Machine translation

Europarl 2021-2022

Adrián Vázquez Barrera

Polytechnic University of Valencia

January 2022

Introducción

En esta práctica se pide realizar una un traductor automático Inglés-Español, basándonos en el corpus Eroparl-v7. Para la que se han entrenado una serie de modelos con métodos estadísticos y neuronales.

De este modo, y partiendo de la información obtenida previamente en las prácticas, se ha hecho un análisis modificando los parámetros correspondientes en Moses y NMT-Keras.

Antes de comenzar, se ha efectuado el correspondiente pre-procesado de los datos, ya que esto es clave para facilitar el entrenamiento. En primer lugar, se ha tenido de tokenizar el corpus, así como una limpieza del mismo, esto ha provocado un descenso de unas tres mil frases entre las treinta y cinco mil originales. Adicionalmente, se han mezclado aleatoriamente las muestras del conjunto de entrenamiento, de este modo conseguimos una mejor distribución a la hora de efectuar la posterior extracción de frases para el conjunto de desarrollo, al mismo tiempo que eliminamos en cierta medida la correlación que estas puedan tener unas con otras.

Como no podía ser de otro modo, el conjunto de desarrollo se ha desactivado para el entrenamiento con NMT-Keras y OpenNMT.

Ejercicios

Ejercicio 1: Moses

EXPERIMENTOS			
Iteraciones (MERT)	n-grama	suavizado	BLEU
7	5	Kneser-Kney	26.20
14	3	Kneser-Kney	25.45
14	4	Kneser-Kney	26.19
14	5	Kneser-Kney	26.40
20	5	Kneser-Kney	26.45

Observamos como los mejores resultados se obtienen a partir de siete iteraciones con 5-gramas rompiendo la barrera del 26 de BLEU. Para la traducción del conjunto oculto ofrecido por el profesor, se ha utilizado esta variante con veinte iteraciones.

No se ha empleado MIRA para el entrenamiento, aunque reduce el tiempo de ejecución, los resultados obtenidos son algo peores, como ya quedo visto en la primera práctica. Además, en este caso, el tiempo de cómputo no era una restricción muy grande al contar con más de un mes para realizar todos los cálculos. El hecho de haber usado Kneser-Kney como método de suavizado, obedece también a las conclusiones de la práctica anterior.

Ejercicio 2: NMT-Keras

A continuación se procederá a repetir el ejercicio anterior utilizando el toolkit NTM-Keras, para entrenar redes neuronales recurrentes con mecanismos de atención. Durante el desarrollo del ejercicio se irán modificando los parámetros disponibles en el toolkit hasta alcanzar una cota de error similar al umbral establecido en el enunciado de la práctica.

Embedding size	BLEU
64	18.32
128	19.71
256	19.57

Si aumenta el tamaño de embedding, observamos como mejora en BLEU obtenido con respecto a 64, que es su valor por defecto. En este caso, es apreciable como a partir de 128 no se logra mejorar más, o al menos no de forma significativa. Por lo tanto, y para ahorrar tiempo de cómputo, se fijará el tamaño a 128.

Encoder/decoder size	BLEU
64	19.71
128	20.45
256	21.55
512	21.49

Es apreciable como a medida que aumentamos el tamaño del encoder/decoder, conseguimos reducir el error del modelo de traducción. Esto es debido a que el corpus es bastante extenso y diverso, al menos ajustado a las capacidades de cómputo disponibles. Por tanto, es de esperar que a medida que se permita mayor espacio de representación, el modelo mejore, aunque como puede verse, no infinitamente. A partir de 256 la variación es prácticamente anecdótica.

Learning Rate	BLEU
0.01	5.77
0.001	21.55
0.0005	20.84

Existen otros optimizadores además de Adam como Adagrad y Adadelta, pero su convergencia es consideradamente más lenta, como ya vimos en la práctica correspondiente. Teniendo en cuenta las limitaciones de tiempo y capacidad de cómputo disponible, se hace inviable utilizarlos.

Ejercicio 3: Tranformer en NMT-Keras

Model size	BLEU
64	9.86
128	10.86
256	5.08

Para poder hacer un análisis justo (en igualdad de condiciones con el ejercicio anterior) y en un tiempo razonable, se ha realizado el experimento con cinco épocas, en este caso vemos como el BLEU baja considerablemente, esto tiene cierto sentido, pues los modelos Transformer necesitan de una gran cantidad de datos (y GPU dedicada) para ofrecer buenos resultados.

De todos modos, una vez hemos fijado que el tamaño de modelo "óptimo" es 128, se ha probado a entrenar con cinco veces más épocas, con un resultado de 7.88, lo que no hace más que confirmar lo mencionado en el párrafo anterior.

Ejercicio 4: OpenNMT

Para este ejercicio, se ha utilizado la versión de TensorFlow del toolkit OpenNMT utilizando la plataforma Google Colaboratory, que cuenta con los recursos suficientes para realizar esta tarea en un tiempo razonable, a diferencia de los equipos domésticos que se han utilizado para esta entrega.

Empleando un modelo mediano, con tamaño de batch de cien, y diez mil iteraciones se ha conseguido llegar hasta un 8.86 de BLEU. Se intentó ejecutar la misma prueba usando Transformers, pero era imposible hacerlo (por la dimensión del corpus) en las mismas condiciones y tomaba demasiado tiempo, tanto que la propia plataforma desactivaba el entorno antes de poder obtener algún resultado.

Conclusiones

Tras la realización de este proyecto se ha podido ver como en algunas ocasiones, los modelos estadísticos pueden ofrecer mejores resultados que los modelos neuronales. En este caso debido a la limitación de hardware (puramente doméstico) y a la (relativa) poca cantidad de muestras disponibles para entrenar.

Es muy probable que con mejores máquinas y muchos más datos pudieran haberse conseguido un mejor BLEU, tal y como nos dice el estado del arte de la traducción automática actualmente. No obstante, esto no hace más que constatar la validez de los modelos probabilísticos en determinados casos como pueda ser este.

Bibliografía

1. Philipp Koehn. MOSES: Statistical Machine Translation System. User Manual and Code Guide. University of Edinburgh. 2014.
http://www.statmt.org/moses/manual/manual.pdf

2. SRILM - The SRI Language Modeling Toolkit
http://www.speech.sri.com/projects/srilm/

3. Philipp Koehn. Europarl: A parallel corpus for statistical machine translation.