Facultat d'Informàtica de Barcelona

Explotación de Wikipedia para el enriquecimiento de un traductor automático

Autor:
Josu Boldoba Trapote

Tutor:
Lluís Màrquez Villodre

FIB - Ingeniería en informática

Proyecto de final de carrera, 37,5 créditos

22 de junio de 2014

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

DATOS DEL PROYECTO

Título del proyecto:	Explotación de Wikipedia para el enriquecimiento de un traductor automático
Nombre del estudiante:	Josu Boldoba Trapote
Titulación:	Ingeniería Informática
Créditos:	37,5
Director:	Lluís Màrquez Villodre
Departamento:	Lenguajes y Sistemas Informáticos
Co-director:	Luis Alberto Barrón Cedeño
Departamento:	Lenguajes y Sistemas Informáticos
Co-directora:	Cristina España-Bonet
Departamento:	Lenguajes y Sistemas Informáticos

MIEMBROS DEL TRIBUNAL *(nombre y firma)*

Presidente:	Lluís Padró Cirera
Vocal:	Joaquim Valls Ribas
Secretario:	Lluís Màrquez Villodre

CALIFICACIÓN

Calificación numérica:

Calificación descriptiva:

Fecha:

Resumen

Esta memoria describe el trabajo realizado para aprovechar la naturaleza multilíngüe de Wikipedia con el fin de construir sistemas de traducción especializados en diferentes dominios de interés que mejoren los resultados obtenidos con un sistema de referencia. Para ello se explican los procedimientos seguidos para extraer un corpus paralelo válido a partir de los artículos de la enciclopedia y el posterior entrenamiento de los traductores. Finalmente se exponen los resultados obtenidos al evaluar los nuevos sistemas y compararlos con el de referencia.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Este proyecto, ubicado en el área del Procesamiento del Lenguaje Natural, se realiza en el marco del proyecto TACARDI, Traducción Automática en Contexto y Aumentada con Recursos Dinámicos de Internet (TIN2012-38523-C02-00)[1], llevado a cabo por los grupos de investigación TALP (Universitat Politècnica de Catalunya) e IXA (Universidad del País Vasco/Euskal Herriko Unibertsitatea). Uno de los objetivos de TACARDI es enriquecer los recursos orientados a los sistemas de traducción automática mediante la generación de corpus paralelos extraídos de Internet.

En este caso se ha fijado Wikipedia como fuente de textos con los que generar dichos corpus. Esta selección se debe a la cantidad de texto que compone Wikipedia y que estos textos se pueden seleccionar atendiendo a su afinidad temática gracias a su naturaleza enciclopédica. De esta forma, se pueden crear traductores "especializados" en un área de conocimiento, haciendo posible que el sistema sepa producir mejor una traducción en el contexto en el que se quiere utilizar. Este proyecto se centra en tres dominios de interés: ciencia, deporte e informática. Respecto al par de idiomas de los traductores, se ha escogido inglés-castellano (en-es), dando prioridad a los traductores que tengan como lengua origen el inglés ya que se ha considerado que son más útiles a la hora de aplicar estos sistemas en la mejora de la Wikipedia.

1.1. Conceptos básicos

En esta memoria se utilizan una serie de términos relacionados con la extracción de texto y los sistemas de traducción. A continuación de definen los conceptos más importantes.

Un *corpus comparable* es un conjunto de textos en dos lenguas que hablan de un mismo tema. Un *corpus paralelo* es una colección de textos en dos lenguas que cumplen la condición de

[1] `http://ixa.si.ehu.es/tacardi`

que cada uno es traducción del otro. En este proyecto se trabaja únicamente a nivel de oración, llamándose *oraciones paralelas* a cada uno de los pares de fragmentos traducidos que compongan los corpus.

Un *modelo de similitud translingüe* permite comparar pares de oraciones en diferentes lenguas con el fin de extraer aquellas que se consideren paralelas. Para aplicar estos modelos es necesario generar una *representación* del texto a comparar, es decir, abstraer su información en las unidades de comparación que utiliza dicho modelo. Finalmente, estas abstracciones se comparan mediante medidas de similitud, que otorgan un resultado numérico a cada par de representaciones comparados.

Un *traductor automático* es un sistema que genera las traducciones de los textos que se le proporciona sin intervención humana. Estos sistemas pueden generar sus resultados de diversas formas. En este proyecto se ha decidido utilizar un sistema de *traducción automática estadística*, que utiliza métodos probabilísticos para decidir cuál es la opción más acertada en cada caso.

Para determinar la calidad de un traductor automático hay que realizar un *proceso de evaluación*, que consiste en comparar el resultado obtenido con los pares de referencia que han sido seleccionados anteriormente. Este proceso asigna una puntuación al traductor determinando cómo de correctas han sido las traducciones con base en las diferentes métricas evaluadas. Entre los diferentes métodos existentes, los traductores entrenados en este proyecto se puntuarán con el modelo BLEU (Bilingual Evaluation Understudy).

1.2. Motivación

Desde que iniciara su recorrido en la web en enero de 2001, Wikipedia ha ido creciendo hasta tener hoy en día, junio de 2014, ediciones en 287 idiomas diferentes con distintos niveles de desarrollo[2]. En ocasiones los contenidos en una lengua son reutilizados para enriquecer otras, característica que hace que Wikipedia sea un recurso a tener en cuenta si se desea obtener de forma automatizada fragmentos de texto traducidos.

Por otra parte, hoy en día es muy común el uso de traductores automáticos, como por ejemplo Google Translate[3] o Bing Translator[4]. Estas herramientas han sido creadas para poder traducir textos de dominio general, es decir, sin ninguna temática específica. Sin embargo, cuando se

[2]Hoy en día hay 276 ediciones activas y 11 de ellas superan el millón de artículos (Wikipedia, 2014b).

[3]`https://translate.google.es` Actualmente soporta traducciones entre 80 lenguas diferentes (Google, 2014).

[4]`http://www.bing.com/translator` Actualmente soporta traducciones entre 45 lenguas diferentes (Bing, 2014).

quiere traducir un texto de un dominio específico, es necesario tener en cuenta su vocabulario y forma de expresarse, siendo más difícil encontrar sistemas que lo hagan. Por ello en este proyecto se quieren usar textos provenientes de Wikipedia que permitan crear sistemas de traducción orientados a enriquecer la enciclopedia utilizando registros propios de cada dominio de la misma. Ello implica tener en cuenta vocabularios y estilos inherentes a los diferentes campos de conocimiento.

1.3. Objetivos

Este apartado describe los objetivos del proyecto. En primer lugar se explican los objetivos generales, que son los que definen a grandes rasgos los hitos planteados. A continuación se exponen los objetivos específicos que se desprenden de los generales. Estos objetivos específicos componen las tareas necesarias para obtener los definidos como principales.

1.3.1. Objetivos generales

1. **Obj_1: Obtener un corpus comparable a partir de Wikipedia.** Partiendo de un par de lenguas y un dominio de interés o área de conocimiento, se trata de identificar los artículos que traten de temas del dominio y que existan en ambas lenguas. Se considerará cubierto en el momento en que se hayan creado las relaciones de qué artículos de la Wikipedia en inglés existen en castellano y son del dominio indicado.

2. **Obj_2: Obtener un corpus paralelo a partir del corpus comparable.** Este objetivo busca la generación de una colección de pares de oraciones paralelas extraídas directamente de los artículos que conforman el corpus comparable de un dominio en Wikipedia. El resultado obtenido al cumplir este objetivo será determinante a la hora de alcanzar el tercer y último objetivo general.

3. **Obj_3: Enriquecer un sistema de traducción automática estadística.** Se trata de obtener traductores especializados en un área de interés a partir de un traductor base entrenado con un corpus sin ningún dominio específico y conseguir que las traducciones de los sistemas entrenados en este proyecto mejoren las realizadas por el sistema seleccionado como base.

1.3.2. Objetivos específicos

1. **Obj_1.1: Analizar la taxonomía de Wikipedia.** Este paso es necesario para poder crear las estrategias de selección de categorías y, por consiguiente, artículos de Wikipedia relacionados con un dominio de interés. En él se obtendrá una descripción de la organización y clasificación de los artículos en la enciclopedia y la forma de ralcionarse con los artículos homólogos en otras lenguas.

2. **Obj_1.2: Extraer los artículos relativos a los dominios de interés.** Una vez conocida la estructura de Wikipedia, este objetivo es necesario para cubrir el primer objetivo general, ya que el resultado de esta extracción será una relación de pares de artículos que pertenecen a un dominio de interés y están relacionados en las Wikipedias en inglés y en castellano.

3. **Obj_2.1: Procesar los artículos.** Antes de poder abstraer y comparar fragmentos de texto, con el fin de crear un corpus paralelo, es necesario obtener los fragmentos de texto que nos interesen de cada artículo. En este paso se espera obtener el contenido de los artículos en texto plano, de forma que se pueda trabajar con él en pasos posteriores.

4. **Obj_2.2: Estimar las similitudes entre oraciones.** Este es el objetivo más crítico de los necesarios para alcanzar el segundo objetivo general ya que los resultados obtenidos aquí determinarán la calidad del corpus creado y, por consiguiente, la de los traductores a entrenar. Una vez alcanzado este hito se habrán calculado las similitudes entre todos los pares de oraciones posibles por cada artículo procesado.

5. **Obj_2.3: Obtener Extraer los pares de oraciones paralelas.** Antes de comenzar el entrenamiento de los traductores, se seleccionarán aquellos pares de oraciones que se consideren paralelos con base en las similitudes otorgadas a cada uno de ellos por los diferentes modelos de similitud translingüe utilizados.

6. **Obj_3.1: Entrenar los traductores.** Se trata de entrenar diversos sistemas de traducción utilizando los corpus paralelos obtenidos de los artículos de Wikipedia. Para ello será necesario seleccionar qué pares de oraciones formarán parte de cada uno de los conjuntos necesarios: el de entrenamiento, el de desarrollo y el de test.

7. **Obj_3.2: Evaluar los traductores.** Último objetivo, se considerará cubierto el mismo una vez se hayan calculado las diferentes métricas de calidad de la traducción de cada uno de los traductores entrenados y del traductor base y se hayan analizado con el fin de poder confirmar que los construidos en este proyecto mejoran al que se use como referencia.

1.4. Organización de la memoria

Esta memoria describe el trabajo llevado a cabo para enriquecer un sistema de traducción automática estadística a partir de pares de oraciones paralelas extraídas de Wikipedia. El resto de la misma se estructura de la siguiente manera:

- El capítulo 2 describe los antecedentes en los campos de la recuperación de información y la traducción automática. Este capítulo permite poner en contexto al lector sobre los métodos aplicados a lo largo del trabajo realizado.
- El capítulo 3 expone los motivos por los que Wikipedia es un recurso adecuado para la obtención de corpus. En él también se explican los pasos llevados a cabo en este proyecto para formar un corpus comparable de la enciclopedia.
- El capítulo 4 explica los modelos de similitud translingüe aplicados en el proyecto y los pasos seguidos desde dicha aplicación hasta la generación de un corpus paralelo. Estos pasos son, básicamente, el análisis de los valores de similitud estimados y la definición de una estrategia para seleccionar aquellos pares considerados paralelos.
- El capítulo 5 indica cómo se han construido los conjuntos de entrenamiento, desarrollo y test para entrenar los sistemas de traducción. Además, muestra la evaluación realizada a dichos sistemas así como un análisis de estos resultados.
- El capítulo 6 contiene la planifiación del proyecto junto con la descripción de los motivos que han retrasado la misma. Por otra parte, tambien recoge un estudio económico que muestra el coste real del trabajo realizado.
- El capítulo 7 recopila las conclusiones alcanzadas con el trabajo realizado e indica los posibles pasos a seguir en el futuro a partir de los resultados aquí alcanzados.

Capítulo 2

Antecedentes

Antes de comenzar a exponer el trabajo desarrollado durante este proyecto es conveniente poner en contexto las tecnologías y metodologías usadas para llevarlo a cabo. Anteriormente se ha dicho que este proyecto está enmarcado en el área del Procesamiento del Lenguaje Natural ya que en él se trata texto escrito por los usuarios de Wikipedia con el fin de que sirva de entrada al entrenamiento de traductores. Sin embargo, atendiendo a los procesos realizados para llevar a buen término el proyecto, los campos más relacionados son la Recuperación de Información (RI) durante la primera fase del proyecto, y la Traducción automática (TA) durante la segunda etapa. En este capítulo se explica el trabajo realizado en estas áreas y qué tecnologías y métodos relacionados con ellas se han utilizado, sirviendo de introducción a las tareas descritas más adelante.

2.1. Recuperación de información

Según Manning *et al.* (2008, pág. 1), "la recuperación de información, como campo académico, es la búsqueda de material (normalmente documentos) de naturaleza no estructurada (normalmente texto) que satisface una necesidad de información desde grandes colecciones (usualmente almacenadas en ordenadores)". En mi caso el objetivo es recopilar las oraciones paralelas contenidas en una serie de artículos de Wikipedia seleccionados previamente, y para ello se ha utilizado un modelo de recuperación, el cuál permite obtener una valoración decimal comprendida entre 0 y 1 que mida cómo de similar es el documento evaluado.

Una vez puestos en el contexto de qué tipo de recuperación se va a realizar, conviene aclarar los métodos con los que se lleva a cabo. Lo primero de todo es incidir en el uso del *índice invertido* para estructurar la información y poder así trabajar con ella. El uso del *índice invertido* consiste en usar un diccionario de términos o *lexicón* que por cada una de sus entradas almacene la lista

de documentos en el que ha aparecido (Manning *et al.*, 2008, pág. 6), en contraposición de lo que haría una indexación normal, la cual almacenaría los documentos como índice y cada una de ellos apuntaría al término que se encuentra en él. Dado que mi objetivo es recuperar oraciones se considera que mis documentos son precisamente eso, oraciones dentro de los pares de artículos de Wikipedia.

Por otra parte hay que destacar el uso del modelo de espacio vectorial para obtener los resultados de las comparaciones de los fragmentos de texto y poder así tomar decisiones sobre ellos. Para aplicar este modelo hay que representar los textos como vectores, pudiendo además ponderar los términos que los conforman. Una vez obtenidos dichos vectores, mediante el cálculo del coseno del ángulo que forman se obtiene la similitud entre ambos (Manning *et al.*, 2008, pág. 109) u otras medidas como son los factores de longitud, ambas explicadas a continuación.

Dadas dos oraciones s y t, el cálculo del coseno se realiza a partir de una representación vectorial ($\vec{s}$ y $\vec{t}$ respectivamente). La medida de similitud de coseno se define como:

$$simiCos(s,t) = \cos(\theta) = \frac{\vec{s} \cdot \vec{t}}{\|\vec{s}\|\|\vec{t}\|} = \frac{\sum_{i=1}^{n} \vec{s}_i \times \vec{t}_i}{\sqrt{\sum_{i=1}^{n} (\vec{s}_i)^2} \times \sqrt{\sum_{i=1}^{n} (\vec{t}_i)^2}} \tag{2.1}$$

es decir, el producto escalar entre los vectores dividido por el producto de sus magnitudes. Por su parte, el modelo de longitud calcula la probabilidad de que t sea traducción de s según sus longitudes ($len(s)$ y $len(t)$ respectivamente) utilizando para ello una distribución normal. Para definir la distribución es necesario calcular previamente qué diferencia de longitud se genera al traducir un conjunto de oraciones desde el idioma origen al idioma destino y obtener el promedio de estas diferencias (μ) y su desviación estándar (σ):

$$simFL(s,t) = \mathrm{e}^{-0{,}5\left(\frac{\frac{len(t)}{len(s)} - \mu}{\sigma}\right)^2} \tag{2.2}$$

2.1.1. Preprocesamiento

Para poder representar el texto de los artículos con los que se va a trabajar es necesario obtener el texto de los mismos y procesarlo de manera que los sistemas sean capaces de tratarlo correctamente, es decir, hay que crear un vocabulario de términos que sirva a los métodos de RI como unidad de comparación. En los modelos de similitud translingüe utilizados para extraer las oraciones paralelas se utilizan diferentes unidades de comparación, como pueden ser los *stems* o los n-gramas, por lo que a continuación se describen las diferentes técnicas utilizadas para obtenerlas a partir del fragmento de texto completo. Cabe destacar que para poder utilizar estos procedimientos es necesario que los textos hayan sido formateados a texto plano.

El primer método es el *case-folding* que, al contrario que la capitalización, supone convertir todos los caracteres del texto a minúscula (Manning *et al.*, 2008, pág. 30). De esta manera se consigue que no se considere diferente un término por el tipo de caracterización utilizado, hecho que podría ocurrir con vocablos que aparezcan al inicio de oraciones, por ejemplo en las oraciones:

"Pienso, luego existo." (René Descartes)

"Verdaderamente tiemblo por mi patria cuando pienso que Dios existe." (Thomas Jefferson)

El término "pienso" aparece en ambas, pero en una su primer carácter se encuentra en mayúscula, hecho por el que si se comparan tal cual, "Pienso"≠"pienso". Sin embargo al aplicar *case-folding*, tenemos que "pienso"="pienso".

El siguiente mecanismo utilizado es la tokenización, es decir, dividir el texto en las unidades significativas mínimas, llamadas *tokens* (Manning *et al.*, 2008, pág. 22). Esto, llevado al caso que nos atañe significa "trocear" el texto separando los signos de puntuación y utilizando los caracteres blancos (espacios y tabuladores) como delimitador del resto del texto. Por ejemplo, la oración "Science is a differential equation. Religion is a boundary condition."[1] (Alan M. Turing) se convertiría en el conjunto {science, is, a, differential, equation, ., religion, is, a, boundary, condition, .}, habiendo aplicado también un plegado de letra.

Por último, es destacable la necesidad de aplicar procesos de *stemming* u "obtención heurística del fragmento de un término resultante de eliminar los prefijos y sufijos que lo componen" (Porter, 1980). El algoritmo, basado en una serie de reglas, permite eliminar las variaciones entre las palabras de forma que converjan en una misma. Esto resulta muy útil para considerar dos palabras cuya única diferencia sea el género o número y tomar ambas como un mismo término. Este proceso se puede aplicar únicamente a textos ya tokenizados, por lo que sirve el conjunto de *tokens* obtenido en el ejemplo anterior para mostrar cómo se realiza. Así, el conjunto {science, is, a, differential, equation, ., religion, is, a, boundary, condition, .} se convierte en {scienc, is, a, differenti, equat, ., religion, is, a, boundari, condit, .}[2]

2.1.2. Recuperación de información translingüe

Ahora que ya está descrita la RI a nivel general, en esta sección se explican los principios básicos de la aplicación de estos métodos a la obtención de información translingüe, es decir, entre diferentes lenguas. Estos trabajos son de suma importancia para la tarea que nos atañe

[1] "La ciencia es una ecuación diferencial. La religión es una condición de frontera."

[2] Resultado obtenido con el algoritmo de *stemming* de Porter.

ya que en este proyecto los textos que se quieren recuperar son pares de oraciones en diferentes idiomas, inglés y castellano, que sean tan similares entre sí que en realidad se trate de una traducción de una a la otra.

Al respecto hay que nombrar los trabajos de Oard y Hackett (1997) respecto al uso de la traducción de documentos como aproximación a la recuperación de información translingüe. En estos primeros trabajos se experimentó con dos tipos de técnicas: la RI con textos en diferentes idiomas y la RI de fragmentos en documentos que habían sido traducidos previamente. Del primer tipo son los modelos aplicados en este trabajo que se basan en la caracterización del texto mediante cognados o n-gramas de caracteres, los cuales se explican en la sección 4.2 detallando cómo se han aplicado en este problema. Respecto al segundo tipo, concluyen que la recuperación de información en documentos previamente traducidos es mejor, por lo que se ha decidido utilizar también métodos de este tipo basados en trabajos posteriores al aqui citado.

2.1.3. Evaluación en RI

Los resultados de las técnicas expuestas en esta sección deben ser analizados y evaluados para determinar si son correctos o no, y, en caso de que más de uno lo sea poder indicar cuál es mejor. Para ello los métodos más utilizados son el cálculo de la precisión y la cobertura del conjunto recuperado (Manning *et al.*, 2008, pág. 154). Para medir estas dos variables es necesario tener dos conjuntos: uno de referencia y otro a evaluar. El primero clasifica los elementos como relevantes o no relevantes, siendo los primeros los casos que interesan. El otro conjunto, por su parte, define cada elemento como recuperado o no recuperado. De esta forma es posible clasificar los elementos obtenidos en 4 categorías:

- Verdaderos positivos (P_v). Son los elementos recuperados que se consideran relevantes.
- Falsos positivos (P_f). Son los elementos recuperados que no son relevantes.
- Verdaderos negativos (N_v). Son los elementos no recuperados que el conjunto de referencia también considera no relevantes.
- Falsos negativos (N_f). Son los elementos no recuperados y que, sin embargo, están considerados relevantes.

A partir de esta clasificación es posible definir las dos medidas de evaluación de la siguiente forma: la *precisión* es la proporción de elementos recuperados que son relevantes; mientras que la *cobertura* mide el porcentaje de elementos relevantes que han sido seleccionados mediante RI.

Además, existe una medida que combina estas dos puntuaciones mediante su media harmónica, de forma que sea más simple la comparación de los diferentes resultados obtenidos (Manning *et al.*, 2008, pág. 156). Esta medida será la que se usará en todo el proyecto para tomar decisiones respecto a los pasos a seguir en el proceso de creación de un corpus paralelo a partir de Wikipedia. Todas estas medidas se definen a continuación partiendo de un conjunto de oraciones O y el listado de referencia R. Primero se escoge de manera un subconjunto de oraciones O_A que consideramos el conjunto de oraciones recuperadas en un artículo. La **precisión** de O_A mide el porcentaje de frases escogidas que son realmente paralelas, es decir:

$$precision(O_A) = \frac{|O_A \cap P|}{|O_A|} \tag{2.3}$$

La **cobertura**, por el contrario, mide el número de oraciones paralelas que han sido recuperadas del total de oraciones que contiene P, es decir:

$$cobertura(O_A) = \frac{|O_A \cap P|}{|P|} \tag{2.4}$$

Estos dos valores se combinan en la **medida** F_1, que se calcula haciendo la media harmónica de los valores de precisión y cobertura del conjunto O_A:

$$F_1 = 2 \cdot \frac{p \cdot c}{p + c} \tag{2.5}$$

2.2. Traducción automática

Desde que a mediados del siglo pasado Warren Weaver propusiera el uso de computadoras para crear sistemas de traducción automática y que la Universidad de Georgetown fuese capaz de traducir 60 oraciones del ruso al inglés de forma automática (Hutchins, 1997), la construcción de los sistemas de traducción ha avanzado mucho.

Al principio los sistemas de traducción trataban de hacer frente al problema de una forma directa traduciendo las palabras una por una y sin tener en cuenta otros parámetros más lingüísticos. Más adelante, hacia finales de los años 70 del siglo pasado, comenzó a tomar fuerza la idea de que estos sistemas fueran capaces de entender cómo funciona el idioma al que se quiere traducir. Con esta idea surgieron los traductores basados en reglas (RBMT), los cuales necesitan de conjunto de normas que indiquen cómo tratar el texto para "transformarlo" en la traducción correcta. Más adelante estos traductores evolucionaron hacia sistemas empíricos, los cuales se basan en una serie de datos y no en las directrices que hayan definido otras personas para traducir de un idioma a otro y pueden ser clasificados, a su vez, en sistemas basados en ejemplos

(EBMT) o sistemas estadísticos (SMT) (España-Bonet, 2008). En esta sección se describen y comparan los sistemas RBMT y SMT ya que son los más representativos de la evolución de los sistemas de traducción.

Un RMBT es un sistema de traducción automática que se vale de las reglas de definidas por un grupo de expertos para tomar las decisiones oportunas a la hora de generar el texto traducido. Por otra parte, un SMT calcula las probabilidades con las que se usa cada posible traducción a una palabra y cómo se ordenan las mismas en una oración a partir de un conjunto de fragmentos paralelos. En este proyecto se ha decidido entrenar un SMT en lugar de programar un RBMT debido a que los primeros tan sólo necesitan un corpus paralelo para poder aprender cómo realizar las traducciones. En la sección siguiente se describe el funcionamiento de un SMT y qué recursos es necesario proveerle para que funcione correctamente.

2.2.1. Sistema básico de traducción

Un sistema de traducción automática estadística implementa la ecuación básica

$$T(f) = \hat{e} = \text{argmax}_{\text{e}}\ P(e|f) = \text{argmax}_{\text{e}}\ P(e)\,P(f|e)\,. \tag{2.6}$$

en las que las probabilidades $P(e)$ and $P(f|e)$ representan le modelo de lenguaje y el modelo de traducción respectivamente. Un *modelo de lenguaje* indica la probabilidad de que una frase cualquiera pertenezca a un determinado idioma, lo que permite evaluar cómo de bien se ha formado una oración en el idioma destino. Por otra parte, el *modelo de traducción* indica cómo de probable es la transformación aplicada a f para convertirse en la traducción e. Además de la consulta de estos dos modelos, en estos sistemas hay un tercer paso encargado de buscar la maximización (*argmax*). Esta tarea la realiza el decodificador (*decoder*) y requiere de una búsqueda en un espacio exponencial en función del tamaño de la entrada del traductor.

Actualmente se usa una generalización, el modelo log-linear (Koehn, 2004), que permite añadir más probabilidades al sistema y no usar únicamente los modelos de lenguaje y de traducción. Partiendo de la aproximación original de los traductores a nivel de oración, este modelo es una extensión que aprovecha el hecho de que la estimación de máxima verosimilitud generaliza a la de máxima entropia, reescribiendose la ecuación (2.6) como un sumatorio pesado de sus logaritmos:

$$\text{argmax}\ P(e|f) = \text{argmax}\ \sum_{m} \lambda_m h_m(f|e)\,. \tag{2.7}$$

Construcción de un SMT

En la práctica para construir un SMT es necesario seguir una serie de pasos que permitan generar los modelos necesarios para el cálculo de las probabilidades que se utilizan durante la tarea de traducción. Las tareas a realizar para ello son:

1. Estimar el modelo de lenguaje del idioma al que va a traducir el sistema.
2. Depurar el corpus de entrenamiento.
3. Alinear el corpus paralelo a nivel de palabra.
4. Construir del modelo de traducción a partir de dicho alineamiento.
5. Ajustar los pesos obtenidos para el modelo de traducción.

En este proyecto se han utilizado los sistemas SRILM[3], GIZA++[4] y MOSES[5]. El primero de ellos es utilizado para la creación de los modelos de lenguaje; el segundo para realizar el alineamiento a nivel de palabra; y el último para calcular el modelo de traducción, construir el sistema y ajustar los parámetros del mismo (Koehn *et al.*, 2007).

Como se ha dicho, la primera etapa a llevar a cabo es construir el modelo de lenguaje del idioma destino, castellano en este caso. Este proceso lo realiza la herramienta `ngram-count` de SRILM que es capaz de contar n-gramas y estimar modelos de lenguaje del texto dado. Con ella se ha construido un modelo de lenguaje de orden 5, es decir, que estima probabilidades para n-gramas desde 1 elemento hasta 5. El modelo de lenguaje del idioma origen no es necesario construirlo ya que un traductor sólo hace uso de este recurso para ver qué traducción es más probable, es decir, para decidir cuales son los n-gramas más probables en el idioma destino y cuál es su orden dentro de la oración.

Para depurar el corpus que se va a emplear en el entrenamiento se ha usado el script `clean-corpus-n.perl` de MOSES, al cual se le indica el nombre del corpus y dos números, el mínimo y el máximo número de tokens que debe contener una frase. En este caso se usarán los valores 1 y 100, de forma que se eliminen las líneas que no contengan ninguna oración y aquellas que sean excesivamente largas. Además, este script normaliza los espacios en blanco, es decir, si encuentra una secuencia de más de un espacio o tabulador, lo convierte en un único espacio en blanco.

[3] `www.speech.sri.com/projects/srilm/`
[4] `www.statmt.org/moses/giza/GIZA++.html`
[5] `http://www.statmt.org/moses/`

Una vez está el modelo de lenguaje construido y el corpus de entrada limpio es el momento de comenzar a entrenar. Inicialmente, con GIZA++, se alinea el corpus a nivel de palabra (Och y Ney, 2003), es decir, para cada palabra en el idioma origen asignan aquellas que son su traducción de forma que cada palabra en el idioma destino esté alineada únicamente con una palabra del origen. Una vez realizado este paso MOSES calcula las tablas de traducción léxica, es decir, las probabilidades de que una palabra sea utilizada como traducción de otra.

Por último, con el *decoder* de MERT[6] se ajustan los pesos del modelo de traducción (Och, 2003) usando como entrada un conjunto de desarrollo que sea disjunto al utilizado para el entrenamiento realizado en los pasos precedentes. Al terminar este proceso se puede considerar que el traductor está listo para ser usado.

Evaluación en traducción automática

Al igual que con los procedimientos de RI, es necesario poder evaluar los traductores entrenados y así tener referencias objetivas para decidir cuál es mejor. Esta evaluación se puede hacer de forma manual, pero eso supone una tarea colosal ya que habría que dedicar muchas horas a comprobar diferentes traducciones y casos intrínsecos al idioma destino. Por ello se han diseñado técnicas que permiten dar una puntuación a los sistemas de traducción de forma totalmente automática.

En este proyecto se ha usado la medida BLEU[7], calculada mediante el script `multi-bleu` proporcionado por MOSES. Esta medida es una de las más usadas hoy en día y se basa en la coincidencia de 4-gramas comparando la traducción realizada de forma automática con una traducción de referencia proporcionada.

[6]Minimum Error Rate Training.

[7]Bilingual Evaluation Understudy

Capítulo 3

Organización de Wikipedia como corpus comparable

En esta sección se definen las estrategias llevadas a cabo para diferenciar los artículos relativos a cada dominio de interés con el fin de poder generar traductores especializados en ellos. Además se explica cómo se han seleccionado los artículos sobre los que finalmente se ha trabajado.

3.1. Creación del entorno de trabajo

Debido a que el trabajo que se va a realizar sobre Wikipedia es de asignación de pares de artículos comunes y comparación de texto, es recomendable utilizar una versión estática de la misma, ya que si se trabaja con la publicación en línea es posible que alguno de los artículos sea modificado e incluso borrado, echando a perder el avance que se haya conseguido hasta ese momento. Para evitar este riesgo se ha decidido aprovechar la publicación periódica que realiza la fundación Wikimedia de Wikipedia (y otros recursos) en diferentes lenguas[1]. En particular, se han tomado los registros disponibles en julio de 2013.

Estos volcados se conforman de ficheros[2] que permiten, tras un procesamiento previo, la creación de las tablas necesarias y la posterior importación de los datos de manera automática. Debido a la naturaleza relacional de estos datos hubo que elegir un sistema de gestión de bases de datos acorde a ello, pero dado el amplio número de posibilidades se decidió seleccionar previamente la herramienta que se iba a utilizar para acceder y explotar los datos almacenados. En

[1] `http://dumps.wikimedia.org/`

[2] Las tablas de páginas se encuentran en formato XML mientras que las que relacionan páginas, es decir, los enlaces, se distribuyen en formato SQL

este caso se encontraron dos buenas opciones: JWPL (Zesch *et al.*, 2008)[3] y WikiXRay (Ortega Soto, 2009)[4]. Ambas bibliotecas proporcionan herramientas para explotar datos de Wikipedia almacenados en bases de datos relacionales. La mayor diferencia se encuentra en el lenguaje para el que han sido creadas: Java en el caso de JWPL y Python en el de WikiXRay. Dado que no hay ninguna desventaja que haga posible una decisión clara hacia alguno de los sistemas, se ha trabajado con JWPL por ser Java el lenguaje utilizado en el proyecto en el que se enmarca este trabajo. Esta biblioteca se distribuye en formato JAR, lo que permite utilizarla configurando únicamente el *classpath* de la máquina virtual de Java en la que se va a desarrollar. Esta selección nos limita la selección del sistema de gestión de bases de datos, por lo que finalmente se decidió utilizar MySQL[5] ya que es el que requiere JWPL.

El último punto importante a tener en cuenta en la creación del entorno de trabajo es la máquina que va a procesar la información. Debido al volúmen de datos que hay que procesar se ha decidido utilizar el *cluster* que gestiona el Laboratorio de Investigación y Desarrollo (RDLab) para el departamento de Lenguajes y Sistemas Informáticos (LSI). En términos generales este conjunto de máquinas se compone de 90 nodos de cómputo con procesadores Intel Xeon[6] a los que se accede mediante 3 servidores de *clustering*, 2 servidores de disco[7] y un *switch* Ethernet de 48 puertos. Respecto al software utilizado, los nodos y servidores utilizan el sistema operativo Ubuntu 12.04 LTS; Open Grid Engine es utilizado para realizar las tareas de *clustering* y los discos están formateados con el sistema de ficheros distribuido Lustre.

3.2. Definición de los dominios de interés

Los dominios de interés se han definido teniendo en cuenta principalmente dos variables: el tamaño y la temática. En el primer punto se ha decidido utilizar áreas que no fuesen muy especializadas, ya que esto provocaría que el subconjunto de artículos que los conformasen fuese muy reducido. En cuanto a la temática, se ha optado por escoger los dominios ya definidos por TACARDI como dominios de interés: deporte (*sp*), ciencia (*sc*) e informática (*cs*).

La primera consecuencia de esta decisión es la necesidad de analizar la categorización de

[3] `https://code.google.com/p/jwpl/`

[4] `http://meta.wikimedia.org/wiki/WikiXRay`

[5] `http://www.mysql.com`

[6] Aunque todos los procesadores son Intel Xeon, la composición de los nodos es bastante heterogénea, usándose desde procesadores QuadCore a HexaCore. Además, existen nodos que usan un procesador y otros dos. En cuanto a la memoria RAM, todos los nodos tienen entre 32 y 128GB.

[7] Los servidores de disco se componen de varios discos SAS de 600Gb a 15.000 rpm montados sobre un array de 4 cabinas. La capacidad de almacenamiento una vez formateado es de 20TB.

artículos de Wikipedia con el fin de obtener los artículos que están relacionados con cada uno de estos temas. La solución más trivial sería acceder a la categoría correspondiente a cada dominio y seleccionar los artículos que contiene, pero este método no tiene sentido ya que no se tienen en cuenta los artículos de categorías más específicas a las que se accede a partir de ésta, omitiendo un gran volumen de datos. Por ello ha sido necesario llevar a cabo diferentes pasos que nos acercaran al conjunto de páginas que nos interesaban.

3.2.1. Exploración del árbol de categorías

Normalmente un sistema de clasificación mediante categorías es jerárquico. En el caso de Wikipedia se supone que esta característica se cumple, ya que provee una página especial llamada "Árbol de categorías"[8] que despliega las páginas y subcategorías de la categoría que se busque. Sin embargo, tras un análisis a esta clasificación, se ve que a menudo una categoría subcategoriza a otra más genérica, provocando ciclos y convirtiendo la estructura en un grafo dirigido con ciclos[9]. Aun con esta peculiaridad, es posible utilizar esta información con el fin de extraer los grupos de categorías mediante un recorrido del grafo a partir de la categoría que define el dominio a definir, tal y como muestra la figura 3.1. Inicialmente consideramos que no se ha visitado ningún nodo excepto la categoría raíz, que además es la única seleccionada como parte del dominio en este punto. A partir de este nodo se realiza una búsqueda en anchura sobre el grafo seleccionando todos los nodos a los que se pueda acceder por este camino.

Una vez realizada la extracción de categorías mediante este método se observó que no era el más adecuado, ya que retornaba cerca del 90 % del total de las categorías. Este resultado se debe a que en ocasiones se relacionan subcategorías de otras temáticas y no sólo las especializaciones de la categoría actual, es decir, los diferentes dominios de interés están fuertemente intersecados entre sí. Un ejemplo de ello se observa en la figura 3.2 que contiene, de forma parcial, los árboles obtenidos para los dominios Deporte y Ciencia. En ella se ve que la categoría "Pirineos" es accesible desde en ambos dominios en menos de 10 pasos[10]:

- Deporte → Deportes → Deportes de montaña → Montañismo → Montañas → Montañas por país → Montañas de Andorra → Pirineos.

- Ciencia → Ramas de la ciencia → Ciencias naturales → Ciencias de la Tierra → Geología → Geología por país → Geología de España → Cordilleras de España → Pirineos.

[8] `http://en.wikipedia.org/w/index.php?title=Special:CategoryTree`

[9] Los propios editores de la Wikipedia en castellano los consideran una consecuencia de que, a menudo, un editor confunde los conceptos de categoría y etiqueta e incluyen categorías generales como subcategorías de otras más específicas. Llaman a este fenómeno "categorización circular" (Wikipedia, 2014a).

[10] Para esta demostración se ha consultado el árbol de categorías de la Wikipedia en castellano el día 26/05/2014.

Extracción de categorías: Dado un grafo de categorías C y una categoría raíz c_r:

// Marcar nodos visitados

Para cada nodo c_i en C:

visitado(c_i) = falso

visitado(c_r) = cierto

// Inicialización del conjunto resultante D y de la cola de nodos pendientes Q

$D = \{c_r\}$

Insertar subcategorías de c_r en la cola Q

// Recorrido en anchura

Mientras Q no está vacía:

c_i = siguiente elemento en Q

Si visitado(c_i) == falso:

visitado(c_i) = cierto

$D = D \cup \{c_i\}$

insertar subcategorías de c_i en la cola Q

Figura 3.1: Algoritmo de selección de categorías mediante recorrido del árbol de un dominio.

Por esta causa el recorrido en anchura del grafo de categorías no es suficiente para determinar qué nodos pertenecen al dominio y cuáles no, motivo por el cuál ha sido necesario experimentar con otras alternativas.

3.2.2. Extracción de las categorías de un dominio

Para resolver el problema expuesto anteriormente se ha mejorado el algoritmo de exploración inpirándose en el propuesto por Cui *et al.* (2008), Classification Tree-Breadth First Search (CT-BFS). Este método, que también recorre el grafo mediante una búsqueda en anchura, propone puntuar las categorías de forma que se pondere su posible pertenencia al dominio.

El sistema de puntuación propuesto ha sido binario, una categoría pertenece al área de interés o no en función de si su nombre contiene alguna palabra clave del dominio en cuestión.

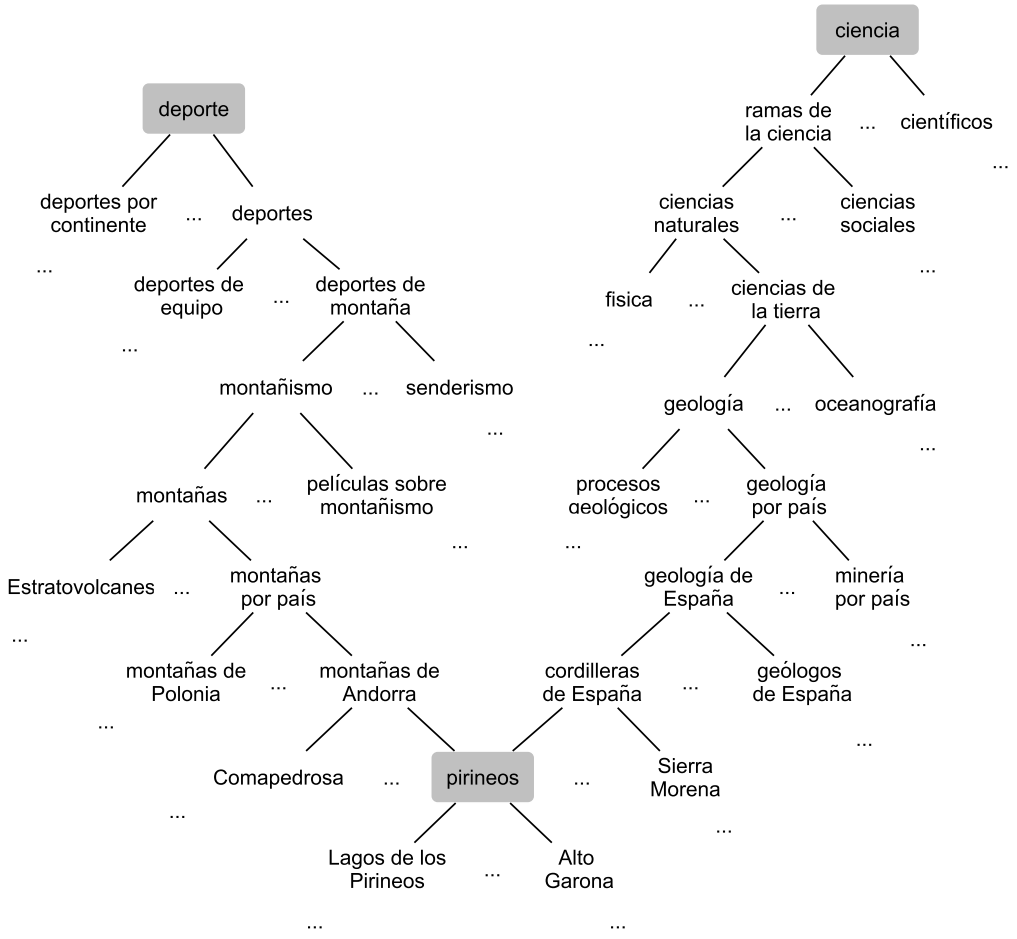


Figura 3.2: Árbol de categorías de los dominios Deporte y Ciencia que muestra la intersección de ambos en la categoría “Pirineos”.

Sin embargo, como este tipo de clasificación puede descartar categorías que sí interesan pero su nombre no es tan determinante, se ha decidido puntuar grupos de categorías de forma que se asigna el porcentaje de categorías positivas que contiene. Los grupos se han formado según la distancia a la que se encuentra la subcategoría de la categoría inical, siendo ésta la única componente del primer grupo. Es decir, dada la categoría raíz c_r y otra categoría c, $c \in C_i \iff d(c_r, c) = i$.

Para aplicar el sistema de puntuación establecido se ha generado un vocabulario por cada dominio e idioma. Tal y como se observa en la tabla 3.1, las categorías seleccionadas como raíz de las áreas de interés contienen varios artículos. Estos artículos se considera que están muy relacionados con el área a explorar, por lo que son una buena opción para generar un vocabulario. Para ello se han realizado las siguientes acciones para cada uno de ellos:

Tabla 3.1: Artículos pertenecientes a la categoría raíz para cada dominio en inglés y castellano en julio de 2013. El texto de estos artículos ha sido utilizado para crear el vocabulario de cada dominio de interés.

	sc	**cs**	**sp**
en	29	4	3
es	3	130	10

1. Se ha extraído el texto de todos los artículos de la categoría raíz.
2. Se ha tokenizado el texto y se han descartado las palabras de paro[11], dígitos y signos de puntuación.
3. Se ha aplicado un proceso de *stemming* (Porter, 1980)[12] a los tokens restantes.
4. Se han considerado sólo los tokens de al menos 4 caracteres y se han ordenado con base en tf [13].

La figura 3.3 muestra cómo puntuar los grupos de categorías según el método descrito anteriormente. El algoritmo procesa cada grupo de manera independiente. En cada grupo se contabilizan el número de elementos que tiene y cuántos de ellos son considerados positivos con el fin de obtener su puntuación ($p = positivos/totales$). Para decidir si un elemento es positivo se le aplica el mismo procesado de texto que se ha usado en la generación del vocabulario y se comprueba si alguno de los *stems* resultantes se encuentra en el listado del vocabulario. Este método retorna la relación de puntuaciones para cada grupo, de forma que se puede graficar la precisión de los mismos y tomar decisiones con base en estos resultados. En la figura 3.4 se pueden ver los resultados obtenidos con este análisis. La característica más evidente es la tendencia descendente de las puntuaciones a medida que se distancia de la categoría raíz. El otro hecho remarcable es que este descenso es brusco desde una distancia relativamente pequeña (en torno al 5).

Con base en estos datos se han definido 3 estrategias de extracción: (a) seleccionar las categorías de todos los grupos hasta que la puntuación sea inferior a 0,6, (b) seleccionar las categorías de todos los grupos hasta que la puntuación sea inferior a 0,5, y (c) seleccionar únicamente las

[11]Nombre que reciben las palabras sin significado como artículos, pronombres o preposiciones. En inglés reciben el nombre de *stopwords*.

[12]Proceso por el que se reduce una palabra a su raíz. De esta manera no se tienen en cuenta rasgos como el género o número a la hora de analizar los tokens. Por ejemplo, "actriz" y "actores" comparten el *stem* "act".

[13]La medida tf (*term frequency*) se obtiene contabilizando el número de apariciones de los términos en un conjunto de documentos.

Puntuación de grupos de categorías: Dados los grupos de categorías C_i y un vocabulario V:

// Inicialización del conjunto de puntuaciones P

$P = \emptyset$;

Para cada C_i:

total = 0

positivos = 0

// Cálculo de puntuación

Para cada categoría c en C_i:

total = total + 1;

Tokenizar y procesar nombre de c para generar la lista de tokens T

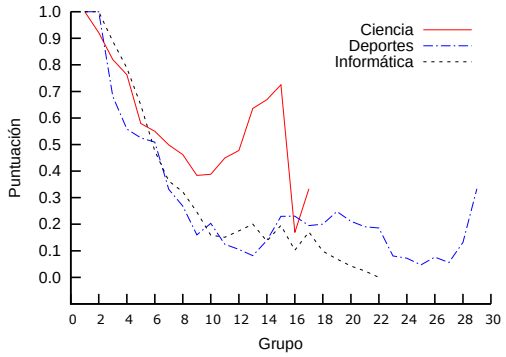
Si algún token de T también pertenece a V:

positivos = positivos + 1

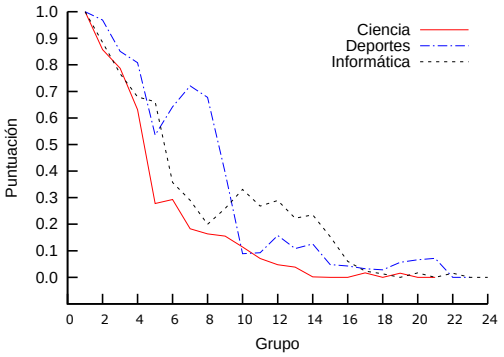
p_i = positivos / total

$P = P \cup \{< i, p_i >\}$

Figura 3.3: Algoritmo de puntuación de grupos de categorías según su afinidad al dominio.



(a) Resultados para la Wikipedia en inglés.



(b) Resultados para la Wikipedia en castellano.

Figura 3.4: Puntuación de grupos de categorías en los dominios ciencia, deporte e informática. Cada grupo de categorías indicado en el eje X está puntuado entre 0 y 1 según lo afines que son al dominio, siendo 0 nada afines y 1 totalmente afines.

categorías marcadas como positivas independientemente del grupo al que pertenezcan. Tras realizar una extracción según los 3 métodos, mediante una inspección visual, se ha observado que el tercer sistema, aún siendo el más preciso en cuanto a los resultados globales, no selecciona categorías que se puedan considerar del dominio dado que sus títulos no contienen palabras del vocabulario. Sin embargo, los otros métodos sí obtienen una buena cobertura sin penalizar en

exceso su precisión. Por este motivo se ha decidido desestimar la extracción (c) y comparar los otros dos conjuntos para finalmente decidir cuál se usará.

Para decidir qué conjunto es mejor se han definido 3 parámetros: (i) el número de categorías extraídas, (ii) la profundidad o distancia que se ha alcanzado, y (iii) el número de artículos que contienen las categorías extraídas. El valor más determinante a la hora de escoger el dominio que se va a usar es el número de artículos extraídos, ya que el tamaño del corpus es directamente proprocional a él. Además hay que tener en cuenta que el número de artículos que realmente se van a utilizar son menos, ya que tan sólo se seleccionarán aquellos que existan en ambas lenguas. En la figura 3.5 está la comparativa de estos parámetros. Se ve que la diferencia de uno o dos niveles de profundidad aumentan considerablemente el tamaño del conjunto, tanto de categorías como de artículos y, además, que el conjunto más restrictivo (puntuación superior a 0,6) extrae muy pocos artículos. Por ello se ha decidido trabajar con las categorías seleccionadas por el método (a) .

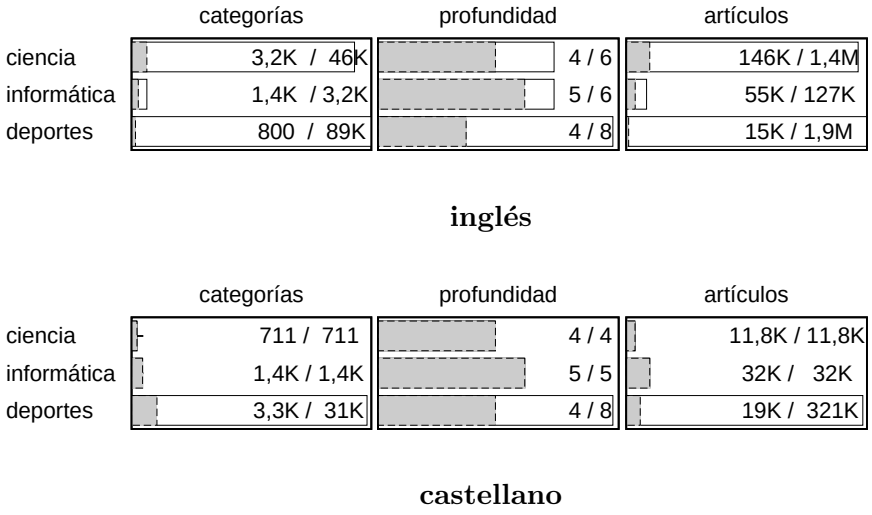


Figura 3.5: Comparación de la selección de categorías y artículos del dominio con los umbrales 0,6 (barra oscura) y 0,5 (barra clara). Observe que las barras de *deporte* correspondientes a categorías y artículos en inglés con umbral=0,5 van más allá de la gráfica.

3.3. Selección de documentos comparables

En esta sección se explican los últimos pasos a seguir para obtener un corpus comparable válido a partir de Wikipedia. Para ello se parte de los listados de categorías descritos en la sección 3.2.2 para cada dominio e idioma y se extraen los identificadores de las páginas de tipo artículo que contiene cada una de ellas.

Tabla 3.2: Pares de artículos seleccionados para cada dominio.

Ciencia	Deporte	Informática
161,130	72,315	18,168

En la figura 3.6 se expone el algoritmo utilizado para obtener los pares de artículos con los que se ha trabajado en cada uno de los dominios. El primer paso es conseguir de forma independiente los identificadores de los artículos contenidos en cada una de las listas para luego buscar aquellos que identifican el mismo artículo en ambos idiomas. La obtención de los conjuntos de artículos para cada listado es trivial utilizando la biblioteca JWPL, ya que abstrae la entidad categoría con la clase `de.tudarmstadt.ukp.wikipedia.api.Category` que implementa la función `getArticlesIds():Set<Integer>`. Después hay que procesar los dos conjuntos obtenidos y ver, en ambos sentidos, cuáles contienen *langlinks*[14] que los conecten a alguno de los artículos del otro idioma. En la tabla 3.2 se muestra la cantidad de pares de artículos extraídos para cada dominio.

[14]Enlaces internos de Wikipedia que conectan una página con la homóloga en otro idioma. Los volcados de Wikimedia contienen tablas que recogen estainformación. `http://www.mediawiki.org/wiki/Manual:Langlinks_table`

Extracción de pares de artículos: Dados dos idiomas i, j y los conjuntos de categorías C_i y C_j:

// Obtener los conjuntos de artículos

$A_i = \emptyset$;

Para cada c en C_i:

    $A_i = A_i \cup c$.getArticlesIDs();

$A_j = \emptyset$;

Para cada c en C_j:

    $A_j = A_j \cup c$.getArticlesIDs();

// Reconocer los pares de artículos

$P_ij = \emptyset$;

Para cada a en A_i:

    $L_a = a$.langlinks();

    Para cada l en L_a:

        Si ($l \in A_j$):

            $P_ij = P_ij \cup < a, l >$;

Para cada a en A_j:

    $L_a = a$.langlinks();

    Para cada l en L_a:

        Si ($l \in A_i$):

            $P_ij = P_ij \cup < l, a >$;

retorna P_ij;

Figura 3.6: Extracción de los artículos de un dominio comunes a dos Wikipedias dados los conjuntos de categorías que definen el dominio.

Capítulo 4

Extracción del corpus paralelo

Para poder entrenar traductores es necesario obtener un corpus paralelo, debiendo estar alineado a nivel de oración en nuestro caso. La peculiaridad de un corpus alineado es que se puede relacionar cada oración del idioma origen con el par suyo en el idioma destino de forma directa al ocupar ambas la misma posición dentro del corpus. En esta sección se exponen los pasos a seguir para obtenerlo a partir del corpus comparable descrito en el capítulo 3.

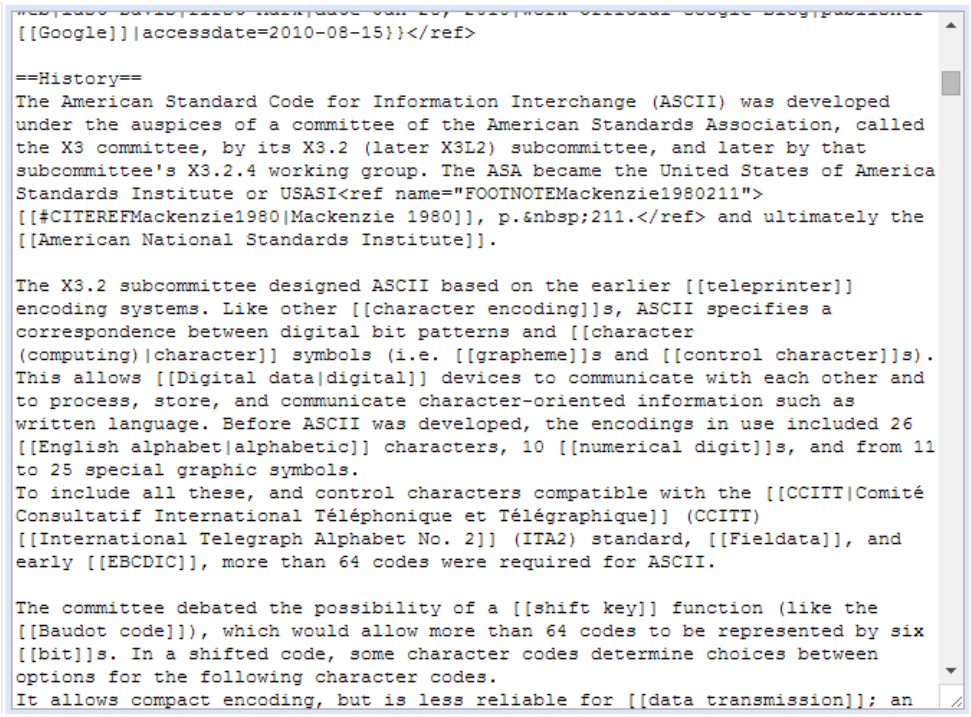
4.1. Preprocesamiento de los textos

En este proyecto han sido necesarios dos tipos de tratamiento de los textos de los artículos obtenidos de Wikipedia. Esto se debe a los requisitos solicitados por los modelos de similitud translingüe seleccionados para generar los corpus deseados (cf. sección 4.2). Los formatos necesarios para los modelos son: texto plano y texto plano traducido. En esta sección se describe la adquisición del texto plano, mientras que la forma en que se ha traducido se expone en el momento en que se necesita dicho formato.

Los artículos de Wikipedia importados a las bases de datos están en *wikitexto*[1], lo que dificulta la extracción de fragmentos en texto plano ya que contiene varios caracteres no deseados. En la figura 4.1 se muestra el wikitexto del artículo "ASCII" almacenado en la base de datos y la visualización que genera Wikipedia al procesarlo. Para poder llevar a cabo la tarea de interpretar el wikitexto que se ha importado a la base de datos se ha utilizado JWPL. Esta biblioteca permite acceder de forma estructurada al contenido de Wikipedia mediante el uso de objetos Java. Además, esta herramienta incorpora un *parser* o analizador sintáctico de páginas que posibilita la inspección de las mismas a nivel de sección, párrafo, lista, tabla o contenidos

[1]El wikitexto se elabora mediante la combinación de texto plano con etiquetas de algún lenguaje de marcado que puedan ser procesadas por un software para wikis. (Wikipedia, 2014c)

multimedia de forma que no es necesario generar gramáticas por parte nuestra que hagan este trabajo.



```
[[Google]]|accessdate=2010-08-15}}</ref>

==History==
The American Standard Code for Information Interchange (ASCII) was developed
under the auspices of a committee of the American Standards Association, called
the X3 committee, by its X3.2 (later X3L2) subcommittee, and later by that
subcommittee's X3.2.4 working group. The ASA became the United States of America
Standards Institute or USASI<ref name="FOOTNOTEMackenzie1980211">
[[#CITEREFMackenzie1980|Mackenzie 1980]], p. 211.</ref> and ultimately the
[[American National Standards Institute]].

The X3.2 subcommittee designed ASCII based on the earlier [[teleprinter]]
encoding systems. Like other [[character encoding]]s, ASCII specifies a
correspondence between digital bit patterns and [[character
(computing)|character]] symbols (i.e. [[grapheme]]s and [[control character]]s).
This allows [[Digital data|digital]] devices to communicate with each other and
to process, store, and communicate character-oriented information such as
written language. Before ASCII was developed, the encodings in use included 26
[[English alphabet|alphabetic]] characters, 10 [[numerical digit]]s, and from 11
to 25 special graphic symbols.
To include all these, and control characters compatible with the [[CCITT|Comité
Consultatif International Téléphonique et Télégraphique]] (CCITT)
[[International Telegraph Alphabet No. 2]] (ITA2) standard, [[Fieldata]], and
early [[EBCDIC]], more than 64 codes were required for ASCII.

The committee debated the possibility of a [[shift key]] function (like the
[[Baudot code]]), which would allow more than 64 codes to be represented by six
[[bit]]s. In a shifted code, some character codes determine choices between
options for the following character codes.
It allows compact encoding, but is less reliable for [[data transmission]]; an
```

(a) Wikitexto del artículo. (Junio de 2013)

History [edit]

The American Standard Code for Information Interchange (ASCII) was developed under the auspices of a committee of the American Standards Association, called the X3 committee, by its X3.2 (later X3L2) subcommittee, and later by that subcommittee's X3.2.4 working group. The ASA became the United States of America Standards Institute or USASI[15] and ultimately the American National Standards Institute.

The X3.2 subcommittee designed ASCII based on the earlier teleprinter encoding systems. Like other character encodings, ASCII specifies a correspondence between digital bit patterns and character symbols (i.e. graphemes and control characters). This allows digital devices to communicate with each other and to process, store, and communicate character-oriented information such as written language. Before ASCII was developed, the encodings in use included 26 alphabetic characters, 10 numerical digits, and from 11 to 25 special graphic symbols. To include all these, and control characters compatible with the Comité Consultatif International Téléphonique et Télégraphique (CCITT) International Telegraph Alphabet No. 2 (ITA2) standard, Fieldata, and early EBCDIC, more than 64 codes were required for ASCII.

The committee debated the possibility of a shift key function (like the Baudot code), which would allow more than 64 codes to be represented by six bits. In a shifted code, some character codes determine choices between options for the following character codes. It allows compact encoding, but is less reliable for data transmission; an

(b) Vista del artículo ya procesado en el navegador.

Figura 4.1: Ejemplo de visualización en Wikipedia del artículo “ASCII” en inglés.

Además del formato en el que se encuentra la información en la base de datos ha sido necesario tener en cuenta que los artículos de Wikipedia pueden contener una serie de secciones especiales que contienen elementos como referencias, notas o enlaces a páginas externas. Estas secciones se han omitido en el procesamiento ya que la información que contienen no es relevante para el objetivo del proyecto. La figura 4.2 contiene el algoritmo utilizado para obtener el texto

plano de un artículo. El método utilizado es muy simple y obtiene todo el texto excepto el de las secciones no deseadas y el texto de las tablas, ya que éstas no suelen tener oraciones. Genera un texto formado por el título del artículo y el texto válido de las secciones no especiales. Por cada sección se añade su título a un conjunto de cadenas de caracteres y las oraciones existentes en sus párrafos y listas. Finalmente se crea un fichero de texto en el que cada cadena de caracteres se escribe en una línea.

Extracción de texto plano: Dado un artículo a y los nombres de las secciones no deseadas S:

// Inicializar la lista de texto T con el título del artículo
$T = \{$título de $a\}$
// Procesar el artículo por secciones
Para cada sección s en el artículo a:
    Si (título de la sección $s \notin S$):
        $T = T \cup \{$título de la sección $s\}$
        Para cada párrafo $p \in s$:
            $T_p = \{$oraciones en $p\}$
            $T = T \cup T_p$
        Para cada lista l en s:
            $T_l = \{$oraciones en l$\}$
            $T = T \cup T_l$
Escribir fichero con los fragmentos de T

Figura 4.2: Extracción de texto plano de un artículo.

4.2. Estimación de similitudes entre oraciones

La anotación de todos los pares paralelos y comparables del corpus de Wikipedia es un trabajo inmenso para realizarlo de manera manual. El primer paso a seguir es calcular la similitud entre todas las oraciones de los artículos para, como segundo paso, seleccionar con base en este resultado aquellas que se consideren paralelas (cf. sección 4.3).

Para estimar la similitud entre dos oraciones se han utilizado dos medidas: coseno y modelo de longitud (cf. sección 2.1). Los vectores representativos de cada texto a comparar se pueden componer de diversas maneras. En este trabajo se han considerado: la tokenización en n-gramas de caracteres, la obtención de pseudocognados y la generación de bolsas de palabras con ambos textos en un mismo idioma.

Tokenización en n-gramas de caracteres (*cng*). Para obtener la representación es ne-

cesario transformar todo el texto a minúsculas para luego dividirlo en grupos de n caracteres, teniendo en cuenta los espacios y puntuación existentes en los fragmentos originales (McNamee y Mayfield, 2004). Aprovechando la variabilidad que ofrece este sistema al modificar el tamaño de los grupos, en el proyecto se ha calculado la similitud para los diferentes pares de oraciones con $n = \{1, 2, \ldots, 5\}$.

Obtención de pseudocognados (*cog*). De acuerdo con The American Heritage dictionary of the English Language (2011), dadas dos palabras en diferentes idiomas, éstas forman un cognado si comparten un origen etimológico; por ejemplo, los vocablos *name* (inglés) y *nombre* (castellano) forman un cognado al derivar ambas del latín *nōmen*. En este trabajo no se ha hecho ningún tipo de análisis lingüístico sobre las palabras de las oraciones, por lo que en lugar de cognados se usa el término pseudocognado para definir aquellos términos que por su composición se puede suponer que comparten este origen etimológico. Basándose en la descripción de Simard *et al.* (1992) se ha definido que una palabra es un pseudocognado si (*i*) con dígitos, (*ii*) es una combinación de letras y números, o (*iii*) si su longitud es igual o mayor a 4. Para representar una oración de esta manera es necesario eliminar los signos de puntuación y diacríticos antes de separar en tokens el texto. Una vez separados se seleccionan sólo aquellos que cumplen los requisitos descritos, truncando a 4 caracteres aquellos que pertenecen a la última condición.

Traducción de un texto y generación de bolsa de palabras (*mono*). Una bolsa de palabras de un texto es el conjunto formado por los términos que aparecen en dicho texto anotando el número de veces de aparición (Manning y Schütze, 1999). La representación que hemos propuesto se basa en Uszkoreit *et al.* (2010) que propone traducir uno de los textos al idioma del otro y representar ambos textos mediante n-gramas de caracteres para luego hacer una comparación monolíngüe. En este caso se ha optado por tokenizar el texto, aplicarle un proceso de *stemming* y agrupar los *stems* resultantes en una bolsa de palabras en lugar de los n-gramas, ya que es más flexible a la hora de ser comparado, supliendo así las posibles deficiencias del sistema utilizado para traducir. Esta representación se ha realizado en ambos sentidos, es decir, traduciendo los textos de castellano a inglés y viceversa. Para obtener los textos traducidos se ha usado un sistema MOSES previamente entrenado con el corpus Europarl v7[2], tal y como se ha explicado en la sección 2.2.1. Dado que los sistemas MOSES deben cargar los modelos de lenguaje y de traducción al inicio de su ejecución para cada texto a traducir es conveniente agrupar los artículos de forma que esta característica no ralentice la traducción de los textos que sean muy cortos y ésta se realice de forma más rápida. Por otra parte también hay que tener en cuenta que si este agrupamiento genera textos muy largos, el proceso sería más costoso,

[2]La versión v7 de Europarl fue publicada en mayo de 2012. `http://www.statmt.org/europarl/v7/`

motivo por el cuál se ha definido un tamaño aproximado de 90 000 líneas para cada fichero de entrada. El proceso consiste en concatenar los artículos de forma que agrupen el número de líneas definido guardando los *offsets*[3] que indican en qué línea comienza cada artículo agrupado. Una vez creados, los nuevos textos se traducen con MOSES y su resultado, que debe ser del mismo tamaño ya que MOSES traduce a nivel de oración, se separan de acuerdo a los *offsets* almacenados anteriormente. De esta manera se consigue que al dividir las traducciones se formen ficheros que corresponden a un sólo artículo en el idioma origen.

Cálculo de longitudes del texto y comparación mediante factores de longitud (*len*). Este método se basa en que al traducir un texto de una lengua a otra, éste aumenta o disminuye su longitud de forma acorde a ciertos factores de longitud. Teniendo en cuenta esto, este modelo calcula cuál es la probabilidad de que un texto sea traducción de otro en función de las longitudes de ambos (Pouliquen *et al.*, 2003). Por lo tanto, la representación del texto es simplemente su longitud en número de caracteres teniendo en cuenta los signos de puntuación, por lo que podemos decir que este es un modelo que no se basa en el léxico a diferencia de las otras expuestas anteriormente. Una vez obtenidos los dos valores, mediante una distribución normal se calcula cómo de probable es la traducción. Un factor de longitud viene definido por la diferencia de longitud del texto traducido respecto al original. Sin embargo, para poder calcular la probabilidad es necesario dar los valores de μ y σ a la distribución normal. Estos valores se calculan a partir de los factores de longitud obtenidos en un corpus de entrenamiento de forma que μ es el promedio de estas diferencias y σ su desviación estandar. En este caso se ha aprovechado el trabajo de Gonzàlez *et al.* (2014) que a partir del corpus WMT13 (Bojar *et al.*, 2013) calcularon los valores $\mu = 1{,}133$ y $\sigma = 0{,}415$ que definen el factor de longitud al traducir del inglés al castellano.

Una vez implementados los diferentes modelos de similitud se ha procedido a aplicarlos sobre todos los pares de oraciones que se han obtenido de los artículos seleccionados. Cada sistema ha sido ejecutado tomando como entrada el conjunto total de pares disponibles, generando así una colección de resultados por cada uno de ellos. En la figura 4.3 se observa cómo se genera una *matriz de similitud* para un par de artículos cualquiera. Inicialmente se toman todos los fragmentos de texto de los artículos, las oraciones en este caso para posteriormente crear una matriz $M_{N \times M}$, siendo N el número de oraciones del primer artículo y M el del segundo. En esta matriz se almacenan los valores de similitud entre los pares de oraciones identificados por i y j respectivamente, es decir, la posición dada por la fila i y la columna j indica la similitud entre

[3] Anglicismo utilizado para referirse al desplazamiento que hay que recorrer en un fichero para acceder a la información deseada.

la oración i en el idioma origen y la oración j en el idioma destino.

Cálculo de similitud para un par de artículos: Dados los artículos a y b:

// Inicializar matriz de similitudes y los conjuntos de oraciones
Crear la matriz $M_{n \times p}$ donde $n = a.longitud$ y $p = b.longitud$.
A =fragmentos de texto del artículo a
B =fragmentos de texto del artículo b
// Calcular las similitudes para todos los pares posibles
Para cada fragmento f_A en A:
    Para cada fragmento f_B en B:
        $M[f_A][f_B] = similitud(f_A, f_B)$;
retorna M;

Figura 4.3: Cálculo de la similitud de oraciones entre dos artículos de Wikipedia.

Para poder ver cómo están repartidos los valores dentro de una matriz se ha implementado una pequeña aplicación que genera un mapa de calor de las similitudes calculadas. En la figura 4.4 se puede apreciar uno de estos mapas de calor. El eje X indexa las oraciones del artículo en castelano, mientras que el eje Y las de inglés, siendo el cuadrado superior izquierdo el correspondiente con la similitud entre los primeros segmentos de cada artículo. En este mapa se ha tomado el tono blanco para indicar que no hay similitud y tonos más oscuros según ésta aumenta. En general se ve que las oraciones con similitudes altas no están agrupadas en una zona específica sino que se reparten de forma más o menos homogénea por toda la matriz representada.

4.3. Extracción del corpus paralelo

En esta sección se exponen los pasos para obtener un conjunto de oraciones paralelas a partir de artículos de Wikipedia. El primer paso es hacer un análisis de las similitudes calculadas. Para ello se ha procedido a hacer una anotación manual de algunos artículos con el fin de tener referencias buenas con las que poder contrastar los resultados.

4.3.1. Anotación manual de artículos comparables

Para poder evaluar la bondad de los modelos de similitud es necesario generar un conjunto de referencia que contenga oraciones paralelas y comparables. Dos oraciones son comparables si contienen texto que, si bien no es una traducción una de la otra, significa prácticamente lo mismo o si una parte es traducción exacta, mientras que la otra no lo es. Un ejemplo de ello es el siguiente par de oraciones

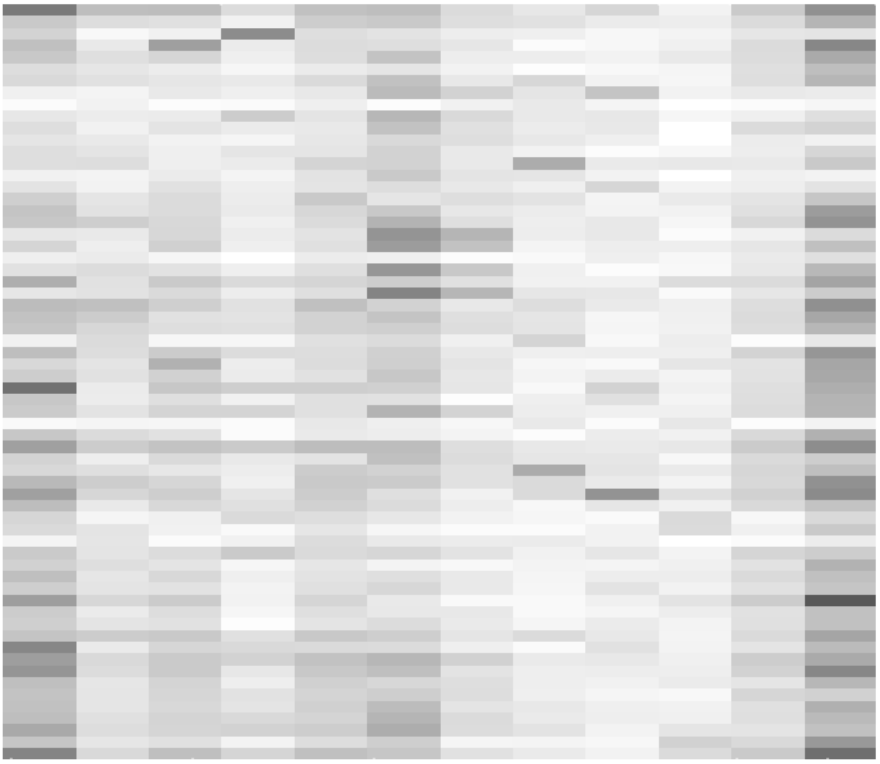


Figura 4.4: Mapa de calor del artículo “GameSpy Industries”, del dominio de Informática. En él se muestran en color oscuro las similitudes más altas y en blanco las más bajas, tomando diferentes tonalidades de gris el resto de ellas.

He also got his first strikeout on that day
También consiguió su primer ponche en ese día, si bien no sería el único de su carrera

Este paso es recomendable que sea manual ya que no se puede tener una certeza absoluta sobre ningún sistema para aceptar los resultados que nos dé sin ningún tipo de supervisión, por ese motivo se han seleccionado algunos pares de artículos de los que un grupo de voluntarios ha extraído oraciones paralelas y comparables. Para seleccionar los pares de artículos se han tenido en cuenta las siguientes métricas sobre los ficheros de las páginas ya preprocesadas:

1. **Longitud de fichero**. Este atributo limita los pares de artículos de forma que ninguno de los ficheros que forman el par tuvieran menos de 3 ni más de 60 líneas. Los artículos restantes se ponderaron por grupos según la longitud del artículo más largo del par, teniendo mayor probabilidad de ser escogido un artículo corto y menor uno largo.

2. **Divergencia entre longitudes**[4]. Con este atributo se eliminan los pares con una divergencia menor a 0,25 y se agrupan el resto en tres grupos (a) $0{,}25 < divergencia <= 0{,}5$, ($b$) $0{,}5 < divergencia <= 0{,}75$ y (c) $0{,}75 < divergencia <= 1{,}0$.

Una vez obtenidos los pares de artículos y organizados en grupos según su longitud y su divergencia se han escogido 300 pares para cada dominio de forma aleatoria. En el algoritmo de selección de estos pares se le ha dado la misma probabilidad a los diferentes grupos de divergencia y mayor probabilidad a los pares con más peso según su longitud. Después de este paso automático se han seleccionado 10 artículos de cada dominio para ser anotados manualmente por dos personas cada uno (cf. anexo A). En la figura 4.5 se puede ver la interfaz gráfica desarrollada anteriormente a este proyecto, utilizada en el proceso de anotación. Esta interfaz muestra los dos ficheros del par en sendos cuadros de texto y el anotador tan sólo tiene que escoger cada par seleccionándolo con el ratón y posteriormente indicar si se trata de una par paralelo o uno comparable mediante el botón correspondiente.

Figura 4.5: Captura de pantalla de la herramienta de anotación (artículo "Carlos Rosa"). En la parte izquierda se muestran las oraciones en inglés, mientras que en la derecha están las de castellano. La herramienta permite seleccionar pares de oraciones y marcarlos como paralelos o comparables.

El resultado de esta anotación se ha guardado para poder evaluar los modelos de similitud y tomar decisiones como qué modelo puede proveer un mejor corpus paralelo o si es necesario combinar los resultados de alguna manera. Para determinar la calidad de las anotaciones se han calculado los acuerdos entre anotadores mediante el coeficiente kappa de Cohen (Cohen, 1960), el cual mide el acuerdo entre dos evaluadores al clasificar N elementos en C categorías, siendo en este caso N el número de pares de oraciones y $C = \{paralelo, comparable, ninguna\}$:

[4]La divergencia de un par de artículos se calcula dividiendo la longitud del artículo más corto entre la del más largo. Cuanto más próximo a 1 sea el valor, más similares son las longitudes.

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{4.1}$$

donde $Pr(a)$ es el acuerdo observado entre los dos evaluadores y $Pr(e)$ la probabilidad hipotética de acuerdo en caso de que los anotadores realicen de forma aleatoria su tarea (Carletta, 1996). En la tabla 4.1 se observan los resultados de la evaluación de la anotación manual realizada. Esta tabla muestra por segmentos los valores del coeficiente kappa para cada anotador (A_1) incluyendo el identificador del otro anotador (A_2) y los identificadores de los artículos anotados por él (ID). Finalmente, para cada anotador, se muestra en negrita el valor promedio de los acuerdos a los que ha llegado con los otros anotadores, de forma que se pueda ver fácilmente si la anotación ha sido correcta o ha habido algún desacuerdo. A la vista de los buenos resultados obtenidos al evaluar el acuerdo entre los anotadores, se ha decidido dar por válidos los pares indicados por ellos como referencia a para evaluar los modelos de similitud.

4.3.2. *Tuning*

Para poder hacer un *tuning* o ajuste de parámetros de forma correcta es necesario evaluar los valores de similitud calculados sobre el conjunto de pares anotados manualmente y así tomar las decisiones más convenientes. La evaluación de las medidas de similitud se ha realizado calculando la precisión y cobertura obtenida por cada una de ellas tomando como referencia los pares de artículos anotados. Como es necesario tener un grupo de candidatos escogidos de forma automática se ha definido el conjunto genérico $U = \{0{,}0,\ 0{,}05,\ \ldots,\ 1{,}0\}$, el cual define los valores umbral para seleccionar los elementos que definen cada subconjunto de similitudes a evaluar. De esta manera, dado un umbral $u \in U$, el grupo de oraciones O_u está formado por todos los elementos con una similitud mayor o igual al umbral, es decir, dada la similitud $x, x \in O_u \iff x \geq u$. A continuación se analizan los resultados de esta evaluación para los 9 modelos de similitud considerados. Se ha seleccionado el máximo alcanzado por F_1 para seleccionar qué umbral es el mejor a aplicar a cada conjunto de similitudes para maximizar los resultados en cada uno de ells. Finalmente se detalla cómo se han utilizado estas medidas y qué mejoras se han probado sobre las similitudes obtenidas.

Análisis de resultados

El primer método calculado ha sido la tokenización de fragmentos de texto en n-gramas de caracteres, utilzando los valores $n = 1 \ldots 5$. Estos modelos se nombran sustituyendo el valor de n en el acrónimo *cng*. En la figura 4.6 podemos observar las gráficas resultantes del análisis.

Tabla 4.1: Acuerdo entre anotadores en la anotación manual de pares de artículos de Wikipedia.

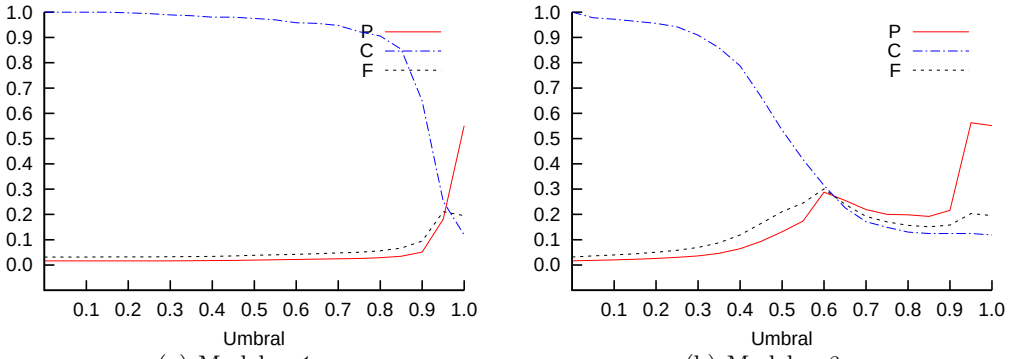
A_1	A_2	**ID**	κ
1	3	1016238	0,35
1	6	2439963	0,67
1	2	2899220	0,84
1	3	3617429	0,71
1	2	5250166	1,00
1	8	5585748	1,00
1	4	941121	0,94
1			0,79
2	1	2899220	0,84
2	1	5250166	1,00
2	8	121982	0,95
2	7	1442469	0,28
2	5	2090638	1,00
2	7	4358215	0,88
2	3	508140	0,49
2	7	741312	0,75
2			0,76
3	1	1016238	0,35
3	1	3617429	0,71
3	2	508140	0,49
3	6	1091160	0,52
3	8	152918	0,86
3	6	2720529	0,76
3	5	3175837	0,70
3	8	948901	0,33
3			0,62

A_1	A_2	**ID**	κ
4	1	941121	0,94
4	9	1077240	0,59
4	6	145321	0,48
4	6	1695687	0,66
4	9	2436477	0,70
4	9	3193649	0,72
4			0,63
5	2	2090638	1,00
5	3	3175837	0,70
5	9	297342	0,78
5	7	3329373	0,84
5	6	5672222	0,82
5	8	5777133	0,65
5			0,76
6	1	2439963	0,67
6	3	1091160	0,52
6	3	2720529	0,76
6	4	145321	0,48
6	4	1695687	0,66
6	5	5672222	0,82
6	7	1475656	0,56
6	7	5124707	0,19
6			0,57

A_1	A_2	**ID**	κ
7	2	1442469	0,28
7	2	4358215	0,88
7	2	741312	0,75
7	5	3329373	0,84
7	6	1475656	0,56
7	6	5124707	0,19
7	8	5344091	0,74
7			0,66
8	1	5585748	1,00
8	2	121982	0,95
8	3	152918	0,86
8	3	948901	0,33
8	5	5777133	0,65
8	7	5344091	0,74
8			0,71
9	4	1077240	0,59
9	4	2436477	0,70
9	4	3193649	0,72
9	5	297342	0,78
9			0,73

Mirando la curva producida por la cobertura se observa que su tendencia es negativa y que cuanto mayor es el valor de n, más desciende. Esto se debe a que cuanto más grandes son los n-gramas, es menos probable que coincidan los obtenidos en el idioma origen con los obtenidos en el idioma destino.

Respecto a la precisión, tiene dos características a comentar: la primera es que a pesar de tener una tendencia ascendente muestra cambios de pendiente a medida que se va incrementando el umbral, y la segunda es que los máximos de la curva se van desplazando hacia umbrales



(a) Modelo *c1g* (b) Modelo *c2g*

(c) Modelo *c3g* (d) Modelo *c4g*

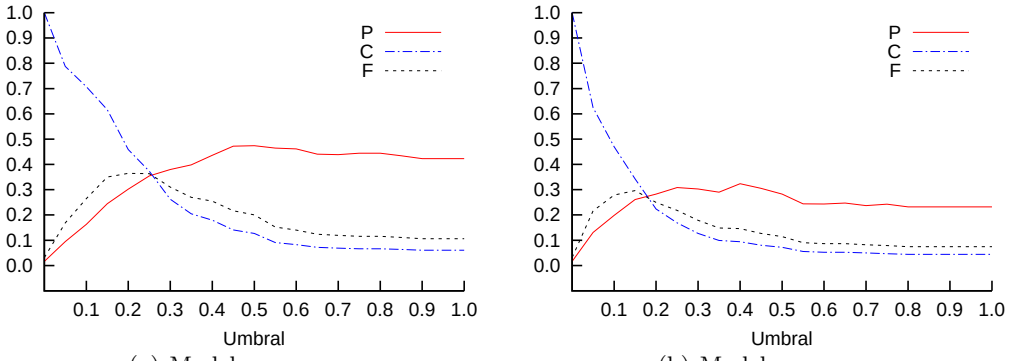
(e) Modelo *c5g*

Figura 4.6: Precisión y cobertura del modelo *cng*.

más bajos a medida que se incrementa n. La variación de la pendiente se explica teniendo en cuenta el comportamiento de los dos parámetros que definen la precisión, el número de elementos correstamente extraídos y el número total de elementos extraídos. Para tener una tendencia ascendente es necesario que se incrementen los casos positivos pero no los casos totales o que se decremente el número de elementos sin perder casos correctos. Mientras que para que decremente debe ocurrir lo contrario. Tomando estas dos aserciones y sabiendo que a medida que se incrementa el umbral el número de elementos seleccionados disminuye es trivial deducir que los cambios de pendiente se deben a la pérdida o ganancia brusca de casos positivos. Un buen ejemplo de esto es la figura 4.6(b) que sufren un cambio a descendente en el umbral 0,6 y luego

otro cambio ascendente con el umbral 0,8. Sin embargo en la figura 4.6(a) se ve que para $n = 1$ no llega a producirse este efecto ya que el tamaño de sus tokens hace que los valores de similitud obtenidos sean muy altos y se acumulen entre 0,8 y 1,0. La segunda característica comentada anteriormente sobre la precisión se debe a dónde se acumulan las similitudes obtenidas. En general, el primer máximo de la curva indica el umbral alrededor del cual se distribuyen la mayor parte de elementos calculados que realmente son paralelos, o al menos comparables. Por ejemplo, en la figura 4.6(c) se ve un primer máximo en el umbral 0,35 que nos indica que aplicando el método *c3g*, los resultados de elementos paralelos se encuentran en torno a dicho valor.

Finalmente, hay que analizar la curva de F_1, ya que indica qué método ha sido mejor sobre el conjunto de artículos anotados, hecho que genera la hipótesis de que cuanto mayor sea el valor máximo de F_1, mejor es el modelo de similitud a la hora de obtener las oraciones paralelas para el corpus que se usará para entrenar los traductores. En el caso de los n-gramas de caracteres se ve claramente que esta curva es muy similar a la de la precisión. En la tabla 4.2 se observa que desde $n = 1$ hasta $n = 3$ el valor máximo de F_1 se incrementa y a partir de ese valor va decreciendo. Por lo tanto el modelo *c3g* es el que mejor resultados da para esta familia de similitudes, aunque los resultados para el resto de valores, excepto $n = 1$, también indican que se podrían obtener pares paralelos utilizándolos como sistema de extracción.



(a) Modelo *mono-en* (b) Modelo *mono-es*

Figura 4.7: Precisión y cobertura del modelo *mono*.

El siguiente modelo a analizar es el propuesto por Uszkoreit *et al.* (2010), que se basa en la comparación monolíngüe de dos textos, uno en el idioma original (también llamado base) y el otro traducido a dicho idioma. En primer lugar, indicar que la figura 4.7(a) muestra que al traducir textos de castellano a inglés para hacer la comparación monolíngüe se obtienen mejores resultados que al hacer la traducción a la inversa (figura 4.7(b)). Respecto a la cobertura, ambas gráficas tienen una curva muy similar que desciende rápidamente al incrementar los umbrales

Tabla 4.2: Valores máximos de F_1 para las estimaciones de similitud realizadas junto con los umbrales en los que se ha obtenido.

Modelo	**F_1**	**Umbral**
c1g	0,2109	0,95
c2g	0,3008	0,60
c3g	0,3642	0,25
c4g	0,3184	0,20
c5g	0,3120	0,15
cog	0,2424	0,30
mono-en	0,3644	0,20
mono-es	0,2963	0,15
len	0,1368	0,90

y más suavemente a partir de los umbrales 0,55 para el modelo *mono-en* y 0,35 para *mono-es*. La curva de la precisión también es muy similar aunque en el caso de tomar español como idioma de comparación (*mono-es*) se vuelve a producir el mismo efecto que con el modelo basado en n-gramas, es decir, a partir de cierto punto sufre un descenso pronunciado (umbral 0,4). Sin embargo la gráfica 4.7(a) se comporta de la manera ideal que se espera en este tipo de experimento, es decir, la precisión va en aumento a medida que se reduce el tamaño del conjunto de elementos escogidos. Finalmente, la tabla 4.2 muestra los umbrales en los que se alcanza la mejor medida F_1 para este modelo. Tal y como se adelantaba antes, el modelo *mono-en* recibe una mejor puntuación entre las dos variantes, siendo ésta mínimamente mejor que el método *c3g*.

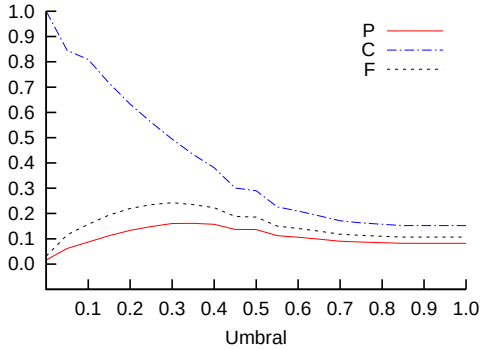


Figura 4.8: Precisión y cobertura del modelo *cog*.

En tercer lugar está el resultado del modelo basado en cognados. La notación usada para

nombrar a este método es *cog*. En la figura 4.8 se observa que las curvas tienen tendencias similares a las del modelo *mono*. Esto se debe a que el tipo de comparación es muy similar ya que, aunque no se traduce ningún artículo, la detección de pseudocognados genera una bolsa de palabras que incluye los valores numéricos y alfanuméricos del texto de forma completa y los alfabéticos truncándolos a 4 caracteres (cf. . sección 4.2) con lo que se consigue que los tokens que no componen palabras se mantengan (numéricos y alfanuméricos) y los que son palabras sólo añadan la información del inicio de su formación (*stems* truncados a 4 caracteres). Esto hace que las bolsas de palabras obtenidas sean bastante similares entre sí a pesar de no haber traducido ningún texto. Por tanto, la cobertura desciende de una manera más o menos contínua mientras que la precisión tiene una tendencia positiva hasta alcanzar un máximo en el umbral 0,35 y luego desciende un poco, manteniéndose en torno al 0,1. Respecto a la medida F_1, su curva es muy similar a la de la precisión, obteniendo su máximo en el umbral 0,3, tal y como muestra la tabla 4.2. Sin embargo éste obtiene una puntuación bastante baja en comparación con los descritos anteriormente.

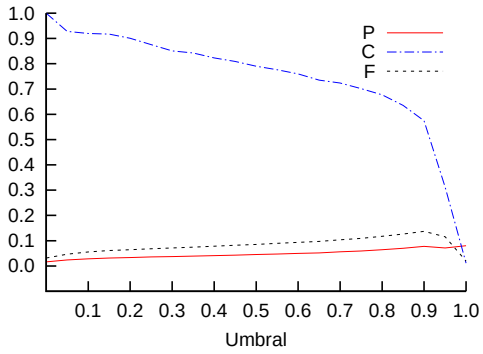


Figura 4.9: Precisión y cobertura del modelo *len*.

El último método es el basado en factores de longitud (*len*). La figura 4.9 muestra los resultados obtenidos, los cuales son muy similares a los calculados con el modelo *c1g*, hecho que se considera normal ya que son los dos modelos que menos información léxica tienen en cuenta para calcular sus valores. La cobertura se mantiene bastante alta hasta llegar al umbral 0,9 en el que cae bruscamente. En cuanto a la precisión, siempre es ascendente, sin presentar ningún tipo de máximo relativo, aunque nunca supera la puntuación 0,1. Esta particularidad junto con su valor máximo de F_1 hace imaginar que este modelo no es de los mejores para obtener corpus paralelos, aunque es un modelo a tener en cuenta para combinar con otros gracias a los resultados obtenidos por Barrón-Cedeño *et al.* (2013) a la hora de identificar fragmentos paralelos con este sistema. En la tabla 4.2 se ve que este modelo de similitud translingüe es el que peor resultados da, con un valor máximo casi una décima inferior que el modelo *c1g*, que

era el que había obtenido la puntuación más baja hasta este momento.

Ajuste de las medidas de similitud

Los resultados obtenidos han sido reveladores para ver que en general todos los modelos de similitud dan resultados similares a excepción de *c1g* y *len*, siendo este último el peor de todos. Por ello se ha decidido realizar algunas combinaciones de los resultados de similitud obtenidos por los modelos con el fin de mejorar los valores obtenidos. Estas combinaciones se notarán como *comb* al referirse a ellas como grupo, y son:

- ***mean***. Esta combinación se realiza calculando el promedio de los modelos léxicos[5]. Es decir, dado el conjunto ML de modelos léxicos y las oraciones a y b, el valor $mean(a, b)$ se define como:

$$mean(a, b) = \frac{\sum_{i=1}^{|ML|} ML_i(a, b)}{|ML|} \tag{4.2}$$

- ***mean_len***. Esta medida se obtiene ponderando los valores del método *mean* por el valor correspondiente que da el modelo de factores de longitud. Es decir, dado el conjunto ML de modelos léxicos y las oraciones a y b, el valor $mean\_len(a, b)$ se define como:

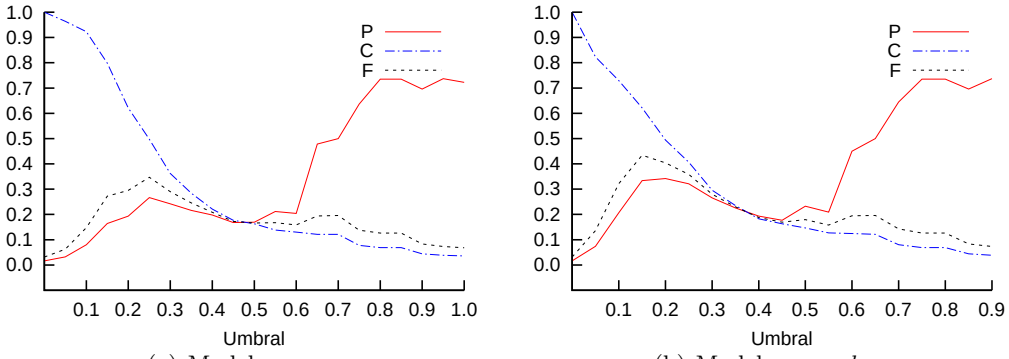
$$mean\_len(a, b) = len(a, b) \cdot \frac{\sum_{i=1}^{|ML|} ML_i(a, b)}{|ML|} \tag{4.3}$$

- ***mean_f***. Este resultado se calcula haciendo la media ponderada de las similitudes dadas por los modelos léxicos. Para ponderar se utiliza el valor de la F_1 máxima obtenida por cada uno de ellos. Es decir, dado el conjunto ML de modelos léxicos, los valores F_1 de los modelos y las oraciones a y b, el valor $mean\_f(a, b)$ se define como:

$$mean\_f(a, b) = \frac{\sum_{i=1}^{|ML|} ML_i(a, b) \cdot F_{1_i}}{|ML|} \tag{4.4}$$

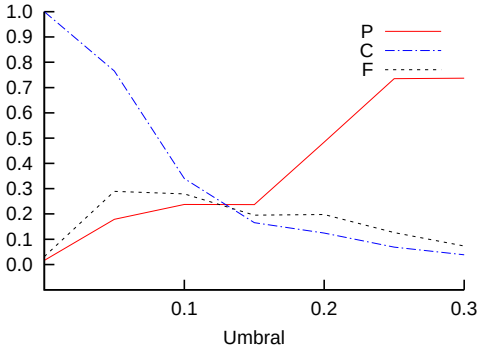
- ***mean_f_len***. Al igual que el método *mean_len*, esta similitud se calcula multiplicando los valores del método *mean_f* por el valor correspondiente al modelo de factores de longitud. Es decir, dado el conjunto ML de modelos léxicos, los valores F_1 de los modelos y las oraciones a y b, el valor $mean\_f\_len(a, b)$ se define como:

$$mean\_f\_len(a, b) = len(a, b) \cdot \frac{\sum_{i=1}^{|ML|} ML_i(a, b) \cdot F_{1_i}}{|ML|} \tag{4.5}$$

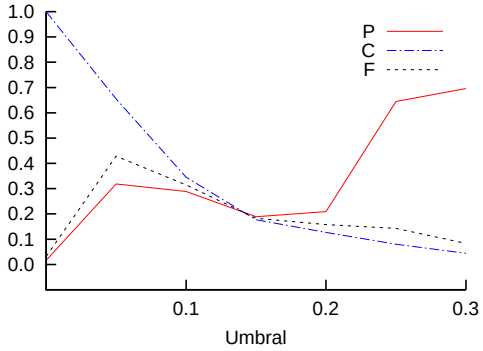


(a) Modelo *mean*

(b) Modelo *mean_len*



(c) Modelo *mean_f*



(d) Modelo *mean_f_len*

Figura 4.10: Precisión y cobertura del modelo *comb*.

En la figura 4.10 se muestra la evaluación de las cuatro combinaciones sobre el mismo conjunto de referencia. Lo primero de todo hay que destacar que a excepción del *mean* el resto de gráficas no muestran valores por encima de ciertos umbrales, ya que no es posible generar ningún subconjunto de oraciones a partir de ese punto[6]. Dejando de lado esta particularidad se observa que todas las combinaciones presentan una curva de cobertura descendente muy similar tal y como se espera. Respecto a las curvas de precisión, su comportamiento también es muy parecido en todas las combinaciones. Todas las curvas se pueden separar en tres tramos: (*i*) ascenso brusco en los umbrales inferiores, (*ii*) descenso leve en los umbrales intermedios y, (*iii*) ascenso de nuevo en los valores más altos. Esto se debe a que las medidas calculadas inicialmente no agrupan las similitudes en el mismo rango de umbrales, sino que unas lo hacen en los valores más bajos y otras en los más altos, lo que provoca ese leve descenso en el tramo (*ii*) . Respecto al valor de F_1, la tabla 4.3 muestra que sólo dos combinaciones (*mean_len* y *mean_f_len*) suponen una mejora considerable respecto a los valores sin combinar, hecho que confirma la hipótesis formulada sobre la ayuda que aporta el modelo *len* al ser combinado conotro método para ajustar las similitudes.

[5]Los modelos definidos como léxicos son: *c1g*, *c2g*, *c3g*, *c4g*, *c5g*, *cog*, *mono-en*, *mono-es*

[6]Estos umbrales son: 0,30 para los casos *mean_f* y *mean_f_len* y 0,90 para el modelo *mean_len*)

Tabla 4.3: Valores máximos de F_1 para las combinaciones de los modelos de similitud aplicados junto con los umbrales en los que se ha obtenido.

Modelo	F_1	**Umbral**
mean	0,3472	0,25
mean_len	0,4344	0,15
mean_f	0,2894	0,05
mean_f_len	0,4286	0,05

4.3.3. Extracción final

Una vez obtenidas todas las similitudes, tanto de forma individual como combinada, es el momento de extraer los fragmentos que compondrán los corpus paralelos que se van a utilizar. Para esta tarea se usa un método de extracción que ordena todos los pares de oraciones de mayor a menor similitud y selecciona las k primeras, que de acuerdo a las métricas calculadas son las que mejor resultado deben dar. Una vez obtenido este conjunto, se extraen todos los pares siguientes que tengan una similitud igual que la del par k-ésimo ya que no pueden ser descartadas simplemente porque en la ordenación hayan acabado en una posición más allá del límite impuesto por k. En este caso se han generado dos resultados uno con $k = 1\,000\,000$ y otro con $k = 2\,000\,000$. En el primer caso una inspección visual ha determinado que el corpus obtenido no era bueno ya que contenía muchos títulos y nombres propios y pocas oraciones completas, por lo que finalmente se ha decidido utilizar los conjuntos de oraciones obtenidos para $k = 2\,000\,000$. A partir de este corpus se han clasificado los pares según el dominio al que pertenecen para poder así utilizarlos de forma que los traductores que los utilicen se "especialicen" en el área que corresponda. En la tabla 4.4 se muestra el número de pares de artículos extraídos para cada dominio según los diferentes modelos de similitud.

Tabla 4.4: Tamaño, en número de oraciones, de los corpus obtenidos para cada dominio según el modelo de similitud.

Modelo	**cs**	**sc**	**sp**
c1g	207 592	1 585 582	404 656
c2g	99 964	745 821	326 882
c3g	96 039	724 210	335 147
c4g	110 701	863 090	394 105
c5g	126 692	1 012 993	466 007
cog	182 981	1 215 008	451 941
len	271 073	1 941 866	550 338
mono-en	211 209	1 367 917	461 731
mono-es	183 439	1 273 509	435 671
mean	154 917	1 098 453	450 933
mean_len	121 697	957 662	390 783
mean_f	153 056	1 085 502	448 076
mean_f_len	121 407	957 967	392 241

Capítulo 5

Enriquecimiento del traductor

En este capítulo se expone el trabajo realizado para alcanzar el último objetivo del proyecto, la obtención de sistemas de traducción entrenados a partir del corpus paralelo extraído durante el trabajo precedente. El proceso consta de 3 pasos: entrenamiento, ajuste y evaluación.

5.1. Definición de los traductores y conjuntos de entrada

El primer paso a seguir antes de comenzar a entrenar traductores es analizar los resultados de la extracción de los corpus paralelos con el fin de decidir cuáles de ellos se utilizarán en el entrenamiento de los traductores. Además de los valores obtenidos al calcular la precisión y cobertura de cada uno de los modelos se han obtenido las intersecciones entre los conjuntos de oraciones extraídos para formar los diferentes corpus paralelos. En general, el tamaño de la intersección de los conjuntos extraídos no superaba el 30 %, aunque si solo se comparaban las extracciones realizadas con las variantes de un mismo modelo de similitud, éste sí era más alto. Por ese motivo se ha decidido clasificar las extracciones en familias, siendo de una misma familia todas aquellas que han sido obtenidas con el mismo modelo de similitud. Así pues tenemos 5 familias: *cng*, *cog*, *mono*, *len* y, por último *comb*, formado por las combinaciones descritas en la sección 4.3.2, es decir *mean*, *mean_len*, *mean_f* y *mean_f_len*. Una vez hecha esta agrupación de modelos, dado que todos ellos se han analizado mediante el cálculo de su precisión y cobertura respecto a un conjunto de artículos anotados manualmente, se ha utilizado dicho análisis para tomar esta decisión, concretamente se ha utilizado el valor F_1 para hacer esta selección, tomando los corpus de aquellos que tuvieran la mayor puntuación dentro de su grupo. La tabla 5.1 muestra cuál ha sido el conjunto seleccionado para cada grupo definido. Cabe destacar que el grupo *len*, que solo contenía un modelo de extracción, ha sido descartado debido a la puntuación tan baja que había obtenido, aunque, tal y como se ha visto con los modelos *mean_f_len* y, especialmente,

Tabla 5.1: Corpus seleccionados para entrenar traductores.

Grupo	**Corpus seleccionado**
cng	*c3g*
cog	*cog*
mono	*mono_en*
len	-
comb	*mean_len*

mean_len, es una medida a tener en cuenta para mejorar otras medidas al combinarlas con ella.

Tras esta decisión tenemos 12 corpus, uno por cada sistema y dominio, con los que entrenar, lo que supone entrenar el mismo número de traductores. Además, se han generado corpus más grandes para ver la importancia del tamaño del corpus en nuestro caso y el efecto que tiene sobre el dominio. Estos nuevos corpus se crean de dos formas: (*a*) uniendo las extracciones de cada sistema para todos los dominios, creando así un nuevo dominio global notado como *wk*, y (*b*) haciendo la unión de los pares obtenidos con todos los modelos de similitud por cada dominio definido, excepto el de factores de longitud; este corpus es nombrado *union*. Finalmente se utiliza otro traductor genérico (*baseline*) que sirve de referencia a la hora de comparar los resultados obtenidos. Este último traductor se describe en la sección correspondiente a la evaluación de los traductores.

Cabe destacar que teniendo en cuenta posibles objetivos posteriores a este trabajo, por ejemplo, ampliar una Wikipedia a partir del contenido de su homóloga en otro idioma, se ha decidido entrenar traductores de inglés a castellano, al ser la primera más prolífica que la segunda. Es decir, en caso de que los nuevos traductores sean evaluados de forma satisfactoria se podrían aprovechar para completar artículos en castellano utilizando información de la enciclopedia anglosajona.

5.2. Generación de los conjuntos de entrenamiento, desarrollo y test

Por otra parte, para entrenar un traductor es necesario tener 3 conjuntos disjuntos de oraciones: (*a*) el de entrenamiento, (*b*) el de desarrollo o ajuste y, (*c*) el de referencia o test. La necesidad de disyunción de estos conjuntos se debe a que de esa manera se puede asegurar que ningún paso del proceso ha sido adulterado al tener información "extra" de ningún paso prece-

dente. Por ejemplo, si se utiliza la oración *"El coche es rojo."* para entrenar y ésta está contenida también en el conjunto de referencia, los resultados de la evaluación estarán sobrevalorados ya que el hecho de que el sistema traduzca correctamente esta entrada durante la evaluación no implica que sepa traducir oraciones similares como *"El coche es azul."* sino que la ha memorizado y es capaz de repetirla directamente. En este proyecto se ha decidido obtener estos tres conjuntos de los pares extraídos anteriormente, ya que en el caso del conjunto de desarrollo es una buena técnica para conseguir la especialización por dominio que se está buscando; y en el caso del de test, ya que los textos de referencia deben pertenecer al dominio del traductor en cuestión, es la mejor opción que se tiene. Los conjuntos de desarrollo y test estan formados por 1000 oraciones cada uno y el de entrenamiento por todas las restantes una vez se eliminan las 2000 seleccionadas. Además, antes de comenzar a manipular los corpus, los fragmentos de texto se han cambiado a minúsculas y se han tokenizado de acuerdo a la tokenización esperada por el sistema MOSES con el que se va a trabajar.

Es necesario que los conjuntos de desarrollo y test sean paralelos realmente y, dado que la selección automática deja fragmentos que no lo son, es necesario seleccionar aquellos que cumplen los requisitos. La primera opción utilizada para obtener estos 2000 pares de oraciones por dominio ha sido seleccionarlos del corpus que se considera está mejor formado según su F_1, el conseguido usando las similitudes de *mean_len*. Para llevar a cabo esta selección se ha hecho un filtrado previo ya que no interesa que estos elementos sean muy pequeños (menos de 4 tokens) ni que estén en un idioma diferente al esperado[1] o se formen con dígitos y símbolos únicamente. Para eliminar los elementos no alfanuméricos y de longitud menor a 4 tokens se han utilizado expresiones regulares, mientas que para escoger aquellos elementos que realmente están en el idioma corecto se ha usado la biblioteca language-detection[2], implementada para Java, que permite identificar el idioma de un texto mediante un clasificador Naive Bayes usando n-gramas de caracteres como entrada (Shuyo, 2010). Una vez hecho el filtrado, se han creado dos conjuntos de 1000 pares de oraciones: para el de test se han seleccionado directamente los pares con una similitud mayor y para el de desarrollo se ha hecho una extracción aleatoria entre los 5000 pares más similares. Basarse en los pares con de similitud más elevada ha tenido como consecuencia obtener un conjunto formado básicamente por títulos y nombres propios, tal y como se puede apreciar en la tabla 5.2 que muestra algunos de los pares seleccionados con este método para los diferentes dominios.

[1] El texto extraído contiene elementos en latín, por ejemplo los nombres científicos de plantas o animales; y citas en otros idiomas.

[2] `http://code.google.com/p/language-detection/`

Tabla 5.2: Fragmentos no deseados para los conjuntos de desarrollo y test obtenidos al seleccionar los pares con similitud más alta.

Dominio	**en**	**es**
cs	f (x) = 2x ˆ 3-4x ˆ 2 + 3x	f (x) = 2x ˆ 3-4x ˆ 2 + 3x
sc	georges auric (1899 – 1983) ;	georges auric (1899 – 1983) ;
sp	copa asobal 2003-2004 ,	campeón copa asobal 2003-2004 ,

En la tarea de ajuste de atributos una entrada compuesta por este tipo de texto hace que el traductor no aprenda correctamente, es decir, aprende a traducir los fragmentos dados, pero no consigue saber cómo se deben ordenar otras oraciones que se le den posteriormente; mientras que durante la evaluación este tipo de texto provocaría una puntuación bastante alta[3] pero que realmente no tiene sentido ya que se evaluan pares que no son traducciones realmente, como por ejemplo los nombres propios.

Debido a este resultado se ha considerado otro método para crear los conjuntos necesarios. Este método se basa en el cálculo de la perplejidad de los pares de oraciones candidatos a formar parte del conjunto de test y del de desarrollo. La perpelejidad (*ppl*), en este caso indica cómo de fluida es la oración dentro de un idioma y para calcularla se ha utilizado el algortimo incluido en las herramientas SRILM[4], cuya fórmula es:

$$ppl = 10^{\left(\frac{-logprob}{palabras-OOV}\right)} \tag{5.1}$$

donde *logprob* es la propabilidad dada por el modelo de lenguaje a la oración, *palabras* es el número de vocablos conocidos por el modelo y 00V[5] el número de términos que no están contemplados en el modelo de traducción.

Para poder calcular las perplejidades necesarias se ha utilizado un modelo de lenguaje construido a partir de un corpus formado por el corpus de Europarl en su versión 7 al que se le han añadido los corpus extraídos de Wikipedia. Dado que esta medida se aplica a una frase en un idioma, el valor obtenido para un par se ha calculado realizando el promedio de la perplejidad de cada una de las oraciones del par en la lengua correspondiente. Una vez hecho este paso de cálculo, los pares se han ordenado de menor a mayor perplejidad, ya que cuanto menor sea ésta, más fluida es la oración en su lengua. Finalmente se han seleccionado los 5000 pares con

[3]BLEU $\approx$ 80 en las pruebas realizadas

[4]SRI Language Model Toolkit es una colección de bibliotecas C++, programas y scripts que permiten trabajar con modelos de lenguaje (Stolcke, 2002). El proyecto está mantenido por SRI International. `http://www.speech.sri.com/projects/srilm/`

[5]Acrónimo de *Out of Vocabulary*

Tabla 5.3: Tamaño de los conjuntos de entrenamiento para cada traductor en número de pares de oraciones.

	cs	**sc**	**sp**	**wk**
c3g	95 715	723 760	334 828	883 366
cog	182 283	1 213 965	451 324	1 430 962
mono_en	210 664	1 367 169	461 237	1 638 777
mean_len	120 835	956 346	389 975	1 160 977
union	577 428	3 847 381	1 181 664	4 948 241

valores más altos para hacer un filtrado manual. Este método se ha aplicado para cada uno de los dominios definidos. Debido a que en el momento de ejecutar este método el proyecto tenía un retraso bastante considerable y que el filtrado manual es largo, se ha decidido reducir el tamaño de los conjuntos, siendo ahora de 900 pares para desarrollo, 300 aportados por cada dominio, y 500 para el de test. En el caso del nuevo dominio *wk*, se utiliza la concatenación de estos conjuntos, siendo por tanto tres veces más grande. Este sistema sí ha permitido obtener unos conjuntos que se consideran buenos para ajustar y evaluar el traductor por lo que han sido los definitivos para esta tarea.

En la tabla 5.3 se muestra el tamaño de los corpus de entrada a los traductores entrenados, tomando el par de oraciones como unidad de medida. En ella se aprecia que Informática ha sido el dominio con menos pares recuperados, mientras que ciencia el que más pares ha obtenido para entrenar sus traductores, hecho lógico ya que la proporción de artículos recuperados para cada dominio cumple la misma proporción (cf. tabla 3.2).

5.3. Evaluación de los traductores

Antes de poder extraer conclusiones sobre cuál es el mejor traductor de los que se deben analizar, y para ello es necesario fijar otro traductor como referencia. El sistema de referencia o *baseline* escogido es un sistema MOSES entrenado con el corpus Europarl en su séptima versión y el corpus de 2009 de News Commentary (NC-2009). La tabla 5.4 muestra los datos relacionados con este sistema de referencia, incluyendo los corpus utilizados para cada paso y el tamaño de los mismos.

Tal y como se ha descrito en la sección 2.2.1, el sistema de evaluación utilizado es BLEU, que se calcula con el script `multi-bleu` de MOSES. Una vez definido el traductor que servirá de referente y la puntuación que se comparará para decidir si los sistemas entrenados en este

Tabla 5.4: Información de los corpus utilizados para entrenar y ajustar el traductor usado como referencia.

	Corpus	Oraciones	Tokens en	Tokens es
Entrenamiento	Europarl-v7	1 965 734	49 093 806	51 575 748
Desarrollo	NC-2009	2525	65 595	68 089

Tabla 5.5: Puntuaciones BLEU de los traductores entrenados y el sistema de referencia. Se han marcado con letra negrita las mejores puntuaciones para cada dominio.

	cs	**sc**	**sp**	**wk**
Europarl v7	27,99	34,00	30,02	30,63
c3g	38,81	40,53	46,94	43,68
cog	57,32	56,17	57,60	58,14
mono_en	54,27	52,96	55,74	55,17
mean_len	56,14	57,40	58,39	58,80
union	**64,65**	**62,95**	**62,65**	**64,47**

proyecto son mejores o no, es necesario traducir los textos de referencia con cada uno de los traductores entrenados además de con el *baseline* ya que BLEU evalúa la calidad de un texto traducido por el sistema de traducción frente al mismo texto traducido anteriormente. Este texto, como es lógico, es el conjunto de test.

La tabla 5.5 muestra las puntuaciones BLEU obtenidas por los nuevos sistemas y por el traductor *baseline*, estando en negrita aquellos que mejor resultado han dado para cada dominio. En los resultados se ve que la unión de todos los corpus extraídos con diferentes modelos de similitud es la mejor opción en todos los casos. Este hecho es bastante lógico ya que por su naturaleza, el corpus de entrenamiento es mayor, consiguiendo así un mayor vocabulario y muchas más referencias a la hora de estimar las probabilidades de los modelos de lenguaje. Por otra parte se percibe que los modelos que utilizan sistemas de bolsa de palabras, bien sean pseudocognados, o bien sean *stems*, dan buenos resultados, incluso mejor que los n-gramas de caracteres. De hecho, estos resultados desbaratan la hipótesis formulada anteriormente de que el valor F_1 está directamente relacionado con la calidad del corpus y, por consiguiente, la del traductor. Por ejemplo, el modelo *c3g* que fue seleccionado con $F_1 = 0{,}36$ ha conseguido una puntuación mejora media respecto al traductor *baseline* de 11,83 puntos, mientras que el mode-

lo *cog*, con $F_1 = 0{,}24$ ha alcanzado una diferencia media respecto a Europal de 24,15, es decir, 2,04 veces mayor incremento que *c3g*. Finalmente, cabe destacar que el modelo de factores de longitud sí es un buen corrector de las medidas de similitud, ya que el modelo *mean_len* ha conseguido una mejora media de 27,02 puntos sobre Europarl, obteniendo un segundo puesto en la comparación de los modelos y el primero si no se tiene en cuenta el corpus *union*, que ha sido generado posteriormente a partir de los anteriores.

Capítulo 6

Planificación y costes

6.1. Planificación

Aunque este proyecto comenzó a realizarse en marzo de 2013, el trabajo se ha hecho de forma intermitente debido a que durante una buena parte del mismo se ha tenido que compaginar con otra actividad laboral. Por ello, en lugar de presentar la planificación durante todo este año se muestra un diagrama de Gantt idealizado al eliminar los periodos de tiempo en los que no se ha dedicado tiempo al proyecto. En la tabla 6.1 se muestra el listado de tareas en el que se ha descompuesto el trabajo a realizar, agrupándolas en grupos según el objetivo que permitían alcanzar; junto con las horas planificadas para acabar cada una de ellas y la acumulación de tiempo en cada punto del proyecto.

La figura 6.1 muestra el diagrama de Gantt correspondiente a la planificación expuesta teniendo en cuenta jornadas de 8 horas y semanas laborables de 40 horas. También se puede apreciar que todas las tareas son secuenciales excepto la anotación manual de artículos (T3.3) que se realiza mientras se están computando las similitudes por lo que cualquier retraso en una de las tareas supone un retraso en el resto de la planificación. De hecho se han producido ciertos retrasos que en su conjunto han supuesto un incremento del tiempo de proyecto en 168 horas. En la figura 6.2 se aprecia en rojo las tareas retrasadas, que han sido: "Selección de categorías" (48 horas), "Extracción de texto" (40 horas) y "Creación de los conjuntos de desarrollo, test y entrenamiento" (80 horas), que supone un incremento de un 24 % respecto al tiempo inicial. El motivo de estos retrasos ha sido que la estrategia decidida originalmente para realizar la tarea no ha funcionado correctamente por lo que se ha tenido que dedicar tiempo a decidir y desarrollar alternativas. La discusión de los métodos que han fallado y la opción utiliza como sustituto se ha expuesto en las secciones. Por otra parte hay que destacar que el cluster de computación de

Tabla 6.1: Estimación temporal de las tareas del proyecto.

ID	Descripción	Horas/Tarea	Total
T1	Preparación del entorno		
T1.1	Entorno de desarrollo	16	16
T1.2	Creación e importación de la base de datos	24	40
T2	Obtención del corpus comparable		
T2.1	Definición de dominios y estrategia	8	48
T2.2	Selección de categorías	80	128
T2.3	Selección de artículos	40	168
T3	Extracción del corpus paralelo		
T3.1	Definición de estrategias	8	176
T3.2	Extracción de texto	64	240
T3.3	Anotación manual de artículos	56	296
T3.4	Estimación de similitudes	72	368
T3.5	Extracción de pares de oraciones	40	408
T3.6	Tuning	88	496
T4	Experimentos		
T4.1	Selección de corpus a usar		
T4.2	Creación de los conjuntos de desarrollo, test y entrenamiento	56	560
T4.3	Entrenamiento de traductores		
T4.4	Evaluación de traductores	32	632
T5	Análisis y conclusiones	56	688

RDLab estuvo fuera de servicio durante aproximadamente 10 días y luego tardó otra semana en volver a funcionar a un ritmo normal. Esta incidencia ha provocado otra semana de retraso que ya no se ha contemplado en el Gantt dado que en este se ha querido exponer únicamente los retrasos provocados por las decisiones tomadas durante el proyecto.

6.2. Estudio económico

Para realizar dicho estudio se han considerado los medios humanos, materiales y tecnológicos utilizados para el desarrollo del proyecto. Para estimar el coste de los recursos humanos, se ha considerado un único perfil de trabajador, el investigador, y se ha considerado que tiene un coste de 16,50 €/hora[1]. Respecto a los materiales, se ha utilizado un portátil con Ubuntu[2], una pantalla adiccional para el portátil y el cluster que gestiona el Laboratorio de Investigación y

[1]Estimación de coste realizada con el salario publicado en Glassdoor para un investigador en la UPC (Glassdoor, 2014)

[2]La versión utilizada ha sido Ubuntu 12.04 LTS

Tabla 6.2: Coste del proyecto

Concepto	**Coste**
Mano de obra	14 124,00 €
Cluster	2386,40 €
Portátil	600,00 €
Pantalla	180,00 €
Total	17 290,40 €

Desarrollo (RDLab). La tabla 6.2 muestra la relación de costes dividida por conceptos junto al montante total. Cabe destacar que el coste del cluster incluye el coste de todos los trabajos que se han enviado sin tener en cuenta si han terminado correctamente o han fallado durante su ejecución, hecho que ha ocurrido en un 22 % de las ocasiones. Por lo tanto, en caso de que todas estas tareas hubieran funcionado bien, el coste del cluster se habría reducido en 525 € y el total del proyecto hubiera sido de 16 795,40 €.

Por otra parte, la figura 6.3 muestra la factura emitida por RDLab con el coste desglosado en colas de nodos utilizados. Para calcular este coste RDLab divide el uso de los nodos en *packs* de 1 hora y 2GB de RAM que tienen un precio de 0,40 € cada uno.

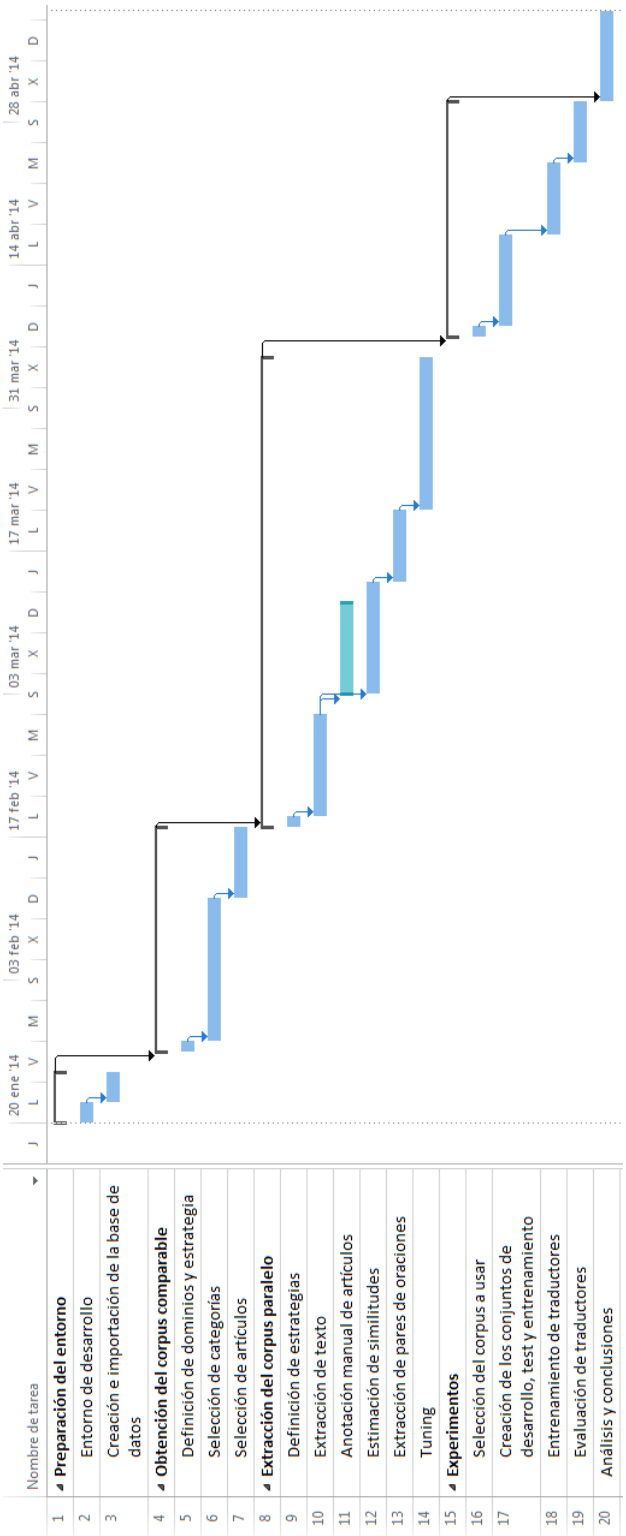


Figura 6.1: Diagrama de Gantt con la planificación inicial del proyecto.Duración estimada: 16 semanas.

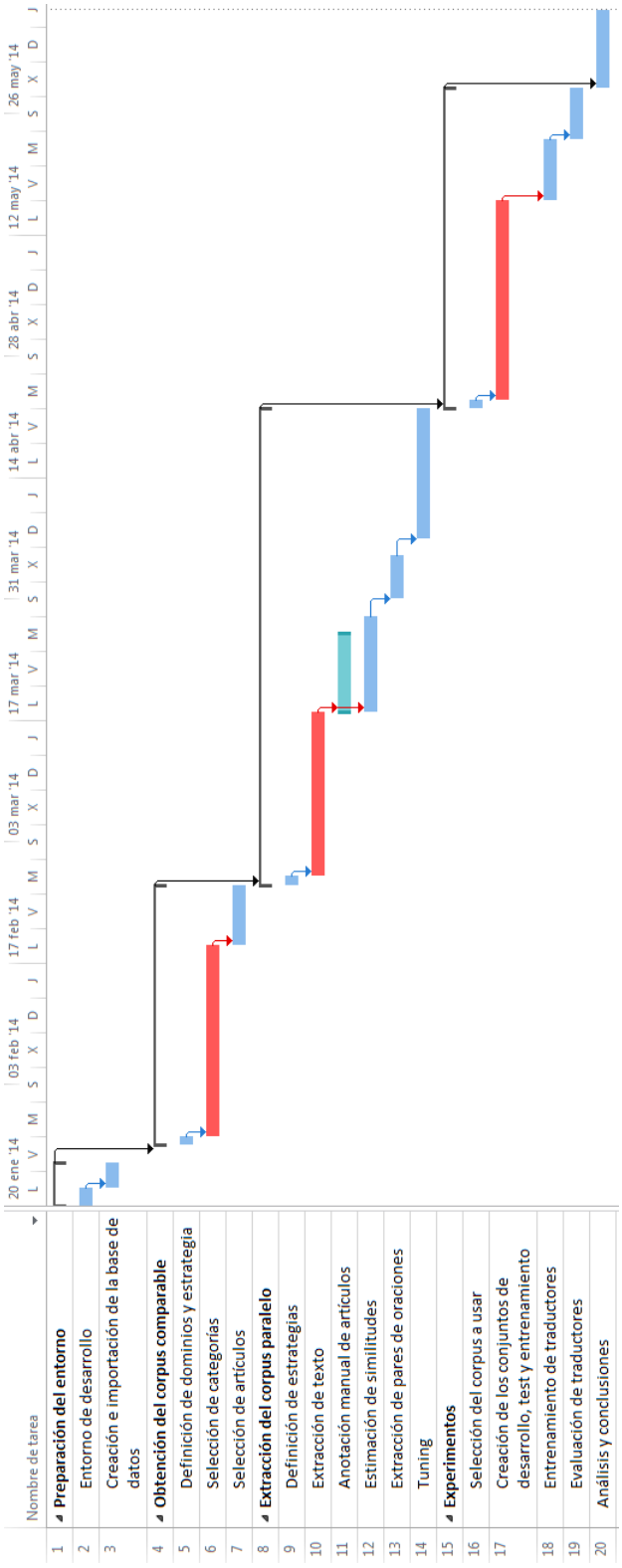


Figura 6.2: Diagrama de Gantt con el desarrollo real del proyecto, que ha supuesto 4 semanas más de trabajo.

Initial Date 2014-01-01
Final Date 2014-12-31

Queue: long — Pack info: RAM 2 GB + CPU Time 1 h

User	#jobs	Total RAM (Gb)	Total time (h)	#packs	Pack price (€)	Total
jboldoba	35	184.39	258.69	259	0.4	103.6 €

Queue: medium — Pack info: RAM 2 GB + CPU Time 1 h

User	#jobs	Total RAM (Gb)	Total time (h)	#packs	Pack price (€)	Total
jboldoba	320	1168.14	1754.85	1755	0.4	702 €

Queue: short — Pack info: RAM 2 GB + CPU Time 1 h

User	#jobs	Total RAM (Gb)	Total time (h)	#packs	Pack price (€)	Total
jboldoba	1292	4781.25	3951.8	3952	0.4	1,580.8 €

Queue: test — Pack info: RAM 2 GB + CPU Time 1 h

User	#jobs	Total RAM (Gb)	Total time (h)	#packs	Pack price (€)	Total
jboldoba	0	0	0	0	0.4	0.00 €

Subtotal: 2,386.4 €

Figura 6.3: Factura emitida por RDLab con el coste del procesamiento en el cluster.

Capítulo 7

Conclusiones

A lo largo de esta memoria se han descrito los pasos y decisiones seguidos para obtener traductores automáticos a partir de textos extraídos de Wikipedia. Además, el desarrollo del mismo ha permitido conocer mejor algunas de las características de Wikipedia y cómo afectan a la recuperación de información, independientemente de para qué fin sea esta minería de datos. En este capítulo se exponen las conclusiones del trabajo realizado a fin de alcanzar cada uno de los objetivos generales del proyecto (cf. sección 1.3), los cuales han sido logrados en su totalidad.

El objetivo **Obj_1** que implicaba la obtención de un corpus comparable a partir de Wikipedia ha sido posiblemente la tarea que mejores resultados ha dado ya que en todos los dominios se ha conseguido una cantidad considerable de pares de artículos. Por otra parte, hay que destacar que el mayor problema encontrado en esta fase ha sido la dificultad para poder diferenciar qué categorías debían ser incluidas en cada dominio. Este problema se deriva de la sobrecategorización de los artículos de Wikipedia y se ve magnificado por la categorización circular que hace que ciertas categorías contengan a otra entre sus subcategorías y sus supercategorías al mismo tiempo y se ha solventado mediante el uso de vocabularios de dominio creados para este fin.

El objetivo **Obj_2**, Obtener unn corpus paralelo a partir del corpus comparable, ha supuesto ciertas complicaciones que se han superado mediante la redefinición de estrategias de procesamiento de texto. Uno de los puntos más costosos ha sido ser capaces de eliminar correctamente el texto que no se deseaba de los artículos, ya que *a priori* no se había definido qué partes interesaban y cuáles no debido al desconocimiento de la estructura de los artículos en la enciclopedia. Tal y como se ha expuesto en la sección 4.1 se extrajeron únicamente las oraciones de los párrafos y listas de todas las secciones junto a los títulos de las mismas, sin embargo, una vez finalizados los experimentos se ha llegado a la conclusión de que los requisitos definidos para escoger una

Tabla 7.1: Tamaño de los conjuntos de entrenamiento de cada traductor añadiendo Europal v7

	cs	**sc**	**sp**	**wk**
c3g	2 061 449	2 689 494	2 300 562	2 849 100
cog	2 148 017	3 179 699	2 417 058	3 396 696
mono_en	2 176 398	3 332 903	2 426 971	3 604 511
mean_len	2 086 569	2 922 080	2 355 709	3 126 711
union	2 543 162	5 813 115	3 147 398	6 913 975

oración han sido un tanto deficientes, provocando un postprocesamiento dificultoso del corpus para poder crear los conjuntos de entrenamiento, desarrollo y test que se iban a utilizar para entrenar.

El objetivo **Obj_3**, enriquecer un sistema de traducción automática estadística, ha supuesto un reto debido a que los corpus conseguidos en el trabajo precedente eran muy densos y necesitaban un postprocesamiento. La idea inicial para este objetivo era utilizar el corpus extraído en los puntos anteriores al corpus del traductor *baseline* (Europarl v7) de forma que realmente se enriqueciera un traductor y no se creara uno sin tener en cuenta el corpus base. En la tabla 7.1 se observa el tamaño de los conjuntos de entrenamiento resultantes de añadir el corpus de Europarl (1 965 734 oraciones) a los conjuntos obtenidos de Wikipedia. Debido a esta cantidad de texto y el tiempo que se disponía para realizar los entrenamientos se ha optado por utilizar únicamente el corpus de WIkipedia. Aún con esta modificación leve del objetivo, se considera ampliamente alcanzado ya que la puntuación BLEU de los mejores traductores entrenados en este trabajo ha superado a la obtenida por el traductor de referencia en una media de 33 puntos (cf. tabla 5.5).

En el terreno personal este proyecto ha supuesto la adquisición de nuevos conceptos sobre dos áreas apenas conocidas por el autor: la recuperación de información y la traducción automática. Si bien había asistido a asignaturas relacionadas con el procesamiento del lenguaje natural y el aprendizaje automático, nunca había llegado a profundizar en estos dos campos. Por otra parte, el desarrollo de este proyecto ha implicado conocer una salida profesional, la investigación, quizás no tan tenida en cuenta durante la carrera, que la misma está más orientada a las salidas dedicadas al desarrollo de sistemas informáticos, desde el diseño a la implementación y a la administración de sistemas.

7.1. Trabajo futuro

Con el trabajo descrito en esta memoria es posible obtener los artículos de Wikipedia que pertenecen a un campo determinado, oraciones paralelas de artículos comparables de la enciclopedia y entrenar traductores de inglés a castellano. A partir de este punto surgen nuevas posibilidades para este proyecto que se presentan a continuación.

- En relación con los procedimientos de extracción del texto plano de los artículos de Wikipedia, se propone buscar alternativas que alivien la carga de trabajo del postprocesado del corpus. Principalmente se recomienda utilizar la biblioteca language-detection (Shuyo, 2010) para obtener únicamente las oraciones que se encuentren en el idioma seleccionado.

- A la vista de los buenos resultados conseguidos con las extracciones realizadas con las similitudes calculadas al combinar los modelos de longitud, especialmente el modelo *len*, se considera oportuno investigar en esta dirección, probando diferentes combinaciones e incluso nuevos modelos de similitud que puedan surgir en un futuro.

- Si bien los resultados obtenidos entrenando únicamente con los corpus extraídos de Wikipedia ya superan las expectativas que se tenía, es necesario terminar la investigación inicial utilizando corpora formada por la unión de los conjuntos extraídos y Europarl.

- En este trabajo se han sentado las bases para construir sistemas de traducción especializados en áreas de conocimiento específicas a partir de Wikipedia. Sin embargo sólo se ha utilizado el par en-es para ello. Debido a que TACARDI amplía el conjunto de lenguas contempladas con el catalán y el euskera, este es un buen punto de partida para trasladarlo a los pares formados por el resto de idiomas.

- En la misma dirección que el trabajo anterior se propone aplicar estos procedimientos a lenguas cuya distancia lingüística[1] sea mayor que la de los idiomas utilizados, por ejemplo, inglés y árabe.

- Por último s2e propone investigar el comportamiento de estos traductores en un entorno real con el fin de poder enriquecer las diferentes Wikipedias a partir de los artículos homólogos en otros idiomas. También se puede pensar en utilizar estos traductores para obtener información en otros medios como puede ser la divulgación científica.

[1] La distancia lingüística es un término usado para referirse a las diferencias lingüísticas entre dos lenguas (Centro Virtual Cervantes, 2014)

Bibliografía

Barrón-Cedeño, A., Màrquez, L., Henríquez Q., C., Formiga, L., Romero, E., y May, J. (2013). Identifying Useful Human Correction Feedback from an On-Line Machine Translation Service. En *IJCAI*.

Bing (2014). Acerca del Traductor de Bing. [http://www.bing.com/translator/help/]. Accedido 22/05/2014.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., y Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. En *Proceedings of the Eighth Workshop on Statistical Machine Translation*, páginas 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., y Tapias, D., editores (2008). *Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.*, **22**(2), 249–254.

Centro Virtual Cervantes (2014). Distancia lingüística. [http://cvc.cervantes.es/ensenanza/biblioteca_ele/diccio_ele/diccionario/distan cialinguistica.htm]. Accedido 20/06/2014.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**(1), 37–46.

Cui, G., Lu, Q., Li, W., y Chen, Y. (2008). Corpus Exploitation from Wikipedia for Ontology Construction. En Calzolari *et al.* (2008), páginas 2126–2128.

España-Bonet, C. (2008). A proposal for an arabic-to-english SMT (Master Thesis). Universitat de Barcelona and Universitat Politècnica de Catalunya.

Glassdoor (2014). Universitat politecnica de catalunya salaries in spain. [http://www.glassdoor.com/Salary/Universitat-Politecnica-de-Catalunya-Salaries-E2 53780.htm]. Accedido 19/06/2014.

Gonzàlez, M., no, A. B.-C., y Màrquez, L. (2014). IPA and STOUT: Leveraging Linguistic and Source-based
Features for Machine Translation Evaluation. En *Ninth Workshop on Statistical Machine Translation (WMT2014). Metrics Task.* , Baltimore, USA.

Google (2014). Inside Google Translate. [http://translate.google.com/about/intl/en_ALL/]. Accedido 22/05/2014.

Hutchins, J. (1997). From First Conception to First Demonstration: the Nascent Years of Machine Translation, 1947-1954. *Machine Translationi*, **12**(3), 195–252.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. En *Proceedings of EMNLP 2004*, Barcelona, Spain.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., y Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. En *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, Prague, Czech Republic.

Manning, C., Raghavan, P., y Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Manning, C. D. y Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

McNamee, P. y Mayfield, J. (2004). Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval*, **7**(1-2), 73–97.

Oard, D. W. y Hackett, P. G. (1997). Document translation for cross-language text retrieval at the university of maryland. En *TREC*, páginas 687–696.

Och, F. (2003). Minimum error rate training in statistical machine translation. En *Proc. of the ACL Conference*.

Och, F. y Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*.

Ortega Soto, J. (2009). *Wikipedia: A Quantitative Analysis*. PhD thesis, Universidad Rey Juan Carlos, Madrid, Spain.

Porter, M. (1980). An Algorithm for Suffix Stripping. *Program*, **14**, 130–137.

Pouliquen, B., Steinberger, R., y Ignat, C. (2003). Automatic Identification of Document Translations in Large Multilingual Document Collections. En *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-2003)*, páginas 401–408, Borovets, Bulgaria.

Shuyo, N. (2010). Language detection library for java. [http://www.slideshare.net/shuyo/language-detection-library-for-java]. Accedido 19/06/2014.

Simard, M., Foster, G. F., y Isabelle, P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. En *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*.

Stolcke, A. (2002). SRILM – An extensible language modeling toolkit. En *Proc. Intl. Conf. on Spoken Language Processing*.

The American Heritage dictionary of the English Language (2011). Houghton Mifflin Harcourt.

Uszkoreit, J., Ponte, J., Popat, A., y Dubiner, M. (2010). Large scale parallel document mining for machine translation. En C.-R. Huang y D. Jurafsky, editores, *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, páginas 1101–1109, Beijing, China. COLING 2010 Organizing Committee.

Wikipedia (2014a). Wikipedia: Categorización. [http://es.wikipedia.org/wiki/Wikipedia:Categorización]. Accedido 6/mar/2014.

Wikipedia (2014b). Wikipedia: Language editions. [http://en.wikipedia.org/wiki/Wikipedia#Language_edition Accedido 17/06/2014.

Wikipedia (2014c). Wikipedia: Wiktexto. [http://es.wikipedia.org/wiki/Wikitexto]. Accedido 12/06/2014.

Zesch, T., Müller, C., y Gurevych, I. (2008). Extracting Lexical Semantic Knowledge from Wikipedia and Wikictionary. En Calzolari *et al.* (2008).

Apéndice A

Artículos anotados manualmente

En las siguientes tablas se incluyen los pares de artículos anotados manualmente (cf. sección 4.3.2) separados según el dominio al que pertenecen. Cada tabla muestra el identificador dado al par de artículos que es el mismo que tiene el artículo en la Wikipedia en castellano y los títulos de las entradas en cada una de las Wikipedias.

Tabla A.1: Relación de los artículos del dominio Informática

ID	Artículo en inglés	Artículo en español
1016238	Knowledge base	Base de conocimiento
1442469	Rail Fence Cipher	Cifrado Rail Fence
152918	GNU C Library	Glibc
297342	OpenGL Utility Library	OpenGL Utility
5250166	Chaocipher	Chaocipher
2439963	Telsecundaria	Telesecundaria
2090638	Smart label	Etiqueta RFID
1695687	Webometrics	Cibermetría
5124707	Pixmania	Pixmania
948901	BLAG Linux and GNU	BLAG Linux and GNU

Tabla A.2: Relación de los artículos del dominio Ciencia.

ID	Artículo en inglés	Artículo en español
3175837	16th meridian west	Meridiano 16 oeste
121982	Larry Wall	Larry Wall
3617429	Marvel Super Heroes: War of the Gems	Marvel Super Heroes: War of the Gems
741312	Egyptan geometry	Geomatría en el Antiguo Egipto
5344091	Blotting paper	Papel secante
1475656	Why I Am Not a Christian	Por qué no soy cristiano
5672222	Swype	Swype
508140	Beta Tauri	Elnath
2436477	Mineral Water	Agua mineral
3329373	Homo sovieticus	Homo sovieticus

Tabla A.3: Relación de los artículos del dominio Deportes.

ID	Artículo en inglés	Artículo en español
5585748	Raúl Orosco	Raúl Orosco
5777133	1993 Men's European Volleyball Chamionship	Campeonato Eurpeo de Voleibol Masculino de 1993
3193649	International Orienteering Federation	Federación Internacional de Orientación
145321	FIFA U-20 World Cup	Copa Mundial de Fútbol Sub-20
1077240	Young rider classification in the Tour de France	Maillot blanco
2829220	Smash (tennis)	Smash (tenis)
4358215	Ron Carter (basketball)	Ron Carter (baloncestista)
1091160	Markel Susaeta	Markel Susaeta
2720529	Dmytro Chygrynskiy	Dmitro Chigrinskiy
941121	Qatar Open (tennis)	Torneo de Doha