Traducción Automática: Trabajo con Europarl

Ramon Ruiz Dolz

December 2018

1 Introducción

En esta memoria se presentan todos los experimentos realizados a modo de trabajo final de la asignatura. Para ello se ha hecho uso del corpus Europarl[1] español-ingles. Este corpus [2] esta formado por muestras de texto paralelo en 11 idiomas extraídos del parlamento europeo. En este trabajo, como se ha mencionado, se ha hecho uso de la dupla español e inglés. Concretamente el propósito de este trabajo es el de construir el mejor traductor de inglés a español posible aplicando las distintas técnicas aprendidas a lo largo del curso.

La idea inicial era la de realizar este trabajo en las máquinas del escritorio virtual (EVIR) del DSIC. Sin embargo, la limitación, tanto de hardware como de la conexión ha hecho realmente incómoda esta tarea debido a su elevada complejidad en relación al material disponible. Además, los modelos entrenados en EVIR tampoco destacaban por su gran calidad. Es por esto que se ha decidido realizar la instalación tanto de SRILM[2] [6] como de MOSES[3] en dos máquinas distintas con tal de poder realizar mayor número de experimentos de mayor complejidad en menor tiempo. Concretamente se han lanzado los experimentos en linux Mint sobre un procesador i7 7700 y un i7 9700k.

2 Ejercicio básico: Proceso experimental

Siguiendo el guión de las prácticas, en esta sección se explicará el proceso experimental realizado, desde la definición de las variables y la preparación de los datos hasta la evaluación de las traducciones realizadas. Se han realizado tres experimentos, un experimento inicial en EVIR y posteriormente otros dos experimentos en las máquinas descritas anteriormente. El experimento lanzado en EVIR, debido a las limitaciones del sistema se ha reducido en replicar la práctica de traducción estadística pero con este corpus de mayor tamaño. Sin embargo, en los otros dos experimentos sí se han podido ajustar y modificar

[1] http://www.statmt.org/europarl/
[2] http://www.speech.sri.com/projects/srilm/
[3] http://www.statmt.org/moses/

parámetros de gran relevancia. Como pueden ser los n-gramas del modelo del lenguaje o el número de iteraciones del ajuste de los pesos.

A continuación se define todo el proceso experimental relacionado con el experimento básico propuesto. En la Figure 1 se puede observar un diagrama del proceso que se ha definido en las siguientes secciones.

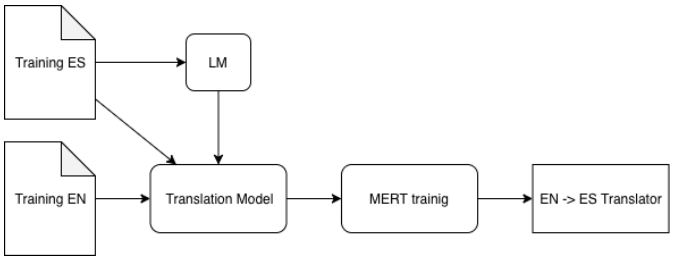

Figure 1: Diagrama del proceso experimental para obtener un traductor estadístico con Moses

2.1 Instalación

Previo al inicio con los experimentos, se ha realizado la instalación de todo el software necesario para la experimentación en dos máquinas distintas. La instalación tanto de MOSES como de SRILM en linux ha consistido en descargar ambos programas y seguir las instrucciones de instalación.

2.2 Definición de las variables de entorno

Ya con todo el software instalado, el primer paso consiste en definir las variables de entorno para poder ejecutar los comandos necesarios para la realización del trabajo. Para ello se han creado un scripts llamado *export.sh* con el siguiente contenido,

```
export PATH=$PATH:/home/raruidol/moses-mt/moses/bin
export PATH=$PATH:/home/raruidol/moses-mt/moses/scripts/training
export PATH=$PATH:/home/raruidol/srilm/bin/i686-m64
export SCRIPTS_ROOTDIR=/home/raruidol/moses-mt/moses/scripts
export GIZA=/home/raruidol/moses-mt/training-tools
export MOSES=/home/raruidol/moses-mt/moses
```

Figure 2: Contenido del script export.sh

De esta forma lanzando únicamente este script se deja totalmente listo el entorno para poder empezar a lanzar experimentos.

2.3 Preparación de los datos

Una vez el entorno definido se ha procedido con la preparación de los datos, tanto de test como de entrenamiento. Para ello se ha descargado el dataset Europarl en inglés y español. Este dataset viene sin ningún tipo de procesado ni alineamiento. Es por esto que en primer lugar se ha pasado la herramienta de tokenización proporcionada por MOSES a los corpus de ambos idiomas. Además del tokenizador también se ha hecho uso de la herramienta para limpiar corpus *clean-corpus-n.perl*. Es importante destacar en esta parte que se han realizado tres limpiezas distintas. Uno de los parámetros que recibe esta herramienta es el tamaño mínimo y máximo admitido. En las prácticas se definía el mínimo como 1 y el máximo como 60. Sin embargo en este trabajo se ha experimentado con distintos máximos como 60, 75, 80 y 100.

Una vez ambos corpus de entrenamiento tokenizados y limpios, estos han sido divididos en dos partes. Una parte mayor para el entrenamiento del modelo de traducción y otra parte de menor tamaño, llamada desarrollo, para el entrenamiento de los pesos del modelo. Concretamente se ha trabajado con conjuntos de desarrollo de 1500, 2500, 3300, 4800 y 6500 lineas.

2.4 Entrenamiento de los modelos de lenguaje

El primer paso de entrenamiento consiste en entrenar el modelo del lenguaje. Esto se realiza mediante SRILM. Concretamente para este trabajo se han construido modelos de trigramas y 5-gramas aplicando suavizado de Kneser-Ney [1]. Para el experimento lanzado en EVIR se ha entrenado un modelo de trigramas, puesto que construir un modelo de 5-gramas quedaba demasiado grande debido al aumento de complejidad ligado al modelo de lenguaje. Es por esto que, siguiendo los buenos resultados obtenidos en las prácticas, en los experimentos lanzados en ambas máquinas propias se ha hecho uso de modelos del lenguaje de 5-gramas.

2.5 Entrenamiento del modelo de traducción

Una vez obtenido el modelo de lenguaje a partir del conjunto de entrenamiento, el segundo paso consiste en entrenar el modelo de traducción. Se hace uso del software GIZA++[4] [3] para construir las tablas de segmentos y los modelos de reordenamiento que correspondan a los dos idiomas escogidos a partir de los conjuntos de entrenamiento de ambos idiomas.

El resultado de este proceso es un modelo de traducción, pero con los pesos sin ajustar. Los resultados obtenidos por este modelo son todavía muy pobres. Es por esto que en la siguiente sección se explica el proceso seguido para realizar el ajuste de estos pesos y mejorar así el comportamiento del modelo de traducción automática.

[4] http://www.statmt.org/moses/giza/GIZA++.html

2.6 Entrenamiento de los pesos del modelo log-lineal

Como ya se ha introducido, el paso final de la fase de entrenamiento del modelo de traducción estadístico consiste en el ajusto de los pesos del modelo. Para ello se ha hecho uso de MERT [4], una técnica de optimización de la calidad de la traducción. Para realizar el ajuste de los pesos se hace uso de los conjuntos de desarrollo separados del corpus de entrenamiento inicialmente. Este proceso es un poco más costoso puesto que consiste en traducir el conjunto de desarrollo cada vez por iteración.

El proceso seguido y las técnicas aplicadas para el entrenamiento se asemejan a las utilizadas en la práctica, es por esto que no se ha profundizado en las explicaciones respectivas.

2.7 Proceso de traducción

Finalmente, haciendo uso del modelo obtenido al finalizar el proceso de entrenamiento se ha utilizado este para traducir los conjuntos de test. Concretamente se disponen de dos conjuntos de test que han sido tokenizados y limpiados de la misma forma que los conjuntos de entrenamiento.

Mediante el decodificador de MOSES se ha realizado una traducción del conjunto de test en inglés al español. El archivo resultante se ha comparado con el conjunto de test en español, pudiendo así obtener la calidad de la traducción. En la siguiente sección se muestran los resultados obtenidos en cada experimento así como la configuración de cada uno de ellos.

2.8 Evaluación de los resultados

En esta sección final del ejercicio básico se presentan las distintas configuraciones estimadas para los distintos experimentos realizados con el corpus Europarl. La métrica empleada para la evaluación de los modelos obtenidos tras el entrenamiento es BLEU [5]. El propósito del uso de esta metrica es el de poder comparar con mayor facilidad los distintos modelos obtenidos y poder determinar cual de ellos alcanza una mayor calidad de las traducciones.

Como ya se ha mencionado, se han lanzado experimentos en tres entornos distintos. Un primer experimento en el entorno EVIR y los demás experimentos en entornos particulares denominados CASA (i7 9700k) y DSIC (i7 7700).

El primer experimento ha sido lanzado en EVIR. Trás varios intentos experimentando falta de espacio en el disco y cortes con la conexión remota, se lanzó un experimento ajustando los parámetros para que fuese factible su finalización. Los parámetros escogidos para el experimento lanzado en EVIR consisten en, un modelo de lenguaje de trigramas, limitar el número de iteraciones máximas de MERT a 5 y un conjunto de desarrollo de 3300 frases. El valor de BLEU obtenido en este experimento es de 25.71, justo lo esperado al aplicar un procedimiento similar al del boletín de prácticas.

Experimento	n-gramas	max_size clean	iter.	talla dev	BLEU
EVIR	3	60	5	3300	25.71

Table 1: Configuración y resultado experimento EVIR

Por otra parte en el entorno CASA se han lanzado varios experimentos. Concretamente se han realizado tres experimentos distintos tratando de ver el comportamiento del modelo entrenado al reducir el número del tamaño de desarrollo y dejando un mayor tamaño para entrenamiento. También se ha probado a aumentar la talla máxima en el proceso de limpieza de corpus.

Experimento	n-gramas	max_size clean	iter.	talla dev	BLEU
CASA	5	80	15(9)	4800	27.38
CASA2	5	80	15(9)	2500	27.63
CASA3	4	100	20(11)	2000	27.72

Table 2: Configuración y resultado experimento CASA

En este caso hemos podido observar que, al reducir el tamaño del conjunto de desarrollo, el modelo toma mayor número de iteraciones para ajustar los pesos sin dispararse este número, y además dispone de un mayor tamaño de entrenamiento, por lo cual obtiene mejor puntuación.

Finalmente, en el entorno DSIC se han lanzado dos experimentos. Un primer experimento tratando de observar el comportamiento del traductor ajustado con un conjunto de desarrollo de mayor tamaño, y por otra parte, habiendo observado los resultados de CASA, se ha lanzado otro experimento similar al de mejor resultado obtenido previamente pero reduciendo la talla de desarrollo ligeramente más.

Experimento	n-gramas	max_size clean	iter.	talla dev	BLEU
DSIC	5	75	15(7)	6500	25.99
DSIC2	4	100	20(9)	1500	28.25

Table 3: Configuración y resultado experimento DSIC

Mediante esta experimentación se ha podido comprobar como, al tomar demasiadas frases para desarrollo, el conjunto de entrenamiento queda demasiado pequeño y los resultados obtenidos empeoran, además de consumir un mayor tiempo el proceso de ajuste de pesos. Por otra parte, reduciendo aún más, a 1500, el número de frases para ajustar los pesos, se ha podido obtener el mejor traductor posible, alcanzando un 28.25 de BLEU. Con este mismo traductor se ha alcanzado una puntuación de 29.21 en la competición realizada en la asignatura, siendo este el segundo mejor traductor estadístico desarrollado.

3 Ejercicios avanzados

En esta sección se presentan los experimentos realizados dentro del marco de ejercicios avanzados. Concretamente se ha realizado un experimento entrenando un post-editor con Moses, otro experimento con el toolkit Thot[5] y finalmente un último experimento haciendo uso de redes neuronales para construir un traductor automático. Para este último experimento se ha hecho uso del toolkit nmt-keras[6] que combina las librerías Theano y Tensorflow.

3.1 Post-edición automática

El primer experimento realizado consiste en el entrenamiento de un traductor automático de post-edición del texto. Concretamente esta técnica consiste en traducir la traducción obtenida con el traductor básico. Para ello, se dispone de dos nuevos corpus de entrenamiento y de test para el post-editado. En nuestro caso particular que se busca obtener un traductor de inglés a español, los corpus en ingles se traducen con el primer traductor básico, y las traducciones obtenidas son usadas como entrada para este nuevo modelo. El proceso a seguir para obtener este traductor con Moses es exactamente igual que en el ejercicio básico. Partiendo del traductor expuesto en Figure 1, en Figure 3 se puede observar la continuación de este proceso mediante el cual se obtiene el traductor de post-edición automática.

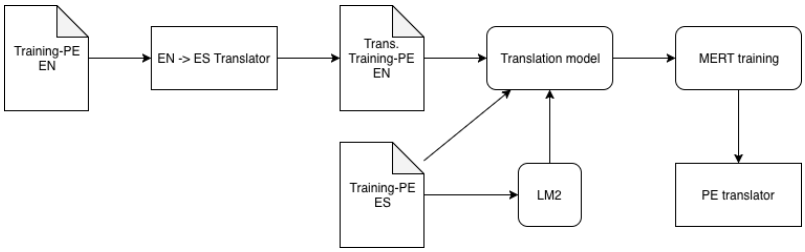

Figure 3: Diagrama del proceso experimental para obtener un traductor de post edición en Moses

Este traductor de post-edición se ha entrenado con los siguientes parámetros,

Experimento	**n-gramas**	**max_size clean**	**iter.**	**talla dev**	**BLEU**
PostEdit	5	100	20(9)	1000	28.5

Table 4: Configuración y resultado experimento EVIR

Como se puede apreciar, el resultado obtenido tras su evaluación mejora ligeramente el BLEU del traductor sin post edición. Los parámetros escogidos

[5] http://daormar.github.io/thot/
[6] https://nmt-keras.readthedocs.io/en/latest/

para la realización de este experimento se han basado en los resultados obtenidos en el proceso de obtención del mejor traductor estadístico con Moses.

3.2 Uso del toolkit Thot

El segundo experimento realizado ha consistido en reproducir un experimento similar al que ha permitido conseguir mayor puntuación con Moses en el ejercicio básico. De esta forma es posible realizar una comparación *justa* entre ambos toolkits.

El proceso experimental seguido con la herramienta Thot es realmente similar al seguido con Moses. En primer lugar se ha realizado un procesado del corpus. Concretamente el corpus se ha tokenizado, posteriormente se ha pasado todo a minúscula y finalmente se ha realizado la limpieza de las frases demasiado largas, de forma similar al experimento básico descrito en este trabajo.

Una vez el corpus ya listo para ser utilizado se ha empezado con el proceso de entrenamiento de los distintos modelos y el ajuste de parámetros. En primer lugar se ha entrenado el modelo de lenguaje del idioma de salida, es decir español. Para ello, el toolkit Thot nos proporciona una serie de herramientas con las cuales a partir del corpus limpio en español se nos permite entrenar un modelo de lenguaje en español. El siguiente paso ha consistido en entrenar un modelo de traducción de inglés a español. Para ello Thot también nos proporciona una serie de herramientas que nos permiten obtener el modelo estadístico. Una vez el modelo de traducción obtenido es necesario realizar el ajuste de los parámetros del modelo. Thot incorpora el método Nelder-Mead de optimización para la realización de este ajuste de los parámetros. Finalmente, antes de proceder a evaluar nuestro modelo es necesario filtrar el modelo de frases. Este paso es importante puesto que el modelo de frases puede contener gran cantidad de información mucha de ella de poco interés. Es por esto que, dado que conocemos el texto sobre el que vamos a aplicar nuestro modelo de traducción, es posible filtrar únicamente los parámetros que sean relevantes para llevar a caso el proceso de traducción.

Ya con el modelo entrenado, ajustado y filtrado, podemos llevar a cabo la traducción. Para ello, Thot también proporciona una serie de herramientas para traducir texto haciendo uso de un modelo previamente entrenado. Finalmente, al comparar la traducción obtenida con la traducción real se han obtenido los siguientes valores de BLEU.

Experimento	**n-gramas**	**max_size clean**	**talla dev**	**BLEU**
Thot1	4	100	1500	24.6

Table 5: Resultados del experimento realizado con Thot

3.3 Traducción automática con Keras

Finalmente, el último experimento realizado dentro del marco de este trabajo ha consistido en el entrenamiento de modelos basados en redes neuronales para

traducción automática. Para ello se ha hecho uso del toolkit nmt-keras como ya se ha comentado anteriormente. Este toolkit permite trabajar con redes neuronales con gran facilidad, puesto que dispone de un fichero de configuración desde el que podemos ajustar gran cantidad de parámetros de la red.

A diferencia de los experimentos realizados en la sesión de prácticas, para la realización de esta parte del trabajo se ha hecho uso del fichero de configuración que viene con la librería como base. Este fichero hace uso de los recursos de la GPU. Es por esto que además de toda la configuración e instalación realizada, se ha configurado la máquina para poder trabajar con la tarjeta gráfica. El hecho de hacer uso de la GPU para el entrenamiento de la red ha permitido acelerar este proceso.

Siguiendo los resultados obtenidos en la práctica como orientación se han lanzado dos experimentos basados en redes neuronales. Ambos experimentos se han configurado con Adam como algoritmo de optimización, 0.001 de factor de aprendizaje y se ha activado la reducción del learning rate en función de los resultados obtenidos en las últimas epochs. Además se han activado tanto capas con ruido gausiano como batch normalization. Lo que se ha variado ha sido el tamaño de los word embeddings y el tamaño de los encoder y decoders. Los resultados obtenidos se pueden observar a continuación.

Experimento	**#Word embeddings**	**#Encoder-decoder**	**BLEU**
nmt-1	64	64	24.3
nmt-2	128	128	29.7

Table 6: Resultados de la experimentación con redes neuronales

4 Conclusiones

La realización de este trabajo la considero realmente útil e interesante por distintos factores. En primer lugar, al hacer uso de un corpus de mayor tamaño al empleado en las sesiones de prácticas se ha podido comprobar el elevado coste temporal del proceso de traducción automática. Es por esto que es de gran importancia decidir cómo configurar cada experimento y no lanzarlos aleatoriamente. Además, tras la decisión de no hacer uso del EVIR, he tenido que instalar todos los distintos toolkits y herramientas, experiencia bastante nutritiva desde un punto de vista práctico. Finalmente me gustaría remarcar como experiencia negativa, el propio EVIR. Pienso que para los requisitos de los experimentos realizados en este trabajo, el escritorio virtual que se nos proporciona queda demasiado corto en cuanto a especificaciones técnicas. No es que funcione muy lentamente, si no que a demás las sesiones son interrumpidas tras el trascurso de un determinado tiempo de inactividad con el peligro de perder todo el trabajo realizado.

References

[1] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *icassp*, volume 1, page 181e4, 1995.

[2] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[3] Franz Josef Och. Giza++: Training of statistical translation models. *http://www. isi. edu/˜ och/GIZA++. html*, 2001.

[4] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[5] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[6] Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*, 2002.