INFORMATION RETRIEVAL : ASSIGNMENT 1

ROLL : MDS201803

NAME : SUBHASISH BASAK

→ Movie Name 1 : When Harry met Sally.

After Case folding & Stop word removal : 'harry', 'met', 'sally'

Movie Name 2 : Harry Potter and the cursed child

After Case folding & Stop word removal : 'harry', 'potter', 'curse', 'child'.

Movie Name 3 : The lost child, Sally.

After Case folding & stop word removal : 'lost', 'child', 'sally'

Common Words : 'harry', 'child', 'sally'

3-gram non positional inverted indices :

$ha	harry		$ch	child
har	harry		chi	child
arr	harry		hil	child
rry	harry		ild	child
ry$	harry		ld$	child
$me	met		$lo	lost
met	met		los	lost
et$	met		ost	lost
$sa	sally		st$	lost
sal	sally			
all	sally			
lly	sally			
ly$	sally			
$po	potter			
pot	potter			
ott	potter			
tte	potter			
ter	potter			
er$	potter			
$cu	curse			
cur	curse			
urs	curse			
rse	curse			
se$	curse			

Term-document inverted index:

We have the following documents:

Doc 1 : 'harry', 'met', 'sally'.
Doc 2 : 'harry', 'potter', 'curse', 'child'.
Doc 3 : 'lost', 'child', 'sally'

Now the term-doc inverted index is as follows:

WORD	FREQUENCY	POSTING LIST
harry	2	1, 2
met	1	1
sally	2	1, 3
potter	1	2
curse	1	2
child	2	2, 3
lost	1	3

Permuterm non positional inverted index.

harry$, arry$h, rryha, ryhar, y$harr, $harry → harry

met$, et$m, t$me, $met → met

sally$, ally$s, llysa, lysal, y$sall, $sally → sally

potter$, otter$p, tterpo, terpot, er$pott, r$potte, $potter → potter

curse$, urse$c, rsecu, secur, e$curs, $curse → curse

child$, hild$c, ildch, ldchi, d$chil, $child → child

lost$, ost$l, stlo, tlos, $lost → lost

Wild card Query 1 : ha*ry Wild card Query 2 : los*

For Query 1 we scan through the 3-gram index list and search for $ha and ry$ and take their intersection.
and for permuterm index we add an end marker to the query and rotate till the * comes to the end, then search it ry$ha as a prefix in the non positional permuterm inverted index.

For Query 2 we scan the 3-gram index for $lo and los and take their intersection.

and for permuterm index we use an end marker and search for $los in the permuterm non positional inverted index.

In the above queries we take the resulting term and search for the document in the original term-document posting list,

while scanning through the permuterm index there are 40 indices to scan. but in 3-gram indices there are 33 indices.

So, 3-grams takes less space compared to permuterm.