Solemne I – 2da Parte: Inteligencia Artificial

Profesor: ***Alejandro Figueroa***

Ponderación: ***2***

Fecha de entrega: ***viernes 5 de septiembre.***

Ayudante: ***Nicolás Olivares***

Método de entrega: ***mail al ayudante (nicolivares@gmail.com)***

Objetivo

Cada alumno, utilizando sus datos etiquetados en la tarea anterior, aprenderá varios modelos de predicción basado en aprendizaje supervisado. Cada modelo permitirá predecir las etiquetas de tripletas nuevas <título, contenido, respuestas> que aparezcan en Yahoo! Answers, ergo que no tengan sus etiquetas manuales.

Para realizar aprendizaje supervisado se necesitan cuatro componentes: un espacio vectorial, una clase de modelo, una metodología experimental y una métrica. En esta tarea, utilizaremos SVM[1] Light como modelos base y F-Score como métrica de desempeño.

Feature Vector

Exploraremos varios espacios vectoriales, en los cuales estudiaremos los diferentes aspectos observados en la tarea anterior. Tomaremos en cuenta cuatro grandes grupos de features:

1. Los que corresponden a la palabra que está siendo clasificada. Acá consideraremos un feature binario que indica si la palabra tiene o no una raíz morfológica detectada por Montylingua. Por ejemplo, la palabra “houses” tiene como raíz “house”, por ende, en ese caso el valor sería “1”. Otro feature binario a nivel de palabra es si esta está escrita completamente en mayúsculas, y obviamente tenemos el similar si está escrita completamente en minúsculas. Otro feature binario indica si la palabra comienza con una letra mayúscula y el resto está escrito en minúscula. Otro feature binario indica si la categoría sintáctica de la palabra es o no “NN” (sustantivo propio). Este último se puede homologar para cada una de las categorías sintáctica que entrega MontyLingua. Finalmente, considerar la palabra que se está analizando, y su largo en carácteres. Consideremos el siguiente ejemplo: “ I think you have bought the cover story this is propaganda , absolutely opposite the truth , the man was trying to rebuild the western alliance in the threat of rising german aggression .” Para la palabra “I”, tendriamos:

[1] http://www.cs.cornell.edu/People/tj/svm_light/
http://svmlight.joachims.org/

TIENE_RAIZ=0

FULL_MAYUSCULAS=1

FULL_MINUSCULAS=0

INICIO_MAYUSCULAS_RESTO_MINUSCULAS=0

CAT_SINTACTICA_PRP=1

PALABRA_I=1

PALABRA_LARGO=1

Por otro lado para la palabra "was", tendríamos:

TIENE_RAIZ=1
FULL_MAYUSCULAS=0

FULL_MINUSCULAS=1

INICIO_MAYUSCULAS_RESTO_MINUSCULAS=0

CAT_SINTACTICA_VBZ=1

PALABRA_was=1

PALABRA_LARGO=3

2. En el segundo grupo de features, modelaremos el contexto de la palabra. Como contexto asumiremos tres palabras a la izquierda, y tres palabras hacia la derecha. Por ejemplo, para la palabra "was" que analizamos anteriormente, tenemos " **, the man was trying to rebuild**". Modelaremos el contexto izquierdo y derecho por separado y de diferentes formas. La primera es mediante las palabras y su frecuencia en el contexto, después modelaremos las categorías sintácticas de las palabras del contexto. Para la palabra "was" tenemos:

IZQUIERDA_,=1

IZQUIERDA_the=1

IZQUIERDA_man=1

DERECHA_trying=1

DERECHA_to=1

DERECHA_rebuild=1

IZQUIERDA_CAT_PUNCT=1

IZQUIERDA_CAT_NN=1

IZQUIERDA_CAT_DET=1

DERECHA_CAT_VBG=1

DERECHA_CAT_TO=1

DERECHA_CAT_VB=1

Para el caso de la palabra "I", tenemos:

DERECHA_think=1

DERECHA_you=1

DERECHA_have=1

DERECHA_CAT_VB=2

DERECHA_CAT_PRN=1

3. En este tercer grupo, modelaremos el contexto global de una palabra. En una página de respuestas, es decir una tripleta <título, contenido, respuestas>, una palabra aparece muchas veces. Uno podría entonces pensar que el contexto de todas sus apariciones es útil para discernir si la palabra es o no una entidad. Para modelar el contexto global, lo haremos de manera similar al punto dos, pero considerando todas las instancias de la palabra en la página. Consideremos el caso de "was":

"*the church , **was** it treason or* "
"womaniser . WWII **was** caused by Hitler "
", *the man* **was** trying to rebuild"

Siguiendo el mismo procesamiento que en el paso dos, obtenemos los siguientes features:

GLOBAL_IZQUIERDA_,=2

GLOBAL_IZQUIERDA_the=2

GLOBAL_IZQUIERDA_man=1

GLOBAL_IZQUIERDA_church=1

GLOBAL_IZQUIERDA_womaniser=1

GLOBAL_IZQUIERDA_WWII=1

GLOBAL_IZQUIERDA_.=1

GLOBAL_DERECHA_trying=1

GLOBAL_DERECHA_to=1

GLOBAL_DERECHA_rebuild=1

GLOBAL_DERECHA_caused=1

GLOBAL_DERECHA_by=1

GLOBAL_DERECHA_it=1

GLOBAL_DERECHA_treason=1

GLOBAL_DERECHA_or=1

GLOBAL_DERECHA_Hitler=1

GLOBAL_IZQUIERDA_CAT_PUNCT=3

GLOBAL_IZQUIERDA_CAT_NN=3

GLOBAL_IZQUIERDA_CAT_DET=2

GLOBAL_IZQUIERDA_CAT_NNP=1

GLOBAL_DERECHA_CAT_PRP=1

GLOBAL_DERECHA_CAT_NN=1

GLOBAL_DERECHA_CAT_CC=1

GLOBAL_DERECHA_CAT_VBD=1

GLOBAL_DERECHA_CAT_PREP=1

GLOBAL_DERECHA_CAT_NNP=1

GLOBAL_DERECHA_CAT_VBG=1

GLOBAL_DERECHA_CAT_TO=1

GLOBAL_DERECHA_CAT_VB=1

4. El cuatro grupo de features considera que sabemos las etiquetas de palabra anterior, por ejemplo en el caso de “was”, sabemos que la palabra anterior es “,”, es decir un token. Considere un feature binario que diga si la palabra anterior es o no una entidad. Para el caso de “was” tendríamos:

CLASE_PALABRA_PREVIA=0

Para su análisis, piense si es posible considerar la clase de la palabra posterior, y que problema traería para un modelo considerar la clase de ambas palabras.

Para cada uno de los cuatro grupos de features, crear un espacio vectorial diferent. Un espacio vectorial se construye mapeando los features a un ID numérico, tome por ejemplo “GLOBAL_IZQUIERDA_WWII” puede ser mapeado a una ID única 1456. Para determinar las ids, se debe recolectar todos los features a lo largo de la colección. En otros términos, debemos recolectar los features para todas las palabras de la colección. Tomemos el primer ejemplo ilustrativo, supongamos que “was” y “I” fueren todas las palabras de nuestra colección. La lista global de features sería:

IZQUIERDA_, IZQUIERDA_the IZQUIERDA_man DERECHA_trying DERECHA_to DERECHA_rebuild IZQUIERDA_CAT_PUNCT IZQUIERDA_CAT_NN IZQUIERDA_CAT_DET DERECHA_CAT_VBG DERECHA_CAT_TO DERECHA_CAT_VB DERECHA_think DERECHA_you DERECHA_have DERECHA_CAT_PRN

Acá feature de esta lista global, se le asigna una ID numérica de 1 hasta “j” (la total de features). Supongamos el caso anterior, podemos arbitrariamente tener:

IZQUIERDA_, 16 IZQUIERDA_the 1 IZQUIERDA_man 3 DERECHA_trying 5 DERECHA_to 7 DERECHA_rebuild 8 IZQUIERDA_CAT_PUNCT 10 IZQUIERDA_CAT_NN 12 IZQUIERDA_CAT_DET 14 DERECHA_CAT_VBG 2 DERECHA_CAT_TO 4 DERECHA_CAT_VB 9 DERECHA_think 11 DERECHA_you 6 DERECHA_have 13 DERECHA_CAT_PRN 15

Finalmente, con estas IDs únicas mapeamos los features a un vector "matemáticamente correcto". Para el caso de la palabra "was":

16:1 1:1 3:1 5:1 7:1 8:1 10:1 12:1 14:1 2:1 4:1 9:1

Para el caso de la palabra "I", tenemos:

11:1 6:1 13:1 9:2 15:1

Ya tenemos los vectores. Sólo dos detalles para tenerlos listos para el formato SVM_LIGHT, el primero que debe ir un número al comienzo indicando la clase de la palabra. Para el caso de "token" usemos el "-1", y para el caso de "named entities" usemos el "+1". Finalmente, al final de la línea podemos agregar comentarios con un "#", es conveniente poner cuatro cosas detrás del comentario: 1) la qid de la pregunta de dónde se sacó la palabra, la palabra, la categoría, y finalmente repetir la categoría. Por ende, nuestros dos vectores ilustrativos quedan:

1 16:1 1:1 3:1 5:1 7:1 8:1 10:1 12:1 14:1 2:1 4:1 9:1 # qid was 1 cat

1 11:1 6:1 13:1 9:2 15:1 #qid I 1 cat

En resumen, debemos tener cuatro archivos de vectores, uno para cada grupo de features.

Cross-validation (Para cada uno de los tres espacios vectoriales por separado)

Una vez obtenidos los vectores hay que dividir los datos en 10 "splits" disjuntos de igual tamaño de acuerdo a sus qids. Es decir, todas las palabras que vienen de una misma qid deben estar en el mismo split. Para hacer está división hay que ir por cada una de las 26 categorías, y aleatoriamente escoger el 10% de las qids y ponerlas en uno de los splits. Los que ya fueron asignados, no son considerados para los próximos splits. Toda qid de una categoría debe estar asignada a un split, y por ende todos sus vectores respectivos pertenecen al split asignado a la qid.

El resultado de este proceso son 10 splits, cada uno contiene el 10% de los datos de cada categoría. Llamemos a cada split S_i, con i=1...10. Por ejemplo, si la categoría deporte tiene 30 qids, cada split va a tener los vectores correspondientes a 3 qids de deporte. Ninguna qid puede estar en dos splits, y toda qid debe estar en uno. Lo mismo se hace para las otras 25 categorías. Finalmente, cada split va a tener los vectores correspondientes a 3*26 qids.

El siguiente paso es generar los datos de entrenamiento. Se generan 10 conjuntos de entrenamiento mediante la unión sistemática de los split. Hágalo de la siguiente forma:

$E_1 = S_2+S_3+S_4+S_5+S_6+S_7+S_8+S_9+S_{10}$

$E_2 = S_1+S_3+S_4+S_5+S_6+S_7+S_8+S_9+S_{10}$

$E_3 = S_1+S_2+S_4+S_5+S_6+S_7+S_8+S_9+S_{10}$

$E_4 = S_1+S_2+S_3+S_5+S_6+S_7+S_8+S_9+S_{10}$

$E_5 = S_1+S_2+S_3+S_4+S_6+S_7+S_8+S_9+S_{10}$

$E_6 = S_1+S_2+S_3+S_4+S_5+S_7+S_8+S_9+S_{10}$

$E_7 = S_1+S_2+S_3+S_4+S_5+S_6+S_8+S_9+S_{10}$

$E_8 = S_1+S_2+S_3+S_4+S_5+S_6+S_7+S_9+S_{10}$

$E_9 = S_1+S_2+S_3+S_4+S_5+S_6+S_7+S_8+S_{10}$

$E_{10} = S_1+S_2+S_3+S_4+S_5+S_6+S_7+S_9+S_9$

Una vez construidos los conjuntos de entrenamiento, se debe generar los modelos. Para ésto, se debe utilizar la siguiente línea de comando svm_multiclass:

```
svm_learn -c 1.0 E_i M_i
```

Se genera un modelo (M_i) por cada uno de los E_i. Para evaluar el modelo debe ejecutarse sobre el dato que no fue considerado en el respectivo E_i, es decir S_i. Las predicciones svm_multiclass las hace mediante la siguiente línea de comando:

```
svm_classify S_i M_i Resultados_i
```

Cuando se tengan los diez archivos de resultados, se debe calcular el desempeño del clasificador SVM para predecir si una palabra es o no una entidad. Con este objetivo, examine los archivos de resultados y compare las etiquetas asignadas manualmente y las que entrega el clasificador. Calcule la accuracy y la matriz de confusión. Además, la precisión, recall y F-Score de ambas clases. Comente acerca de los errores ¿Hay algún patrón? ¿Cómo podría mejorar? Además, grafique la curva ROC para cada uno de los cuatro casos. Si tuviera que combinar dos modelos ¿Por qué pareja optaría? Fundamente de acuerdo a los resultados obtenidos. Calcule la pureza de los clústeres obtenidos

¿Qué sucede cuando una entidad en la pregunta coincide con una en la respuesta? ¿Suelen ser buenas respuestas? ¿Alguna de las 26 categorías es más susceptible a tener entidades? ¿Qué se le ocurre para mejorar sus modelos? Observa algún patrón lingüístico que pudiera explotarse para mejorar el reconocimiento. Es importante que en sus conclusiones considere qué sucede para el caso del español. En esta tarea utilizamos dos modelos de contexto y uno de palabra. ¿Qué pasa con el español y los modelos de la tarea? También considere en sus conclusiones aplicaciones empresariales. Por ejemplo, en las glosas de las transacciones bancarias de una

cartola de cuenta corriente ¿Qué variaciones habría que hacer para reconocer entidades? ¿Qué problema vería con los modelos de contexto presentados en esta tarea? Y en el caso de libros escaneados para una biblioteca ¿Qué problemas avizora que encontraría?

Otro aspecto que es siempre conveniente analizar es el parámetro "C" que utiliza SVM. Por defecto este se usa como "1", pero qué sucede si lo cambiamos sistemáticamente. Para el mejor modelo que ha obtenido, muestre un gráfico del valor de "C" versus el F-Score obtenido.

Preguntas adicionales: ¿Cree que combinando las salidas de los diferentes modelos se podría mejorar los resultados? Si su respuesta es afirmativa, ¿Cómo debería efectuarse esa combinación?, de lo contrario ¿Qué problemas ve?

Además, tenga en cuenta que:

Cada estudiante debe trabajar sobre su propio conjunto de datos. El utilizar los datos de otro compañero automáticamente le hace acreedor de la nota 1 en la tarea, y deberá tener sus propios datos etiquetados para las partes posteriores. Cada estudiante debe solicitar su conjunto personal de datos al ayudante.