MIARFID Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

DSIC DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Statistical Machine Translation

Traductor automático del idioma inglés al español

Toolkit: Moses

Autor: Pascual Andrés Carrasco Gómez

Asignatura: Traducción automática (TA)

Índice

Introducción

En este trabajo se han realizado un conjunto de experimentaciones con el toolkit Moses con la finalidad de crear traductores automáticos basados en modelos estadísticos. El trabajo está compuesto por dos partes, la primera parte corresponde a crear un traductor (básico) del idioma inglés al español, la segunda parte consiste en realizar una serie de experimentaciones para intentar mejorar los resultados obtenidos en el primer apartado.
El corpus utilizado es un subconjunto del corpus europarl (europarl-v7.es-en) compuesto por un conjunto de entrenamiento (pares de frases inglés-español) de 50.000 frases y un conjunto de test (pares de frases inglés-español) de 1.000 frases. Se ha creado una estructura del Corpus con la finalidad de que los scripts que se han implementado sirvan para cualquier Corpus de entrada simplemente estructurando y renombrando los ficheros del corpus de la siguiente forma:

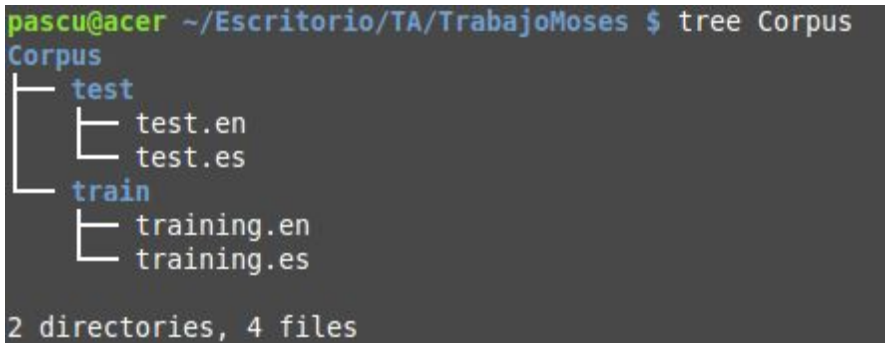

Imagen 1: Estructura del Corpus de entrada para el correcto funcionamiento de los scripts.

En la primera parte del trabajo se ha realizado un estudio de aspectos que se han visto en la práctica de laboratorio de la asignatura, para realizar este estudio se ha optado por reducir el corpus (europarl-v7.es-en) a un conjunto de entrenamiento de 9.500 frases y un conjunto de test de 500 frases. Esta reducción del tamaño del corpus se debe a que una ejecución con la totalidad del corpus tiene un coste temporal de aproximadamente 10 horas en la máquina donde se ha realizado el trabajo (portátil personal). Al finalizar el estudio se han escogido los mejores resultados obtenidos y se ha generado el traductor automático con el conjunto de entrenamiento completo (europarl-v7.es-en).
En la segunda parte del trabajo se ha empleado la totalidad del corpus (europarl-v7.es-en) ya que los resultados los hemos comparado con el resultado final del primer apartado.
Para todas las ejecuciones se ha optado por dividir el corpus de entrenamiento en dos partes, una parte de entrenamiento (85%) para entrenar el modelo en Moses y una parte de desarrollo (15%) para entrenar los pesos del modelo log-lineal mediante MERT (utilizando 5 iteraciones).
En el trabajo se adjuntan todos los scripts que se han utilizado organizados en directorios con la finalidad de que sean consultados si se desconoce algún parámetro en la ejecución del script para ese ejercicio concreto.

Ejercicio Básico

Para realizar esta primera parte se han limpiado, tokenizado y reducido los corpus de entrenamiento y test mediante la ejecución del script “formatear_corpus.sh”.
Los resultados (BLEU) se han obtenido promediados con tres ejecuciones ya que MERT realiza una inicialización aleatoria para obtener los pesos del modelo log-lineal.

1) Estudio de diferentes valores de Ngramas para el modelo de lenguaje de salida (Español)

En este apartado del trabajo se ha estudiado cómo afecta la configuración de Ngramas en la creación del modelo de lenguaje de salida que utiliza Moses para crear el traductor automático. Los scripts que se han utilizado para obtener los resultados de la tabla se encuentran en Basico/Ngramas/ y se comentan a continuación:

- script_train.sh: Crea el traductor automatico variando los valores de Ngramas en la generación del modelo de lenguaje.
- script_test.sh: Realiza la traducción y la evaluación del modelo (BLEU).

Nota: Para realizar esta experimentación se ha escogido una técnica de suavizado por interpolación con método de descuento modified Kneser-Ney (-kndiscount).

Ngramas	BLEU (1)	BLEU (2)	BLEU (3)	BLEU
2	19.87	19.87	20.17	19.97
3	20.46	20.46	20.24	20.39
4	20.27	20.49	20.24	20.33
5	20.20	20.28	20.37	20.28

Tabla 1: Media obtenida sobre la experimentación de la variación del orden de Ngramas.

Los resultados obtenidos no son muy significativos. Se observa que con 2gramas se obtiene una media más pequeña respecto a 3,4, y 5gramas. Para 3,4 y 5gramas prácticamente los resultados son los mismos, por lo tanto vamos a escoger 3gramas como parámetro para obtener el traductor (básico) con todas las muestras de entrenamiento.
Es interesante observar que reduciendo el corpus de entrenamiento (train) a 9.500 pares de frases el BLEU que se obtiene está sobre un 20.39, el BLEU indicativo (obtenido con todas las muestras) que marca el enunciado del trabajo está sobre 25.5, observamos que de 9.500 pares de frases a 50.000 se mejora aproximadamente un 5 de BLEU, esto nos indica que a mayor número de pares de frases de entrenamiento mejor resultado obtenemos (es lógico), pero se observa que la mejora no es muy grande.
En un apartado de ejercicios extra se va a ampliar el conjunto de entrenamiento (train) a 70.000 pares de frases con la finalidad de ver más detalladamente la mejora que hay al incrementar en 20.000 pares de frases el conjunto de entrenamiento.

2) Estudio de la utilización de otras técnicas de suavizado

Los Ngramas nos permiten modelar los modelos de lenguaje con buenos resultados gracias a las técnicas de suavizado que los acompañan. Para este trabajo se ha realizado un estudio de técnicas de suavizado por backoff e interpolación aplicando métodos de descuento: good-turing (default), modified Kneser-Ney (-kndiscount), unmodified Kneser-Ney (-ukndiscount), Witten-Bell (-wbdiscount).
Los scripts que se han utilizado para obtener los resultados de la tabla se encuentran en Basico/Suavizado/ y se comentan a continuación:

- script_train.sh: Crea el traductor automatico utilizando 3gramas y variando las técnicas de suavizado y sus correspondientes métodos de descuento en la generación del modelo de lenguaje.
- script_test.sh: Realiza la traducción y la evaluación del modelo (BLEU).

En la siguiente tabla se muestran los resultados obtenidos en la experimentación:

Backoff				
Descuento	BLEU (1)	BLEU (2)	BLEU (3)	BLEU
good-turing	20.00	20.03	20.18	20.07
kndiscount	20.33	20.22	20.15	20.23
ukndiscount	20.34	20.34	20.34	20.34
wbdiscount	20.37	20.07	20.37	20.27
Interpolate				
Descuento	BLEU (1)	BLEU (2)	BLEU (3)	BLEU
good-turing	20.43	20.28	20.42	20.38
kndiscount	20.46	20.46	20.24	20.39
ukndiscount	20.53	20.76	19.83	20.37
wbdiscount	19.73	20.06	20.21	20.00

Tabla 2: Media obtenida en la experimentación sobre la variación de las técnicas de suavizado.

Se observa que para la técnica de suavizado backoff el mejor resultado obtenido es mediante el método de descuento ukndiscount y el peor resultado es con el método de descuento good-turing. Para la técnica de suavizado interpolate el mejor resultado obtenido es mediante el método de descuento kndiscount y el peor resultado es mediante el método de descuento wbdiscount. La experimentación obtenida sobre la variación de las técnicas de suavizado no es muy significativa, se ha escogido el mejor resultado global que se ha obtenido en la experimentación que ha sido la técnica de suavizado interpolate con el

método de descuento kndiscount como parámetros para obtener el traductor automático (Básico) de este ejercicio.

3) Estudio del parámetro de número de iteraciones en el algoritmo MERT

Para estimar los pesos del modelo log-lineal se necesita un conjunto de desarrollo (dev) y el algoritmo MERT (Minimum Error Rate Training) o el algoritmo MIRA (Margin Infused Relaxed Algorithm), nosotros hemos utilizado el algoritmo MERT en este trabajo.
Como ya se ha comentado anteriormente el conjunto de desarrollo (dev) se ha obtenido separando el conjunto de muestras de entrenamiento (train) en dos partes, una parte para entrenamiento formada por el 85% del conjunto de train original (9.500 pares de frases) y el 15% restante para el conjunto de desarrollo.
El algoritmo MERT es un algoritmo iterativo y en este apartado del ejercicio básico hemos variado el número de iteraciones con valores 5,10 y 15.
Los scripts que se han utilizado para obtener los resultados de la tabla se encuentran en Basico/iteMert/ y se comentan a continuación:

- script_train.sh: Crea el traductor automatico utilizando los parámetros elegidos en los apartados anteriores modificando el párametro de iteraciones de MERT.
- script_test.sh: Realiza la traducción y la evaluación del modelo (BLEU).

En la siguiente tabla se muestran los resultados obtenidos en la experimentación:

Iteraciones	BLEU (1)	BLEU (2)	BLEU (3)	BLEU
5	20.46	20.46	20.24	20.39
10	20.27	20.38	20.46	20.37
15	20.14	20.46	20.38	20.33

Tabla 3: Media obtenida en la experimentación sobre la variación de iteraciones en el algoritmo MERT

Los resultados obtenidos no son significativos, prácticamente nos devuelve el mismo resultado aplicar 5 iteraciones a MERT que 15 iteraciones. Por el coste temporal al aumentar las iteraciones del algoritmo de MERT decidimos quedarnos con 5 iteraciones ya que como se puede observar el resultado es prácticamente el mismo.
Nota: El parámetro de iteraciones del algoritmo MERT junto al número de pares de frases correspondiente al conjunto de desarrollo (dev) son los que provocan que el coste temporal en las ejecuciones explote considerablemente, es importante ser conscientes de los valores de estos dos parámetros.

4) Traductor automático (Básico)

El traductor automático (Básico) del inglés al español es el traductor final que hemos obtenido en este ejercicio básico, para obtener este traductor se han utilizado todos los datos del corpus que nos proporciona el ejercicio, es decir, 50.000 pares de frases como conjunto de entrenamiento (85% train + 15% dev) y 1.000 pares de frases como conjunto de test. El modelo de lenguaje para este ejercicio se basa en las experimentaciones que

hemos realizado en los apartados anteriores, es decir, se utiliza la técnica de suavizado interpolate con método de descuento modified Kneser-Ney. Se han utilizado 5 iteraciones para el algoritmo MERT, la ejecución para obtener el traductor automático (Básico) ha tardado aproximadamente 10 horas en mi equipo (ordenador portátil), por este motivo se presenta un único resultado en este apartado.

Traductor	**BLEU**
Básico (final) del Inglés al Español	27.24

Tabla 4: Resultado (BLEU) obtenido con el Traductor (Básico) del Inglés al Español para el enunciado del ejercicio básico del trabajo.

Con este traductor se ha realizado la traducción de la tarea liberada el día 7 de enero (europarl-v7.es-en-test-hidd1617.tok.clean) que consiste en un nuevo fichero de test (tokenizado y límpio) cuyo BLEU obtenido se muestra en la siguiente tabla:

Traductor	**BLEU**
Básico (final) del Inglés al Español	28.49

Tabla 5: Resultado (BLEU) obtenido con el Traductor (Básico) del Inglés al Español para el conjunto de test europarl-v7.es-en-test-hidd1617.tok.clean.

Ejercicios Extra

El modelo de lenguaje utilizado para las experimentaciones realizadas en este apartado (extra) de la memoria se ha generado con los valores que hemos escogido en la experimentaciones del ejercicio básico (apartado anterior).
Se han realizado 5 iteraciones para entrenar los modelos log-lineales (mediante MERT).
En esta parte del trabajo se ha realizado una única ejecución por apartado ya que se trabaja con todos los datos y como hemos comentado anteriormente cada ejecución tiene una duración aproximadamente de 10 horas.
El corpus se ha limpiado y tokenizado con el script “limpiar_corpus.sh” antes de realizar los apartados del ejercicio extra.

1) Estudio sobre la utilización de pares de palabras o tripletes de palabras como unidades básicas en lugar de palabras para construir un traductor basado en Moses. Por ejemplo: "una casa verde y luminosa" podría ser "#_una una_casa casa_verde verde_y y_luminosa" en el caso de escoger pares de palabras.

Bigramas

Para obtener los resultados para bigramas se han utilizado los ficheros que se encuentran en “Extra/ejercicio1/bigramas/” y se exponen a continuación:

- 2gramas.py: Genera el fichero de bigramas del conjunto de muestras especificado. Para especificar un conjunto hay que descomentar el corpus de entrada, también se ha de descomentar el nombre que recibirá el fichero de salida (son correlativos).
- script_train.sh: Crea el traductor automático utilizando los conjuntos de datos generados como bigramas.
- script_test.sh: Realiza la traducción y la evaluación del modelo (BLEU).

Trigramas

Para obtener los resultados para trigramas se han utilizado los ficheros que se encuentran en “Extra/ejercicio1/trigramas/” y se exponen a continuación:

- 3gramas.py: Genera el fichero de trigramas del conjunto de muestras especificado. Para especificar un conjunto hay que descomentar el corpus de entrada, también se ha de descomentar el nombre que recibirá el fichero de salida (son correlativos).
- script_train.sh: Crea el traductor automático utilizando los conjuntos de datos generados como trigramas.
- script_test.sh: Realiza la traducción y la evaluación del modelo (BLEU).

Unidades básicas	**BLEU**
Bigramas (2gramas)	11.31
Trigramas (3gramas)	4.58

Tabla 6: Resultado (BLEU) al utilizar bigramas y trigramas como unidades básicas.

Se observa que el resultado de la experimentación es peor que el resultado obtenido cuando se trabaja con palabras como unidades básicas (ejercicio básico). En el caso de bigramas vemos que el BLEU baja en 16 puntos y con trigramas el BLEU baja en 22 puntos. Al utilizar como unidades básicas bigramas el modelo de lenguaje obtenido mediante SRILM hace un modelo de trigramas basado en bigramas donde se observa que es más complicado que se repitan las unidades básicas (bigramas) en el modelo. Cuando las unidades básicas son trigramas SRILM hace un modelo de trigramas basado en trigramas donde se observa que es aún más complicado que se repitan las unidades básicas (trigramas) en el modelo.

2) Estudio de la degradación que se produce en un proceso consistente en traducir en un sentido y a continuación traducir el resultado en sentido contrario sucesivas veces.

Para realizar este ejercicio se ha generado un traductor de español a inglés para poder realizar una traducción en ambos sentidos, para realizar este traductor se ha utilizado el mismo script “script_train.sh” del ejercicio básico pero intercambiando el lenguaje de salida por el lenguaje de entrada. El proceso de traducir en un sentido y en sentido contrario se ha realizado dos veces, una vez empezando en sentido de inglés a español y la segunda vez empezando en sentido de español a inglés.
Los scripts que se han utilizado para obtener los resultados de las tablas se encuentran en “Extra/ejercicio2/” y se comentan a continuación:

- script_test_es_en.sh: Realiza la degradación empezando en sentido Español-Inglés.
- res_degradacion_bleu_es_en.txt: Resultados de la ejecución anterior.
- script_test_en_es.sh: Realiza la degradación empezando en sentido Inglés-Español.
- res_degradacion_bleu_en_es.txt: Resultados de la ejecución anterior.

Mostramos los resultados en las siguientes tablas:

Sentido		**BLEU**
Inglés	Español	27.24
Español	Inglés	53.27
Inglés	Español	26.44
Español	Inglés	51.81
Inglés	Español	26.28
Español	Inglés	51.61
Inglés	Español	26.29
Español	Inglés	51.62
Inglés	Español	22.31

Tabla 7: Resultados (BLEU) de la degradación obtenida empezando en sentido Inglés-Español.

Sentido		**BLEU**
Español	Inglés	28.12
Inglés	Español	49.78
Español	Inglés	26.50
Inglés	Español	48.33
Español	Inglés	26.38
Inglés	Español	48.18
Español	Inglés	26.36
Inglés	Español	48.14
Español	Inglés	26.36

Tabla 7: Resultados (BLEU) de la degradación obtenida empezando en sentido Español-Inglés.

En esta experimentación observamos que al traducir del idioma origen al idioma destino el BLEU es el esperado pero al volver a traducir en sentido contrario el BLEU mejora considerablemente (aproximadamente el doble de BLEU), esto se produce porque al traducir del idioma origen al idioma destino hay muchas palabras que se quedan sin traducir, es decir, se quedan en el lenguaje de origen y al realizar la traducción en el sentido contrario esas palabras ya se encuentran bien traducidas. Por otra parte también se observa el proceso de degradación viendo como al realizar la traducción cada vez del idioma origen al idioma destino el BLEU disminuye, esto se debe a que la traducción obtenida cada vez aporta peores resultados y se va acumulando ese error en cada traducción.

3) Construir un traductor que realice una post-edición automática utilizando el corpus intermedio que está aquí: 1) Traducir europarl-v7.es-en-train-red-PE.en con el traductor del ejercicio básico; 2) Construir un traductor con Moses tomando como entradas las traducciones obtenidas de europarl-v7.es-en-train-red-PE.en y como salidas europarl-v7.es-en-train-red-PE.es; 3) Traducir el conjunto de test del ejercicio básico; 4) Comparar los resultados.

En este apartado se ha realizado un traductor que realiza una post-edición automática utilizando el corpus intermedio europarl-v7.es-en-train-red-PE. Se ha cogido el traductor (Básico) del ejercicio básico y se ha traducido europarl-v7.es-en-train-red-PE.en al español, llamemos a esta traducción trad_es. Se ha generado un nuevo traductor del idioma español' al idioma español, para ello se ha tomado como entrada el fichero trad_es y como salida el fichero europarl-v7.es-en-train-red-PE.es, por último se ha realizado la evaluación que consiste en coger el test del inglés del ejercicio básico, traducirlo al español con el traductor (Básico) del ejercicio básico y esta traducción pasarla al traductor español'-español como entrada evaluando la salida sobre el test del español del ejercicio básico.

Los scripts que se han utilizado para obtener los resultados de la tabla se encuentran en "Extra/ejercicio3/" y se comentan a continuación:

- script_test1.sh: Traduce y evalua el test europarl-v7.es-en-train-red-PE.en con el traductor del ejercicio básico.
- script_train.sh: Generar el traductor del español' al español.
- script_test_postedicion.sh: Traduce y evalua el test del inglés al español del ejercicio básico y del español' al español.

El resultado de realizar una post-edición se muestra a continuación:

Traductor	**BLEU**
Post-edición español'-español	27.08

Tabla 8: Resultado (BLEU) del estudio de la post-edición automática.

Observamos que al realizar la post-edición el resultado no mejora respecto al ejercicio básico, una hipótesis a este resultado obtenido es que las frases de la post-edición no guardan una relación significativa con las frases de entrenamiento y test del modelo del traductor inglés-español, hubiera sido más interesante disponer de más pares de frases de test para realizar más de una evaluación y poder obtener unas conclusiones más claras.

4) Repetir el ejercicio básico con el toolkit Thot (https://github.com/daormar/thot/).

Este apartado consiste en obtener un traductor automático del inglés al español con el corpus proporcionado en el ejercicio básico, pero en vez de utilizar el toolkit Moses utilizar el toolkit Thot. La instalación y configuración de Thot ha generado una serie de problemas que con paciencia e investigando se han solucionado, los problemas y su correspondiente soluciones están comentadas dentro del script thot.sh.

Los scripts que se han utilizado para obtener los resultados de la tabla se encuentran en "Extra/ejercicio4/" y se comentan a continuación:

- thot.sh: Es el script con el que se han entrenado los modelos y se obtenido el BLEU. Contiene anotaciones del conjunto de problemas que han aparecido en la instalación y la solución que se ha aplicado para corregirlos.
- salida_thot_sh.txt: Es la salida que se ha obtenido al ejecutar thot.sh.

Nota: Thot devuelve el resultado (BLEU) en un rango entre 0.0 y 1.0.

Traductor	**BLEU**
Thot Ejercicio básico (inglés-español)	22.41

Tabla 9: Resultado (BLEU) del ejercicio básico utilizando el toolkit Thot.

Se han utilizado el conjunto de datos completo proporcionado por el ejercicio básico, es decir, 50.000 pares de frases para entrenamiento (85% train + 15% dev) y 1.000 pares de frases para test. Observamos que el traductor obtenido con Moses funciona mejor ya que

hemos obtenido un 27.24 de BLEU mientras que con el traductor obtenido con Thot se consigue un 22.41 de BLEU.

5) Ejercicio básico ampliando el corpus con los datos proporcionados en el ejercicio 3 (europarl-v7.es-en-train-red-PE).

En este apartado se ha cogido el corpus proporcionado en el ejercicio básico (europarl-v7.es-en-train-red) y se ha unido con el corpus proporcionado en el ejercicio (extra) 3 (europarl-v7.es-en-train-red-PE) obteniendo un corpus resultante de 70.000 pares de frases de entrenamiento (85% train + 15% dev). Con este corpus se ha generado un nuevo traductor del inglés al español, el cual se ha evaluado con el test proporcionado en el ejercicio básico.

Traductor	**BLEU**
Básico (final) del Inglés al Español	27.24
Básico (v2) del Inglés al Español	28.87

Tabla 10: Resultado (BLEU) con un corpus con más datos de entrenamiento.

Observamos que la mejora no es muy significativa incrementando en 20.000 pares de frases el corpus de entrenamiento (train), esto nos permite comprender la complejidad existente en los sistemas de traducción automática, la complejidad viene dada en que por muchos datos de entrenamiento que tengamos siempre van a aparecer palabras no vistas que provocan fallos en la traducción y que nos aparecen reflejados al obtener el BLEU.