3. Resultats de la cerca de corpus xinès-català a Opus

Recurs	Segments bilingües
GNOME	76
KDE4	9.425
OpenSubtitles2016	95.699
OpenSubtitles2018	134.536
Ubuntu	26
TOTAL	**239.762**

Aquests resultats ens mostren tres corpus de l'àmbit del programari i la informàtica (un sistema operatiu lliure i dos entorns d'escriptori) i dos corpus més generals, que poden referir-se a qualsevol tema.

Pel que fa al francès, els resultats a Opus ens mostren que aquests recursos també s'hi troben disponibles. Això ens facilita, d'una banda, la recerca de corpus monolingües en català (llengua

d'arribada en totes dues combinacions) i de l'altra, la cerca de textos per traduir, ja que a totes dues combinacions lingüístiques els àmbits són els mateixos.

Taula 4. Resultats de la cerca de corpus francès-català a Opus

Recurs	Segments bilingües
GNOME	2.147
OpenSubtitles2016	299.172
OpenSubtitles2018	363.408
KDE4	140.334
Ubuntu	6.808
TOTAL	**811.869**

Malauradament, i com ja havíem previst, la combinació de llengües romàniques disposa de quasi 572.107 segments paral·lels més que el corpus xinès-català. Per aquesta raó, i per tal de poder obtenir resultats semblants amb l'anàlisi de les traduccions, s'ha fet recerca per tal de suplir aquesta manca de segments (vegeu els següents apartats). Sabem que la quantitat de segments bilingües per temàtica serà molt diferent en totes dues combinacions lingüístiques, però, també sabem, que la recerca de segments bilingües en xinès-català serà dificultosa.

Pel que fa al corpus reservat per a l'avaluació automàtica de la qualitat (vegeu l'apartat 3.3.1.Mètodes d'avaluació automàtica de TA) s'han seguit les recomanacions de Sánchez (2017) i s'han reservat 2.000 oracions que no s'han inclòs a l'entrenament dels motors. Així, encara que en aquest treball no s'ha dut a terme *tuning* o *optimització* dels motors, és convenient reservar aquest corpus per si, en el futur, es vol realitzar aquesta tasca.

Quant al format dels arxius utilitzats, les dificultats per escollir amb quin format treballar van ser nombroses. Primerament, perquè, depenent del programari a utilitzar, només s'accepta o bé TMX (KantanMT) o bé Moses (MTradumàtica). A més, d'una banda, els arxius TMX partien del català com a direcció de la traducció i no sabíem si això pot afectar els resultats o l'entrenament dels motors. D'altra, els arxius Moses presenten problemes com espais innecessaris després d'apòstrofs i guionets i línies mal codificades (vegeu l'apartat 4.4.3.Modificació dels arxius (TMX i Moses)) i alinear-los per convertir-los en TMX és un procés que resulta lent amb totes les eines amb què s'ha intentat (SDL Trados, LF_Aligner, Wordfast). Finalment, es va optar per utilitzar els arxius en format Moses (ja arreglats) i crear un script en llenguatge JavaScript (vegeu l'apartat 4.4.2.Transformació de formats (Moses i TMX)) que fusionés tots dos arxius i crees un TMX senzill. Per al propòsit d'aquest treball, però, i amb les màquines que ho acceptessin, es va intentar entrenar els motors amb tots dos formats (és el cas de Microsoft Translation Hub i LetsMT).

Finalment, en relació amb l'àmbit dels corpus vam tenir dubtes sobre quina mena de motors es volien entrenar: o bé uns motors d'un àmbit més genèric (incloent-hi tots els arxius descarregats d'Opus), o bé corpus més concrets, d'un àmbit determinat com és la informàtica (en aquest cas, només s'inclourien els recursos de GNOME, KDE4 i Ubuntu). Segons la documentació sobre el tema (vegeu l'apartat 3.3.2.Millorant la qualitat en TA) l'àmbit dels corpus depèn del tipus de text que es vulgui

traduir i, com en aquest punt del treball encara no havíem decidit els textos concrets que s'utilitzarien per a les proves de traducció, es va optar per incloure tots els recursos d'Opus i crear corpus amb més contingut. D'aquesta manera, a més, s'evitaria crear motors acostumats a frases curtes i sense context, que és el tipus de text que s'ha trobat als recursos de l'àmbit informàtic.

4.3.1.2. Recursos amb corpus alineats

Una de les opcions més recomanables era la cerca de recursos semblants a Opus, és a dir, recursos amb corpus en xinès i català. Així, ens vam trobar amb el *Academia Sinica Balanced Corpus of Modern Chinese*, també anomenat *Sinica Corpus*, que consisteix en un gran corpus monolingüe etiquetat de xinès modern (tot i que, amb aquestes combinacions, no ens resultava útil, ja que el corpus monolingüe havia de ser en català, que és la llengua d'arribada). En aquesta línia, també ens vam trobar amb diferents corpus alineats en xinès-anglès (com el del *Center for Chinese Linguistics Corpus*, que no permet descarregar-se).

En aquesta fase, vam considerar l'opció d'utilitzar recursos alineats en xinès-castellà i traduir del castellà al català. Tanmateix, vam descartar-la, atès que no disposem de recursos de TA fiables per dur a terme aquesta tasca (hauríem d'entrenar un motor amb la combinació castellà-català en un àmbit molt específic i això requereix conèixer quina de les eines que analitzem és la més adient per aquest objectiu) i no disposàvem del temps necessari per dur a terme aquesta proposta de manera manual.

4.3.1.3. Alineació de pàgines web

La següent opció que vam plantejar va ser l'alineació de pàgines web, bé per mitjà d'un alineador automàtic com Bitextor (http://bitextor.sourceforge.net/), bé de manera més manual, descarregant la pàgina web completa amb l'ajut de programari concret (pensem en HTTrack, que és gratuït i específic per aquest tipus de tasca) i alineant-ne els resultats.

Malauradament, i deixant de banda algunes excepcions com l'institut Confuci (http://www.confuciobarcelona.cat/) o la pàgina del Grup de recerca en Traducció del xinès al català/castellà de la Universitat Autònoma de Barcelona (http://grupsderecerca.uab.cat/txicc/), no existeixen pàgines amb el mateix contingut en aquestes dues llengües.

La darrera alternativa en aquest sentit era cercar a pàgines informatives (com ara la Viquipèdia) articles en les dues llengües. Tot i això, vam descartar aquesta opció per la seva variació en contingut i mida.

4.3.1.4. Alineació manual d'entrades a diccionaris bilingües

La darrera opció, que ha resultat la que finalment s'ha utilitzat en aquest treball, ha consistit en la creació manual de recursos en format TMX i en format Moses per a la combinació xinès-català. Es va decidir crear en tots dos formats perquè, com ja hem comentat (vegeu l'apartat 4.1.Programari de l'anàlisi) hi ha eines que només accepten o bé un format o bé un altre.

Amb aquesta finalitat, s'ha utilitzat una pàgina col·laborativa de traduccions en línia que permet realitzar cerques de mots concrets en aquesta combinació d'idiomes (tot i que no permet descarregar de manera directa la informació de les memòries de traducció que emmagatzema). En aquest sentit, vam haver de descartar altres recursos com Linguee (https://www.linguee.es/) o Reverso (http://www.reverso.net/text_translation.aspx?lang=ES) perquè no oferien el català en cap de les direccions de la traducció.

Imatge 10. Resultats de la cerca de *sistema* a la pàgina col·laborativa

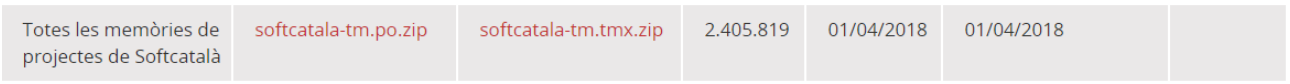
Com es pot veure a la captura de pantalla (vegeu imatge 10), amb la recerca de només una paraula, aquesta pàgina cerca unes nou entrades de segments disponibles a les seves memòries. Això ens permet, amb una llista reduïda de mots, aconseguir molta informació per al TMX/Moses (vegeu l'apartat 4.4.1.Creació de documents amb els resultats de la pàgina col·laborativa i l'apartat 4.4.2.Transformació de formats (Moses i TMX)) que s'ha elaborat.

4.3.2.Corpus monolingües

La recerca de corpus monolingües va resultar més simple que l'anterior, atès que la llengua d'arribada en les dues combinacions és la mateixa. Així, per obtenir textos en català de qualitat i, que a més, fossin de fàcil accés, es va decidir extreure el contingut en català de les TMX (vegeu l'apartat 4.4.2.Transformació de formats (Moses i TMX)) que ofereix Softcatalà al seu web (https://www.softcatala.org/recursos/memories.html). Es tracta de contingut real, revisat i, a més a més, relacionat directament amb part de la temàtica del corpus (el vessant relacionat amb l'àmbit informàtic).

Per dur a terme aquesta tasca, es va descarregar el zip del TMX amb contingut de *Totes les memòries dels projectes de Softcatalà*:

Imatge 11. Descàrrega de TMX a Softcatalà



Per ampliar el contingut en corpus monolingüe es va decidir utilitzar, a més, els Moses en català de Global Voices, Books i Tatoeba (tots disponibles a la pàgina d'Opus) el contingut dels quals és de temàtica general i semblant a OpenSubtitles. El total pel que fa als corpus monolingües, doncs, és el següent:

Taula 5. Corpus monolingües en català

Recurs	**Segments en català**
Global Voices	19.887
Books	4.605
Softcatalà	293.252
Tatoeba	973
TOTAL	**318.717**

4.4.Elaboració dels corpus

Aquesta secció resumeix l'elaboració del corpus, és a dir, la creació, modificació, etc. dels arxius específics per a l'entrenament dels motors del programari d'aquest treball, a partir dels recursos trobats a l'apartat anterior (vegeu l'apartat 4.3.Cerca i obtenció de corpus).

4.4.1.Creació de documents amb els resultats de la pàgina col·laborativa

Per a la creació dels arxius amb els resultats de les cerques de la pàgina web col·laborativa es va decidir crear un script en llenguatge Python. D'aquesta manera es podria automatitzar el procés de

buscar i emmagatzemar els resultats en aquesta pàgina. A més, vam plantejar un tipus de codi que, amb algunes modificacions, serviria per extreure informació de qualsevol pàgina amb característiques semblants (pensen en Linguee o Reverso).

Per realitzar aquesta tasca es va consultar informació sobre com extreure, en llenguatge Python, informació d'una pàgina web (docs.python-guide.org/en/latest/scenarios/scrape/). Pel que fa a les paraules que es van utilitzar per a les cerques, vam decidir aprofitar el vocabulari de l'examen oficial de xinès HSK 6 (VOCABULARIO HSK 6 - Fulls de càlcul de Google, s.d.), un recurs que, d'una banda, recull algunes de les paraules més freqüents en llengua xinesa i, de l'altra, ens proporciona paraules suficients (unes 5.000) per a l'elaboració d'aquests arxius:

Script 1. Arxius monolingües amb resultats de les cerques

`import requests`
 	
`import sys`
 	
`try:`
`  in_zh = open('chinese-words.txt', 'r')`
`except Exception as e:`
`  print e`
`  sys.exit(1)`
	1

`try:`
`  out_zh = open('tfm.zh', 'w')`
`except Exception as e:`
`  print e`
`  sys.exit(1)`
 	
`try:`
`  out_ca = open('tfm.ca', 'w')`
`except Exception as e:`
`  print e`
`  sys.exit(1)`
	2

`for word in in_zh:`
`  query = word.rstrip('\n')`
`  page = requests.get[redacted] + str(query))`
`  tree = html.fromstring(page.content)`
`  translated =`
`tree.xpath('/html/body/div[3]/div/div[2]/div/div/article/div/div[1]/`
`div/ul/li[1]/div[3]/strong/text()')`
`  print translated`
`  if len(translated) != 0:`
`  out_zh.write(query + '\n')`
`  out_ca.write(translated[0].encode('utf-8') +'\n')`
`  num_words = num_words + 1`
	3

```
num_tables = len(tree.xpath('//*[@class="tableRow row-fluid"]'))                    4
    if num_tables != 0:
      #There are context sentences in the page
      for count in range(num_tables):
        #parse catalan text
        cat =
tree.xpath('/html/body/div[3]/div/div[2]/div/div/div/article/div[1]/
div/div[' + str(count + 1) + ']/div[2]/span/span')
        cat_text = cat[0].text_content()
        print cat_text
        out_ca.write(cat_text.encode('utf-8') +'\n')
        #parse chinese text
        chinese_text =
tree.xpath('/html/body/div[3]/div/div[2]/div/div/div/article/div[1]/
div/div[' + str(count + 1) + ']/div[1]/span/span')
        out_zh.write(chinese_text[0].text_content().encode('utf-8')
+'\n')
print "Total number of translated words: ", num_words                              5

in_zh.close()
out_ca.close()
out_zh.close()
```

La primera part d'aquest script (1) s'encarrega de llegir el `'chinese-words.txt'`, un document prèviament creat que conté les entrades en xinès (les paraules del HSK) que volem buscar a la pàgina. A continuació (2) es creen dos documents d'escriptura anomenats `tfm.zh` i `tfm.ca`.

La tercera part del script (3) s'encarrega de llegir el contingut de la pàgina, organitzada segons les seves etiquetes `div,` i de guardar allò que és rellevant (hi ha informació que ens interessa i d'altra que no). Si hi ha contingut (`if len(translated) != 0`), és a dir, si la pàgina retorna informació un cop s'han realitzat les cerques, aquesta s'escriu als arxius d'escriptura, tot separant els resultats per salts de línia. Si, a més, hi ha contingut a l'apartat de les memòries (vegeu imatge 10) aquest també s'afegeix als arxius (4). Finalment, tots dos es tanquen (5).

El problema d'aquesta primera versió del nostre script és que, si la pàgina canvia en algun aspecte (les etiquetes `div`, per exemple), aquest no serà capaç de dur a terme aquest procés. Es va decidir, doncs, crear una segona versió que rebés la informació directament de l'API per a memòries que ofereix la pàgina:

Script 2. Modificació del script de cerques

```
for word in in_zh:                                                                  1
 r =
requests.get([redacted]/gapi/tm?from=zho&dest=cat&phrase=
+ str(word.strip("\n")) + "&format=json")
```

D'aquesta manera (1), a través de l'API, s'assenyala la combinació lingüística en què cal fer la cerca (`from=zho&dest=cat`) i, de la mateixa manera que amb la versió anterior, es demana que els resultats s'emmagatzemin diferenciats per salts de línia (`str(word.strip("\n"))`). Podeu veure el codi complet a l'Annex 1.

4.4.2.Transformació de formats (Moses i TMX)

Un cop obtinguts els arxius en text pla amb la informació de cada cerca (vegeu l'apartat 4.4.1.Creació de documents amb els resultats de la pàgina col·laborativa) calia crear TMX per als programes que no accepten Moses. Així, en un principi, es volia realitzar un alineament manual dels arxius (vegeu l'apartat 4.3.1.1.Opus Corpus).

Aquest mètode, però, va resultar ser massa lent i poc automatitzat i, a més, els errors d'alineació eren considerables. Per aquest motiu, es va decidir crear un script en JavaScript que, a partir dels arxius creats (`tfm.zh` i `tfm.ca`), fos capaç de crear un arxiu TMX nou. Això ens va permetre, a més, crear arxius en TMX amb els Moses descarregats d'Opus (vegeu l'apartat 4.3.1.1.Opus Corpus).

A continuació, s'expliquen algunes de les parts més destacables d'aquest document, que també es pot consultar de manera completa als annexos:

Script 3. Creació de nou TMX

`import fs from 'fs'` `const createItem = (chinese, catalan) => {` ` return `<tu>` ` <tuv xml:lang="zh"><seg>${chinese}</seg></tuv>` ` <tuv xml:lang="ca"><seg>${catalan}</seg></tuv>` `</tu>` `}`	1
`const file = 'out.tmx'` `const header = `<?xml version="1.0" encoding="UTF-8" ?>` `<tmx version="1.4">`	2

`<header creationdate="Sun Jan 3 23:24:06 2016"`
`srclang="ca"`
`adminlang="ca"`
`o-tmf="unknown"`
`segtype="sentence"`
`creationtool="Uplug"`
`creationtoolversion="unknown"`
`datatype="PlainText" />`
	3

`<body>` `const footer = `</body>` `</tmx>`	4

`const outCa = fs.readFileSync('tfm.ca', 'utf8').split('\r')` `const outZh = fs.readFileSync('tfm.zh', 'utf8').split('\r')`	5

`if (outCa.length !== outZh.length) {`
`  console.log('Both files should contain the same amount of examples')`
`  process.exit(2)`
	6

`let words = ''`
`for (let i = 0; i < outZh.length - 1; i++) {`
`  words = words + createItem(outZh[i], outCa[i])`
`}`
`fs.writeFileSync(file, header + words + footer)`
`console.log('[✔] Finished')`
	7

La primera part del script (1) senyala que els valors que rebrem seran `chinese` i `catalan` i que al TMX es crearan elements `<tu>` que contindran unitats de traducció (`<tuv xml:lang)` en què dipositaran la informació que rebin.

Tot seguit (2) s'especifica que el document resultant s'anomenarà `out.tmx` i que a la seva capçalera hi apareixerà la versió d'xml que s'utilitzarà per aquest TMX, la codificació del document i la versió del TMX. Pel que fa a l'element `header` del TMX (3), hi apareixeran els atributs que, segons TMX 1.4b Specification. (s.d.), són necessaris per crear TMX correctes.

A continuació, s'especifica l'obertura i tancament de l'element `<body>` i del TMX (4). En aquest punt, s'indica a l'script (5) que cada element que rebrà es troba als arxius `'tfm.ca'` i `'tfm.zh'` i que, a més, es trobarà separat per salts de línia. També s'ha indicat una regla per a què, si els valors que s'introdueixen a `zh` i `ca` difereixen en quantitat, aparegui un missatge d'error i s'aturi tot el procés (6).

Finalment, s'indica que s'escrigui la informació necessària al document `out.tmx` i s'aturi el procés (7). El document resultant conté entrades de traducció i ofereix l'aspecte següent (vegeu imatge 12):

Imatge 12. Out.tmx

```
<?xml version="1.0" encoding="UTF-8" ?>
<tmx version="1.4">
<header creationdate="Sun Jan  3 23:24:06 2016"
        srclang="ca"
        adminlang="ca"
        o-tmf="unknown"
        segtype="sentence"
        creationtool="Uplug"
        creationtoolversion="unknown"
        datatype="PlainText" />
  <body><tu>
  <tuv xml:lang="zh"><seg>平方</seg></tuv>
  <tuv xml:lang="ca"><seg>quadrat</seg></tuv>
</tu><tu>
  <tuv xml:lang="zh"><seg>冲击波将几百平方英里的森林</seg></tuv>
  <tuv xml:lang="ca"><seg>descendí i tombà arbres al llarg de</seg></tuv>
</tu><tu>
  <tuv xml:lang="zh"><seg>最后,铺两千万平方英尺沥青路的计划</seg></tuv>
  <tuv xml:lang="ca"><seg>Al final, quasi 2 milions de metres quadrats d'asfalt</seg></tuv>
</tu><tu>
  <tuv xml:lang="zh"><seg>建筑面积一千万平方英尺(约九十万平方米), 超高建筑密度。</seg></tuv>
  <tuv xml:lang="ca"><seg>10 milions de metres quadrats en una densitat molt alta,</seg></tuv>
</tu><tu>
  <tuv xml:lang="zh"><seg>两公顷半(注:约一万平方)的空间里有 慢跑道路,餐厅,</seg></tuv>
  <tuv xml:lang="ca"><seg>una hectàrea, pistes per córrer, restaurants,</seg></tuv>
</tu><tu>
```

Quant a l'elaboració de Moses per als corpus monolingües (vegeu l'apartat 4.3.2.Corpus monolingües) s'ha utilitzat Tikal, una aplicació d'Okapi Framework, basada en ordres i multiplataforma, per convertir el TMX al format desitjat. Com Tikal es basa en comandes, només ens ha calgut executar-la i escriure la comanda corresponent:

Imatge 13. Conversió de TMX a Moses

```
                          -xm zn-cat.tmx -2 -sl zh -tl ca
------------------------------------------------------------------------------
Okapi Tikal - Localization Toolset
Version: 2.0.35
------------------------------------------------------------------------------
Extraction to Moses InlineText
Source language: zh
Target language: ca
Default input encoding: windows-1252
Filter configuration: okf_tmx
Input:
Done in 3.355s
```

El resultat són dos fitxers Moses, un per a cada llengua, al mateix directori on es trobava el TMX que s'ha utilitzat per a la conversió.

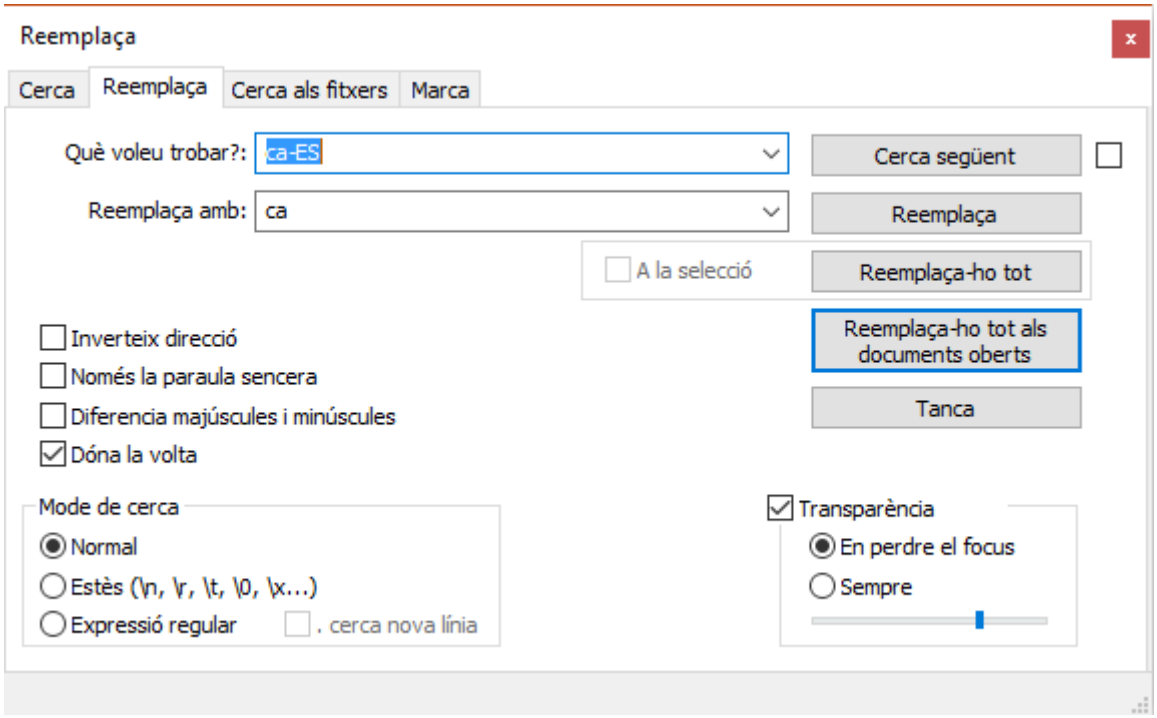
4.4.3.Modificació dels arxius (TMX i Moses)

El programa Notepad++, un reconegut editor de text, s'ha utilitzat per a dues tasques sense les quals no hauria estat possible l'entrenament dels motors: la cerca i substitució de la llengua d'arribada del

TMX del corpus monolingüe abans de la seva conversió a Moses (vegeu l'apartat 4.4.2.Transformació de formats (Moses i TMX)) i la correcció d'errades als Moses descarregats d'Opus.

Així, en primer lloc, es va haver de substituir la llengua d'arribada del TMX descarregat de Softcatalà (vegeu l'apartat 4.3.2.Corpus monolingües), ja que, com aquest arxiu especifica la regió de la llengua d'arribada (ca-ES) Tikal és incapaç de generar l'arxiu en format Moses. Aquesta acció es va realitzar amb un cerca i reemplaça amb Notepad ++, tal com es mostra a la imatge 14:

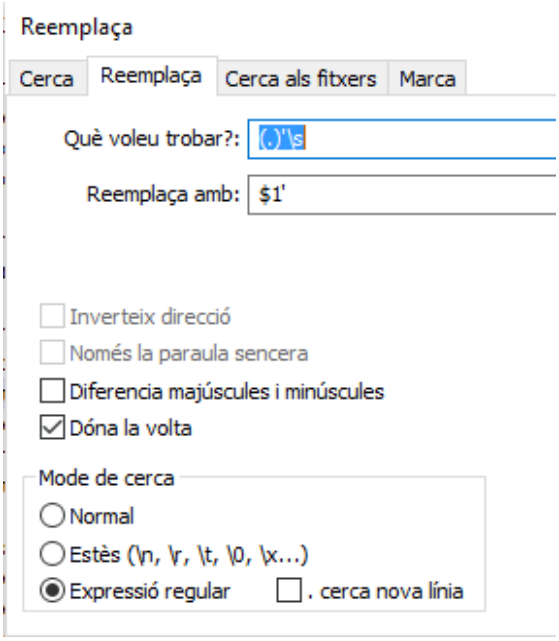
Imatge 14. Cerca i reemplaça al TMX de Softcatalà



Pel que fa als corpus bilingües, en totes dues combinacions, s'han esborrat les primeres línies dels Moses que es van descarregar d'Opus, les que van referència al nom dels traductors. D'aquesta manera, evitem introduir soroll a l'entrenament dels motors.

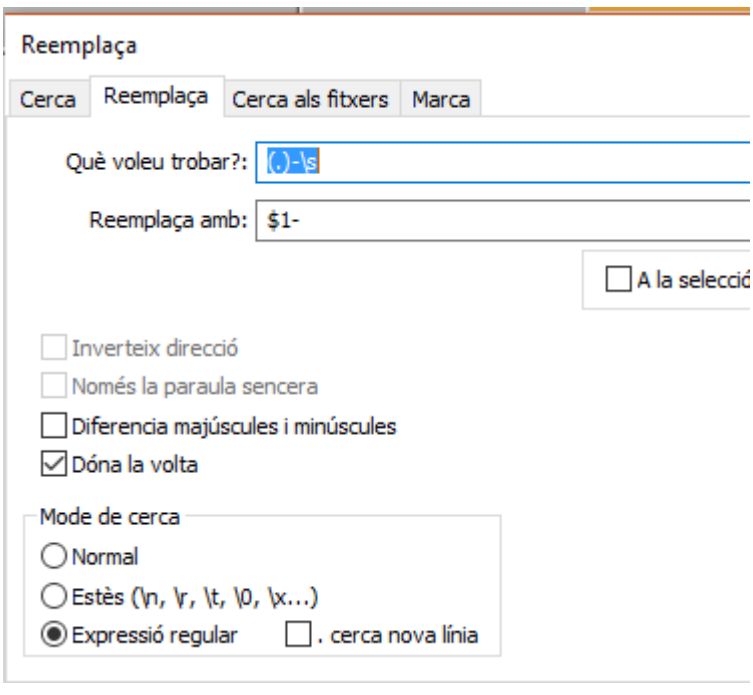
A més, als arxius Moses del KDE4 en francès i català es va detectar un espai addicional entre el signe ortogràfic de l'apòstrof i la lletra següent i entre el guionet i el pronom que acompanya. Aquest error també es va suprimir amb l'ajut d'un cerca i reemplaça. En aquest cas, però, es va marcar la casella d'expressions regulars i se'n va utilitzar una de molt simple:

Imatge 15. Cerca i reemplaça de l'apòstrof al KDE4



Així, aquesta expressió busca qualsevol caràcter que precedeixi un apòstrof i que tingui un espai a continuació i la reemplaça pel mateix caràcter seguit d'apòstrof (marcat com un grup independent a la casella de cerca), però sense l'espai. L'expressió que es va usar per a la substitució dels guionets també va seguir aquesta dinàmica:

Imatge 16. Cerca i reemplaça del guionet al KDE4



Finalment, als arxius Moses de xinès-català, es van haver d'eliminar algunes línies que no es trobaven codificades en *UTF-8* (a diferència de la resta de l'arxiu) i que no vam poder recodificar de cap manera. Van ser les següents:

- De la 5.694 a la 5.701
- De la 4.887 a la 5.361
- De la 11.193 a la 11.226
- De la 12.638 a la 12.666
- De la 13.108 a la 13.112
- De la 16.726 a la 16.738
- De la 17.932 a la 17.946
- De la 27.734 a la 27.770
- De la 86.659 a la 86.681
- De la 106.860 a la 106.912
- De la 106.881 a la 107.011
- De la 107.138 a la 107.181
- De la 119.910 a la 120.753

4.5.Instal·lació dels programes de traducció automàtica

Aquest apartat descriu el procés d'instal·lació de tres dels sis programes que s'analitzen en aquest treball (Machine Translation Training Tool, ModernMT i MTradumàtica), ja que, els altres tres sistemes, funcionen exclusivament en línia i no requereixen instal·lació prèvia. En el cas de Machine Translation Training Tool i MTradumàtica també es podria utilitzar la versió en línia, però, per als propòsits d'aquest treball, hem volgut investigar el procés d'instal·lació de totes dues eines. D'aquesta manera, deixem constància del procediment, i, a més, facilitem la informació als traductors que es plantegin instal·lar i utilitzar al seu ordinador personal aquests programes de traducció.

Així, tot i les seves diferències, cal destacar que dos dels tres programes necessiten un sistema operatiu concret (Ubuntu) i que tots requereixen uns coneixements mínims del terminal de la línia d'ordres per a la seva instal·lació. A més, el procés d'instal·lació, exemples, etc. de les tres eines es troben penjats a Github, un servei de hosting de repositoris.

Un altre punt que volem remarcar abans de començar a descriure el procés d'instal·lació *per se* és que, si bé les eines s'enfoquen a usuaris diferents (MTradumàtica i Machine Translation Training Tool són recursos enfocats a estudiants i professors mentre que ModernMT es presenta com un programa per a empreses) hem considerat que, en els tres casos, el procés d'instal·lació pot resultar excessivament complicat per a algú que no compleixi aquests perfils (és el cas del traductor autònom sense coneixements tecnològics) però que també es beneficiaria dels avantatges que ofereixen.

Finalment, volem subratllar que s'hi han inclòs unes petites conclusions que recullen les nostres sensacions sobre cada eina en concret.

4.5.1.Instal·lació de Machine Translation Training Tool

Per començar el procés d'instal·lació d'aquesta eina, cal, segons les instruccions de Github (Lafuente, R, 2017) haver instal·lat Moses. Així, el primer amb què ens trobem és que la pàgina no ens especifica enlloc on o com cal instal·lar aquest sistema.

Imatge 17. Instal·lació a Ubuntu de Machine Translation Training Tool

On Ubuntu

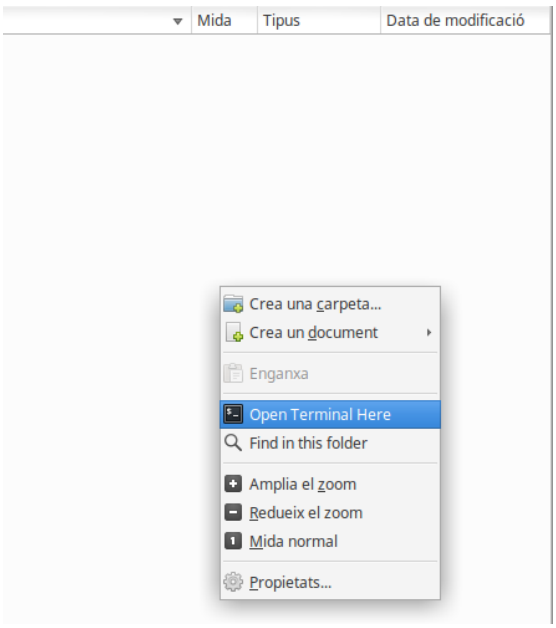
- MOSES (Install with "--with-mm" and "--install-scripts" flags)
- To install its dependencies run

```
python ubuntu_install.py
```

A la pàgina oficial de Moses, però (http://www.statmt.org/moses/?n=Development.GetStarted) es descriu, pas a pas, com cal fer-ho. En aquesta mateixa pàgina trobem que "Moses requires a word alignment tool, such as giza++, mgiza, or Fast Align" (Moses - Development/GetStarted, 2018) i, per aquest motiu, aquest procés s'ha iniciat amb la instal·lació de GIZA++. Ho podem fer de la següent manera:

- Obrim un terminal en una carpeta buida

Imatge 18. Obrir un terminal amb Ubuntu



- Introduïm les següents comandes

```
git clone https://github.com/moses-smt/giza-pp.git
```

```
cd giza-pp
make
```

D'aquesta manera, a la carpeta buida on hem obert el terminal, apareixeran els binaris *~/giza-pp/GIZA++-v2/GIZA++*, *~/giza-pp/GIZA++-v2/snt2cooc.out* i *~/giza-pp/mkcls-v2/mkcls*

- Copiem aquests executables a la mateixa carpeta on posarem el Moses, ja que "these need to be copied to somewhere that Moses can find them" (Moses - Development/GetStarted, 2018). Nosaltres, des de la mateixa terminal on hem compilat el GIZA, pugem dos nivells amunt i creem aquesta carpeta amb les següents comandes

```
cd ..
cd ..
mkdir mosesdecoder
cd mosesdecoder
```

- Un cop dins la carpeta, creem una subcarpeta per aquests executables

```
mkdir tools
```

- Hi copiem els fitxers

```
cp ../giza/giza-pp/GIZA++-v2/GIZA++ ../giza/giza-pp/GIZA++-
v2/snt2cooc.out ../giza/giza-pp/mkcls-v2/mkcls tools/
```

Ara que ja tenim el GIZA++ en una carpeta fàcilment identificable (*tools*) podem instal·lar el Moses a la nostra carpeta *mosesdecoder*. Només ens cal obrir un terminal en aquesta carpeta i anar escrivint les mateixes comandes que apareixen a la seva pàgina oficial.

- Instal·lem els programes addicionals que Moses necessita

```
sudo apt-get install build-essential git-core pkg-config automake
libtool wget zlib1g-dev python-dev libbz2-dev
```

- Descarreguem al directori el codi de Moses

```
git clone https://github.com/moses-smt/mosesdecoder.git
```

- Instal·lem més programes que Moses necessita

```
make -f contrib/Makefiles/install-dependencies.gmake
```

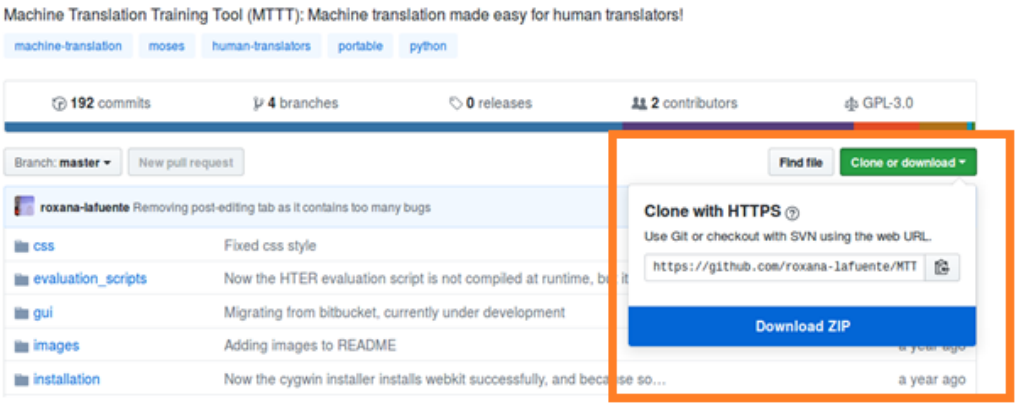
- Compilem Moses

```
./compile.sh --with-mm
```

Aquest procés trigarà una bona estona (dependent de si ho instal·lem en una màquina física o virtual trigarà més o menys). En el nostre cas, han sigut gairebé dues hores.

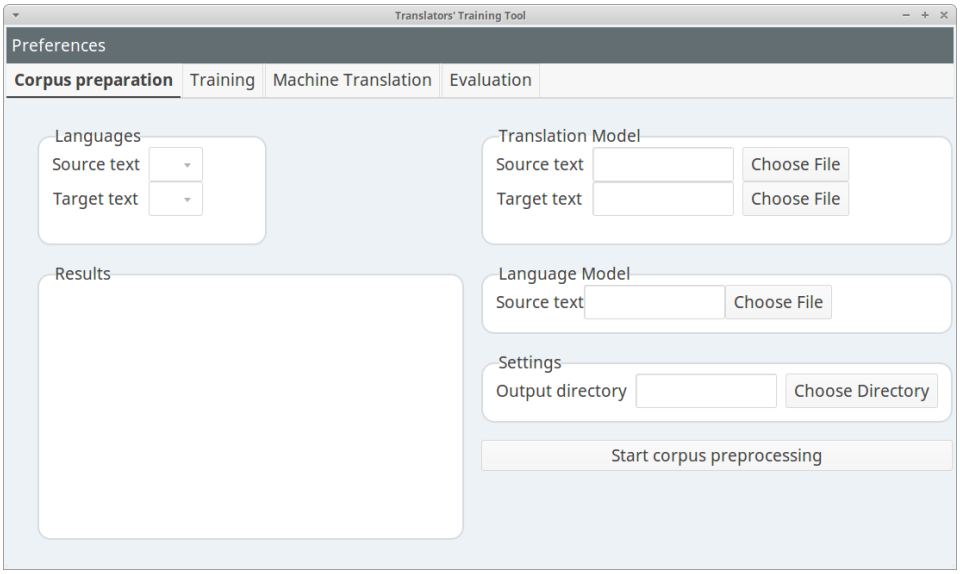
Les instruccions de MTTT no ens ajuden gaire a partir d'aquest pas, ja que no diu enlloc que cal fer a continuació. La següent comanda que apareix a la instal·lació per a Ubuntu ens sembla una mica confosa. Per aquesta raó, vam decidir descarregar tot el contingut de MTTT de Github (vegeu imatge 19). i executar l'ordre `python ubuntu_install.py`, que és la que ens indiquen les instruccions oficials (vegeu imatge 18).

Imatge 19. Descàrrega de MTTT de Github



Durant aquest procés se'ns demana el directori on hem instal·lat Moses, així que, en el nostre cas, només cal buscar la carpeta *mosesdecoder.* Un cop s'ha realitzat aquesta tasca, s'obrirà la interfície gràfica del programa.

Imatge 20. Interfície gràfica de MTTT



Al desplegable de *Languages* podem veure que les llengües amb què treballa aquesta eina són anglès, francès i alemany. Per aquesta raó, i encara que no aparegui de manera explícita a la documentació oficial, no hem considerat entrenar-la, ja que no accepta les nostres combinacions lingüístiques (vegeu l'apartat 4.1.1.Machine Translation Training Tool (MTTT)).

Finalment, les instruccions de la pàgina especifiquen que cal la comanda `python main.py` per executar aquesta eina, o sigui que si la tanquem i la volem tornar a obrir, cal obrir un terminal a la carpeta on s'ha descarregat i escriure aquesta comanda.

4.5.2.Instal·lació de ModernMT

En primer lloc, cal destacar que hi ha diverses maneres d'instal·lar aquest programa. A la taula següent (vegeu taula 6) s'ha fet un resum d'aquests procediments, tot i que, com es veurà a continuació, només hem aconseguit instal·lar l'eina amb un dels tres processos.

Taula 6. Opcions per instal·lar ModernMT

Opció	Requisits	Comentaris
Compilació del codi font	Cal haver clonat o descarregat el projecte de la corresponent pàgina de Github (https://github.com/ModernMT/MMT)	Aquesta és l'opció que hem triat, ja que ens permet utilitzar una configuració més optimitzada per al nostre sistema

Instal·lació dels binaris	Cal haver descarregat els binaris de la pàgina de Github (https://github.com/ModernMT/MMT/releases)	Tot i que aquesta opció no ens va funcionar, deixem explicat el procés d'instal·lació com a referència.
Contenidors Docker	Cal haver clonat o descarregat el projecte de la corresponent pàgina de Github (https://github.com/ModernMT/MMT)	Hem descartat aquesta opció perquè desconeixem el funcionament dels contenidors.

En segon lloc, volem destacar que, a causa de les característiques dels nostres ordinadors personals, no vam poder instal·lar la funcionalitat de traducció neuronal, ja que cal "at least one CUDA-capable GPU" (MMT, 2018).

4.5.2.1.Compilació del codi font

Per instal·lar aquesta opció, que és la que finalment s'ha utilitzat, només cal seguir els passos de la pàgina de Github de l'eina (MMT, 2018). Per fer-ho, però, cal, de manera prèvia, descarregar-se el contingut de la pàgina i obrir un terminal allà on sigui aquest contingut. Un cop realitzats aquests dos passos previs, cal seguir aquestes instruccions:

- Instal·lem una sèrie de prerequisits

```
sudo add-apt-repository ppa:george-edison55/cmake-3.x
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install python-pip
pip install -U requests
sudo apt-get install g++
sudo apt-get install libsnappy-dev zlib1g-dev libbz2-dev
libboost-all-dev libsparsehash-dev cmake openjdk-8-jdk git maven
```

En aquest punt es pot veure que el codi font necessita pip, un administrador de paquets per a Python, i el mòdul requests per a poder executar-se.

- Inicialitzem els submòduls de l'eina

```
cd ModernMT
git submodule init
git submodule Update
```

- Descarreguem aquests submòduls i els compilem

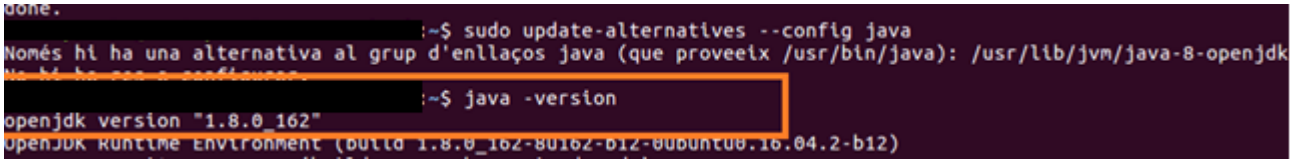
```
cd vendor
./compile
cd ..
```

En aquest punt, les instruccions ens demanen que comprovem la nostra versió de Java. Ho podem fer amb una simple comanda:

```
java -version
```

Com es pot veure a la imatge (vegeu imatge 21), la versió que utilitzem és la 8, la mínima per poder instal·lar aquest programa (MMT, 2018):

Imatge 21. Versió de Java al nostre sistema operatiu



En cas de no tenir-la, però, caldria instal·lar-la amb la comanda següent:

```
sudo add-apt-repository ppa:openjdk-r/ppa && sudo apt-get update &&
sudo apt-get install openjdk-8-jdk
```

- Creem la nostra distribució de MMT

```
cd src
mvn clean install
cd ..
```

Ara que l'eina ja es troba instal·lada es podrien pujar els arxius necessaris i crear un motor de traducció. A la pestanya Wiki de la pàgina de l'eina a Github (https://github.com/ModernMT/MMT/wiki/Create-an-engine-from-scratch) ens expliquen com fer-ho des de 0. També, però, podem crear un motor de prova amb els textos per defecte que ens descarreguem des de la mateixa pàgina web.

- Creem un motor de prova amb la combinació lingüística anglès-italià (amb els textos que es descarreguen per defecte) i l'executem

```
./mmt create en it examples/data/traint
./mmt start
```

- Fem proves (*hello world*) amb aquesta combinació. Les instruccions diuen que, opcionalment, podem especificar-hi el context (en aquest cas, *programming language tutorial*)

```
./mmt translate --context "programming language tutorial" "hello
world"
```

Així, teòricament, el programa hauria de poder entrenar-se amb qualsevol combinació lingüística, tot i que, a la pàgina web oficial (ModernMT, s.d.) remarquen que només es poden crear motors de nou combinacions lingüístiques diferents. Per aquest motiu, no s'han creat motors en aquesta eina, ja que les combinacions lingüístiques que accepta no coincideixen amb els nostres corpus.

4.5.2.2.Instal·lació dels binaris

Tot i que aquesta opció no ens ha servit per instal·lar el programa creiem que deixar documentada la nostra experiència pot ser molt útil per als lectors d'aquest treball. Així doncs, per executar aquesta opció, i com en el cas anterior, ens cal assegurar-nos de tenir instal·lada una versió 8 o superior de java, pip i el mòdul `requests` (vegeu l'apartat 4.5.2.1.Compilació del codi font). També ens cal haver descarregat, de manera prèvia, els binaris de la pàgina de Github (https://github.com/ModernMT/MMT/releases). Un cop hem realitzat aquestes tasques, podem seguir les instruccions de la pàgina:

- Descomprimim un dels binaris (mmt-2.4-ubuntu.tar.gz), esborrem la seva versió comprimida i anem al directori de la pàgina

```
tar xvfz mmt-2.4-ubuntu.tar.gz
rm mmt-*.tar.gz
cd mmt
```

En aquest punt, només caldria, com en el cas anterior, crear un motor de prova i executar-lo. Malauradament, hem intentat aquesta opció diverses vegades i, en tots els casos, ens ha aparegut un error al terminal.

4.5.3.Instal·lació de MTradumàtica

La instal·lació d'aquesta eina ha resultat la més senzilla de totes, ja que només cal seguir les passes que ens indica el *README* de la pàgina a Github (Tradumatica, 2017):

- Obrim un terminal en una carpeta buida i ens descarreguem MTradumàtica

```
git clone https://github.com/tradumatica/mtradumatica.git –
recursive
```

- Entrem al directori on ens hem descarregat l'eina

```
cd mtradumatica
```

- Instal·lem els prerequisits necessaris

```
sudo MTRADUMATICADIR/scripts/run-as-root.sh
```

- Instal·lem la màquina

```
sudo MTRADUMATICADIR/scripts/install.sh
```

Un cop tinguem la màquina instal·lada cada vegada que la vulguem utilitzar caldrà utilitzar la comanda `sudo MTRADUMATICADIR/scripts/startup.sh` i connectar-se a l'enllaç que se'ns indica (http://localhost:8080/).

4.5.4.Conclusions de la instal·lació de les eines

Aquests apartats inclouen les nostres opinions sobre cadascuna de les eines instal·lades. La informació més rellevant, però, es repetirà als resultats i a les conclusions del treball.

4.5.4.1.Conclusions de la instal·lació de MTTT

En primer lloc, creiem que a la pàgina de la documentació oficial s'hauria d'especificar que, com a requisits bàsics, cal tenir instal·lats Moses i Python per a l'execució d'aquest programa. En el cas de Moses seria molt útil proporcionar un vincle a la seva pàgina oficial i una mínima presentació sobre per a què serveix o per a què MTTT ho necessita.

També creiem que s'hauria d'explicar que l'eina només treballa amb tres combinacions lingüístiques, ja que, fins que no hem finalitzat la seva instal·lació, no hem pogut esbrinar aquesta informació fonamental.

Finalment, creiem que, encara que tingui interfície gràfica, seria molt útil que, un cop instal·lada, es pogués executar mitjançant un botó (o semblant) per no haver d'obrir un terminal cada vegada que s'utilitzi.

4.5.4.2.Conclusions de la instal·lació de ModernMT

Primerament, creiem que és un aspecte positiu que aquesta eina es pugui instal·lar de diverses maneres, ja que permet a l'usuari escollir quin mètode li és més convenient. També creiem que és positiu que permeti utilitzar traducció neuronal.

Per contra, per utilitzar la funcionalitat neuronal es necessiten unes característiques molt específiques (targeta gràfica amb CUDA) i, com en el cas anterior es necessiten uns requisits mínims (Python, versió de Java 8 o superior, etc.) per poder executar el programa. La principal diferència, però, és que les instruccions proporcionen els recursos per acomplir aquests mínims (les ordres per actualitzar la versió de Java, per exemple).

Finalment, considerem que a les instruccions de Github s'hauria de remarcar que aquesta eina només funciona amb nou combinacions lingüístiques diferents (com sí que ho fa la pàgina oficial).

4.5.4.3.Conclusions de la instal·lació de MTradumàtica

Cal destacar que aquesta eina ens ha donat diversos errors al llarg del seu procés d'instal·lació. Com vam comunicar-ho als desenvolupadors a través del nostre tutor del treball, però, aquests van modificar les instruccions del *README* i, finalment, vam poder instal·lar correctament la màquina. També volem subratllar que, tot i tractar-se d'una versió d'escriptori, es necessita accés a internet per poder-la utilitzar.

En segon lloc, creiem que es tracta del procés d'instal·lació més senzill de tots els que hem descrit en aquest treball.

4.6.Entrenament dels programes de traducció automàtica

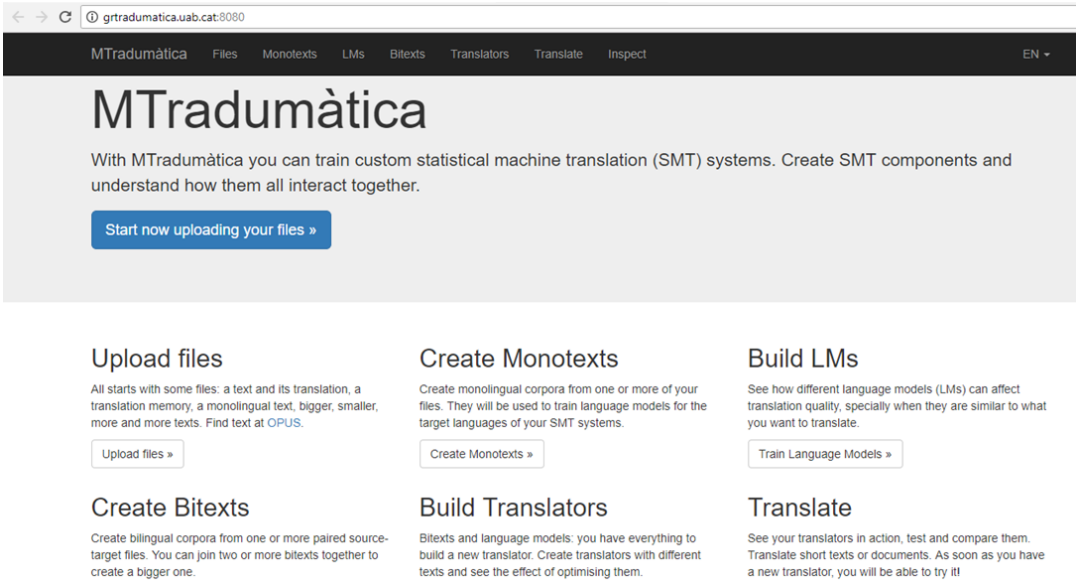
Aquesta secció descriu, pas a pas, com cal entrenar els diferents programes que analitzem en aquest treball. Cal remarcar, però, que només s'han descrit aquells que permeten l'ús de les nostres combinacions lingüístiques i que, per tant, hem utilitzat les opcions disponibles per a aquests parells de llengües. També cal recordar, com ja hem esmentat anteriorment, (vegeu l'apartat 4.3.1.1.Opus Corpus) que no s'ha realitzat optimització dels motors un cop entrenats.

Com ja s'ha fet a l'apartat anterior (vegeu l'apartat 4.5.Instal·lació dels programes de traducció automàtica) s'hi han inclòs unes breus conclusions sobre cadascuna de les eines amb què hem treballat.

4.6.1.Entrenament de MTradumàtica

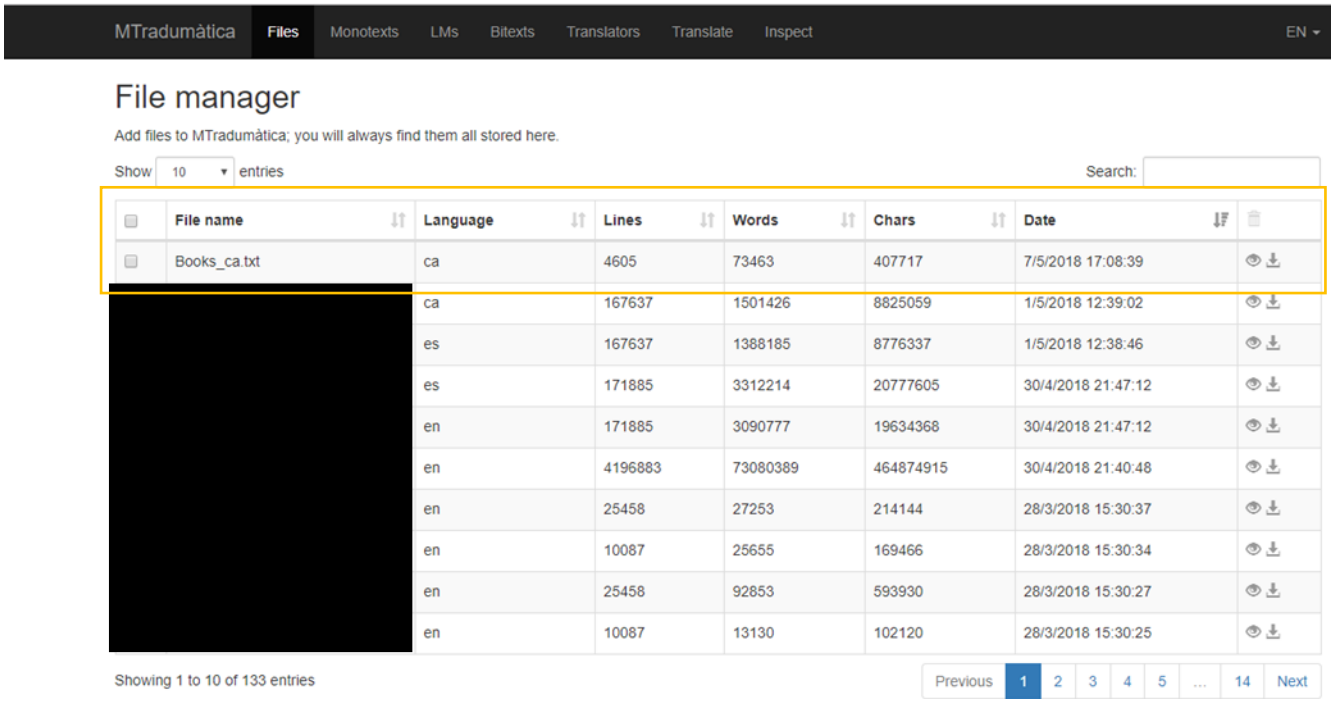
Com ja s'ha dit anteriorment (vegeu l'apartat 4.1.3.MTradumàtica), MTradumàtica té una versió d'escriptori i una versió en línia i encara que aquestes dues versions s'executin de manera diferent (per a la versió en línia només cal anar a l'adreça *www.m.tradumatica.net* i per la versió d'escriptori s'ha d'introduir la comanda `./scripts/startup.sh` i enganxar l'adreça *http://localhost:8080*) totes dues funcionen amb el mateix flux de treball:

Imatge 22. Interfície de MTradumàtica



En primer lloc, cal pujar els arxius (en format Moses o TXT en aquesta eina) a la pestanya *File Manager*. Un cop s'ha realitzat aquesta tasca, ens apareixeran juntament amb la resta d'arxius dels altres usuaris de la pàgina (vegeu imatge 23):

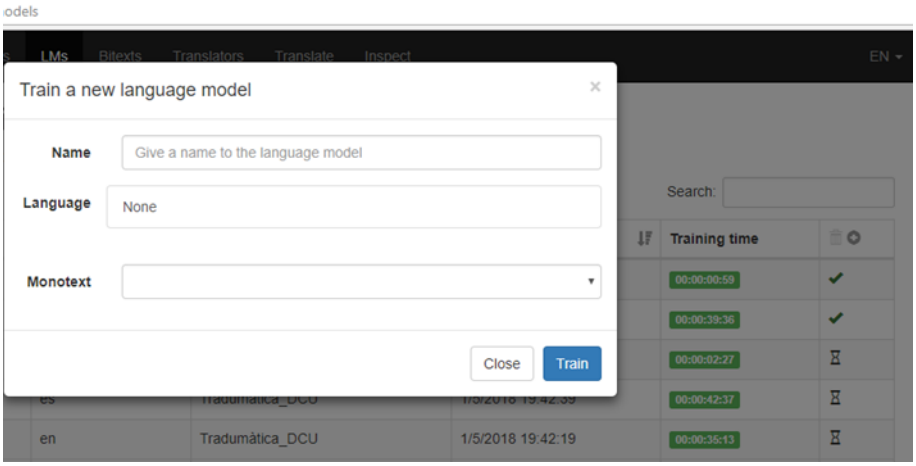
Imatge 23. Pujada d'arxius a MTradumàtica



D'aquesta manera, cal anar seguint els diferents passos que ens indica l'eina fins a arribar a l'apartat *Translators*, on s'entrenarà el motor. A diferència dels altres programes que hem descrit, MTradumàtica ens permet moure'ns lliurement per les diferents pestanyes i afegir i esborrar arxius al nostre gust. Això no vol dir, però, que aquest flux de treball no sigui molt endreçat, és a dir, no es podran entrenar els models de llengua si no hem creat els monotextos a partir dels arxius monolingües que hem pujat prèviament.

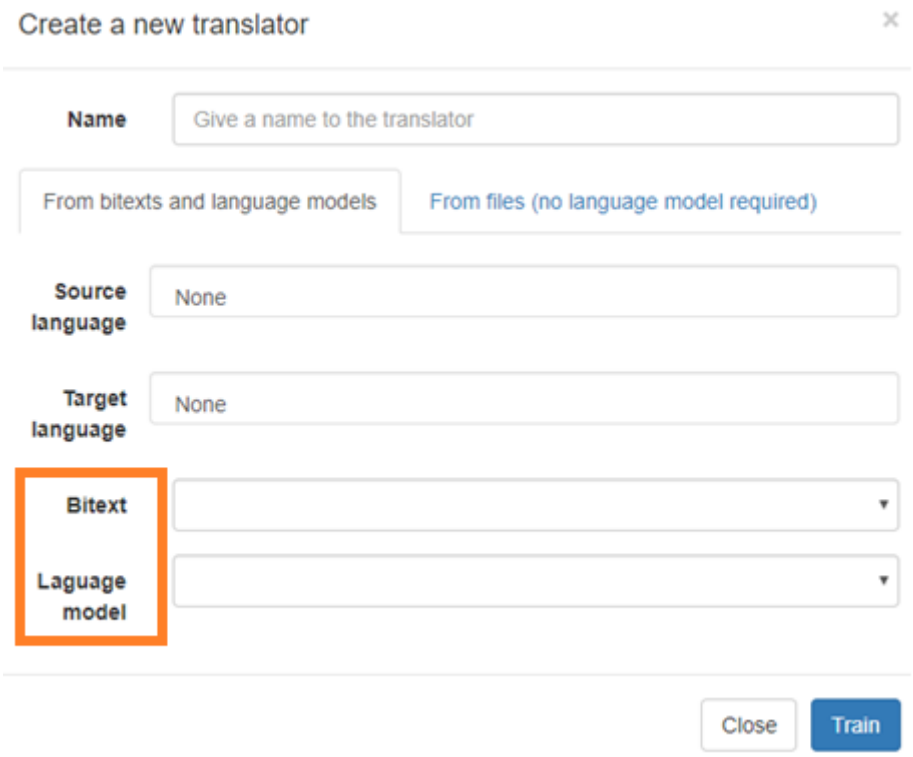
En relació al model de llengua (pestanya *LMs*), volem destacar que és un pas que en les altres eines que hem entrenat no es troba definit de manera explícita (creen el model de manera automàtica). D'aquesta manera, però, podem aprofitar (o no) els recursos bilingües en llengua d'arribada (és a dir, els Moses en català que pertanyen al corpus bilingüe) per tal de millorar els resultats del motor:

Imatge 24. Entrenament del model de llengua a MTradumàtica

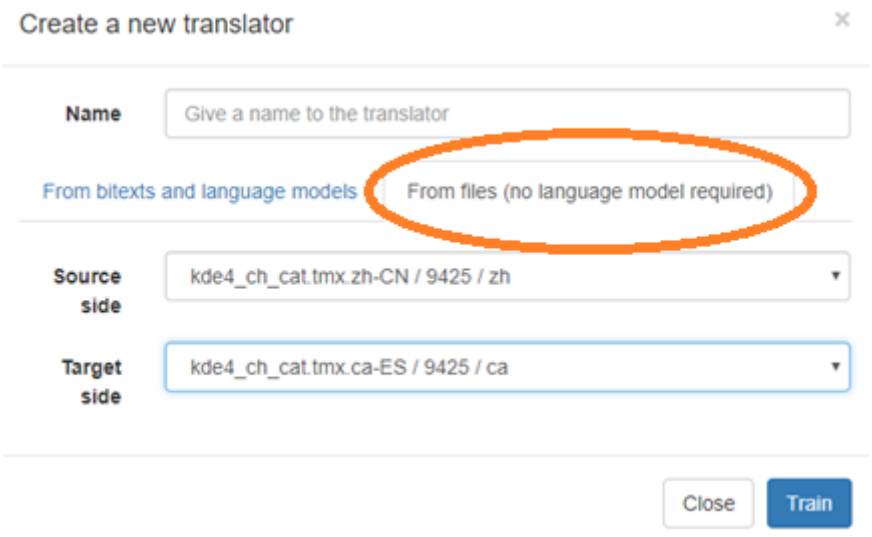


Pel que fa a la pestanya *bitexts*, és on cal verificar els arxius en llengua d'origen i destí que es corresponen entre ells. Un cop realitzada aquesta tasca ja podem passar a la pestanya *Translators*. Els motors, però, es poden entrenar de dues maneres diferents: o bé utilitzant el model de llengua i els bitextos creats anteriorment o bé sense model de llengua, basant-nos en dos arxius Moses del corpus bilingüe:

Imatge 25. Entrenament de MTradumàtica amb model de llengua



Imatge 26. Entrenament de MTradumàtica sense model de llengua



Una altra característica de l'entrenament és que, un cop finalitzat (ha trigat poc més d'una hora amb cadascun dels corpus), MTradumàtica permet optimitzar el motor resultant (botó *Optimize*).

4.6.2.Entrenament de LetsMT

Aquesta eina ens permetia, en un principi, entrenar utilitzant recursos en format Moses i en format TMX. Per aquest motiu, i per al propòsit d'aquest treball, s'ha intentat entrenar amb tots dos tipus de formats.

Primerament, cal situar-se a la pestanya *MT Systems* i anar a l'opció *Create MT System*. Allà, separats per pestanyes, es troben tots els passos que cal anar seguint per entrenar el motor (des de la selecció de llengües fins a l'avaluació automàtica de la traducció). En total, la plataforma distingeix 8 etapes diferents:

Taula 7. Procés d'entrenament de LetsMT

Pestanya de LetsMT	**Funció de la pestanya**
System Properties	Introduir les dades bàsiques del motor
Parallel Corpora	Pujar el corpus paral·lel
Monolingual Corpora	Pujar el corpus monolingüe
Translation exceptions (optional)	Pujar una sèrie d'excepcions que el motor ha de tenir en compte a l'hora de traduir
Terminology (optional)	Pujar terminologia concreta
Advanced options (optional)	Personalitzar (o no) la fase d'automatització del motor
Training	Entrenar el motor
Evaluation	Avaluar la qualitat del motor de manera automàtica

Pel que fa als arxius que s'han pujat, LetsMT calcula el total de segments en corpus paral·lel i monolingüe i els mostra durant el procés d'entrenament:

Taula 8. Corpus a LetsMT

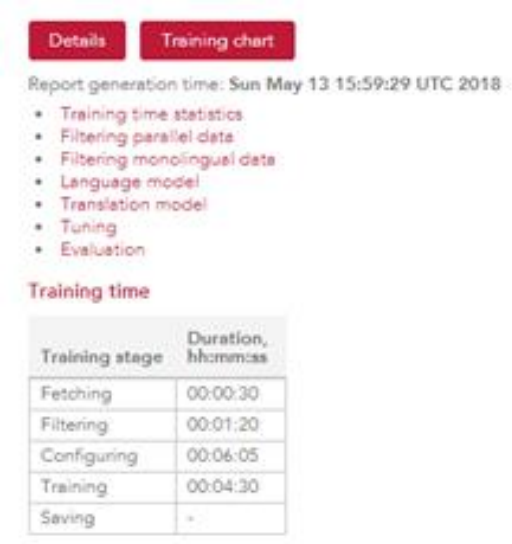
	Xinès-català (segments)	Francès-català (segments)
Corpus paral·lel	327.295	811.869
Corpus monolingüe	507.559	992.505

Ens cal subratllar que l'eina permet utilitzar els recursos públics que es trobin en la mateixa combinació lingüística que el motor que volem entrenar. Malauradament, la versió de prova que hem utilitzat permet un màxim de segments per a l'entrenament i, per tant, no hem pogut utilitzar aquests recursos per millorar el motor resultant.

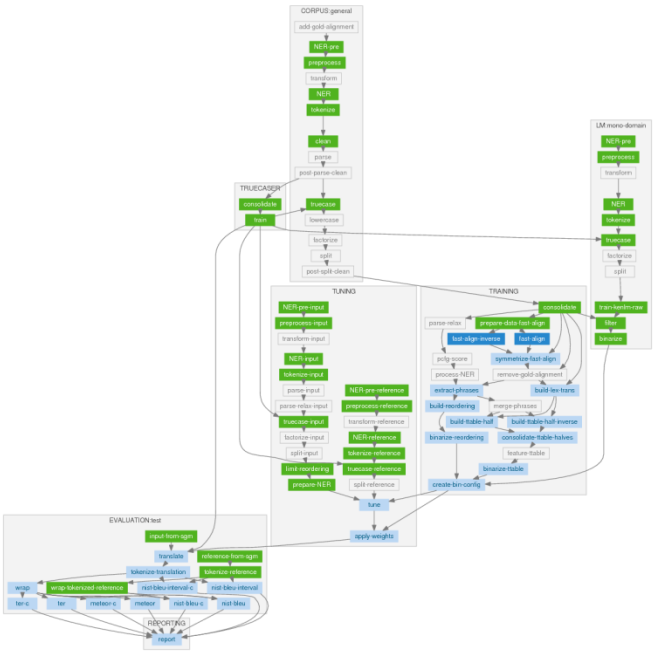
També cal destacar que, si s'intenta entrenar amb arxius en format Moses, el programa detecta aquests arxius com a corpus monolingüe i no com a corpus bilingüe, cosa que ens ha obligat a utilitzar, en totes dues combinacions lingüístiques, la versió del corpus en TMX.

Finalment, pel que fa a l'entrenament *per se*, que ha trigat menys de dues hores en tots dos casos, la pàgina ens ofereix un informe i una gràfica per poder veure el seu progrés en temps real:

Imatge 27. Informe de l'entrenament de LetsMT



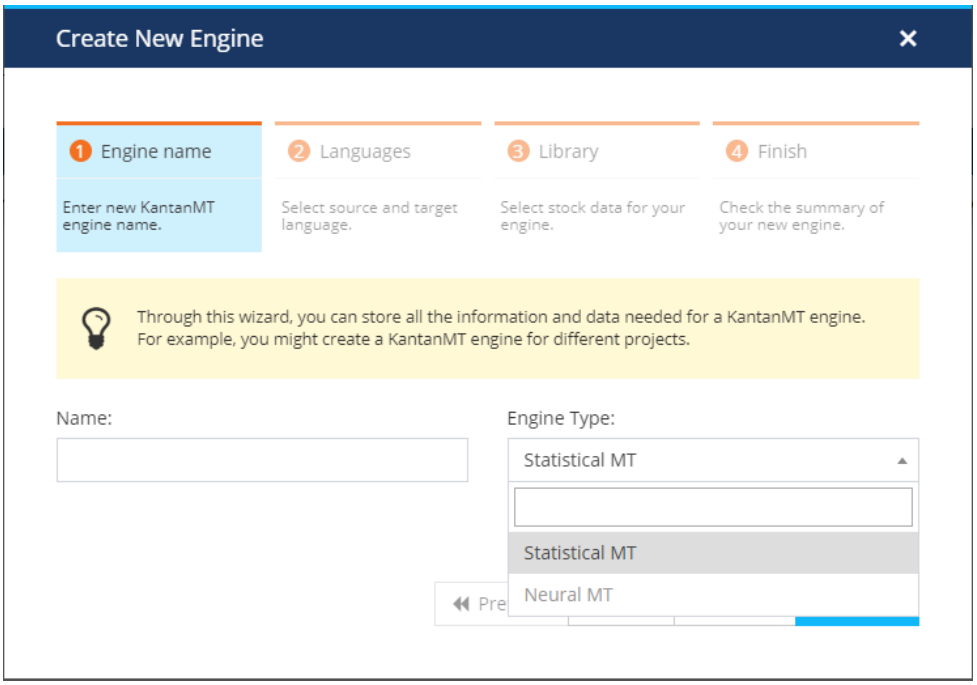
Imatge 28. Gràfica de l'entrenament de LetsMT



4.6.3.Entrenament de KantanMT

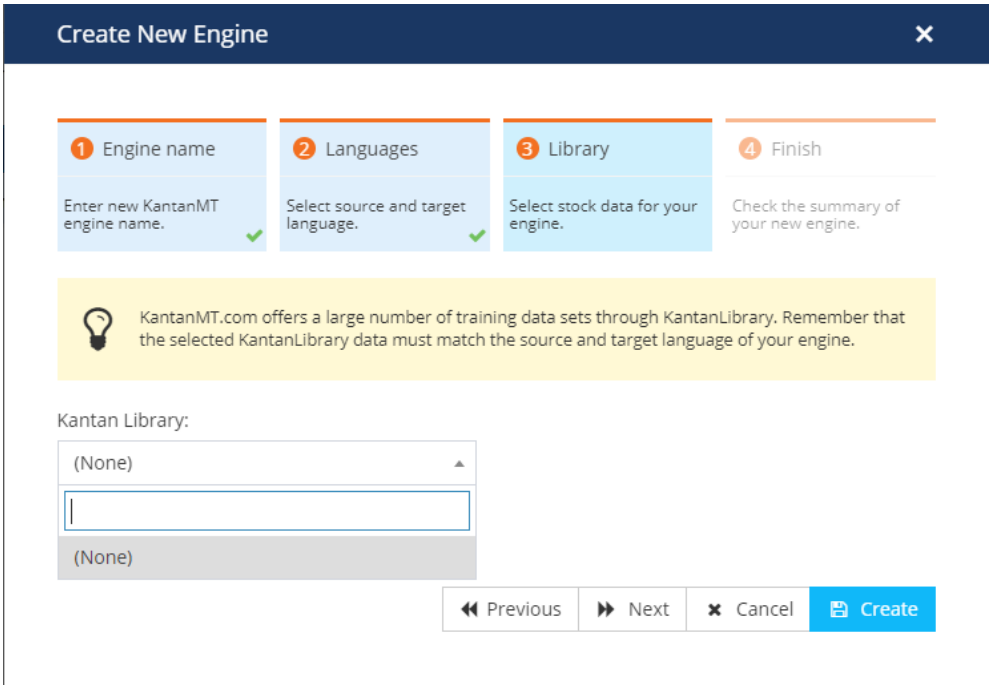
L'entrenament d'aquesta eina s'ha realitzat amb la versió TMX dels nostres corpus, ja que, aquest sistema, no accepta el format Moses. En primer lloc, però, cal crear, des de l'opció *Engines* a la pestanya *Dashboard*, un motor de traducció:

Imatge 29. Creació d'un nou motor amb KantanMT



Durant aquest primer pas, hem d'indicar el nom i el tipus de motor que volem entrenar (encara que, la versió per a docents només permet utilitzar TAE). També cal que indiquem les llengües amb què es treballarà i si es volen utilitzar corpus públics de Kantan (amb les nostres combinacions lingüístiques, però, no hem pogut aprofitar aquesta possibilitat):

Imatge 30. Llibreries de Kantan



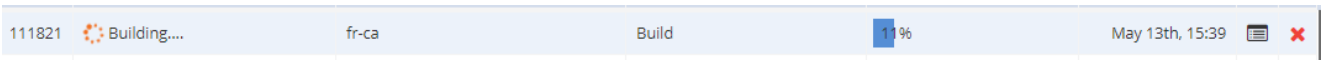
Un cop s'ha creat el motor, cal, des de l'opció *Training* arrossegar els arxius amb què s'entrenarà i clicar l'opció *build*. Aquest sistema també ofereix la possibilitat de pujar als motors arxius ja posteditats (opció *Adapt*) i de personalitzar les regles d'entrenament (opció *Rule Editor*). Per manca de temps i de coneixements específics d'aquests apartats, però, ens hem limitat a entrenar amb les característiques que ofereix KantanMT per defecte. Així, els arxius que s'han pujat per a les diferents combinacions lingüístiques han sigut els següents:

Taula 9. Arxius per l'entrenament de KantanMT

Xinès-català (TMX)	Francès-català (TMX)	Monolingües en català (TXT)
fr_zh_2018	fr_ca_2018	Books_ca
fr_zh_2016	fr_ca_2016	GlobalVoices_ca
Gnome_fr_ca	Gnome_fr_ca	Tatoeba_ca
KDE_zh_ca	KDE_fr_ca	TMX_softacatala_ca
Out	Ubuntu_fr_ca	
Ubuntu_zh_ca		

Finalment, a mesura que el procés d'entrenament es va completant, es pot veure el seu progrés a la barra de la pestanya *Jobs* (cal destacar que, en tots dos casos, ha trigat poc més d'una hora a entrenar-se):

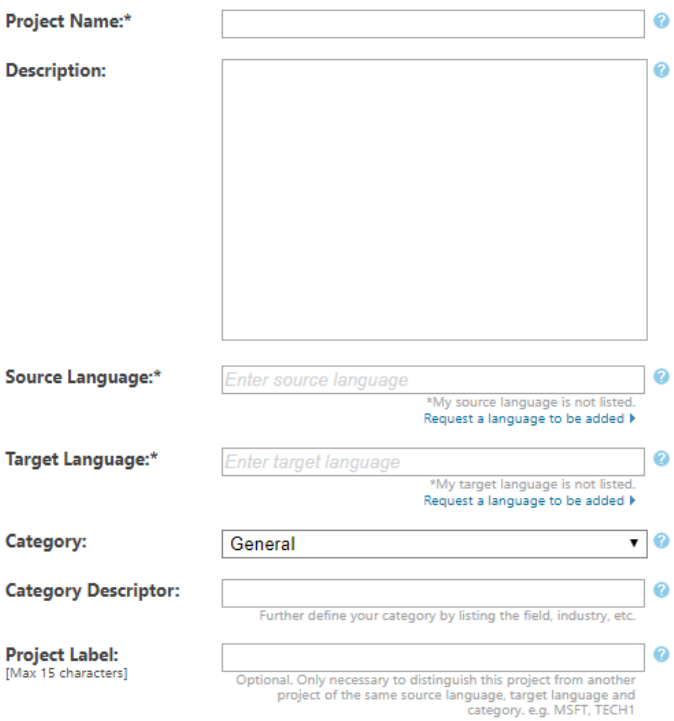
Imatge 31. Progrés de l'entrenament amb KantanMT



4.6.4.Entrenament de Microsoft Translator Hub

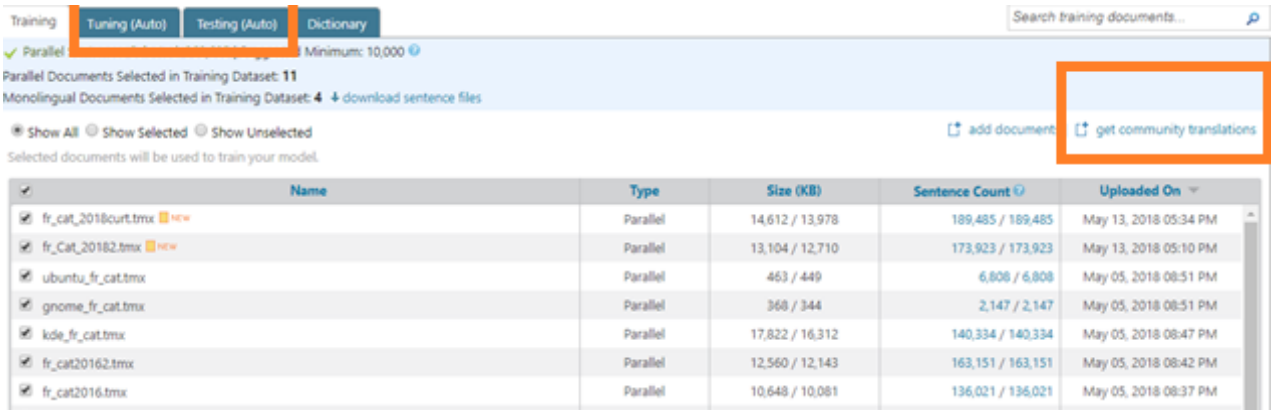
La darrera eina que hem entrenat ens permetia, teòricament, l'entrenament bilingüe dels motors tant amb format TMX com amb format Moses. Així, en primer lloc, ens cal anar a la pestanya *Projects*, a l'opció + *Add Project*. D'aquesta manera, podem definir les configuracions bàsiques del motor que volem entrenar:

Imatge 32. Configuracions dels motors amb Microsoft Translation Hub



Un cop creat, ens apareixerà el projecte a la pestanya *Projects*. Podem fer-hi un clic per definir quins arxius pujarem per entrenar-lo:

Imatge 33. Arxius per a l'entrenament de Microsoft Translation Hub



Com es pot veure a la captura (vegeu imatge 33) a més dels nostres propis arxius, també podem accedir a traduccions de la comunitat Microsoft. Lamentablement, com en el cas de KantanMT (vegeu l'apartat 4.6.3.Entrenament de KantanMT), no existeixen recursos per a les nostres combinacions lingüístiques. A la imatge també es pot veure que Microsoft Translator Hub permet escollir entre la selecció manual i la selecció automàtica (opcions *Tuning* i *Testing*) de les frases per al procés d'optimització i per al procés de *testing*.

Pel que fa a la pujada dels arxius, com en el cas de LetsMT (vegeu l'apartat 4.6.2.Entrenament de LetsMT), Microsoft interpreta el format Moses com un format monolingüe i, per tant, per a l'entrenament d'aquesta eina, hem hagut d'utilitzar la versió en TMX del nostre corpus. A més, no permet pujar arxius amb un pes major a 10 MB, cosa que ens ha obligat a dividir els TMX superiors a aquest pes.

Finalment, pel que fa a l'entrenament, es pot iniciar fent un clic a la pestanya *Start Training* i triga entre 4 i 12 hores en finalitzar (en el nostre cas unes 6 hores). Com amb els programes anteriors, l'eina permet fer un seguiment d'aquest progrés.

4.6.5.Conclusions de l'entrenament dels motors

De la mateixa manera que a l'apartat anterior, aquesta secció pretén destacar els aspectes més rellevants del procés d'entrenament dels motors del programari. De la mateixa manera que abans, però, el contingut imprescindible apareix altre vegada als resultats d'aquest treball (vegeu l'apartat 5.Resultats).

4.6.5.1.Conclusions de l'entrenament de MTradumàtica

Creiem que aquesta és l'eina que més ens ha permès personalitzar el procés d'entrenament. En primer lloc, perquè permet escollir amb quins arxius crear el model de llengua (cosa que les altres eines que hem descrit en aquest treball fan de manera automàtica). En segon, perquè permet escollir com es volen entrenar els motors[1] (amb model de llengua o sense). Finalment, perquè, un cop entrenat el motor, ens permet optimitzar-lo.

Malauradament, no permet utilitzar arxius en format TMX, cosa que obliga a l'usuari a utilitzar format Moses[1]. També creiem que seria positiu que la versió d'escriptori pogués operar sense accés a internet.

[1]1: En el moment de tancar aquest treball (juny 2018), s'anunciava una nova versió del programa que permet el processament de TMX.

4.6.5.2.Conclusions de l'entrenament de LetsMT

Creiem que és positiu que l'eina permeti entrenar els motors amb terminologia i excepcions de traducció específiques. També ens ha agradat que indiqui de manera molt clara els segments que s'han pujat.

El punt negatiu que volem destacar és que no indica de forma específica que els arxius en format Moses es tindran en compte pels corpus monolingüe i no pas pel bilingüe (vegeu l'apartat 4.1.4.LetsMT) i, de fet, no ha sigut fins al moment de voler començar a entrenar el sistema que ens hem adonat d'aquesta realitat.

4.6.5.3.Conclusions de l'entrenament de KantanMT

Considerem que es tracta d'una eina força intuïtiva i considerem molt positiu que hagi entrenat els nostres motors en poc més d'una hora (uns 70 minuts). El principal problema que ens planteja aquesta eina, però, és que no sabem com tracta el model de llengua, ja que, a diferència de MTradumàtica (en què podem escollir) i LetsMT (en què l'eina utilitza de manera automàtica el contingut del corpus bilingüe per a aquest model) no ens permet diferenciar el contingut del corpus bilingüe i monolingüe.

4.6.5.4.Conclusions de l'entrenament de Microsoft Translator Hub

Creiem que, d'una banda, tot i ser l'eina en línia que més ha trigat a entrenar, també és la més permissiva pel que fa al corpus, ja que, a diferència de les anteriors, només recomana 10.000 frases bilingües per entrenar (vegeu imatge 33). També creiem que és positiu que, en cas de voler treballar amb una llengua que no es troba disponible a la llista de selecció, Microsoft ofereixi la possibilitat d'afegir-la (vegeu l'apartat 4.1.6.Microsoft Translator Hub).

De l'altra, hem hagut de modificar els TMX per poder-los pujar per a la fase d'entrenament i, com en el cas de LetsMT, no especifica que el format Moses s'interpretarà com a arxiu monolingüe.

5.Resultats

Aquest apartat es divideix, al seu torn, en dos subapartats que resumeixen tota la part pràctica d'aquest TFM. En primer lloc, es presenten les característiques de cadascuna de les eines, agrupades en taules. Aquestes intenten resumir tots els continguts que hem descobert durant els processos d'investigació, instal·lació i entrenament dels programes. No contenen valoracions personals, encara que sí que destaquen les característiques que hem considerat que cal valorar a l'hora de triar un programa o un altre.

El segon subapartat d'aquests resultats conté l'avaluació automàtica de les traduccions de les eines que hem pogut entrenar. En un principi, volíem utilitzar el recurs Asiya, que conté diverses mètriques d'avaluació diferents, però, malauradament, sembla que no funciona correctament (hem intentat utilitzar tant el recurs en línia com la instal·lació d'escriptori). Per aquest motiu, i per tal de poder

valorar aquests resultats quant a qualitat de la traducció, s'ha utilitzat només la mètrica BLEU (vegeu l'apartat 3.3.1.Mètodes d'avaluació automàtica de TA) amb una altra eina independent.

5.1.Anàlisi de les característiques de les eines

Machine Translation Training Tool		
Característiques	En quin sistema operatiu s'executa?	En Linux i en Windows
	S'utilitza en local o en línia?	En local i en línia
	Es necessita connexió a Internet?	Depèn de si s'utilitza la versió d'escriptori o la versió en línia
	Té interfície gràfica?	Sí
	Quins formats accepta per a l'entrenament?	Suposem que Moses
	Quins formats accepta per a la traducció?	Suposem que Moses
	És programari lliure o privatiu?	Lliure
	Quin tipus de traducció automàtica permet?	Estadística
	Té versió de prova?	No, no en té
	S'ha de pagar per utilitzar aquesta eina?	No, no s'ha de pagar
	Quines llengües suporta?	Anglès, francès i alemany
	Cal algun requisit específic per utilitzar-la?	És recomanable tenir un ordinador amb unes característiques superiors a les normals, ja que la compilació de Moses serà més ràpida.També cal tenir Python
Instal·lació	Hi ha més d'una manera d'instal·lar-la?	Sí, depenent del sistema operatiu
	Cal seguir molts passos?	Sí, i al repositori Github no hi ha una explicació clara de quins passos s'han de seguir per instal·lar-la
	La instal·lació és ràpida?	No, no ho és. La compilació de Moses és força lenta
	Cal coneixements específics per instal·lar-la?	Sí, s'ha de tenir un mínim de coneixements de funcionament del terminal i de programació
	És intuïtiva la instal·lació?	No, no ho és. La manca d'instruccions concretes no ajuda. A més, l'ús del terminal la dificulta.

	Les instruccions d'instal·lació són específiques?	No, no ho són. Cal buscar a diferents pàgines per poder completar la instal·lació
	Es pot instal·lar en un ordinador amb unes característiques comunes?	Sí, però es recomana un ordinador amb un bon processador

ModernMT		
Característiques	En quin sistema operatiu s'executa?	En Ubuntu
	S'utilitza en local o en línia?	En local
	Es necessita connexió a Internet?	No
	Té interfície gràfica?	No
	Quins formats accepta per a l'entrenament?	XML, TMX i TXT. No sabem si n'accepta més
	Quins formats accepta per a la traducció?	Sembla que TXT. No sabem si n'accepta més
	És programari lliure o privatiu?	Lliure, però hi ha una versió per a empreses
	Quin tipus de traducció automàtica permet?	Estadística i neuronal
	Té versió de prova?	No, no en té
	S'ha de pagar per utilitzar aquesta eina?	Si es vol fer servir la traducció automàtica diàriament com a empresa, sí que hi ha un servei que s'ha de pagar. Aquest servei inclou el motor de traducció automàtica ja entrenat i a punt per a executar-lo
	Quines llengües suporta?	El programa es pot configurar de manera que faci totes les combinacions però s'ofereixen l'anglès, el castellà, l'italià, el portuguès, l'alemany, el francès, el neerlandès, el rus, l'àrab i el xinès en la versió per empreses
	Cal algun requisit específic per utilitzar-la?	Cal una plataforma de x86_64, un mínim de 5 GB d'espai a l'ordinador i es recomana tenir una GPU amb una memòria de 8 GB i una targeta CUDA (aquesta última en cas que es faci servir la traducció neuronal). També cal tenir Python i una versió de Java 8 o superior

Instal·lació	Hi ha més d'una manera d'instal·lar-la?	Sí, n'hem trobat 3
	Cal seguir molts passos?	Sí
	La instal·lació és ràpida?	No, no ho és
	Cal coneixements específics per instal·lar-la?	Sí, s'ha de tenir un mínim de coneixements de funcionament del terminal i de programació
	És intuïtiva la instal·lació?	Sí, sí que ho és
	Les instruccions d'instal·lació són específiques?	Sí, hi ha suficients indicacions però, si no es tenen coneixements de programació, és complicat seguir-les
	Es pot instal·lar en un ordinador amb unes característiques comunes?	Sí, però es recomana fer-ho en un servidor

MTradumàtica		
Característiques	En quin sistema operatiu s'executa?	En Ubuntu. Si es fa servir la versió en línia, es pot executar en qualsevol sistema
	S'utilitza en local o en línia?	En local i en línia
	Es necessita connexió a Internet?	Sí, es necessita en les dues versions
	Té interfície gràfica?	Sí
	Quins formats accepta per a l'entrenament?	Moses, TXT
	Quins formats accepta per a la traducció?	HTML, TXT, DOCX, PPTX, XLSL ODT, ODS, ODP i Moses
	És programari lliure o privatiu?	Lliure
	Quin tipus de traducció automàtica permet?	Estadística
	Té versió de prova?	No, no en té
	S'ha de pagar per utilitzar aquesta eina?	No
	Quines llengües suporta?	Vegeu Annex 3
	Cal algun requisit específic per utilitzar-la?	Connexió a internet
Instal·lació	Hi ha més d'una manera d'instal·lar-la?	Sí, també es poden utilitzar contenidors Docker
	Cal seguir molts passos?	No
	La instal·lació és ràpida?	No, no ho és
	Cal coneixements específics per instal·lar-la?	Sí, calen coneixements de funcionament del terminal

	És intuïtiva la instal·lació?	Sí
	Les instruccions d'instal·lació són específiques?	Sí
	Es pot instal·lar en un ordinador amb unes característiques comunes?	Sí
Entrenament	L'entrenament és intuïtiu?	Sí
	S'han de modificar els arxius per entrenar l'eina?	No
	Es pot veure la quantitat de segments que es pugen?	Sí, es poden veure els segments de cada part de l'entrenament (documents inicials, monotextos, etc.)
	Quant triga a entrenar-se amb els nostres corpus?	Poc més d'una hora
	Té avaluació automàtica de la qualitat integrada?	No, no en té
	Es recomana un mínim de segments per entrenar el motor?	No, no és necessari, però sí recomanable
	Es pot utilitzar el contingut del corpus bilingüe per al monolingüe?	Sí

LetsMT		
Característiques	En quin sistema operatiu s'executa?	En qualsevol
	S'utilitza en local o en línia?	En línia
	Es necessita connexió a Internet?	Sí
	Té interfície gràfica	Sí
	Quins formats accepta per a l'entrenament?	TMX, XLIFF, XLZ, Moses, PDF, DOC, TXT, tar, zip i tgz
	Quins formats accepta per a la traducció?	DOC, DOCX, XLSX, PPTX, ODT, ODP, ODS, HTML, HTM, XHTML, XHT, TXT, TMX, XLF, XLIF, XLIFF, SDLXLIFF, TTX, RTF en codificació UTF-8 o UTF-16
	És programari lliure o privatiu?	Privatiu
	Quin tipus de traducció automàtica permet?	Estadística
	Té versió de prova?	Sí
	S'ha de pagar per utilitzar aquesta eina?	Sí
	Quines llengües suporta?	Vegeu Annex 4
	Cal algun requisit específic per utilitzar-la?	Crear-se un compte a la pàgina web de https://www.letsmt.eu/
Entrenament	L'entrenament és intuïtiu?	Sí

	S'han de modificar els arxius per entrenar l'eina?	No
	Es pot veure la quantitat de segments que es pugen?	Sí
	Quant triga a entrenar-se amb els nostres corpus?	Menys de dues hores
	Té avaluació automàtica de la qualitat integrada?	Sí, disposa de BLEU, NIST, TER i METEOR
	Es recomana un mínim de segments per entrenar el motor?	Sí, es recomana que per al corpus bilingüe es disposi d'un milió de segments
	Es pot utilitzar el contingut del corpus bilingüe per al monolingüe?	Sí. Agafa el contingut del corpus bilingüe de manera automàtica

KantanMT		
Característiques	En quin sistema operatiu s'executa?	En qualsevol
	S'utilitza en local o en línia?	En línia
	Es necessita connexió a Internet?	Sí
	Té interfície gràfica	Sí
	Quins formats accepta per a l'entrenament?	TMX, TXT, XLSX, TBX, DOCX, ZIP, GZ, PDF
	Quins formats accepta per a la traducció?	XLIFF TTX TXML TMX EXP XLZ MQXLZ .sub.trg XLF - CAD DOCX PDF ODT DITA XML INX IDML HTML SVG NovaDoc MonTag XML AborText XML TXT XLSX
	És programari lliure o privatiu?	Privatiu

	Quin tipus de traducció automàtica permet?	Estadística i neuronal (encara que la segona no es pot utilitzar a la versió per a docents)
	Té versió de prova?	No
	S'ha de pagar per utilitzar aquesta eina?	Sí
	Quines llengües suporta?	Vegeu Annex **5**
	Cal algun requisit específic per utilitzar-la?	Tenir un compte a https://www.kantanmt.com/ i haver pagat el servei de traducció automàtica
Entrenament	L'entrenament és intuïtiu?	Sí
	S'han de modificar els arxius per entrenar l'eina?	No
	Es pot veure la quantitat de segments que es pugen?	No
	Quant triga a entrenar-se amb els nostres corpus?	Poc més d'una hora
	Té avaluació automàtica de la qualitat integrada?	Sí, disposa de BLEU, TER, F-Measure, Word Count, Unique WC i Mono WC
	Es recomana un mínim de segments per entrenar el motor?	No, no és necessari, però sí recomanable
	Es pot utilitzar el contingut del corpus bilingüe per al monolingüe?	No ho sabem. Suposem que sí i que es fa automàticament, ja que ha tingut molts bons resultats (vegeu l'apartat 5.2.Resultats de l'avaluació automàtica dels motors)

Microsoft Translator Hub		
Característiques	En quin sistema operatiu s'executa?	En qualsevol
	S'utilitza en local o en línia?	En línia
	Es necessita connexió a Internet?	Sí
	Té interfície gràfica	Sí
	Quins formats accepta per a l'entrenament?	TMX, XLF, XFIFF, LCL, XLSX, TXT, HTML, PDF, DOCX, ALIGN, ZIP, GZ i TGZ
	Quins formats accepta per a la traducció?	TMX, XLIFF, TXT, HTML, DOCX, XLSX i PDF
	És programari lliure o privatiu?	Privatiu
	Quin tipus de traducció automàtica permet?	Estadística i neuronal
	Té versió de prova?	Sí
	S'ha de pagar per utilitzar aquesta eina?	Sí

	Quines llengües suporta?	Per a la traducció estadística suporta les llengües següents: Afrikaans Alemany Anglès Àrab Bangla Bosni (llatí) Búlgar Cantonès (tradicional) Català Coreà Crioll haitià Croat Danès Eslovac Eslovè Espanyol Estonià Fijià Filipí Finlandès Francès Gal·lès Grec Hebreu Hindi Hmong Daw Holandès Hongarès Indonesi Islandès Italià Japonès Kiswahili Klingon Klingon (plqaD) Letó Lituà Malai Malgache Maltès Noruec Persa Polonès Portuguès Queretaro Otomi Romanès Rus

		Samoà Serbi Suec Tahití Tailandès Tamil Tongan Turc Txec Ucraïnès Urdu Vietnamita Xinès Yucatec Maya
	Cal algun requisit específic per utilitzar-la?	Tenir un compte d'Outlook
Entrenament	L'entrenament és intuïtiu?	Sí
	S'han de modificar els arxius per entrenar l'eina?	Sí, no poden pesar més de 10 MB
	Es pot veure la quantitat de segments que es pugen?	Sí
	Quant triga a entrenar-se amb els nostres corpus?	Unes 6 hores
	Té avaluació automàtica de la qualitat integrada?	Sí, disposa de BLEU
	Es recomana un mínim de segments per entrenar el motor?	Sí, unes 10.000 línies per al corpus bilingüe
	Es pot utilitzar el contingut del corpus bilingüe per al monolingüe?	Ho desconeixem

5.2.Resultats de l'avaluació automàtica dels motors

Com ja hem comentat en la introducció d'aquest apartat (vegeu l'apartat **5.Resultats**), hem utilitzat la mètrica BLEU per valorar els diferents resultats de cadascuna de les eines que hem entrenat amb les dues combinacions lingüístiques. La següent taula (vegeu taula 10) les resumeix:

Taula 10. Avaluació automàtica amb la mètrica BLEU

Eina	Puntuació corpus xinès-català	Puntuació corpus francès-català
MTradumàtica (model de llengua només amb corpus monolingüe	0,01	23,98
MTradumàtica (model de llengua amb corpus monolingüe i bilingüe)	0,01	25,27
LetsMT	-	-
KantanMT	1,19	72,76
Microsoft Translator Hub	9,27	26,07

Com es pot veure a la taula, en tots els casos, el corpus en francès-català ha obtingut uns resultats molt millors que el de xinès-català. De fet, en el cas de MTradumàtica, la quantitat de xinès als textos ja traduïts era superior a la de català. Deixant de banda aquest resultat, ens trobem amb diferències notables entre els diferents programes.

Així, per exemple, en el cas de KantanMT, en francès s'ha aconseguit una valoració superior al 50%, és a dir, que en aquesta combinació, "less post-editing will be required to achieve publishable translation quality" (www.kantanmt.com : Description of BLEU Score for MT Quality., s.d.). Desconeixem el motiu pel qual aquesta eina en aquesta combinació ha aconseguit tan bons resultats i, en canvi, en xinès-català, es troba en segona posició. Sospitem que el tractament dels recursos pel model de llengua (agafa el contingut del corpus bilingüe pel corpus monolingüe) s'hi troba relacionat, ja que, en el cas de francès-català, la suma de tots dos corpus arriba als 992.505 segments monolingües i, en xinès-català, només fins als 507.559.

La següent eina millor valorada en general és Microsoft Translator Hub i, finalment, MTradumàtica, que s'ha entrenat de dues maneres: utilitzant només el corpus monolingüe i utilitzant el contingut del corpus bilingüe per al model de llengua. En el cas de LetsMT, no se n'han pogut valorar els resultats, ja que, la versió que hem utilitzat (vegeu l'apartat 4.1.4.LetsMT) només permet traduir els deu primer segments d'un document (una quantitat insuficient per poder avaluar-ne la qualitat).

6.Conclusions

En primer lloc, volem destacar que ens ha sorprès la manca de recursos disponibles per al corpus xinès-català. Així, tot i haver creat un mètode eficient per augmentar els segments inicials obtinguts a través d'Opus corpus, la combinació lingüística conté llengües amb recursos tan poc desenvolupats, que, o bé no existeix prou material disponible al web per a aquesta combinació, o bé el material existent no està prou actualitzat.

Pel que fa a les eines que requereixen instal·lació, com ja hem remarcat en apartats anteriors (vegeu l'apartat 4.5.Instal·lació dels programes de traducció automàtica) creiem que aquest tipus de programes requereixen uns coneixements d'informàtica superiors als que sol tenir un traductor autònom corrent. Així, encara que el procés d'instal·lació requereixi pocs passos, des d'un punt de vista de l'optimització del temps, la instal·lació no és pràctica. A més, alguns no disposen d'interfície gràfica (pensem en ModernMT), cosa que pot empitjorar l'experiència de traducció i, en certs casos, és recomanable tenir màquines amb característiques diferents de les d'un ordinador que s'utilitzi en l'àmbit d'usuari.

També creiem que tot i ser programari idealment creat per a traductors de qualsevol combinació, les restriccions que té per a certes llengües no n'afavoreixen l'ús (pensem en MTTT i ModernMT). Per exemple, un traductor de xinès a castellà no podrà utilitzar d'alguns dels programes que s'han analitzat en aquest TFM.

Quant als formats, considerem que és positiu no haver hagut de modificar-los excepte en casos excepcionals (com amb Microsoft Translator Hub, que no accepta arxius superiors a 10 MB) i que totes les eines accepten una quantitat decent de formats diferents, tant per a l'entrenament com per a la traducció *per se*. De la mateixa manera, també creiem que és positiu que totes les eines que s'han entrenat hagin trigat menys de sis hores en fer-ho, encara que el corpus de francès-català contingui quasi un milió de segments bilingües (una quantitat que ens sembla considerable).

Finalment, i després d'haver analitzat els resultats de l'avaluació automàtica de les traduccions, creiem que el fet d'haver hagut d'utilitzar versions de prova (amb restriccions implícites) o de no haver pogut valorar les traduccions amb més mètriques provoca que els nostres resultats pel que fa a la valoració dels motors entrenats no siguin fiables i, encara que el propòsit d'aquest TFM és analitzar les eines d'una manera general, ens hauria agradat haver pogut investigar-les una mica més. Per exemple, el motiu pel qual KantanMT ha obtingut uns resultats tan positius en la combinació francès-català.

Finalment, volem concloure afirmant que aquest treball no només ens ha servit per investigar algunes de les eines que permeten personalitzar els seus motors de TA, també ens ha ajudat a adonar-nos-en de l'estat de la qüestió dels recursos de certes combinacions lingüístiques, de la manca de material de referència de les eines que cal instal·lar des del punt de vista del traductor autònom i de la diversitat de característiques (eines privatives o lliures, en línia o en local, formats i llengües que accepten, etc.) que podem observar analitzant només sis de tots els programes d'aquest tipus existents avui dia. També ens ha servit per aprendre a posar en pràctica els diferents recursos i mètodes que hem vist al

llarg del màster (editors de notes, sistemes operatius, com dividir un TMX, expressions regulars, etc.) i per anar més enllà, desenvolupant els nostres propis recursos durant l'elaboració d'aquest TFM. En altres paraules, ens ha ajudat a sortir de la nostra *zona de confort* i, per aquest motiu, encara que els nostres resultats no siguin totalment conclusius, creiem que és un bon exemple d'un treball destinat a l'anàlisi de productes.

7.Bibliografia

Alonso, J. (2007). Els sistemes de traducció automàtica. *Llengua i ús. Revista tècnica de política lingüística*, 38, 23-32.

Aprender a programar: introducción y conceptos básicos - 1&1. (s.d.). Recuperat de https://www.1and1.es/digitalguide/paginas-web/desarrollo-web/aprender-a-programar-introduccion-y-conceptos-basicos/

Aranberri, N. (2014). Postedició, productivitat i qualitat. *Revista Tradumàtica: tecnologies de la traducció, 0*(12), 471-477.

Babych, B. (2014). Mètriques d'avaluació automatitzada de TA i les seves limitacions. *Revista Tradumàtica: Tecnologies de La Traducció, 0*(12), 464-470.

Bahdanau, D., Cho, K. i Bengio, Y. (2014). *Neuronal Machine Translation by Jointly Learning to Align and Translate*. Recuperat de http://arxiv.org/abs/1409.0473

Bouillon, P., Estrella, P., Lafuente, R., i Girletti, S. (2017). MTTT – Machine Translation Training Tool: A tool to teach MT, Evaluation and Post-editing, 18.

Catalunya, I. O. (s.d.). *Sistemes operatius monolloc*. Recuperat 29 abril 2018, de http://ioc.xtec.cat/materials/FP/Materials/2201_SMX/SMX_2201_M02/web/html/WebContent/u1/a2/continguts.html

Casacuberta, F. i Peris, A. (2017). Traducció automàtica neuronal. *Tradumàtica: traducció i tecnologies de la informació i la comunicació, 0*(15), 66-74.

Doğru, G, Martín-Mor, A. i Ortiz-Rojas, S. (2017). *MTradumàtica: Free Statistical Machine Translation Customisation for Translators*. Recuperat de https://ufal.mff.cuni.cz/eamt2017/user-project-product-papers/papers/user/EAMT2017_paper_75.pdf

Forcada, M. L. (2017). Making sense of neuronal machine translation. *Translation Spaces*, 6:2, p. 291–309.

García Varea, I. (2003). *Traducción automàtica estadística: modelos de traducción basados en máxima entropía y algoritmos de búsqueda*. Recuperat de https://ruidera.uclm.es/xmlui/handle/10578/959

Gavrila, M. i Vertan, C. (2011). *Training Data in Statistical Machine Translation– The More, the Better?* –. Recuperat de http://www.aclweb.org/anthology/R11-1077

Gehring, J., Auli, M., Grangier, D., Yarats, D. i Dauphin, Y. N. (2017). *Convolutional Sequence to Sequence Learning*. Recuperat de http://arxiv.org/abs/1705.03122

Ginestí Rosell, M., i Forcada Zubizarreta, M. L. (2009). *La traducció automàtica en la pràctica: aplicacions, dificultats i estratègies de desenvolupament. En: Caplletra : Revista Internacional de Filologia, 2009, No. 46: 43*. Recuperat de http://roderic.uv.es/handle/10550/48542

Görög, A. (2014). Quantificació i avaluació comparativa de la qualitat: el Dynamic Quality Framework de TAUS. *Revista Tradumàtica: Tecnologies de La Traducció*, *0*(12), 443-454.

Haddow, B. i Koehn, P. (2012). *Analysing the Effect of Out-of-Domain Data on SMT Systems*. Recuperat de http://www.aclweb.org/anthology/W12-3154

Hearne, M. i Way, A. (2011). *Statistical Machine Translation: A Guide for Linguists and Translators*. Recuperat de http://www.computing.dcu.ie/~away/CA446/SMTforLinguists.pdf

Introduction to ICTCLAS. (s.d.). Recuperat de http://sewm.pku.edu.cn/QA/reference/ICTCLAS/FreeICTCLAS/English.html

Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press.

Koehn, P. (2018). *Moses User Manual and Code Guide*. Recuperat de http://www.statmt.org/moses/manual/manual.pdf

Lafuente, R. (2017). *Machine Translation Training Tool (MTTT): Machine translation made easy for human translators!* Python. Recuperat de https://github.com/roxana-lafuente/MTTT /

Lopez, A. (2008). Statistical Machine Translation. *ACM Comput. Surv.*, *40*(3), 8:1–8:49.

Martín-Mor, A. (2016). *MTradumàtica: Statistical Machine Translation Customisation for Translators*. Recuperat de http://www.skase.sk/Volumes/JTI12/pdf_doc/02.pdf

Martín-Mor, A. i Peña-Irles, Víctor (2017). Creació d'un motor de TAE especialitzat en farmàcia i medicina per a la combinació romanés-castellà. Linguamática, 9(2), 45-53. https://doi.org/10.21814/lm.9.2.254

Martín-Mor, A. i Piqué, R. (2017). MTradumàtica i la formació de traductors en Traducció Automàtica Estadística. *Revista Tradumàtica: Tecnologies de La Traducció*, *0*(15), 97-115.

Martín-Mor, A., Piqué, R. i Sánchez-Gijón, P. (2016). *Tradumàtica: tecnologies de la traducció: Biblioteca de traducció i interpretació 21*. Vic: Eumo

Microsoft Corporation. (2018). *Microsoft Translator Hub User Guide*. Recuperat de https://hub.microsofttranslator.com/Help/Download/Microsoft%20Translator%20Hub%20User%20Guide.pdf

MMT: Neuronal Adaptive Machine Translation that adapts to context and learns from corrections. (2018). *ModernMT*. Recuperat de https://github.com/ModernMT/MMT

Moses - Development/GetStarted. (s.d.). Recuperat 6 maig 2018, de http://www.statmt.org/moses/?n=Development.GetStarted

ModernMT. (s.d.). Recuperat de https://www.modernmt.eu/

MTradumàtica (s.d.). Recuperat de http://grtradumatica.uab.cat:8080/index

NLPIR 汉语分词系统. (s.d.). Recuperat de http://ictclas.nlpir.org/

Nolla, F. C., i Abril, Á. P. (2017). Traducció automàtica neuronal. *Revista Tradumàtica: tecnologies de la traducció, 0*(15), 66-74.

O'Brien, S. (2012). *Towards a Dynamic Quality Evaluation Model for Translation*. Recuperat de http://www.jostrans.org/issue17/art_obrien.pdf

OPUS - an open source parallel corpus. (s.d.). Recuperat de http://opus.nlpl.eu/

Peña-Irles, V. i Martín- Mor, A. (2017). *Entrenament de motors de traducció automàtica estadística especialitzats en farmàcia i medicina entre el castellà i el romanés*. Recuperat de https://ddd.uab.cat/record/178210

Pérez, M. I. (2016). *Phrase-based statistical machine translation: explanation of its processes and statistical models and evaluation of the English to Spanish translations produced*. Recuperat de http://rua.ua.es/dspace/handle/10045/56346

Polo, L. R. (2013). Els lenguatges controlats i la documentació tècnica: millorar la traducibilitat. *Revista Tradumàtica: tecnologies de la traducció, 0*(10), 192-204.

Sánchez, F. (2017). «Diapositives de l'assignatura Traducció Automàtica Estadística». Màster en Tradumàtica: Tecnologies de la Traducció. Universitat Autònoma de Barcelona.

Simon Montserrat, A. i Llisterri, J. (2017). *Problemes lingüístics de la traducció automàtica entre l'anglès i el japonès*. Recuperat de https://ddd.uab.cat/record/180150

SL, U. T. (s.d.). *Ubuntu 17.10 (64-bit) para Ubuntu* - Descargar. Recuperat 29 abril 2018, de https://ubuntu.uptodown.com/ubuntu

TAUS (2016). *Example-based machine translation*. Recuperat de https://www.taus.net/knowledgebase/index.php/Example-based_machine_translation

TAUS (2016). *Statistical machine translation*. Recuperat de https://www.taus.net/knowledgebase/index.php/Statistical_machine_translation

TAUS (2017). *Pre-editing*. Recuperat de https://www.taus.net/knowledgebase/index.php/Pre-editing

Tilde MT. (s.d.). Recuperat de https://www.letsmt.eu

Tiedemann, J. (2012). *Parallel Data, Tools and Interfaces in OPUS*. Recuperat de http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf

The Stanford Natural Language Processing Group. (s.d.). Recuperat de https://nlp.stanford.edu/software/segmenter.shtml

TMX 1.4b Specification. (s.d.). Recuperat de http://www.ttt.org/oscarstandards/tmx/#creationtool

Torres Hostench, O., Cid-Leal, P., Presas, M., Piqué, R., Sánchez-Gijón, P. i Aguilar Amat, A. (2016). *L'ús de traducció automàtica i postedició a les empreses de serveis lingüístics de l'Estat espanyol*. Recuperat de https://ddd.uab.cat/record/166753?ln=ca

Traducción automàtica - EU Law and Publications. (s.d.). Recuperat de https://publications.europa.eu/en/publication-detail/-/publication/8c9d8a8a-a759-44a5-9e22-b6ebbf9705d5/language-es

Tradumatica. (2017). *Contribute to mtradumatica development by creating an account on GitHub. JavaScript*. Recuperat de https://github.com/tradumatica/mtradumatica

Translator Hub - Microsoft Translator. (s.d.). Recuperat de https://www.microsoft.com/en-us/translator/hub.aspx

Ubuntu. (s.d.). Recuperat de https://www.softcatala.org/programes/ubuntu/

Vasiļjevs, A., Skadiņš, R. i Tiedemann, J. (2012). *LetsMT!: A Cloud-Based Platform for Do-It-Yourself Machine Translation*. Recuperat de http://www.aclweb.org/anthology/P12-3008

VOCABULARIO HSK 6 - Fulls de càlcul de Google. (s.d.). Recuperat de https://docs.google.com/spreadsheets/d/1m6_X1v2q2kzEm1ku4Jo6voyt9qVWrb8LdhRvVJ9OBlA/edit#gid=0

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W. i Dean, J. (2016). *Google's Neuronal Machine Translation System: Bridging the Gap between Human and Machine Translation*. Recuperat de http://arxiv.org/abs/1609.08144

www.kantanmt.com : Description of BLEU Score for MT Quality. (s.d.). Recuperat de https://www.kantanmt.com/whatisbleuscore.php

Yuste, E. (2012). La postedició en el flux de producció de contingut multilingüe: tendències, actors i implicacions tecnològiques. *Revista Tradumàtica: Tecnologies de La Traducció*, *0*(10), 157-165.

Annex 1

Script que agafa informació de les cerques de la pàgina

```
# -*- coding: utf-8 -*-
"""
Created on Sat Apr 7 19:25:29 2018

@author: [redacted]
"""

from lxml import html
import requests



import sys
import time


try:
  in_zh = open('chinese-words.txt', 'r')
except Exception as e:
  print e
  sys.exit(1)


try:
  out_zh = open('tfm.zh', 'w')
except Exception as e:
  print e
  sys.exit(1)

try:
  out_ca = open('tfm.ca', 'w')
except Exception as e:
  print e
  sys.exit(1)

num_words = 0

for word in in_zh:
  if num_words % 100 == 0:
    print "Waiting a bit"
    time.sleep(15)

  query = word.rstrip('\n')
  page = requests.get("https://[redacted]/zh/ca/" + str(query))
  tree = html.fromstring(page.content)
  translated =
tree.xpath('/html/body/div[3]/div/div[2]/div/div/article/div/div[1]/div
/ul/li[1]/div[3]/strong/text()')
  print translated
  if len(translated) != 0:
    #Translation found
```

```
    out_zh.write(query + '\n')
    out_ca.write(translated[0].encode('utf-8') +'\n')
    num_words = num_words + 1
    #Get context sentences
    num_tables = len(tree.xpath('//*[@class="tableRow row-fluid"]'))
    if num_tables != 0:
      #There are context sentences in the page
      for count in range(num_tables):
        #parse catalan text
        cat =
tree.xpath('/html/body/div[3]/div/div[2]/div/div/div/article/div[1]/div
/div[' + str(count + 1) + ']/div[2]/span/span')
        cat_text = cat[0].text_content()
        print cat_text
        out_ca.write(cat_text.encode('utf-8') +'\n')
        #parse chinese text
        chinese_text =
tree.xpath('/html/body/div[3]/div/div[2]/div/div/div/article/div[1]/div
/div[' + str(count + 1) + ']/div[1]/span/span')
        out_zh.write(chinese_text[0].text_content().encode('utf-8')
+'\n')


#section =
tree.xpath('/html/body/div[3]/div/div[2]/div/div/div/article/h3')
#all_tables = tree.xpath('//*[@class="tableRow row-fluid"]')
print "Total number of translated words: ", num_words

in_zh.close()
out_ca.close()
out_zh.close()
```

Script que agafa informació de l'API de la pàgina

```
import requests

import sys



try:

 in_zh = open('chinese-words.txt', 'r')

except Exception as e:

 print(e)

 sys.exit(1)



try:
```

```
 out_zh = open('tfm.zh', 'w')

except Exception as e:

 print(e)

 sys.exit(1)



try:

 out_ca = open('tfm.ca', 'w')

except Exception as e:

 print(e)

 sys.exit(1)



for word in in_zh:

 r = re-
quests.get("https://glosbe.com/gapi/tm?from=zho&dest=cat&phrase=" +
str(word.strip("\n")) + "&format=json")

 for example in r.json()["examples"]:

  out_zh.write(example["first"].encode('utf-8') + "\n")

  out_ca.write(example["second"].encode('utf-8') + "\n")



in_zh.close()

out_ca.close()

out_zh.close()

print("End")
```

Annex 2

```
import fs from 'fs'
const createItem = (chinese, catalan) => {
 return `<tu>
 <tuv xml:lang="zh"><seg>${chinese}</seg></tuv>
 <tuv xml:lang="ca"><seg>${catalan}</seg></tuv>
</tu>`
}
const file = 'out.tmx'
const header = `<?xml version="1.0" encoding="UTF-8" ?>
<tmx version="1.4">
<header creationdate="Sun Jan 3 23:24:06 2016"
     srclang="ca"
     adminlang="ca"
     o-tmf="unknown"
     segtype="sentence"
     creationtool="Uplug"
     creationtoolversion="unknown"
     datatype="PlainText" />
 <body>`

const footer = `</body>
</tmx>`
const outCa = fs.readFileSync('tfm.ca', 'utf8').split('\n')
const outZh = fs.readFileSync('tfm.zh', 'utf8').split('\n')
if (outCa.length !== outZh.length) {
 console.log('Both files should contain the same amount of examples')
 process.exit(2)
}
let words = ''
// - 1 because UNIX convention: latest line in the file will be empty
for (let i = 0; i < outZh.length - 1; i++) {
 words = words + createItem(outZh[i], outCa[i])
}
// Write result to file
fs.writeFileSync(file, header + words + footer)
console.log('[✔] Finished')
```

Annex 3

Llengües que suporta MTradumàtica

Afrikaans
Shqip
አማርኛ
العربية
Aragonés
Հայերեն
Avesta
Aymar aru
Azərbaycan dili
Euskara
Беларуская мова
Bislama
Bosanski
Brezhoneg
Български език
Català
Нохчийн мотт
ChiChewa
中文
Чăваш чĕлхи
Corsu
Hrvatski
Čeština
Dansk
Nederlands
English
Esperanto
Eesti
Ɛvegbe
Føroyskt
Vosa Vakaviti
Suomi
Français
Galego
ქართული
Deutsch
Ελληνικά
Avañe'ẽ

ગુજરાતી
עברית
हिन्दी
Magyar
Bahasa Indonesia
Gaeilge
Íslenska
Italiano
日本語
Basa Jawa
Қазақ тілі
Кыргызча
Коми кыв
Kikongo
Kuanyama
Latine
Luganda
Lingála
ພາສາລາວ
Lietuvių kalba
Latviešu valoda
Gaelg
Македонски јазик
Fiteny malagasy
മലയാ ളം
Malti
Te reo Māori
Монгол хэл
Diné bizaad
IsiNdebele
नेपाली
Owambo
Norsk bokmål
Norsk nynorsk
IsiNdebele
Occitan
فارسی
Język polski
پښتو
Português
Runa Simi
Rumantsch grischun
Ikirundi

Română
Р у с с к и й
संस्कृतम्
Sardu
С р п с к и
Gàidhlig
ChiShona
සිංහල
Slovenčina
Slovenski
Soomaaliga
Sesotho
Español
Basa Sunda
Kiswahili
SiSwati
Svenska
தமிழ்
తెలుగు
ไทย
བོད་ཡིག
Wikang Tagalog
Setswana
Faka Tonga
Türkçe
Xitsonga
Twi
Українська мова
اردو
Tshivenḓa
Tiếng Việt
Cymraeg
Wollof
Frysk
IsiXhosa
ייִדיש
Yorùbá
IsiZulu

Annex 4

Llengües que suporta LetsMT

Afar
Afrikaans
Aghem
Akan
Albanès
Alemany
Alsacià
Alt sòrab
Amhàric
Anglès
Àrab
Armeni
Assamese
Asturià
Asu
Atlas central Tamazight
Atlas central Tamazight (Tifinagh)
Atlas central Tamazight (llatí)
Atlas central Tamazight (àrab)
Azerbaidjan (ciríl·lic)
Àzeri (llatí)
Àzeriès
Bafia
Baix sòrab
Bamanankan
Bamanankan (llatí)
Bangla
Basaa
Basc
Bashkir
Bemba
Bena
Bielorús
Birmà
Blin
Bodo
Bosnio (llatí)
Bosnià
Bosnià (ciríl·lic)

Bretó
Búlgar
Caixmir
Caixmir (Devanagari)
Caixmir (Perso-àrab)
Català
Cherokee
Cherokee
Chiga
Coreà
Cornish
Cors
Croat
Danès
Dari
Divehi
Duala
Dzongkha
Edo
Embu
Eslau eclesiàstic
Eslovac
Eslovè
Espanyol
Esperanto
Estonià
Ewondo
Feroès
Filipí
Finlandès
Francès
Frisó occidental
Friülà
Fulah
Fulah
Gallec
Gal·lès
Ganda
Gaèlic escocès
Georgià
Grec
Groenlànic
Guaraní
Gujarati

Gusii
Hausa
Hausa (llatí)
Hawaiana
Hebreu
Hindi
Holandès
Hongarès
Ibibio
Igbo
Indonesi
Interlingua
Inuktitut
Inuktitut (Syllabics)
Inuktitut (llatí)
Ioruba
Irlandès
IsiZulu
Islandès
Italià
Japonès
Javanès
Javanès
Javanès (javanès)
Jola-Fonyi
K'iche '
K'iche '
Kabuverdianu
Kabyle
Kako
Kalenjin
Kamba
Kannada
Kanuri
Kazakh
Khmer
Kikuyu
Kinyarwanda
Kirguizistan
Kiswahili
Konkani
Koyra Chiini
Koyraboro Senni
Kurd central

Kurd central
Kwasio
Lakota
Langi
Lao
Letó
Lingala
Lituà
Llatí
Llegat xinès (simplificat)
Llegat xinès (tradicional)
Luba-Katanga
Luo
Luxemburguès
Luyia
Macedoni
Machame
Makhuwa-Meetto
Makonde
Malai
Malayalam
Malgache
Maltès
Manipuri
Manx
Maori
Mapudungun
Marathi
Masai
Mazanderani
Meru
Meta'
Mohawk
Mongol
Mongol
Mongol (mongol tradicional)
Morisyen
Mundang
N'ko
Nama
Nepalès
Ngiemboon
Ngomba
North Ndebele

Northern Luri
Noruec
Noruec bokmål
Noruec nynorsk
Nuer
Nyankole
Occità
Odia
Oromo
Ossetia
Ovella
Paixto
Papiamento
Persa
Polonès
Portuguès
Prussià
Punjabi
Punjabi
Quechua
Ripuari
Romansh
Romanès
Rombo
Rundi
Rus
Rwa
Saho
Sakha
Samburu
Sami (Inari)
Sami (Lule)
Sami (Skolt)
Sami (sud)
Sami septentrional
Sango
Sangu
Sena
Serbi
Serbi (ciríl·lic)
Serbi (llatí)
Sesotho
Sesotho sa Leboa
Setswana

Shambala
Shona
Shona (llatí)
SiSwati
Sindhi
Sindhi
Sindhi (Devanagari)
Sinhala
Soga
Somali
Sud de Ndebele
Suec
Sànscrit
Síriac
Tachelhit
Tachelhit (Tifinagh)
Tachelhit (llatí)
Tadjik (ciríl·lic)
Tailandès
Taita
Tajik
Tamazight marroquí estàndard
Tamazight marroquí estàndard (Tifinagh)
Tamil
Tasawaq
Telugu
Teso
Tibetà
Tigre
Tigrinya
Tongan
Tsonga
Turc
Turcman
Txec
Txetxènia
Tàtar
Ucraïnès
Uigur
Urdu
Uzbek
Uzbeka (ciríl·lic)
Uzbeka (llatí)
Uzbeka (persa-àrab)

Vai
Vai (Vai)
Vai (llatí)
Venda
Vietnamita
Volapük
Vunjo
Walser
Wolaytta
Wolof
Xinès
Xinès (simplificat)
Xinès (tradicional)
Yangben
Yi
Yiddish
Zarma

Annex 5

Llengües que suporta KantanMT

Abkhaz
Albanès
Alemany
Anglès
Àrab
Armeni
Àzeriès
Bangla
Basc
Bielorús
Birmà
Búlgar
Català
Chuvash
Coreà
Croat
Danès
Danès
Eslovac

Eslovè
Espanyol
Estonià
Filipí
Finlandès
Francès
Gallec
Georgià
Gujarati
Hindi
Holandès
Hongarès
Irlandès
Italià
Japonès
Japonès
Letó
Lituà
Maltès
Marathi
Nepalès
Noruec
Persa
Polonès
Portuguès
Punjabi
Quichua
Romanès
Rus
Sinhala
Suahili
Suec
Tailandès
Tamil
Telugu
Turc
Txec
Ucraïnès
Udmurt
Urdu
Uzbek
Vietnamita
Xinès