Traducción estadística basada en frases: MOSES

Ramon Ruiz Dolz

December 2018

1 Introducción

El propósito de esta practica es, mediante MOSES, construir un sistema de traducción automática español-ingles de mayor precisión posible. Para ello, basándonos en la teoría estadística hemos entrenado distintos modelos y posteriormente hemos realizado una comparativa entre ellos. Se he hecho uso de la métrica BLEU para la comparativa.

2 Experimentos

En esta practica se han realizado siete experimentos diferentes con el objetivo de comparar la influencia de determinados parámetros en los modelos resultantes. Previo al comienzo de la experimentación se han realizado dos tareas comunes a todos ellos con tal de poder trabajar con comodidad. En primer lugar se han definido las variables de entorno, comunes a todos los experimentos realizados en esta sesión. Además de esto, también se han *limpiado* los corpus de test y de entrenamiento. Con *limpiar* nos referimos a realizad un nivelado y homogeneizar ambos ficheros para poder trabajar de forma adecuada. Siguiendo el guión del boletín, a continuación se explica el proceso experimental seguido en cada uno de estos experimentos.

2.1 Experimento inicial

El experimento inicial que se propone en el boletín esta dividido en cinco pasos principales. El primero de ellos consiste en entrenar el modelo del lenguaje de salida, es decir, el inglés. Para ello se hace uso de la librería SRILM [10] para construir el modelo de tri-gramas a partir del corpus inglés ya limpio. Para este experimento inicial el modelo de tri-gramas se ha hecho uso del suavizado de Kneser-Ney [7]. Ya con el modelo de tri-gramas terminado, se ha entrenado el modelo de traducción en sí. Para ello se ha construido una tabla con los segmentos o frases. Además también se han generado los modelos de reordenado a partir de los datos de train. Para todo esto se hace uso de GIZA++ [2], el modelo resultante se guarda como *moses.ini* a partir del cual se realizará el ajuste de los pesos. Antes de ajustar los pesos es importante obtener el corpus

de desarrollo y dejarlo limpio de igual forma que los anteriores. Ya con el corpus de desarrollo preparado, se hace uso de la técnica MERT [8] para realizar el ajuste de los pesos del modelo, se asigna un numero máximo de iteraciones a 5 para evitar el sobrecoste temporal de esta parte. Finalmente, una vez los pesos están ajustados, para cerrar este experimento se ha realizado el proceso de traducción con tal de comprobar el funcionamiento del modelo entrenado. Para ello se ha hecho uso de los datos de test en español, de esta forma, se traducen estos datos al ingles usando el modelo de traducción entrenado y se realiza una evaluación comparando el resultado obtenido con el resultado correcto.

Partiendo de este experimento inicial, el resto de experimentos consisten principalmente en modificar alguno de los pasos del experimento y observar su repercusión en la evaluación final, en las siguientes secciones se explican los principales cambios respecto a este experimento.

2.2 Experimento ajuste de pesos

Este experimento consiste en comprobar el rendimiento del modelo entrenado sin realizar el ajuste de pesos mediante MERT [8]. Por lo tanto en este experimento, todos los pasos son iguales al anterior caso, únicamente saltando el ajuste de pesos y, por lo tanto, realizando la traducción del test con el modelo *moses.ini* generado inicialmente.

2.3 Experimento número máximo de iteraciones

El experimento del número de iteraciones consiste básicamente en analizar la repercusión que tiene sobre el resultado traducido, realizar un aumento en el número de iteraciones máximas en el paso de ajuste de parámetros usando MERT [8]. El hecho de aumentar el número de iteraciones ha repercutido de forma bastante notable en el tiempo de ajuste. Concretamente se ha experimentado aumentando este número a 7 y a 10. Es importante también remarcar que, debido al pequeño tamaño del corpus, en el experimento lanzado con 10 iteraciones el sistema se ha detenido antes de llegar a la décima iteración.

2.4 Experimento n-gramas

Además de los experimentos anteriores, también se ha probado el efecto en el modelo de traducción, el entrenamiento con modelos distintos a los tri-gramas. Concretamente se ha probado con bi-gramas, 4-gramas y 5-gramas. Concretamente, en el paso inicial de construcción del modelo de lenguaje de salida mediante SRILM [10] se ha modificado la instrucción para crear tres nuevos modelos con los n-gramas deseados. Posteriormente, se han realizado los pasos con los mismos parámetros que en experimento inicial hasta la evaluación.

2.5 Opcional: Experimento MIRA

El propósito de este experimento consiste en comprobar el desempeño del uso del *Margin Infused Relaxed Algorithm* propuesto en [5]. Concretamente se hace uso de la funcionalidad implementada en Moses de *k-best batch MIRA Tuning* [3, 4] como alternativa al uso de MERT [8]. Este algoritmo consiste en un entrenamiento online haciendo uso de las k mejores listas como una aproximación al espacio de búsqueda real del *decoder*. Concretamente MIRA permite trabajar con características más grandes y obtener mejores resultados generalmente cuando el contador de características pasa de 10. Para lanzar este experimento se ha añadido la instrucción *–batch-mira* en el momento de realizar el ajuste de pesos.

2.6 Opcional: Experimento suavizados

En el experimento de suavizados se ha probado a trabajar con modelos del lenguaje basdos en tri-gramas con distintos suavizados respecto al experimento inicial. Como se indicó anteriormente, en el experimento inicial, cuando se construye el modelo de lenguaje mediante SRILM [10] se hace uso del suavizado propuesto por Kneser y Ney en [7]. En este experimento, además de este suavizado se han construido otros dos modelos de lenguaje basados en tri-gramas haciendo uso del suavizado Good-Turing [6] y Witten-Bell [1]. A continuación se repasan brevemente el funcionamiento de los métodos de suavizado empleados.

Good-Turing consiste en estimar una probabilidad p_0 para todas las muestras no observadas. A partir de esta probabilidad p_0 obtenida a partir de $\frac{N_1}{N}$ se reestiman las probabilidades de las muestras observadas aplicando ya esta probabilidad como suavizado y redistribuyendo la masa de probabilidad entre las muestras no observadas. Puesto que Good-Turing es el suavizado por defecto de SRILM, para aplicar este suavizado basta con no especificar suavizado en el momento de construir el modelo de tri-gramas.

Witten-Bell, por otra parte es un método de suavizado dónde la masa de probabilidad a distribuir entre las muestras no observadas se calcula en base a T, el número total de tri-gramas, en este caso, distintos observados en el conjunto. Para aplicar el descuento de Witten-Bell se debe añadir *-wbdiscount* a la linea de comandos.

Finalmente, el suavizado Knesser-Ney consiste en, partiendo del descuento absoluto, realizar modificaciones sobre la distribución de probabilidad suavizadora. En este modelo de suavizado se propone tener en cuenta el contexto en el que aparecen las palabras a la hora de suavizar los n-gramas, puesto que es posible que una palabra aparezca mucho acompañada de una determinada palabra pero sin embargo que sea muy poco probable su aparición junto a otra palabra distinta. Para hacer uso de este suavizado en la construcción del modelo de tri-gramas es necesario poner la instrucción *-kndiscount* en la linea de comandos.

2.7 Opcional: Experimento monótono

Cerrando la sección de experimentación, se ha realizado un último experimento modificando el método de alineamientos en el proceso de traducción automática. Concretamente, la principal diferencia entre un alineamiento monótono y un no-monótono consiste en que mientras un alineamiento no monótono permite reordenar los bloques, un alineamiento monótono no realiza esta reordenación. De esta forma en el caso de que los distintos idiomas mantengan una ordenación sintáctica diferente, esto no se podrá ver reflejado en la traducción obtenida. Para que la traducción se realice sin reordenamiento de bloques, es decir que sea monótono, hay que modificar el parámetro *distortion limit* a 0 dentro del fichero *moses.ini*, es decir el modelo.

3 Resultados

En esta sección se presentan los resultados obtenidos para cada uno de los experimentos realizados comparados frente a los resultados obtenidos en el experimento inicial. En este experimento se han obtenido los siguientes resultados,

Exper.	**Pesos**	**Max. Iter**	**N-gramas**	**Suavizado**	**Monotonía**	**BLEU**
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08

Table 1: Resultados experimento inicial

En las siguientes secciones se realizará la comparación con los experimentos pertinentes. Para evaluar los modelos se hace uso de BLEU [9] una métrica bastante estandarizada para la evaluación de modelos de traducción automática.

3.1 Experimento ajuste de pesos

En la siguiente tabla se puede observar la diferencia en los resultados obtenidos para un modelo con ajuste de pesos o sin ajuste de pesos.

Exper.	**Pesos**	**Max. Iter**	**N-gramas**	**Suavizado**	**Monotonía**	**BLEU**
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08
Pesos	No	5	Tri-gramas	Kneser-ney	No	87.96

Table 2: Resultados experimento pesos

El modelo con ajuste de pesos obtiene una mejor puntuación de BLEU, puesto que se puede considerar que ha afinado más las probabilidades estimadas mediante este ajuste de pesos.

3.2 Experimento número máximo de iteraciones

En este apartado se presentan los resultados obtenidos al modificar el número de iteraciones máximas en el proceso de ajuste de los pesos.

Exper.	Pesos	Max. Iter	N-gramas	Suavizado	Monotonía	BLEU
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08
Iter1	MERT	7	Tri-gramas	Kneser-ney	No	91.64
Iter2	MERT	10	Tri-gramas	Kneser-ney	No	91.81

Table 3: Resultados experimento iteraciones.

Como se puede observar, los valores son muy parejos. Esto no implica una diferencia significativa puesto que, el proceso de ajuste de pesos mediante MERT tiene una serie de parámetros aleatorios que pueden variar en cada ejecución. Sin embargo, se imagina que no existe mejoría al aumentar el número de iteraciones máximo puesto que el experimento se ha realizado con un corpus de tamaño reducido.

3.3 Experimento n-gramas

A continuacion se presentan las puntuaciones de los modelos entrenados con 2, 3, 4 y 5-gramas.

Exper.	Pesos	Max. Iter	N-gramas	Suavizado	Monotonía	BLEU
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08
Gram1	MERT	5	Bi-gramas	Kneser-ney	No	90.48
Gram2	MERT	5	4-gramas	Kneser-ney	No	91.90
Gram3	MERT	5	5-gramas	Kneser-ney	No	91.11

Table 4: Resultados experimentos n-gramas.

Como se puede observar, al crear el modelo con bi-gramas la puntuación decrece. Esto se debe a que con bi-gramas no somos capaces de capturar toda la información que se consigue con los tri-gramas. Los experimentos con 4-gramas y 5-gramas no varían tanto la puntuación, concretamente con 4-gramas se obtiene prácticamente la misma. Sin embargo con los modelos de 5-gramas vuelve a decrecer ligeramente, esto puede ser debido al incremento de la complejidad de estos modelos respecto a los anteriores.

3.4 Opcional: Experimento MIRA

Para el experimento realizado modificando el método de ajuste de pesos podemos observar la comparativa en la siguiente tabla.

Exper.	Pesos	Max. Iter	N-gramas	Suavizado	Monotonía	BLEU
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08
MIRA	MIRA	5	Tri-gramas	Kneser-ney	No	91.81

Table 5: Resultados experimento MIRA.

Como se puede apreciar, los resultados obtenidos haciendo uso de MERT o de MIRA para el ajuste de pesos en este experimento no ha tenido ninguna repercusión determinante en el modelo final obtenido.

3.5 Opcional: Experimento suavizados

Con este experimento se ha tratado de comparar los distintos métodos de suavizado más comunes en el mundo de los modelos de lenguaje. En la siguiente tabla se pueden observar los resultados obtenidos de comparar los métodos de Good-Turing, Kneser-Ney y Witten-Bell.

Exper.	**Pesos**	**Max. Iter**	**N-gramas**	**Suavizado**	**Monotonía**	**BLEU**
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08
suav1	MERT	5	Tri-gramas	Good-Turing	No	90.59
suav2	MERT	5	Tri-gramas	Witten-Bell	No	91.88

Table 6: Resultados experimento suavizados.

En este caso se puede observar que los resultados obtenidos mediante Kneser-Ney y Witten-Bell son muy similares. Sin embargo, al construir los modelos de lenguaje mediante el suavizado más sencillo, Good-Turing, si que se ha visto notablemente disminuido el valor de BLEU asociado al modelo de traducción obtenido en ese caso.

3.6 Opcional: Experimento monótono

Finalmente se ha lanzado el experimento comparando un metodo de alineacion no monótono con uno monótono.

Exper.	**Pesos**	**Max. Iter**	**N-gramas**	**Suavizado**	**Monotonía**	**BLEU**
1	MERT	5	Tri-gramas	Kneser-ney	No	92.08
mono	MERT	5	Tri-gramas	Kneser-ney	Sí	90.11

Table 7: Resultado experimento alineamiento monótono.

Como podemos observar, el hecho de no reordenar los bloques traducidos obtenidos hace bajar notablemente la puntuación BLEU entre los modelos conseguidos. Esto es debido a que entre el inglés y el español, muchas veces varía la ordenación de los elementos en las frases y, en el modelo de alineamiento monótono, esto no se tiene en cuenta.

4 Conclusiones

Mediante esta práctica se ha hecho uso de MOSES, uno de los softwares de traducción automática estadística mejor considerados. Además se ha podido observar el impacto que tienen distintos parámetros en los modelos resultantes.

Se han lanzado gran cantidad de experimentos mediante los cuales también ha sido posible observar las relaciones que puede haber entre distintos parámetros a la hora de hallar un modelo más preciso.

References

[1] Timothy C Bell, John G Cleary, and Ian H Witten. *Text compression*, volume 348. Prentice Hall Englewood Cliffs, NJ, 1990.

[2] Francisco Casacuberta and Enrique Vidal. Giza++: Training of statistical translation models, 2007.

[3] Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics, 2012.

[4] David Chiang, Yuval Marton, and Philip Resnik. Online large-margin training of syntactic and structural translation features. In *Proceedings of the conference on empirical methods in natural language processing*, pages 224–233. Association for Computational Linguistics, 2008.

[5] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(Jan):951–991, 2003.

[6] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3/4):237–264, 1953.

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

[8] 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.

[9] 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.

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