Lingüística Computacional

Algoritmo de Viterbi

22 de marzo de 2020

La desambiguación precisa (i) construir un modelo (hallar las probabilidades de transición y de emisión en una base de datos) y (ii) aplicarlo a las nuevas cadenas de texto. La primera tarea es relativamente fácil. La segunda puede resultar más laboriosa. Por ello recomiendo encarecidamente que se pongan ya con la implementación del algoritmo de Viterbi. El pseudocódigo del algoritmo pueden encontrarlo en Jurafsky, en la Wikipedia en inglés y en infinidad de páginas web. Incluso pueden encontrar implementaciones en Python. Construir su propia función a partir del pseudocódigo es la mejor manera de entender y aprender. Siguen algunas observaciones. Se entenderán mejor si tienen el ejemplo de Jurafsky y el pseudocódigo del algoritmo delante.

1.- A la función hay que pasarle: (i) las probabilidades de transición entre categorías, (ii) las probabilidades de emisión, (iii) la cadena de texto a etiquetar.

Las probabilidades de transición, obtenidas a partir de una base de datos ya etiquetada, por ejemplo con Freeling, pueden estar en un diccionario, por ejemplo: `probabilidades_transicion[('D','N')]=0.003`. Las claves son tuplas de etiquetas y el valor es la probabilidad. En el ejemplo, encontramos que en la base de datos la probabilidad de que a un Determinante le siga un Nombre es de 0.003.

Las probabilidades de emisión, obtenidas a partir de la misma base de datos, pueden estar también en un diccionario. Por ejemplo:

`probabilidades_emision[('D','la')]=0.004`

indicaría que hemos encontrado en la base de datos que un Determinante se realiza como 'la' con la probabilidad 0.004.

La cadena a etiquetar puede ser un string, en cuyo caso hay que segmentarlo en palabras o enviar ya una lista de palabras –digamos `cadena`–.

La función necesita también tener todas las categorías empleadas en el etiquetado –digamos `categorias`–. Se pueden pasar como una lista o extraerla a partir de las claves de los diccionarios anteriores.

2.- A partir de la longitud de las palabras (los datos de observación) y de la longitud de las categorías hay que rellenar dos tablas: (a) una almacenará las probabilidades y (b) otra almacenará los punteros hacia atrás para ir guardando el camino de máxima probabilidad. Esta parte es la más nueva para nostros.

Para almacenar valores numéricos en una tabla (a) creamos un diccionario –digamos `tabla`– por defecto con lambda:0. Para la tabla (b) lo que queremos es guardar el valor de una determinada celda –digamos fila j, columna k–, es

decir una tupa (j,k). Creamos un diccionario por defecto –digamos `back`– con lambda:().

3.- Empezamos rellenando la primera columna: las probabilidades de emisión de la primera palabra para cada una de las categorías. En este caso vamos a simplificar las probabilidades de transición asumiendo que todas las categorías tienen la misma probabilidad de ser iniciales de cadena. El valor será simplemente el de las probabilidades de emisión. Los punteros hacia atrás en este caso serán al inicio.

¿Cómo procedemos en nuestra escritura de código? Supongamos que hemos asignado la longitud de las categorías a la variable `lc`. Podemos iniciar el rellenado de ambos diccionarios así:

```
for j in range(lc):
   tabla[(1,j)]=probabilidades_emision[(categorias[j],cadena[0])]
   back[(1,j)]=(0,j)
```

4.- Ya tenemos los valores de la primera palabra. A partir de aquí procedemos a rellenar todas las restantes celdas. Lo hacemos con dos bucles. El bucle externo recorre todas las palabras de la cadena a partir de la segunda -la primera ya está-. Hacemos un recorrido numérico: si `l` es la longitud de la cadena, nuestro recorrido será por el rango(2,l+1). El bucle interno recorre todas las categorías. Podría ser algo como esto:

```
for j in range(2,l+1):
   for k in range(lc):
```

Ahora tienen que rellenar el valor de cada celda para `tabla` y `back`. Para cada categoría, el valor es el máximo de multiplicar (a) las probabildades de emisión por (b) las probabilidades de transición.

El valor (a) es el que obtenemos en el diccionario y es fijo para categoría y palabra.

El valor (b) hay que computarlo a partir del diccionario de transiciones y de los valores de la celda que corresponde a cada categoría en la situación inmediatamente anterior. Es decir, hay que introducir otro bucle que recorra todas las celdas inmediatamente anteriores (los valores de todas las categorías para la palabra inmediatamente anterior) y que multiplique su valor por el de las transiciones del diccionario. De todos los valores nos quedamos con el máximo. La celda de la que procede el máximo es la que guardamos en `back`.

5.- Una vez finalizado el proceso, la secuencia de categorías más probables se obtiene a partir de las celdas de `back` empezando desde el final.