Traducción Automática

Traducción estadística basada en frases: MOSES

Victoria Beltrán Domínguez

Diciembre 2021

Índice

1. Introducción

En esta práctica nuestro objetivo es experimentar con la herramienta Moses [1] para construir sistemas de traducción a partir de conjuntos de pares de frases bilingües.

Para ello, lo primero que se hará es seguir la guía detallada en el marco de las prácticas. Seguidamente, se procederá a realizar los ejercicios propuestos, analizando los resultados obtenidos.

2. Descripción del trabajo realizado en la sesión

En esta sección se procede a describir brevemente el trabajo realizado en la sesión de prácticas.

Lo primero debemos hacer es definir ciertas variables de entorno para facilitar el acceso a las herramientas Moses y auxiliares. Antes de continuar, y para asegurar la correcta instalación, realizamos una pequeña prueba con modelos ya entrenados para ver si saca traducciones correctamente.

Una vez ya tenemos las variables de entorno definidas y nos hemos cerciorado de que Moses funciona correctamente, procedemos a empezar desde cero con el entrenamiento de nuestro modelo. Para ello, lo primero que debemos hacer es bajar el conjunto de entrenamiento indicado en la práctica, tokenizarlo (en el caso de que no lo estuviera) y limpiarlo los datos.

Una vez tenemos los datos de entrenamiento preparados, debemos entrenar los modelos de lenguaje de salida. Esto se consigue mediante el uso de la herramienta SRILM [2]. Más concretamente, mediante el comando *ngram-count* creamos un modelo de trigramas con suavizado por interpolación y descuento de Kneser-Ney a partir de los datos de entrenamiento. Guardaremos el modelo en *turista.lm*.

Una vez tenemos el modelo de trigramas, pasamos a construir la tabla de segmentos y los modelos de reordenamiento. Para ello, utilizaremos el *software* GIZA++ [3]. Con este, vamos a obtener diferentes modelos de ordenamiento y la tabla de segmentos a partir de las alineaciones computadas entre las frases de entrenamiento de los idiomas origen y destino y el modelo previamente entrenado de trigramas.

En la última parte de entrenamiento, deberemos bajarnos un conjunto de desarrollo, limpiarlo, y entrenar con Mert con el fin de obtener los pesos óptimos para cada uno de los modelos ya entrenados.

Finalmente, utilizaremos un conjunto de test (limpio) para traducir las frases y las compararemos para obtener su puntuación BLEU. Esto lo conseguiremos mediante Moses. Esto nos da como resultado una puntuación BLEU de 91.92.

3. Resolución ejercicios propuestos

3.1. Ejercicio 1

Probar el modelo obtenido en el apartado 6 sin ajuste de pesos.

Realizando el proceso explicado en el anterior apartado sin ajustar los pesos de los modelos obtenidos, nos da una puntuación BLEU de 88.42. Como cabía esperar, la puntuación BLEU empeora, pues todos los modelos pre-generados tienen probabilidades no óptimas.

3.2. Ejercicio 2

Probar para valores más altos del número máximo de iteraciones del MERT (aparatado 7)

Anteriormente, probando con 5 iteraciones, se obtuvo una puntuación de 91.92. Se ha experimentado dos veces con un máximo de 10 iteraciones, convergiendo una vez en las 8 iteraciones y obteniendo un resultado BLEU de aproximadamente 92.0 y otra vez incluso convergiendo en la iteración 6 y dando un resultado de aproximadamente 91.95. Sin embargo, vemos que el aumento de iteraciones no supone un aumento significantivo de la puntuación BLEU mientras que el tiempo de computo sí aumenta significativamente.

3.3. Ejercicio 3

Probar distintos valores de n-gramas (apartado 5)

Tal como vemos en la tabla, probando con distintos valores de n-gramas, vemos como la puntuación BLEU va mejorando a medida que el tamaño de los N-gramas aumenta, hasta que obtenemos el mejor valor para 4-gramas y empieza a bajar de nuevo para n-gramas de tamaño 5 y 6. Por lo tanto, podríamos decir que el tamaño óptimo para este conjunto de datos es de 4.

N-gramas	BLEU
2	91.12
3	91.93
4	92.46
5	92.42
6	92.40

3.4. Ejercicio 4 (opcional)

Probar MIRA para el entrenamiento de los pesos del modelo log-lineal (ver manual de Moses)

Con el fin de experimentar con MIRA, debemos añadir a la orden *mert-moses* el argumento *–batch-mira*. Haciéndolo, obtenemos una puntuación BLEU de 90.93, puntuación que no consigue alcanzar el mejor resultado obtenido hasta el momento.

3.5. Ejercicio 5 (opcional)

Probar otras técnicas de suavizado (ver manual de SRILM)

Probando con diferentes combinaciones de *Backoff* e *Interpolation*, combinados con métodos de descuento *Wittenbell* y *Kneser-Ney*, obtenemos los siguientes resultados:

	Backoff	**Interpolation**
Wittenbell	91.92	91.47
KneserNey	91.84	91.81

Partiendo de las pruebas realizadas, el mejor método de suavizado para este ejercicio es *Backoff*. Además, de entre los métodos de descuento probados, es *Wittenbell* el que consigue igualar el resultado. Sin embargo, ninguno consigue obtener el resultado comparable destacable BLEU (todos rondan un número cercano y podría deberse al factor aleatorizado de Mert).

3.6. Ejercicio 6 (opcional)

Probar moses monótono (ver manual de Moses)

Con el fin de probar Moses monótono, tal y como indica la página 67 del manual de Moses, simplemente debemos añadir el argumento *-distortion-limit 0* en el comando que obtiene los modelos de ordenamiento y tablas de segmentos. Obtenemos un resultado de 90.87, uno de los peores resultados obtenido a lo largo de toda la práctica. Esto no es un hecho que sorprenda, pues el sistema no está intentando reordenar las traducciones.

4. Conclusiones

En esta práctica, se ha experimentado con la creación de modelos de traducción utilizando *Moses*. Como conclusiones a los ejercicios realizados en esta práctica, podemos destacar que el ajuste de pesos de los modelos contribuye bastante en la puntuación *BLEU*, que debemos encontrar un equilibrio entre el número máximo de iteraciones y el tiempo de cómputo, que el valor óptimo para los n-gramas es de 4 y que el mejor método de suavizado para este problema es *Backoff* con *Wittenbell* (aunque para los dos últimas conclusiones sin una diferencia destacable en la puntuación BLEU). No podemos afirmar que estas conclusiones sean generalizables, pues el conjunto de datos es muy pequeño y los resultados son muy variables en cada ejecución.

5. Bibliografía

[1] Philipp Koehn y col. «Moses: Open Source Toolkit for Statistical Machine Translation.» En: *ACL*. Ed. por John A. Carroll, Antal van den Bosch y Annie Zaenen. The Association for Computational Linguistics, 2007.

[2] Andreas Stolcke. «SRILM – An extensible language modeling toolkit». En: *IN PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (ICSLP 2002*. 2002, págs. 901-904.

[3] Franz Josef Och y Hermann Ney. «A Systematic Comparison of Various Statistical Alignment Models». En: *Computational Linguistics* 29.1 (2003), págs. 19-51.