DATA603 – HW2: MapReduce (Harry Potter)

Nick Devroye

DOB used in code: **November 11, 2002**

Overview

This submission follows the HW2 prompt precisely:

- **DOB → Book/Pages:**
  Month 11 ⇒ $\lceil 11/2 \rceil$ = **Book 6** (Half-Blood Prince).
  Day 11 ⇒ **pages 11–20** (`file1.txt`).
  Year 2002 ⇒ "102" ⇒ **pages 102–111** (`file2.txt`).
- **MapReduce word count:** Applied to `file1.txt`.
- **Non-English detection:** Applied to `file2.txt` using `pyspellchecker`.
- **Deliverables here:** Code listing and result tables imported from CSVs.

How to Reproduce

Run the script (Python 3) in the same directory as Book 6 PDF:

```
pip install PyPDF2 pyspellchecker
python HP_MapReduce.py
```

This will write: `file1.txt`, `file2.txt`, and the CSVs used below.

1 Code

The full Python used is shown here for grading transparency.

HP_MapReduce.py

```
"""
DATA603 - HW2: MapReduce (Harry Potter)
DOB: November 11, 2002

Steps:
  1) DOB => Book 6 (Half-Blood Prince), pages 11-20 and 102-111
  2) Extract those pages into file1.txt and file2.txt
  3) MapReduce word count on file1
  4) pyspellchecker to detect non-English tokens in file2
  5) Print results and save CSVs (for LaTeX inclusion)
"""

import os
import re
from collections import Counter, defaultdict
from multiprocessing import Pool, cpu_count
from spellchecker import SpellChecker
import PyPDF2
import math

#  Configuration
# Path to Half-Blood Prince PDF
PDF_PATH = "half-blood-prince.pdf"
# Offset between printed pages and PDF internal index
OFFSET = 0
# Output folder
OUTDIR = "./out"

#  DOB => Book and Page Ranges
BIRTH_MONTH = 11
BIRTH_DAY = 11
BIRTH_YEAR = 2002

BOOK_NUMBER = math.ceil(BIRTH_MONTH / 2) if BIRTH_MONTH >= 8 else BIRTH_MONTH  # => 6
FILE1_START = BIRTH_DAY                 # => page 11
FILE2_START = int("1" + str(BIRTH_YEAR)[-2:])  # => page 102
SPAN = 10                               # always 10 pages
```

```
38
39 #  PDF Extraction
40 def extract_pages_to_text(pdf_path, start_printed, span, offset):
41     reader = PyPDF2.PdfReader(pdf_path)
42     texts = []
43     for p in range(start_printed, start_printed + span):
44         idx = (p - 1) + offset
45         page = reader.pages[idx]
46         texts.append(page.extract_text() or "")
47     return "\n".join(texts)
48
49 #  MapReduce Word Count
50 WORD_RE = re.compile(r"[A-Za-z']+")
51
52 def tokenize(text):
53     return [w for w in WORD_RE.findall(text)]
54
55 def mapper(lines):
56     out = []
57     for line in lines:
58         for tok in tokenize(line):
59             w = tok.lower().strip("'")
60             if w and w not in ("'", "'"):
61                 out.append((w, 1))
62     return out
63
64 def chunkify(lst, n):
65     k = max(1, len(lst) // n) if lst else 1
66     for i in range(0, len(lst), k):
67         yield lst[i:i+k]
68
69 def shuffle(mapped_items):
70     grouped = defaultdict(int)
71     for k, v in mapped_items:
72         grouped[k] += v
73     return grouped
74
75 def mapreduce_wordcount_text(text, processes=None) -> Counter:
```

```
    lines = text.splitlines()
    nprocs = processes or max(1, min(cpu_count(), 8))
    chunks = list(chunkify(lines, nprocs))
    with Pool(processes=nprocs) as pool:
        mapped_lists = pool.map(mapper, chunks)
    combined = []
    for lst in mapped_lists:
        combined.extend(lst)
    grouped = shuffle(combined)
    return Counter(grouped)

#  Preprocess for Spellchecker
def preprocess_for_spellcheck(s):
    s = s.replace("’", "'").replace("‘", "'").replace("—", "-").replace("–", "-")
    s = re.sub(r"[^A-Za-z']+", " ", s)
    s = re.sub(r"(?<=[a-z])(?=[A-Z])", " ", s)
    tokens = s.split()
    cleaned = [t for t in tokens if not (len(t) >= 3 and t.isupper())]
    return " ".join(cleaned)

#  Non-English Detection
spell = SpellChecker()

def detect_non_english_with_spellchecker(text, min_count=2) -> Counter:
    wc = mapreduce_wordcount_text(text)
    counts = Counter()
    for tok, c in wc.items():
        if c >= min_count and tok not in spell:
            counts[tok] = c
    return counts

#  Main
def main():
    os.makedirs(OUTDIR, exist_ok=True)

    print(f"[DOB] 11/11/2002 => Book {BOOK_NUMBER} (Half-Blood Prince)")
    print(f"[Pages] file1: {FILE1_START}-{FILE1_START+SPAN-1} | file2: {FILE2_START
    }-{FILE2_START+SPAN-1}")
```

```
    # Extract pages and save file1.txt + file2.txt
    text1 = extract_pages_to_text(PDF_PATH, FILE1_START, SPAN, OFFSET)
    text2_raw = extract_pages_to_text(PDF_PATH, FILE2_START, SPAN, OFFSET)

    file1_path = os.path.join(OUTDIR, "file1.txt")
    file2_path = os.path.join(OUTDIR, "file2.txt")
    with open(file1_path, "w", encoding="utf-8") as f:
        f.write(text1)
    with open(file2_path, "w", encoding="utf-8") as f:
        f.write(text2_raw)

    # MapReduce wordcount on file1
    wc1 = mapreduce_wordcount_text(text1)
    top_words = wc1.most_common(40)
    all_words = wc1.most_common()

    # Non-English tokens on file2
    text2 = preprocess_for_spellcheck(text2_raw)
    noneng = detect_non_english_with_spellchecker(text2, min_count=2)

    # Print results
    print("\n=== Word Count (Top 40) — file1.txt ===")
    for w, c in top_words:
        print(f"{w:20s} {c:5d}")

    print("\n=== Non-English Tokens (pyspellchecker) — file2.txt ===")
    for t, c in noneng.most_common():
        print(f"{t:20s} {c:5d}")

    # Save CSVs for LaTeX
    wc_csv = os.path.join(OUTDIR, "file1_wordcount_top40.csv")
    wc_csv_all = os.path.join(OUTDIR, "file1_wordcount_all.csv")
    ne_csv = os.path.join(OUTDIR, "file2_nonenglish_pyspellchecker.csv")
    with open(wc_csv, "w", encoding="utf-8") as f:
        f.write("word,count\n")
        for w, c in top_words:
            f.write(f"{w},{c}\n")
```

```
    with open(wc_csv_all, "w", encoding="utf-8") as f:
        f.write("word,count\n")
        for w, c in all_words:
            f.write(f"{w},{c}\n")
    with open(ne_csv, "w", encoding="utf-8") as f:
        f.write("token,count\n")
        for t, c in noneng.most_common():
            f.write(f"{t},{c}\n")

if __name__ == "__main__":
    main()
```

2 Results

CSV outputs are read directly for accurate, reproducible tables.

2.1 Word Count (Top 40) from `file1.txt`

Word	Count
the	166
a	84
and	58
of	57
to	49
he	46
in	40
i	39
said	36
s	36
minister	35
prime	33
was	31
her	30
that	29
you	28
it	27
his	22
with	20
as	20

Word	Count
but	20
t	20
fudge	19
she	19
had	18
for	17
we	17
at	16
not	15
be	15
him	15
scrimgeour	15
narcissa	14
all	13
an	13
from	13
up	12
who	12
they	12
into	11

2.2 Non-English Tokens from `file2.txt` (pyspellchecker)

Token	Count
hermione	29
malfoy	23
borgin	19
weasley	12
malfoy's	6
knockturn	4
hagrid	3
ofthe	3
fleur	3
thankyou	2
weasleys	2
tonks	2

Notes

- **MapReduce:** Implemented with `multiprocessing` (map → shuffle → reduce) on normalized tokens.
- **Non-English heuristic:** A simple dictionary membership check with `pyspellchecker` flags likely proper nouns (names, places, spells) and filtered noise via light preprocessing (token cleanup).
- **DOB requirement:** DOB is explicitly embedded as a code comment and used to compute the book and page windows.

All Word Counts

the	166	for	17	around	8
a	84	we	17	here	8
and	58	at	16	do	8
of	57	not	15	down	8
to	49	be	15	me	7
he	46	him	15	has	7
in	40	scrimgeour	15	what	7
i	39	narcissa	14	is	7
said	36	all	13	very	7
s	36	an	13	just	7
minister	35	from	13	them	7
prime	33	up	12	which	7
was	31	who	12	man	7
her	30	they	12	were	7
that	29	into	11	door	7
you	28	so	10	back	7
it	27	ve	10	light	7
his	22	on	10	re	6
with	20	can	9	must	6
as	20	well	9	out	6
but	20	this	9	then	6
t	20	there	9	other	6
fudge	19	bella	9	did	6
she	19	now	8	if	6
had	18	been	8	over	6

thought	6	are	4	barely	3
my	6	course	4	place	3
looked	6	room	4	through	3
eyes	6	hand	4	magic	3
behind	6	upon	4	looking	3
already	6	m	4	portrait	3
fox	6	smile	4	ll	3
named	5	d	4	clearly	3
last	5	by	4	green	3
time	5	voice	4	their	3
go	5	old	4	like	3
got	5	windows	4	or	3
have	5	your	4	curtains	3
seemed	5	wait	4	let	3
woman	5	long	4	shacklebolt	3
rather	5	two	4	speak	3
yes	5	river	4	merely	3
right	5	black	4	side	3
still	5	bank	4	another	3
moment	5	figure	4	flash	3
first	5	cloak	4	street	3
its	5	hood	4	houses	3
turned	5	followed	4	darkness	3
second	5	stood	4	lord	3
under	5	sister	4	moved	3
asked	5	snape	4	going	2
wand	5	off	3	tell	2
cissy	5	giant	3	hurricane	2
dark	5	really	3	these	2
too	4	magical	3	public	2
almost	4	say	3	injuries	2
after	4	amelia	3	stopped	2
no	4	bones	3	muggles	2
though	4	momentarily	3	saw	2
office	4	any	3	happened	2
won	4	maybe	3	department	2
alone	4	about	3	don	2
wasn	4	one	3	losing	2

law	2	wizard	2	nose	2
may	2	moments	2	some	2
person	2	new	2	grass	2
because	2	robes	2	hooded	2
spinning	2	hair	2	thin	2
see	2	pair	2	air	2
gets	2	holding	2	take	2
toward	2	told	2	few	2
catching	2	heard	2	set	2
didn	2	er	2	listen	2
day	2	busy	2	caught	2
listening	2	security	2	arm	2
dementors	2	drew	2	narrow	2
once	2	himself	2	across	2
anymore	2	replied	2	road	2
creatures	2	happy	2	brick	2
people	2	imperius	2	inthe	2
mist	2	curse	2	streaming	2
made	2	highly	2	broken	2
think	2	work	2	pursuer	2
whole	2	without	2	face	2
wizarding	2	auror	2	wouldn	2
community	2	could	2	house	2
fortnight	2	surely	2	together	2
words	2	attempted	2	slightly	2
sitting	2	shall	2	threw	2
anything	2	case	2	bellatrix	2
kind	2	asthough	2	books	2
nothing	2	fire	2	sofa	2
little	2	slowly	2	cast	2
silver	2	stepped	2	wormtail	2
dumbledore	2	spinner	2	pointed	2
twice	2	end	2	small	2
will	2	away	2	suppose	1
more	2	dirty	2	caused	1
immediately	2	between	2	westcountry	1
rufus	2	chimney	2	temper	1
again	2	apart	2	rising	1

every	1	giants	1	throat	1
pace	1	when	1	effort	1
hetook	1	wanted	1	hisbowler	1
infuriating	1	grand	1	hat	1
discover	1	effect	1	murder	1
reason	1	misinformation	1	newspapers	1
terrible	1	working	1	diverted	1
disasters	1	clock	1	anger	1
able	1	teams	1	ournewspapers	1
worse	1	obliviators	1	middle	1
than	1	trying	1	aged	1
being	1	modify	1	lived	1
thegovernment	1	thememories	1	nastykilling	1
fault	1	mostof	1	lot	1
miserably	1	regulation	1	publicity	1
excuse	1	control	1	police	1
barked	1	creaturesrunning	1	baffled	1
positively	1	somerset	1	sighed	1
stamping	1	find	1	killed	1
anddown	1	disaster	1	thatwas	1
trees	1	furiously	1	locked	1
uprooted	1	deny	1	inside	1
roofs	1	morale	1	know	1
ripped	1	pretty	1	exactlywho	1
lampposts	1	low	1	us	1
bent	1	ministry	1	further	1
horrible	1	whatwith	1	thenthere	1
death	1	head	1	emmeline	1
eaters	1	enforcement	1	vance	1
sfollowers	1	wethink	1	hear	1
suspect	1	murdered	1	oh	1
involvement	1	gifted	1	cornerfrom	1
tracks	1	witch	1	matter	1
hit	1	evidence	1	fact	1
invisiblewall	1	sheput	1	papers	1
whatinvolvement	1	real	1	field	1
grimaced	1	fight	1	breakdown	1
used	1	cleared	1	order	1

backyard	1	ofinvisible	1	sent	1
enough	1	swooping	1	heretonight	1
primeminister	1	towns	1	bring	1
swarming	1	countryside	1	date	1
attacking	1	spreading	1	recent	1
peopleleft	1	despair	1	events	1
center	1	hopelessness	1	introduce	1
happier	1	voters	1	successor	1
sentence	1	feel	1	busyat	1
would	1	quite	1	much	1
unintelligible	1	faint	1	ugly	1
tothe	1	something	1	wearing	1
wiser	1	yourresponsibility	1	longcurly	1
guard	1	dear	1	wig	1
prisoners	1	honestly	1	digging	1
azkaban	1	ofmagic	1	ear	1
cautiously	1	sacked	1	point	1
wearily	1	three	1	quill	1
deserted	1	days	1	catchingfudge	1
theprison	1	ago	1	eye	1
joined	1	screaming	1	finishingaletter	1
pretend	1	resignation	1	wish	1
blow	1	never	1	luck	1
sense	1	known	1	sounding	1
dawning	1	united	1	bitter	1
horror	1	term	1	beenwriting	1
youtell	1	brave	1	past	1
drain	1	attempt	1	budge	1
hope	1	lost	1	prepared	1
happiness	1	despite	1	persuade	1
breeding	1	hisindignation	1	boy	1
causing	1	position	1	might	1
sank	1	placed	1	success	1
weak	1	feltfor	1	subsided	1
kneed	1	shrunken	1	aggrieved	1
nearest	1	opposite	1	silence	1
chair	1	sorry	1	wasbroken	1
idea	1	finally	1	suddenly	1

spoke	1	keen	1	unlocked	1
crisp	1	yellowish	1	interrupted	1
official	1	wire	1	shortly	1
requesting	1	rimmed	1	watched	1
meeting	1	spectacles	1	added	1
urgent	1	certain	1	pointing	1
kindlyrespond	1	rangy	1	swept	1
fine	1	loping	1	acrossthem	1
distractedly	1	grace	1	get	1
flinchedas	1	even	1	business	1
flames	1	hewalked	1	need	1
grate	1	slight	1	discuss	1
emerald	1	limp	1	fullest	1
rose	1	immediate	1	height	1
revealed	1	impression	1	amperfectly	1
heart	1	shrewdness	1	thank	1
disgorging	1	toughness	1	cut	1
later	1	understood	1	poor	1
onto	1	why	1	lookout	1
theantique	1	preferred	1	themuggles	1
rug	1	leader	1	put	1
feet	1	dangeroustimes	1	secretary	1
hesitation	1	how	1	outer	1
same	1	politely	1	getting	1
watching	1	grasped	1	rid	1
arrival	1	briefly	1	kingsley	1
straighten	1	scanning	1	suggesting	1
dust	1	pulled	1	hotly	1
longblack	1	outawand	1	efficient	1
look	1	everything	1	getsthrough	1
foolish	1	striding	1	rest	1
lion	1	andtapping	1	flicker	1
streaks	1	keyhole	1	ofasmile	1
gray	1	lock	1	trained	1
mane	1	click	1	assigned	1
oftawny	1	mind	1	yourprotection	1
bushy	1	ratherthat	1	declared	1
eyebrows	1	remained	1	putyour	1

decide	1	stranglethree	1	sort	1
works	1	best	1	spot	1
coldly	1	remove	1	exchanged	1
am	1	frommuggle	1	incredulous	1
problem	1	society	1	lookwith	1
continues	1	while	1	manage	1
excellent	1	ministeranxiously	1	kindly	1
lamely	1	shrugged	1	thetrouble	1
tohear	1	moving	1	wizards	1
herbert	1	fireplace	1	brightgreen	1
chorley	1	keep	1	vanished	1
junior	1	posted	1	mchapter	1
continued	1	developments	1	miles	1
entertaining	1	least	1	chilly	1
impersonating	1	probably	1	pressed	1
duck	1	come	1	against	1
reacted	1	personally	1	drifted	1
poorly	1	send	1	wound	1
performed	1	consented	1	overgrown	1
saidscrimgeour	1	stay	1	rubbish	1
addled	1	advisory	1	strewn	1
brains	1	capacity	1	banks	1
dangerous	1	unsuccessful	1	immense	1
only	1	toothache	1	relic	1
quacking	1	rummaging	1	disusedmill	1
weakly	1	pocketfor	1	reared	1
bit	1	mysterious	1	shadowy	1
ofarest	1	powder	1	ominous	1
easy	1	ministergazed	1	sound	1
drink	1	hopelessly	1	thewhisper	1
team	1	hadfought	1	water	1
healers	1	suppress	1	sign	1
st	1	evening	1	life	1
mungo	1	burst	1	scrawny	1
hospital	1	heaven	1	thathad	1
maladies	1	sake	1	slunk	1
examining	1	rewizards	1	hopefully	1
far	1	domagic	1	fish	1

chipwrappings	1	otherwrenched	1	succeeding	1
tall	1	listened	1	hold	1
faintpop	1	decision	1	armand	1
slim	1	leave	1	swinging	1
appeared	1	gained	1	faced	1
edge	1	top	1	each	1
froze	1	where	1	trust	1
wary	1	line	1	trusts	1
fixed	1	oldrailings	1	doesn	1
strangenew	1	separated	1	believe	1
phenomenon	1	cobbled	1	mistaken	1
bearings	1	therows	1	panted	1
quick	1	rows	1	eyesgleamed	1
strides	1	dilapidated	1	check	1
rustling	1	dull	1	indeed	1
louderpop	1	blind	1	plan	1
materialized	1	lives	1	anyone	1
harsh	1	contempt	1	betrayal	1
cry	1	muggle	1	snarled	1
startled	1	dunghill	1	beneath	1
crouching	1	our	1	threateningly	1
flat	1	ever	1	laughed	1
theundergrowth	1	foot	1	own	1
leapt	1	slipped	1	breathed	1
hiding	1	gap	1	note	1
yelp	1	rustyrailings	1	ofhysteria	1
fell	1	hurrying	1	brought	1
ground	1	dartingthrough	1	knife	1
dead	1	alley	1	therewas	1
animal	1	identical	1	burned	1
toe	1	streetlamps	1	rushed	1
dismissively	1	women	1	ahead	1
perhaps	1	running	1	rubbing	1
quarry	1	betweenpatches	1	keeping	1
paused	1	deep	1	distance	1
scrambling	1	prey	1	deeper	1
fallen	1	justas	1	desertedlabyrinth	1
seized	1	corner	1	hurried	1

towering	1	gave	1	armchair	1
mill	1	lookof	1	rickety	1
hover	1	drowned	1	table	1
admonitory	1	opening	1	grouped	1
finger	1	wider	1	pool	1
footsteps	1	lightfell	1	dim	1
echoed	1	pleasant	1	byacandle	1
cobbles	1	surprise	1	filled	1
passedboarded	1	severus	1	lamp	1
until	1	strained	1	hung	1
reached	1	whisper	1	ceiling	1
whereadim	1	surgent	1	neglect	1
glimmered	1	allow	1	usually	1
downstairs	1	pass	1	inhabited	1
knocked	1	hoodedsister	1	gestured	1
before	1	invitation	1	aside	1
cursing	1	curtly	1	sat	1
breath	1	passed	1	staring	1
hadcaught	1	mouth	1	white	1
waiting	1	curling	1	trembling	1
panting	1	mocking	1	hands	1
breathing	1	closed	1	clasped	1
smell	1	snap	1	lap	1
carried	1	directly	1	lowered	1
night	1	tiny	1	fair	1
breeze	1	feeling	1	withheavily	1
afterafew	1	ofa	1	lidded	1
seconds	1	padded	1	strong	1
movement	1	cell	1	jaw	1
opened	1	walls	1	gaze	1
crack	1	completely	1	stand	1
sliver	1	covered	1	settling	1
seen	1	most	1	armchairopposite	1
hairparted	1	ofthem	1	sisters	1
sallow	1	bound	1	aren	1
pale	1	brown	1	quietly	1
shine	1	leather	1	counting	1
blonde	1	threadbare	1	vermin	1

arewe	1	frozen	1	wore	1
wall	1	realized	1	unpleasantsimper	1
bang	1	guests	1	left	1
ahidden	1	snapelazily	1	caressing	1
flew	1	crept	1	encased	1
open	1	hunchbacked	1	bright	1
revealing	1	steps	1	glove	1
staircase	1	theroom	1		
manstood	1	watery	1		