Artículo DocToVec

En este artículo explico el trabajo realizado durante esta práctica y los resultados obtenidos.

Para hacer esta practica lo primero que se ha hecho ha sido dividir el trabajo a realizar en diferentes sub-tareas. Se decidió dividir el trabajo en tres partes: preparación de ficheros, creación del modelo y testeo. De este modo podemos ejecutar cualquiera de estas tareas por separado sin tener que ejecutar todo continuamente.

Preparación de ficheros

Lo primero ha sido conseguir el fichero en el que están todas las palabras. El fichero “Context(todos)”. Después se ha dividido el fichero en dos partes, una para entrenar el modelo y otra para probarlo después.

Después de dividirlo hemos clasificado todas las palabras de acuerdo a su contexto. Para clasificarlas hemos tenido que buscar cada una de las palabras en el fichero “wei_ili_to_domains”. Además hemos sustituido el código de la palabra que estamos clasificando por la propia palabra. Para hacer esto hemos utilizado el código y lo hemos buscado en el fichero “wei_eng-30_variant”.

De este modo y tras ejecutar el programa procesador.py obtenemos:

ctx_00001740-n
00001740-n#n#wf0#2 be#v#wf3#1
perceive#v#wf4#1 know#v#wf6#1
infer#v#wf8#1 infer#v#wf8#1 infer#v#wf8#1
have#v#wf10#1 own#a#wf12#1
distinct#a#wf13#1 distinct#a#wf13#1
existence#n#wf14#1 living#a#wf16#1

ctx_00001740-n**#factotum**
entity#n#wf0#2 be#v#wf3#1 perceive#v#wf4#1
know#v#wf6#1 infer#v#wf8#1 infer#v#wf8#1
infer#v#wf8#1 have#v#wf10#1 own#a#wf12#1
distinct#a#wf13#1 distinct#a#wf13#1
existence#n#wf14#1 living#a#wf16#1

Para preparar los datos he pasado el procesador sobre los ficheros context(train) y context(test).

Creación del modelo.

La creación del modelo ha sido el tema mas importante del ejercicio realizado. La dificultad estaba en elegir correctamente que valores asignar a cada variable, siendo el resultado muy cambiante en función de esta elección.

Por ejemplo para este ejercicio he creado 8 modelos. Los modelos han variado el numero de epochs, la utilización del label “factotum” el *size* del algoritmo…

Un consejo por si se pretende replicar el ejercicio es tener cuidado con el parámetro size del algoritmo y el número de epochs. Puede retrasar mucho la ejecución del programa.

Testeo

En principio se pensó en utilizar algunos programas que existían para hacer el testeo, pero la obligación de escribirlo todo en un formato detallado, obligó a desarrollar un programa de testeo propio. Esta decisión ahorro mucho tiempo de desarrollo y nos permite hacer cualquier modificación mediante parámetros a la hora de testear el modelo.

El testeo solo compara el label que esperamos recibir con el label que en realidad recibimos. El programa de testeo tiene una modificación que permite comprobar el label esperado con los primeros diez recibidos.

Además el programa de testeo, al igual que el programa de crear el modelo, tiene un parámetro para que las palabras con label “factotum” no sean tomadas en cuenta.

Resultados

En las siguientes tablas se pueden ver los resultados obtenidos a lo largo de las diferentes pruebas que se han hecho para intentar conseguir los mejores resultados. Se puede observar como se han ido ajustando parámetros y como no se ha podido superar el 67% de acierto para los 10 primeros labels y el 26% teniendo en cuenta solo el primer label.

Ademas se ha visto que no sirve poner muchos epochs ya que el resultado también tiende a empeorar.

Nombre	Epochs	Size	Window	Negative	Min Count	Resultados (1)	Resultados (10)
PequeñoSinFactotum	5	200	10	5	2	11 %	39 %
PequeñoConFactotum	5	200	10	5	2	8 %	33 %
v1-MenosSizeyNegative	5	**50**	10	1	2	**25 %**	**67 %**
v2-MasWindow	5	50	**20**	1	2	24 %	66 %
v2-MenosWindow	5	50	**10**	1	2	23 %	66 %
v3-SoloSize	5	**50**	**10**	5	2	18 %	48 %
v4-MasMinCount	5	50	10	1	**20**	19 %	55 %
v5-MasEpochs	**25**	50	10	1	2	**26 %**	**61 %**

Nombre	Epochs	Size	Window	Negative	Min Count	Resultados (1)	Resultados (10)
MuyMuyGrande SinFactotum	150	50	10	1	1	22 %	56 %
MuyMuyGrande ConFactotum	150	50	10	1	1	19 %	47 %