



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Departamento de sistemas informáticos y
computación

Traducción estadística basada en frases: Moses

Práctica Traducción automática

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

Autor

Francisco Javier Gil-Terrón Rodríguez

2021 - 2022

Tabla de contenidos

1.	Introducción	3
2.	Modelo sin ajuste de pesos	3
3.	Variación nº iteraciones MERT	4
4.	Variación valores n-gramas	4
5.	Evaluación MIRA	5
6.	Técnicas de suavizado	6
7.	Moses monótono	7
8.	Bibliografía	7

1. Introducción

En esta práctica trataremos de construir un sistema de traducción mediante el uso de la herramienta Moses, que creará un modelo de traducción automática a partir de un conjunto de frases en dos idiomas, en este caso, español e inglés.

Para ello trabajaremos sobre el conjunto de datos EuTrans y comenzaremos limpiando los datos para ambos idiomas de manera que las frases queden alineadas (ya viene tokenizado).

A continuación, se entrena el modelo de lenguaje para el inglés utilizando SRILM, que será un modelo de trigramas con método de suavizado Kneser-Ney e interpolación, y tras esto se alineará a nivel de palabra ambos conjuntos (español e inglés) para construir la tabla de segmentos (phrases) y los modelos de reordenamiento empleando el software GIZA. Una vez hecho esto, ya se puede entrenar los pesos del modelo log-lineal para lo que se ha hecho uso de Mert, limitado a 5 iteraciones.

Finalmente, se utilizará Moses con los modelos creados con anterioridad y se contrastará la salida con el conjunto de test para obtener el BLEU, que en esta primera aproximación ha obtenido un valor de 91.76.

2. Modelo sin ajuste de pesos

En este apartado se probará el modelo de traducción sin ajuste de pesos, esto es, saltándonos el paso de entrenar los pesos del modelo log-lineal con Mert.

Bajo esta premisa, haremos uso de los pesos iniciales a partir de los cuales se han entrenado los empleados en la versión inicial, que siguiendo las instrucciones del boletín se encontrarán en la ruta `/train/work/model/moses.ini`.

Con esta aproximación se ha obtenido un BLEU de 88.42, que como era de esperar, es inferior al original.

Modelo con pesos ajustados	Modelo sin ajuste de pesos
91.76	88.42

Tabla 1 – Valores de BLEU en función del ajuste de los pesos del modelo log-lineal

3. Variación nº iteraciones MERT

Esta sección tratará de evaluar los resultados obtenidos tras entrenar los pesos del modelo log-lineal con Mert, incrementando el número máximo de iteraciones, pues recordemos que inicialmente se ha limitado a 5 iteraciones y en mi caso no ha alcanzado la convergencia (consideraremos que no ha convergido si ejecuta el número máximo de iteraciones establecido), ya que Mert introduce cierta aleatoriedad que hace que el algoritmo converja en un rango de iteraciones y no en una determinada.

La primera prueba se ha realizado incrementando el máximo a 7 iteraciones, y en este caso tampoco ha alcanzado la convergencia, pero sí que ha obtenido un mayor valor de BLEU, en concreto, 92.10.

Tras esta prueba se incrementó el máximo a 10 iteraciones, donde convergió a la cuarta iteración, y mejorando ligeramente el BLEU, llegando a ser el máximo alcanzado hasta este punto con un valor de 92.15.

5 iteraciones (sin converger)	7 iteraciones (sin converger)	10 iteraciones (tras convergencia)
91.76	92.10	92.15

Tabla 2 – Valores de BLEU en función de iteraciones de Mert

Tras estas pruebas, se tomará valor de iteraciones máximas 10 para asegurar que converja el algoritmo y obtener los mejores resultados posibles, que, en mi caso, ha sido suficiente para converja en las futuras pruebas.

4. Variación valores n-gramas

Hasta ahora hemos trabajado sobre un modelo de lenguaje de trigramas, y en esta sección se evaluarán los resultados obtenidos con otros modelos de n-gramas.

Empezaremos con bigramas y tras crear el modelo de lenguaje, se entrenarán los pesos con Mert hasta convergencia. El BLEU obtenido para el modelo de bigramas fue de 88.96, el más bajo obtenido en las pruebas de n-gramas.

Este mismo proceso se ha repetido para 4-gramas y 5-gramas, cuyos valores de BLEU están más en la línea de los trigramas, siendo respectivamente 91.72 y 92.15, una vez más, entrenados los pesos con Mert hasta convergencia.

Los resultados se pueden ver reflejados en la siguiente tabla.

3-grama	2-grama	4-grama	5-grama
92.15	88.96	91.72	92.15

Tabla 3 – Valores de BLEU en función del tamaño de n-grama

5. Evaluación MIRA

A continuación, probaremos a estimar los pesos del modelo log-lineal con una alternativa a Mert: Mira. Para ello, bastará con añadir el argumento `--batch-mira` al ejecutar la instrucción para el entrenamiento.

Los valores de BLEU obtenidos estaban en torno a 90.95, que no llega a igualar los que alcanza Mert.

Mert	Mira
92.15	90.95

Tabla 4 – Valores de BLEU en función del algoritmo de estimación de pesos

Mira, por su parte, requiere de más iteraciones para converger, aunque cabe destacar que las iteraciones de Mira fueron considerablemente más rápidas, y, en conjunto era más rápido que Mert, aunque se debe tener en consideración que las pruebas han sido realizadas desde polilabs y que la carga del sistema en un momento dado podría haber modificado el tiempo real de un algoritmo respecto al otro.

6. Técnicas de suavizado

Tal como se citó en la introducción, el tipo de suavizado que se empleó inicialmente es el de Kneser-Ney con interpolación. Probaremos ahora, siguiendo con el modelo de lenguaje de trigramas y entrenando los pesos con Mert hasta convergencia, otras alternativas de suavizado.

La primera alternativa que se probó fue la seguir empleando Kneser-Ney, pero esta vez con backoff en lugar de interpolación, donde logró un mejor resultado que con interpolación, llegando a un valor de BLEU de 92.28 y siendo el más alto hasta ahora. Para ello, únicamente fue necesario eliminar el argumento

-interpolate, ya que backoff es la opción predeterminada que trae Moses.

A continuación, se evaluará el suavizado de Written-Bell, tanto con backoff como con interpolación. Para ello, se reemplazará el argumento -kndiscount por -wbdiscout, y se añadirá la opción -interpolate para su respectivo caso de prueba. Los valores obtenidos de BLEU fueron respectivamente 91.97 y 91.52, ambos más o menos similares al del caso original, pero sin alcanzar el de Kneser-Ney con backoff.

Por último, se probó a añadir el suavizado de Laplace o 'añade uno', pero únicamente con backoff, ya que, tal como citaba la documentación de Moses, sólo está dicha opción implementada. En este caso, el argumento a utilizar es -addsmooth 1, y el valor de BLEU obtenido es 84.41.

	Interpolación	Backoff
Kneser-Ney	92.15	92.28
Written-Bell	91.52	91.97
Laplace	-	88.42

Tabla 5 – Valores de BLEU en función del método de suavizado

Con estos resultados podemos deducir la gran importancia que puede llegar a tener un correcto suavizado para obtener los mejores resultados posibles en una traducción automática.

7. Moses monótono

La última prueba que se propone en la presente práctica es la de estimar el valor de BLEU obtenido empleando Moses monótono, lo que significa que la traducción resultante no tendrá reordenación.

Esto se realizará sobre el modelo de lenguaje inicial, con suavizado Kneser-Ney e interpolación y entrenando los pesos del modelo log-lineal con Mert hasta convergencia, y para ello se añadirá el argumento `-distortion-limit 0`.

El BLEU obtenido con Moses monótono fue de 91.83, que como era de esperar, es ligeramente inferior a la versión original.

Moses dinámico	Moses monótono
92.15	91.83

Tabla 6 – Valores de BLEU en función de la capacidad de reordenación

8. Bibliografía

Casacuberta Francisco, & Domingo Miguel. (2021). *Boletín de la práctica*. Universitat Politècnica de València.
<https://www.prhlt.upv.es/~fcn/Students/ta/p1mt.pdf>

Deniz, Y., & Andreas, S. (2007). *ngram-discount*. SRI International.
<http://www.speech.sri.com/projects/srilm/manpages/ngram-discount.7.html>

European Union. (2018). *Moses - Main/HomePage*. LGPL.
<https://www.statmt.org/moses/>