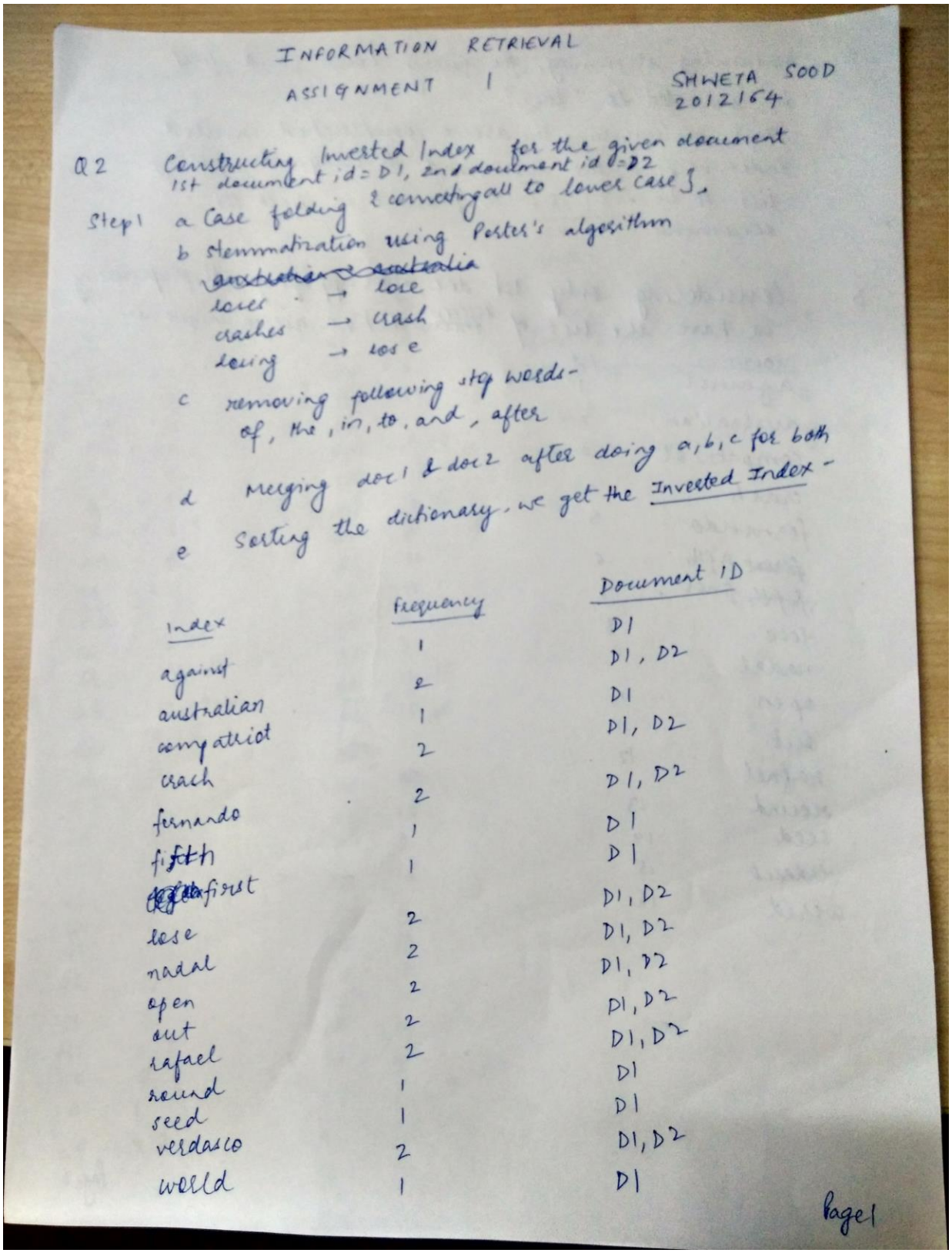
INFORMATION RETRIEVAL ASSIGNMENT 1

SHWETA SOOD
2012164

Q2

Doc 1 as D1: “World’s fifth seed Rafael Nadal crashes out of the Australian Open after losing in the first round against compatriot Fernando Verdasco.”

Doc 2 as D2: “Rafael Nadal loses to Fernando Verdasco and crashes out of the Australian Open”



INFORMATION RETRIEVAL ASSIGNMENT 1

SHWETA SOOD
2012164

Q2 Constructing Inverted Index for the given document
1st document id = D1, 2nd document id = D2

Step 1 a Case folding (converting all to lower case).

b stemmatization using Porter's algorithm

~~australian~~ → ~~australia~~
loser → lose
crashes → crash
losing → lose

c removing following stop words-
of, the, in, to, and, after

d merging doc1 & doc2 after doing a, b, c for both

e Sorting the dictionary, we get the Inverted Index -

Index	Frequency	Document ID
against	1	D1
australian	2	D1, D2
compatriot	1	D1
crash	2	D1, D2
fernando	2	D1, D2
~~fifth~~	1	D1
~~first~~	1	D1
lose	2	D1, D2
nadal	2	D1, D2
open	2	D1, D2
out	2	D1, D2
rafael	2	D1, D2
round	1	D1
seed	1	D1
verdasco	2	D1, D2
world	1	D1

Page1

2 Assuming stemming, for query "Loses", will first be converted to "lose".
Next, on searching the above constructed inverted index, we find, the term and its corresponding posting list to be - D1, D2 ⇒ present in both the documents

3 Considering only 1st doc, after applying all preprocessing we have the list of ~~terms~~ terms that we assign unique ids-

term	id
against	1
australian	2
compatriot	3
crash	4
fernando	5
~~first~~ fifth	6
~~fifth~~ first	7
lose	8
nadal	9
open	10
out	11
rafael	12
round	13
seed	14
verdasco	15
world	16

Bigram for doc 1

Bigram	ID of words
$a	1,2
$c	3,4
$f	5,6,7
$d	8
$n	9
$o	10,11
$r	12,13
$s	14
$v	15
$w	16
ad	9
ae	12
af	12
ag	1
ai	1
al	2,9
an	2,5
ar	4,15
at	3
au	2
co	3,15
cr	4
d$	13,14,16
da	9,15
do	5
e$	8
ed	14
ee	14
el	12
en	10
er	5,15
fa	12
fe	5
fi	6,7
ft	6
ga	1
h$	4,6
ia	2
if	6
in	1
io	3
ir	7
l$	9,12
ld	16
li	2
lo	8
mp	3
n$	2
na	5
nd	5,13
ns	1
o$	5,15
om	3
op	10
or	16
os	8
ot	3
ou	11,13
pa	3
pe	10
ra	2,14,12
rd	15
ri	3
rl	16
rn	5
ro	13
rs	7
sc	15
se	8,14
sh	4
st	1,2,7
t$	1,3,7,11
th	6
tr	2,3
un	13
us	2
ut	11
ve	15
wo	16

The index here, points to id of words assigned, which in turn point to doc ids { 2 level structure }

Bigram for only doc 2, after applying all preprocessing, we have the list of terms that we assign unique ids—

term	id
australian	1
crash	2
fernando	3
ice	4
nadal	5
open	6
out	7
rafael	8
verdasco	9

Bigram for doc 2

$a	1
$c	2
$f	3
$l	4
$n	5
$o	6,7
$r	8
$v	9
ad	5
ae	5
af	8
al	1,5
an	1,3
as	2,9
au	1
co	9
cr	2
da	5,9
do	3
e$	4
el	8
en	6
er	3,9
fa	8
fe	3
h$	2
ia	1
l$	5,8
li	1
lo	4
n$	1
na	3
nd	3
o$	3,9
op	6
ps	4
ou	7
pe	6
ra	1,2,8
rd	9
rn	3
sc	9
se	4
sh	4
st	1
t$	7
tr	1
us	1
ut	7
ve	9

Consolidated bigram inverted index for both docs = bigram for doc1

This is because, after preprocessing, all terms in doc2 are already in doc1.

∴ Consolidated bigram inverted index is already drawn {given for doc1}

Answering queries

I "Rafeal" → (preprocessed) "rafeal"

r, ra, af, fe, ea, al, l
↓ ↓ ↓ ↓ ↓ ↓ ↓
1,2,3 2,4,12 12 5 NULL (not found) 2,9 9,12

frequency of 12 is the most, which points to the word rafael & hence answer of the query is Doc1 & Doc2 {by looking at inverted index}
∴ Inspite of spelling error, we could get the right answer.

II "Australia" → (preprocessed) "australia"

$a, au, us, st, tr, ra, al, li, ia, a$
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
1,2 2 2 1,2,7 2,3 2,4,12 2,9 2 2 NULL/ not found

The most frequent id is 2 ∋ pointing to word australian & hence answer of query by looking at inverted index is Doc1 & Doc2.