Hashing y LSH 2020-2C Contamos con una colección de 1300 millones de tweets con fake news. Queremos usarlospara detectar otros tweets que puedan contener mensajes parecidos para poder filtrarlos.El objetivo final de nuestra aplicación es dado un tweet buscar cuáles son tweets parecidos ennuestra base de 1300 millones de tweets usando LSH para luego decidir si el tweet es o no unfake news. Si hay tweets muy similares al nuevo en nuestra base de datos, lo categorizamoscomo fake news; caso contrario, no. El criterio que fijamos es que si el tweet actual tiene unasemejanza mayor o igual a 0.73 con alguno de nuestra base de fake news entonces es un fakenews.Se decide usar la distancia de Jaccard como métrica para la construcción de nuestro esquemade LSH.Queremos que si la semejanza entre tweets es mayor o igual a 0.55 tengamos más de un 70%de probabilidad de que sean candidatos. Por otro lado, si la semejanza es menor o igual a 0.05queremos tener menos de un 1% de probabilidad de que sean candidatos a ser comparados. En base a esta información le pedimos que responda las siguientes preguntas:

1.¿Cuántos minhash hacen falta y con qué tipo de esquema (b y r) ? Detalle los cálculosrealizados y estime cantidad de falsos positivos y falsos negativos que vamos a tenersobre una base de 10000 tweets nuevos. (30 puntos

El LSH para la distancia de Jaccard se define como H(d1,d2,p1,p2). La distancia de los datos se define como d(a,b) = 1 - semejanza(a,b). La probabilidad de un positivo real en un minhash particular es de 1-d1, y la de positivo falso es 1-d2. Se quiere que entonces que la probabilidad de haya una colision con semejanza mayor o igual a 0.55 sea mayor a 0.7, y que la probabilidad de que haya una colision cuando la semejanza es menor a 0.05 sea menor a 0.01. Se usará el esquema de AND seguido por OR. Puede verse entonces que:

d1 = 1-0.55 = 0.45
d2 = 1-0.05 = 0.95
p1 = 1-(1-(1-d1)^r)^b = 1-(1-0.55^r)^b = p1(r,b) > 0.7
p2 = 1-(1-(1-d2)^r)^b = 1-(1-0.05^r)^b = p2(r,b) < 0.01

Se buscan r y b:

p1(1,2) = 0.79 bien | p2(1,2) = 0.098 mal
p1(2,2) = 0.51 mal | p2(2,2) = 0.005 bien
p1(2,4) = 0.76 bien | p2(2,4) = 0.00996 bien <- r = 2, b = 4

Se tiene entonces que para que se cumplan las probabilidades pedidas se debe tener r=2 y b=4, teniendo así un total de 8 minhashes.
La probabilidad de un falso negativo es 1-p1=1-0.76=0.24
La probabilidad de un falso positivo es p2=0.00996
Sobre 10000 tweets nuevos se tendrán entonces:
10000 (producto) 0.24 = 2400 falsos negativos aproximadamente
10000 (producto) 0.00996 =(aprox)100 falsos positivos aproximadamente

El (producto) utilizado para el cálculo de estimación de falsos positivos y negativos se debe a que el asterisco es un caracter delimitador de región en la que aplicar italics, por lo que no se ve.

2.Describa la etapa de pre-procesamiento en la cual tiene que recorrer los 1300 millonesde tweets y crear una única tabla de hash. Recuerde que puede usar diagramas,esquemas y pseudocódigo para hacer más clara su explicación. Debe ser claroindicando cómo queda conformado el esquema LSH que va a usar en el punto 3. (35puntos)

Para poder comparar algún tweet con los que ya se saben que son falsos se deben tener estos guardados en una tabla de hash, es decir, se deben guardar en una tabla que cumpla los requerimientos conseguidos en el punto 1. Esto implica que se deben tener 4 grupos de 2 minhashes cada uno, y un hash universal para claves de tamaño fijo para poder obtener la posicion final en la que se guardara el tweet (para cada grupo). Se tiene entonces que el proceso de guardado de un tweet en la tabla es el siguiente:

Para guardar un tweet en la tabla se lo debe otorgar para que esta realice los cálculos necesarios para poder indicar los lugares en los que este debe ser almacenado. Una vez recibido un tweet el proceso es el siguiente:

Se separa el tweet en shingles de alguna cantidad arbitraria de caracteres (Ej: un tweet "Hola" en 2-shinles se separa como {H, Ho, ol, la, a}), luego se hashea para ese tweet cada uno de esos shingles con un hash universal para claves de tamaño fijo, y se toma como valor final de ese minhash el menor de los hashes generados, esto se realiza para cada minhash del grupo (es decir, 2). Luego se toman los valores generados por los minhashes del grupo y se rehashean nuevamente con una función universal para claves de tamaño fijo (las de tipo (a.x[0]+b.x[1] + ... +cte.x[n-1])mod p mod m), lo cual otorga la posición de la tabla en la que se guardará el tweet.

Este proceso se repite luego para cada grupo de minhashes, es decir, se realiza un total de 4 veces, generando así 4 posiciones de la tabla, y guardándolo en (como mucho, asumiendo que no colisionan entre sí los hashes finales) 4 posiciones distintas.

Una vez realizado esto se guardó correctamente el tweet en la tabla de hash. Para terminar el preprocesamiento se debe repetir el proceso para cada tweet con el que se quieran realizar luego comparaciones.

Si se ve este proceso para un tweet x se tiene por ejemplo:

b1: h(h1(x),h2(x)) = 0
b2: h(h3(x),h4(x)) = 3
b3: h(h5(x),h6(x)) = 5
b4: h(h7(x),h8(x)) = 0
Donde hi(x) y h(x) son de la forma ya dada de hash universal para claves de tamaño fijo.

La tabla insertando este tweet quedará como:

0 |x|
1 | |
2 | |
3 |x|
4 | |
5 |x|

Si se inserta otro tweet y:

b1: h(h1(x),h2(x)) = 3
b2: h(h3(x),h4(x)) = 1
b3: h(h5(x),h6(x)) = 4
b4: h(h7(x),h8(x)) = 5

La tabla insertando este tweet quedará como:

0 |x|
1 |y|
2 | |
3 |x, y|
4 |y|
5 |x, y|

Este proceso se repetirá para cada tweet insertado

El formato de negrita aplicado en el ejemplo de generación de shingles se debe a que el símbolo de pesos genera dicho formato, y se utilizó ese caracter para completar el espacio "nulo" del primer y último

3.Describir cómo se hace la predicción de si un tweet es fake news o no utilizando el esquema LSH propuesto. (35 puntos

Para saber si un tweet es falso se debe realizar el mismo proceso que para el pre-procesamiento de los tweets con los que se va a comparar. Generará b posiciones de la tabla distintas, que fueron generadas cada una gracias al hasheo de los resultados de r minhashes distintos. Puede verse entonces que se generarán 4 posiciones de la tabla, y se retornará la unión de todos los tweets que se encuentran en los buckets de esas posiciones.

Una vez obtenidas estas colisiones, se decidirá si el tweet es falso o no. Como la tabla retorna los tweets falsos a los que el tweet recibido se parece, lo más probable es que si retorna 1 o más tweets entonces el recibido sea también falso, y en caso de no retornar nada lo más probable es que no sea falso.

Un ejemplo de ejecución de esto sería:

Tabla:
0 |a, b|
1 |a, c|
2 |c|
3 | |
4 |b|
5 |d|

Si se recibe un tweet x tal que:
b1: h(h1(x),h2(x)) = 2
b2: h(h3(x),h4(x)) = 1
b3: h(h5(x),h6(x)) = 3
b4: h(h7(x),h8(x)) = 5

Se retornará como similares el conjunto {c} U {a, c} U {} U {d} = {a, c, d}
Como es similar a tweets falsos se considera que x es falso. Si el conjunto retornado estuviera vacío entonces sería considerado no falso.

In [ ]: